

# SIEMENS

## SINUMERIK

### SINUMERIK 840D sl / 828D Grundfunktionen

Funktionshandbuch

Gültig für

Steuerung  
SINUMERIK 840D sl / 840DE sl  
SINUMERIK 828D

Software Version  
CNC-Software 4.5 SP1

07/2012

6FC5397-0BP40-3AA0


#### Vorwort


A2: Diverse NC/PLC-Nahtstellensignale und Funktionen	1
A3: Achsüberwachungen, Schutzbereiche	2
B1: Bahnsteuerbetrieb, Genauhalt, LookAhead	3
B2: Beschleunigung	4
F1: Fahren auf Festanschlag	5
G2: Geschwindigkeiten, Soll-/Istwertsysteme, Regelung	6
H2: Hilfsfunktionsausgaben an PLC	7
K1: BAG, Kanal, Programm-betrieb, Reset-Verhalten	8
K2: Achsen, Koordinaten-systeme, Frames	9
N2: Not-Halt	10
P1: Planachsen	11
P3: PLC-Grundprogramm für SINUMERIK 840D sl	12
P4: PLC für SINUMERIK 828D	13
R1: Referenzieren	14
S1: Spindeln	15
V1: Vorschübe	16
W1: Werkzeugkorrektur	17
Z1: NC/PLC-Nahtstellen-signale	18
Anhang	A


## Rechtliche Hinweise

### Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 <b>GEFAHR</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>wird</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>WARNUNG</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>kann</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>VORSICHT</b>
mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

<b>VORSICHT</b>
ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

<b>ACHTUNG</b>
bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.


Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

### Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

### Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 <b>WARNUNG</b>
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

### Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

### Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

# Vorwort

## SINUMERIK-Dokumentation

Die SINUMERIK-Dokumentation ist in folgende Kategorien gegliedert:

- Allgemeine Dokumentation
- Anwender-Dokumentation
- Hersteller/Service-Dokumentation

## Weiterführende Informationen

Unter dem Link [www.siemens.com/motioncontrol/docu](http://www.siemens.com/motioncontrol/docu) finden Sie Informationen zu folgenden Themen:

- Dokumentation bestellen / Druckschriftenübersicht
- Weiterführende Links für den Download von Dokumenten
- Dokumentation online nutzen (Handbücher/Informationen finden und durchsuchen)

Bei Fragen zur Technischen Dokumentation (z. B. Anregungen, Korrekturen) senden Sie bitte eine E-Mail an folgende Adresse:

[docu.motioncontrol@siemens.com](mailto:docu.motioncontrol@siemens.com)

## My Documentation Manager (MDM)

Unter folgendem Link finden Sie Informationen, um auf Basis der Siemens Inhalte eine OEM-spezifische Maschinen-Dokumentation individuell zusammenstellen:

[www.siemens.com/mdm](http://www.siemens.com/mdm)

## Training

Informationen zum Trainingsangebot finden Sie unter:

- [www.siemens.com/sitrain](http://www.siemens.com/sitrain)  
SITRAIN - das Training von Siemens für Produkte, Systeme und Lösungen der Automatisierungstechnik
- [www.siemens.com/sinutrain](http://www.siemens.com/sinutrain)  
SinuTrain - Trainingssoftware für SINUMERIK

## FAQs

Frequently Asked Questions finden Sie in den Service&Support Seiten unter Produkt Support. <http://support.automation.siemens.com>

## SINUMERIK

Informationen zu SINUMERIK finden Sie unter folgendem Link:

[www.siemens.com/sinumerik](http://www.siemens.com/sinumerik)

## Zielgruppe

Die vorliegende Druckschrift wendet sich an:

- Projekteure
- Technologen (von Maschinenherstellern)
- Inbetriebnehmer (von Systemen/Maschinen)
- Programmierer

## Nutzen

Das Funktionshandbuch beschreibt die Funktionen, so dass die Zielgruppe die Funktionen kennt und auswählen kann. Es befähigt die Zielgruppe, die Funktionen in Betrieb zu nehmen.

## Standardumfang

In der vorliegenden Dokumentation ist die Funktionalität des Standardumfangs beschrieben. Ergänzungen oder Änderungen, die durch den Maschinenhersteller vorgenommen werden, werden vom Maschinenhersteller dokumentiert.

Es können in der Steuerung weitere, in dieser Dokumentation nicht erläuterte Funktionen ablauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei der Neulieferung bzw. im Servicefall.

Ebenso enthält diese Dokumentation aus Gründen der Übersichtlichkeit nicht sämtliche Detailinformationen zu allen Typen des Produkts und kann auch nicht jeden denkbaren Fall der Aufstellung, des Betriebes und der Instandhaltung berücksichtigen.

## Technical Support

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet unter <http://www.siemens.com/automation/service&support>

## Informationen zu Struktur und Inhalt

### Aufbau

Das vorliegende Funktionshandbuch ist wie folgt aufgebaut:

- Innentitel (Seite 3) mit dem Titel des Funktionshandbuchs, den SINUMERIK-Steuerungen sowie der Software und Version, für die diese Ausgabe des Funktionshandbuchs gültig ist, und der Übersicht der einzelnen Funktionsbeschreibungen.
- Funktionsbeschreibungen in der Reihenfolge ihrer alphanumerischen Kurzzeichen (z. B. A2, A3, B1 etc.)
- Anhang mit:
  - Abkürzungsverzeichnis
  - Dokumentationsübersicht
- Begriffsverzeichnis

---

### Hinweis

Ausführliche Daten- und Alarm-Beschreibungen siehe:

- für Maschinen- und Settingdaten:
    - Ausführliche Maschinendaten-Beschreibung (nur elektronisch auf DOConCD oder DOConWEB)
  - für NC/PLC-Nahtstellensignale:
    - Funktionshandbuch Grundfunktionen; NC/PLC-Nahtstellensignale (Z1)
    - Funktionshandbuch Erweiterungsfunktionen; NC/PLC-Nahtstellensignale (Z2)
    - Funktionshandbuch Sonderfunktionen; NC/PLC-Nahtstellensignale (Z3)
  - für Alarme:
    - Diagnosehandbuch
- 

## Schreibweise von Systemdaten

Für Systemdaten gelten in dieser Dokumentation folgende Schreibweisen:

Signal/Datum	Schreibweise	Beispiel
NC/PLC-Nahtstellensignale	... NC/PLC-Nahtstellensignal: <Signaladresse> (<Signalname>)	Ist die neue Getriebestufe eingelegt, dann werden vom PLC-Programm die folgenden NC/PLC-Nahtstellensignale gesetzt: DB31, ... DBX16.0-2 (Istgetriebestufe A bis C) DB31, ... DBX16.3 (Getriebe ist umgeschaltet)
Maschinendatum	... Maschinendatum: <Typ><Nummer> <Vollständiger Bezeichner> (<Bedeutung>)	Masterspindel ist die abgelegte Spindel im Maschinendatum: MD20090 \$MC_SPIND_DEF_MASTER_SPIND (Löschstellung Masterspindel im Kanal)
Settingdatum	... Settingdatum: <Typ><Nummer> <Vollständiger Bezeichner> (<Bedeutung>)	Die logische Masterspindel ist enthalten im Settingdatum: SD42800 \$SC_SPIND_ASSIGN_TAB[0] (Spindelnummernumsetzer)

**Hinweis**

**Signaladresse**

Die Funktionsbeschreibungen enthalten als <Signaladresse> eines NC/PLC-Nahtstellensignals nur die für SINUMERIK 840D sl gültige Adresse. Die Signaladresse für SINUMERIK 828D ist den Datenlisten "Signale an/von ..." am Ende der jeweiligen Funktionsbeschreibung zu entnehmen.

---

**Mengengerüst**

Erläuterungen bezüglich der NC/PLC-Nahtstelle gehen von der absoluten maximalen Anzahl folgender Komponenten aus:

- Betriebsartengruppen (DB11)
- Kanäle (DB21, ...)
- Achsen/Spindeln (DB31, ...)

**Datentypen**

In der Steuerung werden die folgenden elementaren Datentypen verwendet:

Typ	Bedeutung	Wertebereich
INT	Ganzzahlige Werte mit Vorzeichen	-2147483648 ... +2147483647
REAL	Zahlen mit Dezimalpunkt nach IEEE	$\pm(2,2 \cdot 10^{-308} \dots 1,8 \cdot 10^{308})$
BOOL	Wahrheitswerte TRUE (1) und FALSE (0)	1, 0
CHAR	Zeichen ASCII	entsprechend Code 0 ... 255
STRING	Zeichenkette, Zeichenzahl in [...]	maximal 200 Zeichen (keine Sonderzeichen)
AXIS	nur Achsnamen (Achsadressen)	alle im Kanal vorhandenen Achsadressen
FRAME	geometrische Angaben für Verschieben, Drehen, Skalieren, Spiegeln	

Arrays können nur aus gleichen elementaren Datentypen gebildet werden. Es sind maximal 3-dimensionale Arrays möglich.

# Inhaltsverzeichnis

	<b>Vorwort</b> .....	<b>3</b>
<b>1</b>	<b>A2: Diverse NC/PLC-Nahtstellensignale und Funktionen</b> .....	<b>33</b>
1.1	Kurzbeschreibung .....	33
1.2	NC/PLC-Nahtstellensignale .....	34
1.2.1	Allgemeines .....	34
1.2.2	Bereitschaftssignale an PLC .....	36
1.2.3	Statussignale an PLC .....	36
1.2.4	Signale an/von Bedientafelfront .....	37
1.2.5	Signale an Kanal .....	39
1.2.6	Signale an Achse/Spindel .....	40
1.2.7	Signale von Achse/Spindel .....	50
1.2.8	Signale an Achse/Spindel (Digitale Antriebe) .....	51
1.2.9	Signale von Achse/Spindel (Digitale Antriebe) .....	53
1.3	Funktionen .....	56
1.3.1	BildschirmEinstellungen .....	56
1.3.2	Einstellungen für Evolventen-Interpolation .....	57
1.3.3	DEFAULT-Speicher aktivieren .....	60
1.3.4	PLC-Variable lesen und schreiben .....	61
1.3.5	Zugriffsschutz über Kennwort und Schlüsselschalter .....	65
1.3.5.1	Zugriffsschutz über Kennwort und Schlüsselschalter .....	65
1.3.5.2	Kennwort .....	66
1.3.5.3	Schlüsselschalter-Stellungen (DB10, DBX56.4 bis 7) .....	67
1.3.5.4	Parametrierbare Schutzstufen .....	68
1.3.6	"Parken" einer Maschinenachse .....	69
1.3.7	Umschaltung von Motor- / Antriebsparameterdatensätzen .....	70
1.3.7.1	Allgemeine Informationen .....	70
1.3.7.2	Gültigkeit und Format der Anforderungs- / Anzeige-Schnittstellen .....	71
1.3.7.3	Anforderung eines neuen Motor- und/oder Antriebsparameterdatensatzes .....	72
1.3.7.4	Anzeige des aktiven Motor- und/oder Antriebsparameterdatensatzes .....	72
1.3.7.5	Beispiel .....	73
1.3.7.6	Übersicht der Schnittstellen .....	74
1.3.7.7	Randbedingungen .....	75
1.4	Beispiele .....	76
1.5	Datenlisten .....	78
1.5.1	Maschinendaten .....	78
1.5.1.1	Anzeige-Maschinendaten .....	78
1.5.1.2	NC-spezifischen Maschinendaten .....	78
1.5.1.3	Kanal-spezifische Maschinendaten .....	78
1.5.1.4	Achs-/Spindel-spezifische Maschinendaten .....	79
1.5.2	Systemvariablen .....	80
1.5.3	Signale .....	80
1.5.3.1	Signale an NC .....	80
1.5.3.2	Signale von NC .....	80

1.5.3.3	Signale an Bedientafelfront .....	81
1.5.3.4	Signale von Bedientafelfront .....	81
1.5.3.5	Signale an Kanal .....	81
1.5.3.6	Signale von Kanal .....	82
1.5.3.7	Signale an Achse/Spindel .....	82
1.5.3.8	Signale von Achse/Spindel .....	83
<b>2</b>	<b>A3: Achsüberwachungen, Schutzbereiche .....</b>	<b>85</b>
2.1	Kurzbeschreibung .....	85
2.1.1	Achsüberwachungen.....	85
2.1.2	Schutzbereiche .....	85
2.2	Achsüberwachungen.....	86
2.2.1	Konturüberwachung .....	86
2.2.1.1	Konturfehler.....	86
2.2.1.2	Schleppabstandsüberwachung .....	87
2.2.2	Positionier-, Stillstands- und Klemmungsüberwachung.....	89
2.2.2.1	Zusammenhang zwischen Positionier-, Stillstands- und Klemmungsüberwachung.....	89
2.2.2.2	Positionierüberwachung.....	90
2.2.2.3	Stillstandsüberwachung .....	91
2.2.2.4	Parametersatzabhängige Genauhalt- und Stillstandstoleranz.....	92
2.2.2.5	Klemmungsüberwachung.....	93
2.2.3	Drehzahlsollwertüberwachung .....	101
2.2.4	Istgeschwindigkeitsüberwachung .....	103
2.2.5	Messsystem-Überwachung.....	104
2.2.5.1	Gebergrenzfrequenzüberwachung .....	106
2.2.5.2	Plausibilitätskontrolle bei Absolutwertgebern.....	107
2.2.5.3	Anwenderspezifische Fehlerreaktionen .....	110
2.2.6	Endschalter-Überwachung.....	112
2.2.6.1	Hardware-Endschalter .....	112
2.2.6.2	Software-Endschalter.....	113
2.2.7	Überwachung der Arbeitsfeldbegrenzung.....	115
2.2.7.1	Allgemein.....	115
2.2.7.2	Arbeitsfeldbegrenzung im BKS .....	117
2.2.7.3	Arbeitsfeldbegrenzung im WKS/ENS.....	120
2.2.8	Ausschalten aller Überwachungen: "Parken" .....	123
2.3	Schutzbereiche .....	124
2.3.1	Allgemeines.....	124
2.3.2	Schutzbereichsarten .....	125
2.3.3	Definition per Teileprogrammanweisung .....	128
2.3.4	Definition per Systemvariable .....	132
2.3.5	Aktivieren und Deaktivieren von Schutzbereichen .....	134
2.3.6	Schutzbereichsverletzung und zeitweise Freigabe einzelner Schutzbereiche .....	139
2.3.7	Einschränkungen bei Schutzbereichen.....	143
2.4	Randbedingungen.....	145
2.4.1	Achsüberwachungen.....	145
2.5	Beispiele.....	146
2.5.1	Achsüberwachungen.....	146
2.5.1.1	Arbeitsfeldbegrenzung im WKS/ENS.....	146
2.5.2	Schutzbereiche .....	149
2.5.2.1	Schutzbereichsdefinition und Aktivierung .....	149



2.6	Datenlisten .....	159
2.6.1	Maschinendaten.....	159
2.6.1.1	NC-spezifische Maschinendaten .....	159
2.6.1.2	Kanal-spezifische Maschinendaten .....	160
2.6.1.3	Achs-/Spindel-spezifische Maschinendaten .....	161
2.6.2	Settingdaten.....	162
2.6.2.1	Achs-/Spindel-spezifische Settindaten .....	162
2.6.3	Signale .....	163
2.6.3.1	Signale an Kanal.....	163
2.6.3.2	Signale von Kanal .....	164
2.6.3.3	Signale an Achse/Spindel.....	165
<b>3</b>	<b>B1: Bahnsteuerbetrieb, Genauhalt, LookAhead.....</b>	<b>167</b>
3.1	Kurzbeschreibung .....	167
3.2	Genauhaltbetrieb .....	171
3.3	Bahnsteuerbetrieb.....	176
3.3.1	Allgemeine Funktionalität.....	176
3.3.2	Geschwindigkeitsabsenkung gemäß Überlastfaktor.....	178
3.3.3	Überschleifen .....	180
3.3.3.1	Überschleifen nach Wegkriterium (G641).....	183
3.3.3.2	Überschleifen unter Einhaltung definierter Toleranzen (G642/G643) .....	185
3.3.3.3	Überschleifen mit maximal möglicher Achsdynamik (G644) .....	189
3.3.3.4	Überschleifen tangentialer Satzübergänge (G645) .....	193
3.3.4	LookAhead.....	194
3.3.4.1	Standardfunktionalität .....	194
3.3.4.2	Freiformflächenmodus: Erweiterungsfunktion .....	201
3.4	Dynamikanpassungen .....	206
3.4.1	Glättung der Bahngeschwindigkeit .....	206
3.4.2	Anpassung der Bahndynamik.....	210
3.4.3	Ermittlung der Dynamikgrenzwerte.....	214
3.4.4	Zusammenwirken der Funktionen "Glättung der Bahngeschwindigkeit" und "Anpassung der Bahndynamik".....	215
3.4.5	Dynamikmodus für Bahninterpolation.....	219
3.4.6	Freiformflächenmodus: Grundfunktionen .....	222
3.5	Kompressor-Funktionen.....	226
3.5.1	NC-Satz-Kompression .....	226
3.5.2	Zusammenfassung kurzer Spline-Sätze.....	229
3.6	Kontur-/Orientierungstoleranz.....	232
3.7	Toleranz und Komprimierung von G0-Sätzen .....	236
3.8	RESET-Verhalten .....	239
3.9	Randbedingungen.....	240
3.9.1	Satzwechsel und Positionierachsen .....	240
3.9.2	Satzwechselferzögerung.....	240
3.9.3	Überschleifen und Repositionieren (REPOS).....	241
3.10	Datenlisten .....	242
3.10.1	Maschinendaten.....	242
3.10.1.1	Allgemeine Maschinendaten.....	242
3.10.1.2	Kanal-spezifische Maschinendaten .....	242

3.10.1.3	Achs-/Spindel-spezifische Maschinendaten .....	243
3.10.2	Settingdaten .....	244
3.10.2.1	Kanal-spezifische Settingdaten .....	244
3.10.3	Signale .....	244
3.10.3.1	Signale von Kanal .....	244
3.10.3.2	Signale von Achse/Spindel .....	244
<b>4</b>	<b>B2: Beschleunigung .....</b>	<b>245</b>
4.1	Kurzbeschreibung .....	245
4.1.1	Allgemeines .....	245
4.1.2	Merkmale .....	245
4.2	Funktionen .....	247
4.2.1	Beschleunigung ohne Ruckbegrenzung (BRISK/BRISKA) (kanal-/achsspezifisch) .....	247
4.2.1.1	Allgemeine Informationen .....	247
4.2.1.2	Parametrierung .....	249
4.2.1.3	Programmierung .....	250
4.2.2	Konstantfahrzeit (kanalspezifisch) .....	251
4.2.2.1	Allgemeine Informationen .....	251
4.2.2.2	Parametrierung .....	252
4.2.3	Beschleunigungsanpassung (ACC) (achsspezifisch) .....	253
4.2.3.1	Allgemeine Informationen .....	253
4.2.3.2	Programmierung .....	253
4.2.4	Beschleunigungsreserve (kanalspezifisch) .....	254
4.2.4.1	Allgemeine Informationen .....	254
4.2.4.2	Parametrierung .....	254
4.2.5	Begrenzung der Bahnbeschleunigung (kanalspezifisch) .....	255
4.2.5.1	Allgemeine Informationen .....	255
4.2.5.2	Parametrierung .....	255
4.2.5.3	Programmierung .....	255
4.2.6	Bahnbeschleunigung für Echtzeitereignisse (kanalspezifisch) .....	256
4.2.6.1	Allgemeine Informationen .....	256
4.2.6.2	Programmierung .....	258
4.2.7	Beschleunigung bei programmiertem Eilgang (G00) (achsspezifisch) .....	259
4.2.7.1	Allgemeine Informationen .....	259
4.2.7.2	Parametrierung .....	259
4.2.8	Beschleunigung bei aktiver Ruckbegrenzung (SOFT/SOFTA) (achsspezifisch) .....	260
4.2.8.1	Allgemeine Informationen .....	260
4.2.8.2	Parametrierung .....	260
4.2.9	Beschleunigungsüberhöhung bei nicht tangentialen Satzübergängen (achsspezifisch) .....	261
4.2.9.1	Allgemeine Informationen .....	261
4.2.9.2	Parametrierung .....	261
4.2.10	Beschleunigungsreserve für die Radialbeschleunigung (kanalspezifisch) .....	261
4.2.10.1	Allgemeine Informationen .....	261
4.2.10.2	Parametrierung .....	263
4.2.11	Ruckbegrenzung bei Bahninterpolation (SOFT) (kanalspezifisch) .....	263
4.2.11.1	Allgemeine Informationen .....	263
4.2.11.2	Parametrierung .....	265
4.2.11.3	Programmierung .....	266
4.2.12	Ruckbegrenzung bei Einzelachsinterpolation (SOFTA) (achsspezifisch) .....	266
4.2.12.1	Parametrierung .....	266
4.2.12.2	Programmierung .....	267
4.2.13	Begrenzung des Bahnrucks (kanalspezifisch) .....	268

4.2.13.1	Allgemeine Informationen .....	268
4.2.13.2	Parametrierung .....	268
4.2.13.3	Programmierung .....	269
4.2.14	Bahnruck für Echtzeitereignisse (kanalspezifisch) .....	270
4.2.14.1	Allgemeine Informationen .....	270
4.2.14.2	Programmierung .....	271
4.2.15	Ruck bei programmiertem Eilgang (G00) (achsspezifisch) .....	272
4.2.15.1	Allgemeine Informationen .....	272
4.2.15.2	Parametrierung .....	272
4.2.16	Rucküberhöhung bei nicht krümmungsstetigen Satzübergängen (achsspezifisch) .....	272
4.2.16.1	Allgemeine Informationen .....	272
4.2.16.2	Parametrierung .....	273
4.2.17	Geschwindigkeitsabhängige Ruckadaption (achsspezifisch) .....	273
4.2.18	Ruckfilter (achsspezifisch) .....	276
4.2.18.1	Allgemeine Informationen .....	276
4.2.18.2	Parametrierung .....	278
4.2.19	Geknickte Beschleunigungskennlinie .....	279
4.2.19.1	Anpassung an die Motorkennlinie.....	279
4.2.19.2	Auswirkungen auf die Bahnbeschleunigung.....	282
4.2.19.3	Ersatzkennlinie.....	283
4.2.19.4	Parametrierung .....	285
4.2.19.5	Programmierung .....	286
4.2.19.6	Randbedingungen.....	287
4.2.20	Beschleunigung und Ruck bei JOG-Bewegungen.....	288
4.2.20.1	Parametrierung .....	288
4.2.20.2	Randbedingungen.....	290
4.3	Beispiele.....	292
4.3.1	Beschleunigung .....	292
4.3.1.1	Bahngeschwindigkeitsverlauf.....	292
4.3.2	Ruck .....	294
4.3.2.1	Bahngeschwindigkeitsverlauf.....	294
4.3.3	Beschleunigung und Ruck .....	295
4.3.4	Geknickte Beschleunigungskennlinie .....	297
4.3.4.1	Aktivierung .....	297
4.4	Datenlisten .....	299
4.4.1	Maschinendaten.....	299
4.4.1.1	NC-spezifische Maschinendaten .....	299
4.4.1.2	Kanal-spezifische Maschinendaten .....	299
4.4.1.3	Achs-/Spindel-spezifische Maschinendaten .....	300
4.4.2	Settingdaten.....	301
4.4.2.1	Kanal-spezifische Settingdaten.....	301
4.4.3	Systemvariablen .....	301
<b>5</b>	<b>F1: Fahren auf Festanschlag .....</b>	<b>303</b>
5.1	Kurzbeschreibung .....	303
5.2	Allgemeine Funktionalität.....	304
5.2.1	Programmierung .....	304
5.2.2	Funktionsablauf.....	307
5.2.2.1	Anwahl .....	307
5.2.2.2	Festanschlag wird erreicht.....	308
5.2.2.3	Festanschlag wird nicht erreicht .....	310

5.2.2.4	Abwahl.....	312
5.2.3	Verhalten bei Satzsuchlauf .....	313
5.2.4	Verhalten bei Reset und Funktionsabbruch.....	317
5.2.5	Verhalten bezüglich anderer Funktionen .....	318
5.2.6	Settingdaten .....	319
5.2.7	Systemvariable.....	320
5.2.8	Festanschlagsalarme .....	321
5.2.9	Fahren mit begrenztem Moment/Kraft FOC.....	323
5.3	Beispiele.....	327
5.4	Datenlisten .....	329
5.4.1	Maschinendaten .....	329
5.4.1.1	Achs-/Spindel-spezifische Maschinendaten .....	329
5.4.2	Settingdaten .....	329
5.4.2.1	Achs-/Spindel-spezifische Settingdaten.....	329
5.4.3	Signale .....	330
5.4.3.1	Signale an Achse/Spindel .....	330
5.4.3.2	Signale von Achse/Spindel .....	330
<b>6</b>	<b>G2: Geschwindigkeiten, Soll-/Istwertsysteme, Regelung .....</b>	<b>331</b>
6.1	Kurzbeschreibung .....	331
6.2	Geschwindigkeiten, Verfahrbereiche, Genauigkeiten .....	332
6.2.1	Geschwindigkeiten .....	332
6.2.2	Verfahrbereiche.....	334
6.2.3	Positioniergenauigkeit der Steuerung .....	334
6.2.4	Eingabe-/Anzeigefeinheit, Rechenfeinheit.....	335
6.2.5	Normierung physikalischer Größen der Maschinen- und Settingdaten .....	337
6.3	Metrisches-/Inch-Maßsystem .....	341
6.3.1	Umrechnung des Grundsystems mittels Teileprogramm.....	341
6.3.2	Manuelle Umschaltung des Grundsystems .....	346
6.3.3	FGROUP und FGREF.....	352
6.4	Soll-/Istwertsystem .....	355
6.4.1	Allgemeines.....	355
6.4.2	Sollwert- und Geberzuordnung .....	358
6.4.3	Anpassungen der Motor/Last-Verhältnisse.....	361
6.4.4	Drehzahlsollwertausgabe.....	365
6.4.5	Istwertverarbeitung.....	367
6.4.6	Istwertauflösung .....	369
6.4.6.1	Funktionsbeschreibung .....	369
6.4.6.2	Beispiel: Linearachse mit Linearmaßstab .....	371
6.4.6.3	Beispiel: Linearachse mit rotatorischem Geber am Motor .....	371
6.4.6.4	Beispiel: Linearachse mit rotatorischem Geber an der Maschine .....	373
6.4.6.5	Beispiel: Rundachse mit rotatorischem Geber am Motor .....	374
6.4.6.6	Beispiel: Rundachse mit rotatorischem Geber an der Maschine.....	375
6.4.6.7	Beispiel: Vorsatz-Getriebe mit Geber am Werkzeug .....	376
6.5	Regelung .....	377
6.5.1	Allgemeines.....	377
6.5.2	Parametersätze des Lagereglers .....	381
6.6	Optimierung der Regelung .....	383
6.6.1	Lagereglers Lagesollwertfilter: Symmetrierfilter.....	383

6.6.2	Lageregler Lagesollwertfilter: Ruckfilter.....	387
6.6.3	Lageregelung mit PI-Regler .....	389
6.7	Datenlisten .....	391
6.7.1	Maschinendaten.....	391
6.7.1.1	Anzeige-Maschinendaten .....	391
6.7.1.2	NC-spezifische Maschinendaten .....	391
6.7.1.3	Kanal-spezifische Maschinendaten .....	391
6.7.1.4	Achs-/Spindel-spezifische Maschinendaten .....	392
<b>7</b>	<b>H2: Hilfsfunktionsausgaben an PLC .....</b>	<b>395</b>
7.1	Kurzbeschreibung .....	395
7.1.1	Funktion .....	395
7.1.2	Definition einer Hilfsfunktion .....	396
7.1.3	Übersicht der Hilfsfunktionen .....	397
7.2	Vordefinierte Hilfsfunktionen .....	404
7.2.1	Übersicht: vordefinierten Hilfsfunktionen .....	404
7.2.2	Übersicht: Ausgabeverhalten.....	418
7.2.3	Parametrierung .....	421
7.2.3.1	Gruppenzuordnung .....	421
7.2.3.2	Typ, Adresserweiterung und Wert .....	422
7.2.3.3	Ausgabeverhalten .....	423
7.3	Anwenderdefinierte Hilfsfunktionen .....	428
7.3.1	Parametrierung .....	430
7.3.1.1	Maximale Anzahl anwenderdefinierter Hilfsfunktionen .....	430
7.3.1.2	Gruppenzuordnung .....	430
7.3.1.3	Typ, Adresserweiterung und Wert .....	431
7.3.1.4	Ausgabeverhalten .....	432
7.4	Assoziierte Hilfsfunktionen.....	433
7.5	Typ-spezifisches Ausgabeverhalten .....	435
7.6	Prioritäten des parametrierten Ausgabeverhaltens .....	437
7.7	Programmierung einer Hilfsfunktion .....	439
7.8	Programmierbare Ausgabedauer.....	441
7.9	Hilfsfunktionsausgabe an die PLC .....	443
7.10	Hilfsfunktionen ohne Satzwechselfverzögerung.....	444
7.11	M-Funktion mit implizitem Vorlaufstopp.....	445
7.12	Verhalten bei Überspeichern .....	446
7.13	Verhalten bei Satzsuchlauf .....	447
7.13.1	Hilfsfunktionsausgabe bei Satzsuchlauf Typ 1,2 und 4 .....	447
7.13.2	Zuordnung einer Hilfsfunktion zu mehreren Gruppen .....	449
7.13.3	Zeitstempel der aktiven M-Hilfsfunktion.....	451
7.13.4	Ermittlung der Ausgabe-Reihenfolge.....	451
7.13.5	Ausgabeunterdrückung von Spindel-spezifischen Hilfsfunktionen .....	453
7.13.6	Hilfsfunktionsausgabe bei Satzsuchlauf Typ 5 (SERUPRO).....	457
7.13.7	SERUPRO-Ende-ASUP.....	462
7.14	Implizit ausgegebene Hilfsfunktionen .....	469

7.15	Informationsmöglichkeiten .....	471
7.15.1	Gruppenspezifische modale M-Hilfsfunktionsanzeige .....	471
7.15.2	Abfrage von Systemvariablen .....	473
7.16	Randbedingungen .....	475
7.16.1	Allgemeine Randbedingungen .....	475
7.16.2	Ausgabeverhalten .....	476
7.17	Beispiele .....	478
7.17.1	Erweiterung von vordefinierten Hilfsfunktionen .....	478
7.17.2	Definition von Hilfsfunktionen .....	480
7.18	Datenlisten .....	484
7.18.1	Maschinendaten .....	484
7.18.1.1	NC-spezifische Maschinendaten .....	484
7.18.1.2	Kanal-spezifische Maschinendaten .....	484
7.18.2	Signale .....	485
7.18.2.1	Signale an Kanal .....	485
7.18.2.2	Signale von Kanal .....	485
7.18.2.3	Signale an Achse/Spindel .....	487
7.18.2.4	Signale von Achse/Spindel .....	487
<b>8</b>	<b>K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten.....</b>	<b>489</b>
8.1	Kurzbeschreibung .....	489
8.2	Betriebsartengruppe (BAG).....	493
8.2.1	BAG-Stop .....	496
8.2.2	BAG-Reset .....	497
8.3	Betriebsarten und Betriebsartenwechsel .....	498
8.3.1	Überwachungen und Verriegelungen der einzelnen Betriebsarten .....	503
8.3.2	Betriebsartenwechsel.....	504
8.4	Kanal .....	506
8.4.1	Globale Startsperrung für Kanal.....	509
8.5	Programmtest.....	511
8.5.1	Programmbearbeitung ohne Sollwertausgaben.....	511
8.5.2	Programmbearbeitung im Einzelsatzbetrieb .....	513
8.5.3	Programmbearbeitung mit Probelaufvorschub .....	515
8.5.4	Teileprogrammsätze ausblenden.....	517
8.6	Werkstücksimulation .....	519
8.7	Satzsuchlauf.....	521
8.7.1	Ablauf für Satzsuchlauf vom Typ 1, 2 und 4 .....	523
8.7.2	Satzsuchlauf im Zusammenhang mit weiteren NCK-Funktionen .....	525
8.7.2.1	ASUP nach und bei Satzsuchlauf .....	525
8.7.2.2	PLC-Aktionen nach Satzsuchlauf .....	526
8.7.2.3	Spindelfunktionen nach Satzsuchlauf .....	527
8.7.2.4	Lesen von Systemvariablen bei Satzsuchlauf .....	528
8.7.3	Automatischer Start eines ASUP nach Satzsuchlauf .....	528
8.7.4	Kaskadierter Satzsuchlauf .....	530
8.7.5	Beispiele zum Satzsuchlauf mit Berechnung.....	533
8.8	Satzsuchlauf Typ 5 SERUPRO.....	537
8.8.1	Funktionsbeschreibung .....	537

8.8.2	REPOS.....	541
8.8.2.1	Wiederaufsetzen nach SERUPRO-Suchziel gefunden .....	541
8.8.2.2	Wiederanfahren an Kontur mit gesteuertem REPOS .....	551
8.8.3	Beschleunigungsmaßnahmen über MD .....	554
8.8.4	SERUPRO-ASUP .....	555
8.8.5	Self-Acting SERUPRO .....	559
8.8.6	Programmabschnitt für Wiederaufsetzen sperren .....	561
8.8.7	Besonderheiten im Zielsatz des Teileprogramms.....	565
8.8.7.1	STOPRE im Zielsatz des Teileprogramms .....	565
8.8.7.2	SPOS im Zielsatz.....	566
8.8.8	Verhalten bei Power On, Betriebsartenwechsel und RESET .....	566
8.8.9	Randbedingungen.....	567
8.8.9.1	Fahren auf Festanschlag (FXS).....	567
8.8.9.2	Fahren mit begrenztem Moment/Kraft (FOC).....	567
8.8.9.3	Synchronspindel .....	568
8.8.9.4	Kopplungen und Master-Slave.....	568
8.8.9.5	Achsfunktionen.....	572
8.8.9.6	Getriebestufenwechsel .....	573
8.8.9.7	Überlagerte Bewegung .....	574
8.8.9.8	NC/PLC-Nahtstellensignale .....	574
8.8.9.9	Flexibilisierung der Grundeinstellung.....	575
8.8.10	Systemvariable .....	576
8.9	Programmbetrieb .....	577
8.9.1	Grundstellungen.....	577
8.9.2	Anwahl und Starten des Teileprogramms bzw. Teileprogrammsatzes .....	581
8.9.3	Teileprogrammunterbrechung.....	582
8.9.4	RESET-Kommando .....	584
8.9.5	Programmmzustand.....	585
8.9.6	Kanalzustand .....	586
8.9.7	Reaktionen auf Bedienungs- oder Programmaktionen.....	588
8.9.8	Teileprogramm starten.....	589
8.9.9	Zeitdiagramm-Beispiel für einen Programmablauf .....	590
8.9.10	Programmsprünge .....	591
8.9.10.1	Rücksprung auf Programmanfang .....	591
8.9.11	Programmteilwiederholungen .....	594
8.9.11.1	Übersicht.....	594
8.9.11.2	Einzelner Teileprogrammsatz .....	595
8.9.11.3	Teileprogrammbereich ab einem Start-Label .....	596
8.9.11.4	Teileprogrammbereich zwischen einem Start-Label und End-Label .....	597
8.9.11.5	Teileprogrammbereich zwischen einem Start-Label und dem Schlüsselwort: ENDLABEL .....	598
8.9.12	Ereignisgesteuerte Programmaufrufe.....	599
8.9.12.1	Funktion .....	599
8.9.12.2	Parametrierung .....	603
8.9.12.3	Programmierung .....	609
8.9.12.4	Randbedingungen.....	610
8.9.12.5	Beispiele.....	610
8.9.13	Beeinflussung von Stopp-Ereignissen durch Stopp-Delay-Bereiche.....	612
8.10	Asynchrone Unterprogramme (ASUPs), Interruptroutinen .....	616
8.10.1	Funktion .....	616
8.10.1.1	Allgemeine Funktionalität.....	616
8.10.1.2	Ablauf einer Interruptroutine im Programmbetrieb.....	619

8.10.1.3	Interruptroutine mit REPOSA .....	620
8.10.1.4	NC-Verhalten .....	621
8.10.2	Parametrierung .....	622
8.10.3	Programmierung .....	626
8.10.4	Randbedingungen .....	628
8.10.5	Beispiele .....	629
8.11	Anwenderspezifisches ASUP für RET und REPOS .....	630
8.11.1	Funktion.....	630
8.11.2	Parametrierung .....	630
8.11.3	Programmierung .....	632
8.12	Einzelatz .....	633
8.12.1	Decodier-Einzelatz SBL2 mit impliziten Vorlaufstop .....	634
8.12.2	Einzelatzstopp: Unterdrückung über SBLOF .....	634
8.12.3	Einzelatzstopp: situationsabhängig verhindern.....	637
8.12.4	Einzelatzverhalten in BAG mit Typ A/B.....	639
8.13	Programmbeeinflussung .....	640
8.13.1	Funktionenwahl über Bedientafelfront oder über PLC.....	640
8.13.2	Aktivierung von Ausblendebenen .....	641
8.13.3	Größenanpassung des Interpolationspuffers.....	643
8.13.4	Darstellungsweise der Programmanzeige über eine zusätzliche Basis-Satzanzeige .....	645
8.13.5	Basis-Satzanzeige bei ShopMill/ShopTurn.....	646
8.13.6	Aufbau für einen DIN-Satz .....	648
8.13.7	Abarbeiten von Extern.....	652
8.13.8	Abarbeiten von externen Unterprogrammen.....	654
8.14	Systemeinstellungen für Hochlauf, RESET/Teileprogrammende und Teileprogramm-Start....	658
8.14.1	Werkzeugrückzug nach POWER ON mit Orientierungstransformation.....	663
8.14.2	Werkzeugrückzug in Unterbetriebsart JOG-Retract .....	665
8.15	Ersetzung von Funktionen durch Unterprogramme .....	670
8.15.1	Übersicht .....	670
8.15.2	Ersetzung von M-, T/TCA- und D/DL-Funktionen .....	671
8.15.2.1	Ersetzung von M-Funktionen .....	671
8.15.2.2	Ersetzung von T/TCA- und D/DL-Funktionen .....	674
8.15.2.3	Systemvariable.....	676
8.15.2.4	Beispiel: Ersetzung einer M-Funktion .....	678
8.15.2.5	Beispiel: Ersetzung einer T- und D-Funktion .....	681
8.15.2.6	Verhalten im Konfliktfall.....	682
8.15.3	Ersetzung von Spindel-Funktionen .....	683
8.15.3.1	Allgemeines.....	683
8.15.3.2	Ersetzung von M40 - M45 (Getriebestufenwechsel).....	685
8.15.3.3	Ersetzung von SPOS, SPOSA, M19 (Spindelpositionieren).....	686
8.15.3.4	Systemvariable.....	687
8.15.3.5	Beispiel: Getriebestufenwechsel.....	688
8.15.3.6	Beispiel: Spindelpositionieren .....	690
8.15.4	Eigenschaften der Unterprogramme .....	694
8.15.5	Randbedingungen .....	696
8.16	Programmlaufzeit / Werkstückzähler .....	697
8.16.1	Programmlaufzeit .....	697
8.16.2	Werkstückzähler.....	705
8.17	Datenlisten .....	710



8.17.1	Maschinendaten.....	710
8.17.1.1	Allgemeine Maschinendaten.....	710
8.17.1.2	Kanal-spezifische Maschinendaten .....	711
8.17.1.3	Achs-/Spindel-spezifische Maschinendaten .....	715
8.17.2	Settingdaten.....	715
8.17.2.1	Kanal-spezifische Settingdaten.....	715
8.17.3	Signale .....	716
8.17.3.1	Signale an NC.....	716
8.17.3.2	Signale an BAG .....	716
8.17.3.3	Signale von BAG.....	716
8.17.3.4	Signale an Kanal.....	717
8.17.3.5	Signale von Kanal.....	717
8.17.3.6	Signale an Achse/Spindel.....	719
8.17.3.7	Signale von Achse/Spindel .....	719
<b>9</b>	<b>K2: Achsen, Koordinatensysteme, Frames.....</b>	<b>721</b>
9.1	Kurzbeschreibung .....	721
9.1.1	Achsen .....	721
9.1.2	Koordinatensysteme .....	723
9.1.3	Frames .....	725
9.2	Achsen .....	729
9.2.1	Übersicht.....	729
9.2.2	Maschinenachsen .....	730
9.2.3	Kanalachsen .....	731
9.2.4	Geometrieachsen.....	731
9.2.5	Umschaltbare Geometrieachsen .....	732
9.2.6	Zusatzachsen.....	737
9.2.7	Bahnachsen .....	737
9.2.8	Positionierachsen.....	738
9.2.9	Hauptlaufachsen .....	739
9.2.10	Synchronachsen .....	740
9.2.11	Achskonfiguration .....	742
9.2.12	Link-Achsen .....	745
9.3	Nullpunkte und Referenzpunkte.....	746
9.3.1	Bezugspunkte im Arbeitsraum.....	746
9.3.2	Lage der Koordinatensysteme und Referenzpunkte .....	748
9.4	Koordinatensysteme .....	749
9.4.1	Übersicht.....	749
9.4.2	Maschinenkoordinatensystem (MKS) .....	752
9.4.3	Basiskoordinatensystem (BKS) .....	754
9.4.4	Additive Korrekturen .....	756
9.4.5	Basis-Nullpunktsystem (BNS).....	759
9.4.6	Einstellbares Nullpunktsystem (ENS) .....	761
9.4.7	Werkstückkoordinatensystem (WKS) .....	763
9.5	Frames .....	764
9.5.1	Frame-Arten .....	764
9.5.2	Frame-Komponenten .....	765
9.5.2.1	Translation .....	765
9.5.2.2	Feinverschiebung.....	766
9.5.2.3	Drehung: Übersicht (nur Geometrieachsen).....	767

9.5.2.4	Drehung mit Euler-Winkeln: ZY'X"-Konvention (RPY-Winkel).....	768
9.5.2.5	Drehung mit Euler-Winkeln: ZX'Z"-Konvention .....	773
9.5.2.6	Drehung in beliebiger Ebene.....	775
9.5.2.7	Skalierung .....	776
9.5.2.8	Spiegelung .....	777
9.5.2.9	Verkettungsoperator.....	777
9.5.2.10	Programmierbare Achsbezeichner .....	777
9.5.2.11	Koordinatentransformation.....	779
9.5.3	Frames in der Datenhaltung und aktive Frames.....	779
9.5.3.1	Übersicht .....	779
9.5.3.2	Aktivierung von Datenhaltungsframes .....	781
9.5.3.3	NCU-Globale Frames.....	783
9.5.4	Framekette und Koordinatensysteme .....	783
9.5.4.1	Übersicht .....	783
9.5.4.2	Relative Koordinatensysteme .....	785
9.5.4.3	Projektierbares ENS.....	787
9.5.4.4	Handverfahren im ENS-Koordinatensystem .....	788
9.5.4.5	Unterdrückung von Frames.....	789
9.5.5	Frames der Framekette.....	791
9.5.5.1	Übersicht .....	791
9.5.5.2	Einstellbare Frames \$P_UIFR[n] .....	791
9.5.5.3	Kanal Basisframes \$P_CHBFR[n] .....	792
9.5.5.4	NCU-globale Basisframes \$P_NCBFR[n] .....	793
9.5.5.5	Gesamt-Basisframe \$P_ACTBFRAME .....	795
9.5.5.6	Programmierbarer Frame \$P_PFRAME .....	796
9.5.5.7	Kanalspezifische Systemframes .....	799
9.5.6	Implizite Frame-Änderungen.....	803
9.5.6.1	Frames beim Umschalten von Geometrieachsen.....	803
9.5.6.2	Frames bei An- und Abwahl von Transformationen .....	807
9.5.6.3	Adaptionen von aktiven Frames .....	829
9.5.6.4	Mapped Frames .....	830
9.5.7	Vordefinierte Frame-Funktionen .....	834
9.5.7.1	Inverses Frame .....	834
9.5.7.2	Additives Frame in der Framekette.....	838
9.5.8	Funktionen .....	840
9.5.8.1	Setzen von Nullpunkten, Werkstück- und Werkzeugvermessung .....	840
9.5.8.2	Externe Nullpunktverschiebung über Systemframe.....	840
9.5.8.3	Werkzeugträger.....	842
9.5.9	Unterprogramme mit Attribut SAVE .....	855
9.5.10	Datensicherung .....	856
9.5.11	Positionen in den Koordinatensystemen.....	857
9.5.12	Steuerungsverhalten .....	858
9.5.12.1	POWER ON .....	858
9.5.12.2	Betriebsartenwechsel.....	858
9.5.12.3	RESET, Teileprogrammende .....	859
9.5.12.4	Teileprogrammstart.....	862
9.5.12.5	Satzsuchlauf.....	862
9.5.12.6	REPOS.....	862
9.6	Werkstücknahes Istwertsystem.....	863
9.6.1	Übersicht .....	863
9.6.2	Benutzung des werkstücknahen Istwertsystems .....	863
9.6.3	Besondere Reaktionen.....	865

9.7	Randbedingungen.....	868
9.8	Beispiele.....	869
9.8.1	Achsen .....	869
9.8.2	Koordinatensysteme .....	872
9.8.3	Frames .....	873
9.9	Datenlisten .....	876
9.9.1	Maschinendaten.....	876
9.9.1.1	Anzeige-Maschinendaten .....	876
9.9.1.2	NC-spezifischen Maschinendaten .....	877
9.9.1.3	Kanal-spezifische Maschinendaten .....	877
9.9.1.4	Achs-/Spindel-spezifische Maschinendaten .....	878
9.9.2	Settingdaten.....	878
9.9.2.1	Kanal-spezifische Settingdaten.....	878
9.9.3	Systemvariablen .....	879
9.9.4	Signale .....	881
9.9.4.1	Signale von Kanal .....	881
9.9.4.2	Signale an Achse/Spindel.....	881
9.9.4.3	Signale von Achse/Spindel .....	881
<b>10</b>	<b>N2: Not-Halt.....</b>	<b>883</b>
10.1	Kurzbeschreibung .....	883
10.2	Normen .....	884
10.3	Not-Halt-Stellteile .....	886
10.4	Not-Halt-Ablauf.....	887
10.5	Not-Halt-Quittierung .....	889
10.6	Datenlisten .....	891
10.6.1	Maschinendaten.....	891
10.6.1.1	Achs-/Spindel-spezifische Maschinendaten .....	891
10.6.2	Signale .....	891
10.6.2.1	Signale an NC.....	891
10.6.2.2	Signale von NC .....	891
10.6.2.3	Signale an BAG .....	891
<b>11</b>	<b>P1: Planachsen .....</b>	<b>893</b>
11.1	Kurzbeschreibung .....	893
11.2	Geometrieachse als Planachse definieren .....	896
11.3	Maßangaben von Planachsen .....	898
11.4	Datenlisten .....	905
11.4.1	Maschinendaten.....	905
11.4.1.1	Kanal-spezifische Maschinendaten .....	905
11.4.1.2	Achs-/Spindel-spezifische Maschinendaten .....	905
<b>12</b>	<b>P3: PLC-Grundprogramm für SINUMERIK 840D sl.....</b>	<b>907</b>
12.1	Kurzbeschreibung .....	907
12.2	Eckdaten der PLC-CPU .....	909
12.3	PLC-Betriebssystemversion.....	910

12.4	PLC-Betriebsartenschalter .....	911
12.5	Ressourcen (Timer, Zähler, FC, FB, DB, Peripherie) reservieren .....	912
12.6	Inbetriebnahme Hardware-Konfiguration der PLC-CPU .....	913
12.7	Inbetriebnahme PLC-Programm .....	914
12.7.1	Installation des Grundprogramms .....	914
12.7.2	Anwendung des Grundprogramms .....	914
12.7.3	Versionskennzeichnungen .....	915
12.7.4	Maschinenprogramm .....	916
12.7.5	Datensicherung .....	916
12.7.6	PLC-Serieninbetriebnahme, PLC Archive .....	917
12.7.7	Software-Hochrüstung .....	920
12.7.8	Peripheriebaugruppen (FM-, CP-Baugruppen) .....	920
12.7.9	Fehler-Beseitigung .....	921
12.8	Ankopplung der PLC-CPU .....	922
12.8.1	Allgemeines .....	922
12.8.2	Eigenschaften der PLC-CPU .....	922
12.8.3	Nahtstelle mit integrierter PLC .....	922
12.8.4	Diagnosepuffer der PLC .....	925
12.9	Struktur der Nahtstelle .....	926
12.9.1	Nahtstelle PLC/NCK .....	926
12.9.2	Nahtstelle PLC/HMI .....	933
12.9.3	Nahtstelle PLC/MCP/BHG .....	938
12.10	Struktur und Funktionen des Grundprogramms .....	941
12.10.1	Anlauf und Synchronisation NCK-PLC .....	942
12.10.2	Zyklischer Betrieb (OB 1) .....	943
12.10.3	Zeitalarm Bearbeitung (OB 35) .....	945
12.10.4	Prozessalarm Bearbeitung (OB 40) .....	946
12.10.5	Diagnosealarm, Baugruppenausfall Bearbeitung (OB 82, OB 86) .....	946
12.10.6	Verhalten bei NCK-Ausfall .....	947
12.10.7	Funktionen des Grundprogramms mit Aufruf vom Anwenderprogramm .....	949
12.10.8	Symbolische Programmierung des Anwenderprogramms mit Nahtstellen-DB .....	952
12.10.9	M-Dekodierung nach Liste .....	953
12.10.10	PLC-Maschinendaten .....	957
12.10.11	Projektierung von Maschinensteuertafel, Bedienhandgerät, Direkttasten .....	962
12.10.12	Umschaltung von Maschinensteuertafel, Bedienhandgerät .....	972
12.11	SPL für Safety Integrated .....	975
12.12	Belegungsübersicht .....	976
12.12.1	Belegung: NCK/PLC-Nahtstelle .....	976
12.12.2	Belegung: FB/FC .....	976
12.12.3	Belegung: DB .....	976
12.12.4	Belegung: Timer .....	978
12.13	Speicherbedarf des PLC-Grundprogramms .....	979
12.14	Rahmenbedingungen und NC-VAR-Selector .....	983
12.14.1	Rahmenbedingungen .....	983
12.14.1.1	Programmier- und Parametrierwerkzeuge .....	983
12.14.1.2	Notwendige SIMATIC-Dokumentation .....	985
12.14.1.3	Relevante SINUMERIK-Dokumente .....	985

12.14.2	NC-VAR-Selector .....	986
12.14.2.1	Übersicht .....	986
12.14.2.2	Funktionsbeschreibung .....	989
12.14.2.3	Inbetriebnahme, Installation .....	998
12.15	Bausteinbeschreibungen .....	999
12.15.1	FB 1: RUN_UP Grundprogramm, Anlaufteil .....	999
12.15.2	FB 2: GET NC-Variable lesen .....	1008
12.15.3	FB 3: PUT NC-Variable schreiben .....	1016
12.15.4	PI-Dienste .....	1025
12.15.4.1	FB 4: PI_SERV PI-Dienst anfordern .....	1025
12.15.4.2	Liste verfügbarer PI-Dienste .....	1028
12.15.4.3	PI-Dienst: ASUP .....	1029
12.15.4.4	PI-Dienst: CANCEL .....	1030
12.15.4.5	PI-Dienst: CONFIG .....	1031
12.15.4.6	PI-Dienst: DIGION .....	1031
12.15.4.7	PI-Dienst: DIGIOF .....	1031
12.15.4.8	PI-Dienst: FINDBL .....	1032
12.15.4.9	PI-Dienst: LOGIN .....	1032
12.15.4.10	PI-Dienst: LOGOUT .....	1033
12.15.4.11	PI-Dienst: NCRES .....	1033
12.15.4.12	PI-Dienst: SELECT .....	1033
12.15.4.13	PI-Dienst: SETUDT .....	1034
12.15.4.14	PI-Dienst: SETUFR .....	1034
12.15.4.15	PI-Dienst: RETRAC .....	1035
12.15.4.16	PI-Dienst: CRCEDN .....	1035
12.15.4.17	PI-Dienst: CREACE .....	1036
12.15.4.18	PI-Dienst: CREATO .....	1037
12.15.4.19	PI-Dienst: DELECE .....	1037
12.15.4.20	PI-Dienst: DELETO .....	1038
12.15.4.21	PI-Dienst: MMCSEM .....	1038
12.15.4.22	PI-Dienst: TMCRTO .....	1039
12.15.4.23	PI-Dienst: TMFDPL .....	1040
12.15.4.24	PI-Dienst: TMFPBP .....	1042
12.15.4.25	PI-Dienst: TMGETT .....	1043
12.15.4.26	PI-Dienst: TMMVTL .....	1044
12.15.4.27	PI-Dienst: TMPOSM .....	1046
12.15.4.28	PI-Dienst: TMPCIT .....	1047
12.15.4.29	PI-Dienst: TMRASS .....	1047
12.15.4.30	PI-Dienst: TRESMO .....	1048
12.15.4.31	PI-Dienst: TSEARC .....	1049
12.15.4.32	PI-Dienst: TMCRMT .....	1052
12.15.4.33	PI-Dienst: TMDLMT .....	1053
12.15.4.34	PI-Dienst: POSMT .....	1053
12.15.4.35	PI-Dienst: FDPLMT .....	1054
12.15.5	FB 5: GETGUD GUD-Variable lesen .....	1057
12.15.6	FB 7: PI_SERV2 (PI-Dienst anfordern) .....	1065
12.15.7	FB 9: MzuN Bedieneinheitenumschaltung .....	1066
12.15.8	FB 10: Sicherheits-Relais (SI-Relais) .....	1072
12.15.9	FB 11: Bremsentest .....	1075
12.15.10	FB 29: Diagnose Signalrekorder und Datentrigger .....	1081
12.15.11	FC 2: GP_HP Grundprogramm, zyklischer Teil .....	1085
12.15.12	FC 3: GP_PRAL Grundprogramm, alarmgesteuerter Teil .....	1087

12.15.13	FC 5: GP_DIAG Grundprogramm, Diagnosealarm und Baugruppenausfall .....	1090
12.15.14	FC 6: TM_TRANS2 Transfer-Baustein für Werkzeugverwaltung und Multitool.....	1092
12.15.15	FC 7: TM_REV Transfer-Baustein für Werkzeugwechsel mit Revolver .....	1093
12.15.16	FC 8: TM_TRANS Transfer-Baustein für Werkzeugverwaltung .....	1097
12.15.17	FC 9: ASUP Start von asynchronen Unterprogrammen .....	1104
12.15.18	FC 10: AL_MSG Fehler- und Betriebsmeldungen .....	1107
12.15.19	FC 12: AUXFU Aufrufschnittstelle für Anwender bei Hilfsfunktionen .....	1109
12.15.20	FC 13: BHGDisp Displaysteuerung für Bedienhandgerät.....	1111
12.15.21	FC 17: YDelta Stern-Dreieck-Umschaltung .....	1116
12.15.22	FC 18: SpinCtrl Spindelsteuerung .....	1119
12.15.23	FC 19: MCP_IFM Übertragung der MCP-Signale an die Nahtstelle.....	1131
12.15.24	FC 21: Transfer Datenaustausch PLC-NCK.....	1138
12.15.25	FC 22: TM_DIR Richtungsauswahl für Werkzeugverwaltung.....	1146
12.15.26	FC 24: MCP_IFM2 Übertragung der MCP-Signale an die Nahtstelle.....	1149
12.15.27	FC 25: MCP_IFT Übertragung der MCP-/BT-Signale an die Nahtstelle.....	1153
12.15.28	FC 26: HPU_MCP Übertragung der HT8-Signale an die Nahtstelle.....	1156
12.15.28.1	Übersicht der NC/PLC-Nahtstellensignale von HT 8 .....	1162
12.15.28.2	Übersicht der NC/PLC-Nahtstellensignale an HT 8.....	1164
12.15.29	FC 19, FC 24, FC 25, FC 26 Beschreibung Quellcode .....	1164
12.15.30	FC 1005: AG_SEND Übergibt Daten an Ethernet-CP .....	1166
12.15.31	FC 1006: AG_RECV Empfängt Daten von Ethernet-CP .....	1168
12.16	Signal-/Datenbeschreibungen.....	1169
12.16.1	Nahtstellensignale NCK/PLC, HMI/PLC, MSTT/PLC .....	1169
12.16.2	Dekodierte M-Signale.....	1169
12.16.3	G-Funktionen .....	1170
12.16.4	Meldesignale im DB 2 .....	1172
12.17	Programmiertipps mit STEP 7.....	1173
12.17.1	Daten kopieren.....	1173
12.17.2	ANY und POINTER.....	1174
12.17.2.1	Verwendung von POINTER und ANY im FC.....	1174
12.17.2.2	Verwendung von POINTER und ANY im FB .....	1175
12.17.2.3	Variable POINTER bzw. ANY für Übergabe an FC oder FB .....	1177
12.17.3	Multi-Instanz-DB.....	1179
12.17.4	Strings .....	1180
12.17.5	Ermittlung von Offsetadressen auf Datenbaustein-Strukturen .....	1181
12.17.6	Aufrufe von FBs .....	1182
12.18	Datenlisten .....	1184
12.18.1	Maschinendaten.....	1184
12.18.1.1	NC-spezifische Maschinendaten .....	1184
12.18.1.2	Kanal-spezifische Maschinendaten .....	1184
<b>13</b>	<b>P4: PLC für SINUMERIK 828D.....</b>	<b>1185</b>
13.1	Überblick .....	1185
13.1.1	PLC-Firmware .....	1185
13.1.2	PLC-Anwendernahtstelle .....	1186
13.1.2.1	Zyklisch ausgetauschte Daten .....	1188
13.1.2.2	Alarmer und Meldungen.....	1188
13.1.2.3	Remanente Daten .....	1189
13.1.2.4	Nichtremanente Daten .....	1189
13.1.2.5	PLC-Maschinendaten.....	1189
13.1.3	PLC-Eckdaten .....	1190

13.1.4	PLC Peripherie, schnelle Onboard-Ein-/Ausgänge .....	1190
13.1.5	PLC-Toolbox .....	1191
13.1.5.1	Stern/Dreieck-Umschaltung .....	1191
13.2	Programming Tool PLC828 .....	1192
13.3	Programmierung .....	1193
13.3.1	Einführung.....	1193
13.3.1.1	Wichtige Begriffe.....	1193
13.3.1.2	Projekt anlegen/öffnen .....	1196
13.3.1.3	Programmorganisation durch das Programming Tool.....	1197
13.3.1.4	Schnelle Onboard-Ein- und Ausgänge .....	1199
13.3.2	Speicher des Zielsystems .....	1201
13.3.2.1	Speicherart.....	1201
13.3.2.2	Adressbereiche des Zielsystems .....	1201
13.3.2.3	Adressierung.....	1204
13.3.2.4	Datentypen.....	1207
13.3.2.5	Konstanten.....	1210
13.3.2.6	Datenbausteine.....	1211
13.3.2.7	Sondermerker und ihre Funktionen .....	1225
13.3.3	Operationsumfang .....	1226
13.3.4	Datenklassen .....	1226
13.3.4.1	Definition der Datenklassen.....	1226
13.3.4.2	Baustein einer Datenklasse zuordnen .....	1230
13.3.4.3	Datenklasse(n) laden in CPU.....	1231
13.3.4.4	Datenklasse(n) laden aus CPU.....	1233
13.3.4.5	Vergleich zwischen Offline- und Online-Projekt.....	1234
13.3.4.6	Löschen im Zielsystem .....	1236
13.3.5	Operanden umverdrahten.....	1236
13.4	Test- und Beobachtungsfunktionen .....	1240
13.4.1	Programmstatus.....	1240
13.4.1.1	Statusdefinition.....	1240
13.4.1.2	Voraussetzungen der Statusaktualisierung .....	1242
13.4.1.3	Einfluss des Betriebszustandes des Zielsystems .....	1242
13.4.1.4	Kommunikation und Zyklus.....	1243
13.4.1.5	Status aktualisieren.....	1243
13.4.1.6	Simulieren von Prozessbedingungen .....	1244
13.4.1.7	Prüfen von Querverweisen und verwendeten Elementen .....	1244
13.4.2	Programmstatus im KOP-Programmeditor .....	1245
13.4.2.1	Programmstatus anzeigen .....	1245
13.4.2.2	Eigenschaften der Anzeige .....	1247
13.4.2.3	Einschränkungen .....	1247
13.4.2.4	Anzeige des Programmstatus anpassen .....	1249
13.4.3	Anzeigen des Status in einer Statustabelle .....	1250
13.4.3.1	Eigenschaften einer Statustabelle .....	1250
13.4.3.2	Statustabelle öffnen .....	1252
13.4.3.3	Mit mehreren Statustabellen arbeiten.....	1252
13.4.3.4	Statustabelle anlegen .....	1253
13.4.3.5	Statustabelle bearbeiten .....	1254
13.4.3.6	Datenformate .....	1256
13.4.3.7	Statustabelle einschalten .....	1257
13.4.3.8	Arbeiten mit Testfunktionen in der Statustabelle .....	1257
13.4.4	Zyklen ausführen .....	1258

13.5	Datenschnittstelle .....	1259
13.5.1	Nahtstelle PLC-NCK .....	1259
13.5.1.1	Betriebsartensignale .....	1260
13.5.1.2	NC-Kanal-Signale .....	1261
13.5.1.3	Achs- und Spindelsignale.....	1262
13.5.1.4	Allgemeine NCK-Signale.....	1263
13.5.1.5	Schneller Datenaustausch PLC-NCK .....	1264
13.5.2	Nahtstelle PLC-HMI .....	1264
13.6	Funktionsschnittstelle.....	1265
13.6.1	NC-Variable lesen/schreiben .....	1265
13.6.1.1	Anwenderschnittstelle .....	1265
13.6.1.2	Auftragsspezifizierung.....	1265
13.6.1.3	Auftragsmanagement: Auftrag starten .....	1267
13.6.1.4	Auftragsmanagement: Warten auf Auftragsende.....	1268
13.6.1.5	Auftragsmanagement: Auftragsabschluss .....	1268
13.6.1.6	Auftragsmanagement: Ablaufdiagramm .....	1269
13.6.1.7	Auftragsauswertung .....	1270
13.6.1.8	Bedienbare Variablen.....	1271
13.6.2	Programminstanz-Dienste (PI-Dienste) .....	1279
13.6.2.1	Anwenderschnittstelle .....	1279
13.6.2.2	PI-Dienste.....	1281
13.6.3	PLC-Anwenderalarme.....	1284
13.6.3.1	Anwenderschnittstelle .....	1284
13.6.4	PLC-Achssteuerung .....	1288
13.6.4.1	Überblick .....	1288
13.6.4.2	Anwenderschnittstelle: Vorbereiten der NC-Achse als PLC-Achse.....	1290
13.6.4.3	Anwenderschnittstelle: Funktionalität.....	1292
13.6.4.4	Spindel positionieren.....	1293
13.6.4.5	Spindel drehen .....	1295
13.6.4.6	Spindel pendeln .....	1297
13.6.4.7	Teilungsachse .....	1299
13.6.4.8	Positionierachse metrisch .....	1301
13.6.4.9	Positionierachse Inch .....	1303
13.6.4.10	Positionierachse metrisch mit Handradüberlagerung .....	1305
13.6.4.11	Positionierachse Inch mit Handradüberlagerung.....	1307
13.6.4.12	Spindel drehen mit automatischer Getriebestufenwahl .....	1309
13.6.4.13	Spindel drehen mit konstanter Schnittgeschwindigkeit [m/min].....	1311
13.6.4.14	Spindel drehen mit konstanter Schnittgeschwindigkeit [feet/min].....	1313
13.6.4.15	Fehlermeldungen .....	1315
13.6.5	Starten von ASUPs .....	1316
13.6.5.1	Allgemeines.....	1316
13.6.5.2	Auftragsstart.....	1317
13.6.5.3	Auftragsergebnis .....	1317
13.6.5.4	Signalfluss .....	1318
<b>14</b>	<b>R1: Referenzieren.....</b>	<b>1319</b>
14.1	Kurzbeschreibung .....	1319
14.2	Achsspezifisches Referenzieren.....	1321
14.3	Kanalspezifisches Referenzieren.....	1324
14.4	Referenzpunktfahren aus dem Teileprogramm (G74) .....	1326



14.5	Referenzieren bei inkrementellen Messsystemen .....	1327
14.5.1	Hardware-Signale .....	1327
14.5.2	Nullmarkenauswahl mit BERO.....	1328
14.5.3	Zeitlicher Ablauf .....	1330
14.5.4	Phase 1: Fahren auf den Referenznocken .....	1331
14.5.5	Phase 2: Synchronisation mit der Nullmarke.....	1334
14.5.6	Phase 3: Fahren auf den Referenzpunkt.....	1339
14.6	Referenzieren bei abstandscodierten Referenzmarken .....	1342
14.6.1	Allgemeiner Überblick .....	1342
14.6.2	Grundlegende Parametrierung .....	1343
14.6.3	Zeitlicher Ablauf .....	1345
14.6.4	Phase 1: Überfahren der Referenzmarken mit Synchronisation .....	1346
14.6.5	Phase 2: Fahren auf Zielpunkt.....	1348
14.7	Referenzieren über Istwertabgleich .....	1351
14.7.1	Istwertabgleich auf das referenzierende Messsystem.....	1351
14.7.2	Istwertabgleich bei Messsystemen mit abstandscodierten Referenzmarken.....	1352
14.8	Referenzieren im Nachführbetrieb .....	1354
14.9	Referenzieren bei Absolutwertgebern .....	1358
14.9.1	Informationen zur Justage .....	1358
14.9.2	Justage durch Eingabe der Referenzpunktverschiebung.....	1360
14.9.3	Justage durch Eingabe des Referenzpunktwertes .....	1361
14.9.4	Automatische Justage mittels Messtaster .....	1363
14.9.5	Justage mittels BERO.....	1365
14.9.6	Referenzpunktfahren bei Absolutwertgebern .....	1366
14.9.7	Referenzpunktfahren bei rotatorischen Absolutwertgebern mit Nullmarkenersatz .....	1366
14.9.8	Automatische Gebertausch-Erkennung.....	1369
14.9.9	Einschalten des Messsystems.....	1370
14.9.10	Nicht unterstützte Referenziervarianten .....	1372
14.10	Automatisches Wiederherstellen des Maschinenbezugs .....	1373
14.10.1	Automatisches Referenzieren.....	1374
14.10.2	Restaurieren der Istposition .....	1375
14.11	Randbedingungen.....	1378
14.11.1	Große Verfahrbereiche .....	1378
14.12	Datenlisten .....	1380
14.12.1	Maschinendaten.....	1380
14.12.1.1	NC-spezifische Maschinendaten .....	1380
14.12.1.2	Kanal-spezifische Maschinendaten .....	1380
14.12.1.3	Achs-/Spindel-spezifische Maschinendaten .....	1380
14.12.2	Signale .....	1381
14.12.2.1	Signale an BAG .....	1381
14.12.2.2	Signale von BAG.....	1382
14.12.2.3	Signale an Kanal.....	1382
14.12.2.4	Signale von Kanal.....	1382
14.12.2.5	Signale an Achse/Spindel.....	1382
14.12.2.6	Signale von Achse/Spindel.....	1382
<b>15</b>	<b>S1: Spindeln .....</b>	<b>1383</b>
15.1	Kurzbeschreibung .....	1383

15.2	Betriebsarten .....	1384
15.2.1	Übersicht .....	1384
15.2.2	Betriebsartenwechsel .....	1385
15.2.3	Steuerbetrieb .....	1386
15.2.4	Pendelbetrieb .....	1390
15.2.5	Positionierbetrieb .....	1390
15.2.5.1	Allgemeine Funktionalität .....	1390
15.2.5.2	Positionieren aus der Drehung .....	1397
15.2.5.3	Positionieren aus dem Stillstand .....	1402
15.2.5.4	Signal "Spindel in Position" für Werkzeugwechsel .....	1406
15.2.6	Achsbetrieb .....	1407
15.2.6.1	Allgemeine Funktionalität .....	1407
15.2.6.2	Impliziter Übergang in den Achsbetrieb .....	1411
15.2.7	Spindel-Grundstellung .....	1415
15.3	Referenzieren / Synchronisieren .....	1416
15.4	Projektierbare Getriebeanpassungen .....	1421
15.4.1	Getriebestufen für Spindeln und Getriebestufenwechsel .....	1421
15.4.2	Spindelgetriebestufe 0 .....	1433
15.4.3	Ermittlung der Spindelgetriebestufe .....	1436
15.4.4	Parametersatzanwahl beim Getriebestufenwechsel .....	1437
15.4.5	Vorsatz-Getriebe .....	1440
15.4.6	Nicht quittierter Getriebestufenwechsel .....	1441
15.4.7	Getriebestufenwechsel mit Pendelbetrieb .....	1443
15.4.8	Getriebestufenwechsel auf Festposition .....	1449
15.4.9	Projektierbare Getriebestufe bei M70 .....	1456
15.4.10	Unterdrückung des Getriebestufenwechsels bei DryRun, Programmtest und SERUPRO ....	1458
15.5	Weitere projektierbare Anpassungen der Spindelfunktionalität .....	1461
15.6	Umschaltbare Spindeln .....	1463
15.7	Programmierung .....	1467
15.7.1	Programmierung aus dem Teileprogramm .....	1467
15.7.2	Programmierung über Synchronaktionen .....	1471
15.7.3	Programmierung von Spindelsteuerungen über PLC mit FC18 - nur 840D sl .....	1471
15.7.4	Spezielle Spindelbewegungen über PLC-Schnittstelle .....	1472
15.7.5	Programmierung von extern (PLC, HMI) .....	1478
15.8	Spindelüberwachungen .....	1479
15.8.1	Zulässige Drehzahlbereiche .....	1479
15.8.2	Achse/Spindel steht .....	1480
15.8.3	Spindel im Sollbereich .....	1480
15.8.4	Minimale/Maximale Drehzahl der Getriebestufe .....	1482
15.8.5	Diagnose von Spindeldrehzahlbegrenzungen .....	1483
15.8.6	Maximale Drehzahl der Spindel .....	1485
15.8.7	Maximale Gebergrenzfrequenz .....	1486
15.8.8	Zielpunktüberwachung .....	1488
15.8.9	M40: Automatische Getriebestufenauswahl bei Drehzahlen außerhalb projektierter Schaltschwellen .....	1489
15.9	Spindel mit SMI 24 (Weiss-Spindel) .....	1492
15.9.1	Allgemeine Informationen .....	1492
15.9.2	Sensordaten .....	1493
15.9.3	Spannzustand .....	1495

15.9.4	Weitere Antriebsparameter .....	1497
15.10	Randbedingungen.....	1498
15.10.1	Ändern von Regelparametern.....	1498
15.11	Beispiele.....	1499
15.11.1	Automatische Getriebestufenanwahl (M40).....	1499
15.12	Datenlisten .....	1500
15.12.1	Maschinendaten.....	1500
15.12.1.1	NC-spezifische Maschinendaten .....	1500
15.12.1.2	Kanal-spezifische Maschinendaten .....	1500
15.12.1.3	Achs-/Spindel-spezifische Maschinendaten .....	1501
15.12.2	Settingdaten.....	1503
15.12.2.1	Kanal-spezifische Settingdaten .....	1503
15.12.2.2	Achs-/Spindel-spezifische Settingdaten .....	1503
15.12.3	Signale .....	1504
15.12.3.1	Signale an Achse/Spindel.....	1504
15.12.3.2	Signale von Achse/Spindel.....	1505
<b>16</b>	<b>V1: Vorschübe .....</b>	<b>1507</b>
16.1	Kurzbeschreibung .....	1507
16.2	Bahnvorschub F .....	1509
16.2.1	Vorschubart G93, G94, G95 .....	1511
16.2.2	Vorschubart G96, G961, G962, G97, G971 .....	1514
16.2.3	Vorschub bei Gewindschneiden (G33, G34, G35) .....	1518
16.2.3.1	Programmierbarer Einlauf- und Auslaufweg bei G33, G34 und G35 .....	1520
16.2.3.2	Linear progressive/degressive Gewindesteigungsänderung bei G34 und G35 .....	1522
16.2.3.3	Schnellrückzug während Gewindecneiden .....	1524
16.2.4	Vorschub bei Gewindebohren ohne Ausgleichsfutter (G331, G332).....	1529
16.2.5	Vorschub bei Gewindebohren mit Ausgleichsfutter (G63).....	1531
16.3	Vorschub für Positionierachsen (FA) .....	1532
16.4	Vorschubbeeinflussung.....	1534
16.4.1	Vorschubsperrung und Vorschub/Spindel Halt.....	1534
16.4.2	Vorschubkorrektur über Maschinensteuertafel .....	1536
16.4.3	Programmierbare Vorschubkorrektur .....	1540
16.4.4	Probelaufvorschub .....	1541
16.4.5	Mehrere Vorschubwerte in einem Satz.....	1543
16.4.6	Feste Vorschubwerte .....	1549
16.4.7	Programmierbare Vorschubverläufe.....	1551
16.4.8	Vorschub für Fase/Rundung FRC, FRCM .....	1552
16.4.9	Satzweiser Vorschub FB.....	1555
16.4.10	Beeinflussung der Einzelachsdynamik .....	1556
16.5	Randbedingungen.....	1562
16.6	Datenlisten .....	1563
16.6.1	Maschinendaten.....	1563
16.6.1.1	NC-spezifische Maschinendaten .....	1563
16.6.1.2	Kanal-spezifische Maschinendaten .....	1564
16.6.1.3	Achs-/Spindel-spezifische Maschinendaten .....	1564
16.6.2	Settingdaten.....	1565
16.6.2.1	Kanal-spezifische Settingdaten.....	1565

16.6.2.2	Achs-/Spindel-spezifische Settingdaten.....	1565
16.6.3	Signale .....	1565
16.6.3.1	Signale an Kanal .....	1565
16.6.3.2	Signale von Kanal .....	1566
16.6.3.3	Signale an Achse/Spindel .....	1566
16.6.3.4	Signale von Achse/Spindel .....	1566
<b>17</b>	<b>W1: Werkzeugkorrektur .....</b>	<b>1567</b>
17.1	Kurzbeschreibung .....	1567
17.2	Werkzeug .....	1571
17.2.1	Allgemeines.....	1571
17.2.2	Korrekturspeicherstruktur.....	1573
17.2.3	Verrechnung der WZ-Korrektur.....	1575
17.2.4	Adresserweiterung für die NC-Adressen T und M .....	1575
17.2.5	Freie D-Nummernvergabe .....	1577
17.2.6	Korrektursatz bei Fehler im Werkzeugwechsel.....	1585
17.2.7	Definition der Wirkung der Werkzeugparameter.....	1588
17.3	Flache D-Nummern-Struktur .....	1589
17.3.1	Allgemeines.....	1589
17.3.2	Neuanlegen einer D-Nummer (Korrektursatz) .....	1589
17.3.3	D-Nummer programmieren .....	1590
17.3.4	T-Nummer programmieren.....	1593
17.3.5	M6 programmieren.....	1594
17.3.6	Programmtest.....	1594
17.3.7	Werkzeugverwaltung oder "Flache D-Nummern" .....	1595
17.4	Werkzeugschneide.....	1596
17.4.1	Allgemeines.....	1596
17.4.2	WZ-Parameter 1: Werkzeugtyp .....	1598
17.4.3	WZ-Parameter 2: Schneidenlage.....	1602
17.4.4	WZ-Parameter 3 - 5: Geometrie - Werkzeuglängen .....	1604
17.4.5	WZ-Parameter 6 - 11: Geometrie - Werkzeugform.....	1605
17.4.6	WZ-Parameter 12 - 14: Verschleiß - Werkzeuglängen .....	1607
17.4.7	WZ-Parameter 15 - 20: Verschleiß - Werkzeugform .....	1607
17.4.8	WZ-Parameter 21 - 23: Basismaß / Adaptermaß .....	1608
17.4.9	WZ-Parameter 24: Freischneidwinkel .....	1609
17.4.10	Werkzeuge mit relevanter Schneidenlage .....	1611
17.5	Werkzeugradiuskorrektur-2D (WRK) .....	1612
17.5.1	Allgemeines.....	1612
17.5.2	Anwahl der WRK (G41/G42).....	1613
17.5.3	An- und Abfahrverhalten (NORM/KONT/KONTC/KONTT).....	1614
17.5.4	Weiches An- und Abfahren .....	1620
17.5.4.1	Funktion.....	1620
17.5.4.2	Parameter.....	1621
17.5.4.3	Geschwindigkeiten .....	1630
17.5.4.4	Systemvariablen.....	1632
17.5.4.5	Randbedingungen.....	1632
17.5.4.6	Beispiele.....	1634
17.5.5	Abwahl der WRK (G40).....	1637
17.5.6	Korrektur an Außenecken .....	1637
17.5.7	Korrektur an Innenecken.....	1642

17.5.8	Kollisionsüberwachung und Flaschenhalserkennung.....	1644
17.5.9	Sätze mit veränderlichem Korrekturwert.....	1646
17.5.10	Werkzeugradiuskorrektur konstant halten .....	1648
17.5.11	Alarmverhalten.....	1651
17.5.12	Schnittpunktverfahren für Polynome.....	1652
17.5.13	G461/G462: Erweiterung An- / Abfahrstrategie .....	1652
17.6	Orientierbare Werkzeugträger .....	1657
17.6.1	Allgemeines.....	1657
17.6.2	Kinematische Zusammenhänge und Maschinenaufbau.....	1665
17.6.3	Schrägbearbeitung mit 3 + 2 Achsen.....	1674
17.6.4	Maschine mit drehbarem Werkstück .....	1675
17.6.5	Vorgehen bei Benutzung der orientierbaren Werkzeugträger .....	1680
17.6.6	Programmierung .....	1685
17.6.7	Randbedingungen und Steuerungsverhalten für Orientierungen .....	1686
17.7	Schneidendaten-Modifikation bei drehbaren Werkzeugen.....	1689
17.7.1	Funktion .....	1689
17.7.2	Ermittlung des Drehwinkels .....	1689
17.7.3	Schneidenlage, Schnittrichtung und Winkel bei Drehwerkzeugen .....	1690
17.7.4	Modifikationen bei der Drehung von Drehwerkzeugen.....	1692
17.7.5	Schneidenlage bei Fräs- und Bohrwerkzeugen.....	1695
17.7.6	Modifikationen bei der Drehung von Fräs- und Bohrwerkzeugen .....	1696
17.7.7	Parametrierung .....	1697
17.7.8	Programmierung .....	1698
17.7.9	Beispiel.....	1701
17.8	Inkrementell programmierte Korrekturwerte .....	1703
17.8.1	G91 Erweiterung .....	1703
17.8.2	Bearbeitung in Richtung der Werkzeugorientierung.....	1704
17.9	Werkzeuggrundorientierung.....	1707
17.10	Werkzeugkorrektur-Sonderbehandlungen.....	1712
17.10.1	Relevante Settingdaten.....	1712
17.10.2	Werkzeuglängen spiegeln (SD42900 \$SC_MIRROR_TOOL_LENGTH) .....	1713
17.10.3	Verschleißlängen spiegeln (SD42920 \$SC_WEAR_SIGN_CUTPOS) .....	1714
17.10.4	Werkzeuglänge und Ebenenwechsel (SD42940 \$SC_TOOL_LENGTH_CONST) .....	1716
17.10.5	Werkzeugtyp (SD42950 \$SC_TOOL_LENGTH_TYPE) .....	1718
17.10.6	Werkzeuglängen im WKS unter Berücksichtigung der Orientierung .....	1719
17.10.7	Werkzeuglängenoffsets in Werkzeugrichtung .....	1719
17.11	Summen- und Einrichtekorrekturen .....	1725
17.11.1	Allgemeines.....	1725
17.11.2	Funktionale Beschreibung .....	1726
17.11.3	Aktivierung .....	1729
17.11.4	Beispiele.....	1737
17.11.5	Erweiterungen der Werkzeuglängenbestimmung.....	1739
17.11.5.1	Korrekturen einsatzort-werkstückspezifisch einrechnen .....	1739
17.11.5.2	Funktionalität der einzelnen Verschleißwerte .....	1744
17.12	Mit Werkzeugumgebungen arbeiten.....	1748
17.12.1	Allgemeines.....	1748
17.12.2	Abspeichern mit TOOLENV .....	1748
17.12.3	Werkzeugumgebung löschen .....	1751
17.12.4	Wieviele und welche Umgebungen sind gespeichert? .....	1752

17.12.5	Lesen T, D, DL aus einer Werkzeugumgebung.....	1753
17.12.6	Werkzeuglängen, Werkzeuglängenkomponenten lesen .....	1754
17.12.7	Werkzeugkomponenten ändern.....	1760
17.13	Werkzeuglängen L1, L2, L3 Zuordnung: LENTOAX.....	1767
17.14	Randbedingungen.....	1771
17.14.1	Flache D-Nummern-Struktur .....	1771
17.14.2	SD42935 Erweiterungen.....	1771
17.15	Beispiele.....	1772
17.15.1	Orientierbare Werkzeugträger .....	1772
17.15.1.1	Beispiel: Orientierbare Werkzeugträger.....	1772
17.15.1.2	Beispiel Orientierbare Werkzeugträger mit drehbarem Tisch.....	1773
17.15.1.3	Beispiel Werkzeuggrundorientierung.....	1776
17.15.1.4	Korrekturen einsatzort- und werkstückspezifisch einrechnen .....	1776
17.15.2	Beispiele 3-6: Funktion SETTCOR für WZ-Umgebungen .....	1779
17.16	Datenlisten .....	1786
17.16.1	Maschinendaten.....	1786
17.16.1.1	NC-spezifische Maschinendaten .....	1786
17.16.1.2	Kanal-spezifische Maschinendaten .....	1786
17.16.1.3	Achse-/Spindel-spezifische Maschinendaten .....	1788
17.16.2	Settingdaten .....	1788
17.16.2.1	Kanal-spezifische Settingdaten.....	1788
17.16.3	Signale .....	1789
17.16.3.1	Signale von Kanal.....	1789
<b>18</b>	<b>Z1: NC/PLC-Nahtstellensignale .....</b>	<b>1791</b>
18.1	Diverse Nahtstellensignale und Funktionen (A2).....	1791
18.1.1	Signale von PLC an NC (DB10).....	1791
18.1.2	Anwahl-/Statussignale von HMI an PLC (DB10).....	1791
18.1.3	Signale von NC an PLC (DB10).....	1792
18.1.4	Signale an Bedientafelfront (DB19) .....	1796
18.1.5	Signale von Bedientafelfront (DB19).....	1803
18.1.6	Signale an Kanal (DB21, ...).....	1808
18.1.7	Signale von Kanal (DB21, ...).....	1809
18.1.8	Signale an Achse/Spindel (DB31, ...).....	1810
18.1.9	Signale von Achse/Spindel (DB31, ...).....	1822
18.2	Achsüberwachungen, Schutzbereiche (A3).....	1833
18.2.1	Signale an Kanal (DB21, ...).....	1833
18.2.2	Signale von Kanal (DB21, ...).....	1834
18.2.3	Signale an Achse/Spindel (DB31, ...).....	1836
18.2.4	Signale von Achse/Spindel (DB31, ...).....	1838
18.3	Bahnsteuerbetrieb, Genauhalt und LookAhead (B1).....	1839
18.3.1	Signale von Kanal (DB21, ...).....	1839
18.3.2	Signale von Achse/Spindel (DB31, ...).....	1839
18.4	Fahren auf Festanschlag (F1).....	1841
18.4.1	Signale an Achse/Spindel (DB31, ...).....	1841
18.4.2	Signale von Achse/Spindel (DB31, ...).....	1843
18.5	Hilfsfunktionsausgaben an PLC (H2).....	1844
18.5.1	Signale an Kanal (DB21, ...).....	1844

18.5.2	Signale von Kanal (DB21, ...)	1844
18.5.3	Signale von Achse/Spindel (DB31, ...)	1850
18.6	BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)	1851
18.6.1	Signale an BAG (DB11)	1851
18.6.2	Signale von BAG (DB11)	1857
18.6.3	Signale an Kanal (DB21, ...)	1862
18.6.4	Signale von Kanal (DB21, ...)	1867
18.6.5	Signale an Achse/Spindel (DB31, ...)	1881
18.6.6	Signale von Achse/Spindel (DB31, ...)	1882
18.7	Achsen, Koordinatensysteme, Frames (K2)	1884
18.7.1	Signale an Achse/Spindel (DB31, ...)	1884
18.8	Not-Halt (N2)	1885
18.8.1	Signale an NC (DB10)	1885
18.8.2	Signale von NC (DB10)	1886
18.9	PLC-Grundprogramm (P3)	1887
18.10	Referenzpunktfahren (R1)	1888
18.10.1	Signale an Kanal (DB21, ...)	1888
18.10.2	Signale von Kanal (DB21, ...)	1889
18.10.3	Signale an Achse/Spindel (DB31, ...)	1890
18.10.4	Signale von Achse/Spindel (DB31, ...)	1891
18.11	Spindeln (S1)	1893
18.11.1	Signale an Achse/Spindel (DB31, ...)	1893
18.11.2	Signale von Achse/Spindel (DB31, ...)	1900
18.12	Vorschübe (V1)	1912
18.12.1	Signale an Kanal (DB21, ...)	1912
18.12.2	Signale an Achse/Spindel (DB31, ...)	1921
18.12.3	Signale von Achse/Spindel (DB31, ...)	1927
<b>A</b>	<b>Anhang</b>	<b>1929</b>
A.1	Liste der Abkürzungen	1929
A.2	Dokumentationsübersicht	1939
	<b>Glossar</b>	<b>1941</b>
	<b>Index</b>	<b>1965</b>





# A2: Diverse NC/PLC-Nahtstellensignale und Funktionen

# 1

## 1.1 Kurzbeschreibung

### Inhalt

Die Nahtstelle PLC/NCK wird einerseits durch eine Datenschnittstelle und andererseits durch eine Funktionsschnittstelle gebildet. In der Datenschnittstelle sind Status- und Steuersignale, Hilfs- und G-Funktionen enthalten, während über die Funktionsschnittstelle Aufträge von der PLC an den NCK übergeben werden.

In der vorliegenden Beschreibung wird die Funktionalität von Nahtstellensignalen beschrieben, die von allgemeiner Bedeutung sind und die in den funktionspezifischen Beschreibungen nicht beschrieben sind:

- Asynchroner Ereignisse
- Zustandssignale
- PLC-Variable (lesen und schreiben)

## 1.2 NC/PLC-Nahtstellensignale

### 1.2.1 Allgemeines

#### NC/PLC-Nahtstelle

Die NC/PLC-Nahtstelle besteht aus den Teilen:

- Datenschnittstelle
- Funktionsschnittstelle

#### Datenschnittstelle

Die Datenschnittstelle dient zur Koordination der Komponenten:

- PLC-Anwenderprogramm
- NC
- HMI (Bedienkomponente)
- MSTT (Maschinensteuertafel)

Der Datenaustausch wird durch das PLC-Grundprogramm organisiert.

#### Zyklischer Signalaustausch

Folgende Nahtstellensignale werden vom PLC-Grundprogramm zyklisch, d. h. im Taktraster des OB1, übertragen:

- NC- und Bedientafelfront-spezifische Signale
- BAG-spezifische Signale
- Kanal-spezifische Signale
- Achs-/Spindel-spezifische Signale

#### NC- und Bedientafelfront-spezifische Signale (DB10)

PLC an NC:

- Signale für die Beeinflussung der CNC-Ein- und -Ausgänge
- Schlüsselschalter-Signale (und Kennwort)

NC an PLC:

- Istwerte der CNC-Eingänge
- Sollwerte der CNC-Ausgänge
- Bereitschaftssignale (Ready-Signale) von NC und HMI
- Statussignale der NC (Alarmsignale)

### Kanalspezifische Signale (DB21, ...)

PLC an NC:

- Steuersignal "Restweg löschen"

NC an PLC:

- Statussignale der NC (NCK-Alarm steht an)

### Achs-/Spindel-spezifische Signale (DB31, ...)

PLC an NC:

- Steuersignale an Achse/Spindel (z. B. Nachführbetrieb, Reglerfreigabe, ...)
- Steuersignale an Antrieb (Byte 20, 21)

NC an PLC:

- Statussignale von Achse/Spindel (z. B. Lageregler aktiv, Stromregler aktiv, ...)
- Statussignale vom Antrieb (Byte 93, 94)

### Funktionsschnittstelle

Die Funktionsschnittstelle wird durch Funktionsbausteine (FB) und Funktionen (FC) gebildet. Über die Funktionsschnittstelle werden Funktionsanforderungen z. B. zum Verfahren von Achsen, von der PLC an die NC übergeben.

### Literatur

- Beschreibung des PLC-Grundprogramms:
  - Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)
- Beschreibung des ereignisgesteuerten Signalaustausches (Hilfs- und G-Funktionen):
  - Funktionshandbuch Grundfunktionen; Hilfsfunktionsausgabe an PLC (H2)
- Übersicht aller Nahtstellensignale, Funktionsbausteine und Datenbausteine:
  - Listenhandbuch 2

## 1.2.2 Bereitschaftssignale an PLC

### DB10 DBX104.7 (NCK-CPU ready)

Die NCK-CPU ist betriebsbereit und meldet sich zyklisch bei der PLC.

### DB10 DBX108.3 (HMI - CPU an BTSS ready)

SINUMERIK Operate ist betriebsbereit und meldet sich zyklisch bei der NC.

### DB10 DBX108.5 (Antriebe im zyklischen Betrieb)

Voraussetzung: Bei **allen** Maschinenachsen der NC befinden sich die zugehörigen Antriebe im zyklischen Betrieb, d. h. sie tauschen zyklisch mit der NC PROFIdrive-Telegramme aus.

### DB10 DBX108.6 (Antrieb ready)

Voraussetzung: Für **alle** Maschinenachsen der NC steht bei den zugehörigen Antrieben und auch den Fremdantrieben über PROFIBUS die Betriebsbereitschaft an:

DB31, ... DBX93.5 == 1 (Drive Ready)

### DB10 DBX108.7 (NC ready)

Die NC ist betriebsbereit.

## 1.2.3 Statussignale an PLC

### DB10 DBX103.0 (Ferndiagnose aktiv)

Die HMI-Komponente meldet an die PLC, dass die Ferndiagnose (Option!) aktiv ist, d. h. die Bedienung der Steuerung erfolgt über einen externen PC.

### DB10 DBX109.6 (Luft-Temperatur-Alarm)

Die Umgebungstemperatur- oder die Lüfterüberwachung hat angesprochen.

### DB10 DBX109.7 (NCK-Batterie-Alarm)

Die Batteriespannung ist unter den Grenzwert abgesunken. Die Steuerung kann weiterhin betrieben werden. Ein Ausschalten der Steuerung oder Ausfall der Versorgungsspannung führt zum Datenverlust.

### DB10 DBX109.0 (NCK-Alarm steht an)

Die NC meldet, dass mindestens ein NC-Alarm ansteht. An der kanalspezifischen Nahtstelle kann abgefragt werden, welche Kanäle betroffen sind und ob ein Bearbeitungsstillstand ausgelöst wurde.

### DB21, ... DBX36.6 (Kanalspezifischer NCK-Alarm steht an)

Die NC meldet an die PLC, dass für den betroffenen Kanal mindestens ein NC-Alarm ansteht (siehe auch DB21, ... DBX36.7)

### DB21, ... DBX36.7 (NCK-Alarm mit Bearbeitungsstillstand steht an)

Die NC meldet an die PLC, dass für den betroffenen Kanal mindestens ein NCK-Alarm ansteht, der die aktuelle Teileprogrammbearbeitung unter- bzw. abgebrochen hat.

## 1.2.4 Signale an/von Bedientafelfront

### DB19 DBX0.0 (Bildschirm hell)

Die Dunkelschaltung des Bildschirms wird außer Kraft gesetzt.

### DB19 DBX0.1 (Bildschirm dunkel)

Der Bildschirm der Bedientafel wird dunkel gesteuert.

Bei aktiver Dunkelsteuerung durch das Nahtstellensignal gilt:

- Eine Hellsteuerung über die Tastatur (siehe unten) ist damit nicht mehr möglich.
- Bereits der erste Tastendruck an der Bedientafelfront löst eine Bedienhandlung aus.

---

#### Hinweis

Um unbeabsichtigte Bedienhandlungen bei Dunkelsteuerung des Bildschirms über das Nahtstellensignal auszuschließen, wird empfohlen **gleichzeitig** die Tastatur zu sperren:

DB19 DBX0.1 = 1 **UND** DB19 DBX0.2 = 1 (Tastensperre)

---

### Dunkelsteuerung über die Tastatur bzw. automatischer Bildschirmschoner

Wird in der parametrierten Zeit (Default = 3 Minuten):  
MD9006 \$MM\_DISPLAY\_BLACK\_TIME (Zeit für Bildschirmdunkelschaltung)  
keine Taste an der Bedientafelfront betätigt, wird der Bildschirm automatisch dunkel gesteuert.

Die Hellsteuerung erfolgt mit dem ersten Tastendruck nach Dunkelsteuerung. Dieser Tastendruck löst keine Bedienhandlung aus.

Parametrierung:

DB19 DBX0.1 = 0

MD9006 \$MM\_DISPLAY\_BLACK\_TIME > 0

### DB19 DBX0.2 (Tastensperre)

Alle Eingaben über die angeschlossene Tastatur werden gesperrt.

### DB19 DBX 0.3 / 0.4 (Cancel-Alarme löschen / Recall-Alarme löschen) (HMI Advanced)

Anforderung zum Löschen aller aktuell anstehenden Alarme mit Löschkriterium Cancel bzw. Recall. Das Löschen der Alarme wird über folgende Nahtstellensignale quittiert:

- DB19 DBX20.3 (Cancel-Alarm gelöscht)
- DB19 DBX20.4 (Recall-Alarm gelöscht)

### DB19 DBX0.7 (Istwerte in WKS, 0=MKS)

Umschalten der Istwertanzeige zwischen Maschinen- und Werkstückkoordinatensystem:

- DB19 DBX0.7 = 0: Maschinenkoordinatensystem (MKS)
- DB19 DBX0.7 = 1: Werkstückkoordinatensystem (WKS)

### DB19 DBB13 (Steuerung des Datei-Transfers über Festplatte) (nur HMI Advanced)

Auftragsbyte zur Steuerung des Datei-Transfers über Festplatte. Die Aufträge beziehen sich auf die Anwender-Steuerdatei in den Nahtstellensignalen:

DB19 DBB16 (Teile-Programmhandling: Nummer der Steuerdatei für Anwender-Dateinamen)

DB19 DBB17 (Teile-Programmhandling: Index der zu übertragenden File aus der Anwender-Liste)

#### **DB19 DBB16 (Steuerung des Datei-Transfers über Festplatte) (nur HMI Advanced)**

Steuerbyte des Datei-Transfers über Festplatte zur Beschreibung des Index zur Steuerdatei (Jobliste). Diese Datei wird behandelt entsprechend des Auftrags im Nahtstellensignal:

DB19 DBB13

#### **DB19 DBB17 (Teileprogrammhandling: Index des zu übertragenden Files aus der Anwenderliste)**

Steuerbyte des Datei-Transfers über Festplatte zur Angabe in welcher Zeile der Anwender-Steuerdatei die zu übertragene Steuerdatei steht.

#### **DB19 DBB26 (Teileprogrammhandling: Status)**

Zustandsbyte für den aktuellen Zustand der Datenübertragung bei "Anwahl", "Laden" bzw. "Entladen" oder ob die Datenübertragung fehlerhaft war.

#### **DB19 DBB27 (Error Programmhandling)**

Ausgabebyte für die Fehlerwerte der Datenübertragung über Festplatte.

### **1.2.5 Signale an Kanal**

#### **DB21, ... DBX6.2 (Restweg löschen)**

Mit der steigenden Flanke des Nahtstellensignals wird im entsprechenden Kanal der NC auf der programmierten Bahn mit der aktuell wirksame Bahnbeschleunigung angehalten. Anschließend wird der noch nicht verfahrenende Bahnrestweg gelöscht und der Satzwechsel zum nächsten Teileprogrammsatz freigegeben.

## 1.2.6 Signale an Achse/Spindel

### DB31, ... DBX1.0 (Antriebstest Fahrfreigabe)

Werden Maschinenachsen von speziellen Testfunktionen wie z. B. "Funktionsgenerator" verfahren, wird für die Verfahrbewegung eine explizite Antriebstest-spezifische Freigabe angefordert:

DB31, ... DBX61.0 = 1 (Antriebstest Fahrenanforderung)

Die Verfahrbewegung wird ausgeführt, sobald die Fahrfreigabe erfolgt ist:

DB31, ... DBX1.0 == 1 (Antriebstest Fahrfreigabe)

<b>ACHTUNG</b>
----------------

Es liegt in der alleinigen Verantwortung des Maschinenherstellers / Inbetriebnehmers durch geeignete Maßnahmen / Prüfungen sicherzustellen, dass die Maschinenachse im Rahmen des Antriebstests ohne Gefahr für Mensch und Maschine verfahren werden kann.
--

### DB31, ... DBX1.3 (Achsen- / Spindelsperre )

#### Achsensperre bei stillstehender Maschinenachse

Es wird keine Verfahranforderung (manuell oder automatisch) ausgeführt bei stillstehender Maschinenachse und NC/PLC-Nahtstellensignal:

DB31, ... DBX1.3 == 1 (Achsen- / Spindelsperre).

Die Verfahranforderung bleibt erhalten. Wird die Achsensperre bei anstehender Verfahranforderung aufgehoben DB31, ... DBX1.3 = 0 wird die Verfahrbewegung ausgeführt.

#### Achsensperre bei fahrender Maschinenachse

Bei fahrender Maschinenachse und NC/PLC-Nahtstellensignal DB31, ... DBX1.3 == 1 wird die Verfahrbewegung der Maschinenachse über die aktuell wirksame Bremskennlinie Achs-spezifisch oder, falls sie Teil einer interpolatorischen Bahnbewegung oder Kopplung ist, Bahn- oder kopplungsspezifisch, auf Stillstand abgebremst.

Die Verfahrbewegung wird fortgesetzt, wenn die Achsensperre bei weiterhin anstehender Verfahranforderung aufgehoben wird: DB31, ... DBX1.3 = 0.

#### Spindelsperre

Die Reaktion ist abhängig von der aktuellen Spindelbetriebsart:

- Steuerbetrieb: Drehzahlsollwert = 0
- Positionierbetrieb: siehe oben "Achsensperre"



**DB31, ... DBX1.4 (Nachführbetrieb)**

"Nachführbetrieb" ist nur wirksam in Zusammenhang mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX2.1 (Reglerfreigabe)

DB31, ... DBX2.1	DB31, ... DBX1.4	Funktion
1	unwirksam	Normalbetrieb (Maschinenachse in Regelung)
0	1	Nachführen
0	0	Halten

**Funktion: Nachführen**

Beim Nachführen wird die Sollposition der Maschinenachse kontinuierlich auf die Istposition nachgeführt (Sollposition = Istposition).

Für das Nachführen sind folgende Nahtstellensignale zu setzen:

DB31, ... DBX2.1 = 0 (Reglerfreigabe)

DB31, ... DBX1.4 = 1 (Nachführbetrieb)

Rückmeldung:

DB31, ... DBX61.3 = 1 (Nachführen aktiv)

**Hinweis**

Mit dem Setzen der Reglerfreigabe aus dem Nachführbetrieb erfolgt bei aktivem Teileprogramm NC-intern ein Wiederanfahren der zuletzt programmierten Position (REPOSA: Anfahren auf einer Geraden mit allen Achsen). In allen anderen Fällen beginnen alle weiteren Verfahrbewegungen an der aktuellen Istposition.

Während des "Nachführens" sind Klemmungs- oder Stillstandsüberwachung **inaktiv**.

**Funktion: Halten**

Beim Halten wird die Sollposition der Maschinenachse nicht auf die Istposition nachgeführt. Wird die Maschinenachse von der Sollposition weg bewegt, baut sich ein Schleppabstand (Differenz zwischen Soll- und Istposition) auf, der beim Setzen der Reglerfreigabe vom Lageregler ohne Einhaltung der axialen Beschleunigungskennlinie "schlagartig" ausgeregelt wird.

Für das Halten sind folgende Nahtstellensignale zu setzen:

DB31, ... DBX2.1 = 0 (Reglerfreigabe)

DB31, ... DBX1.4 = 0 (Nachführbetrieb)

Rückmeldung:

DB31, ... DBX61.3 = 0 (Nachführen aktiv)

Während des "Haltens" sind Klemmungs- oder Stillstandsüberwachung **aktiv**.

### ACHTUNG

Das Ausregeln der Soll-/Istdifferenz nach Setzen der Reglerfreigabe bei Funktion: "Halten" erfolgt direkt durch den Lageregler, d.h. ohne Einhaltung der axialen Beschleunigungskennlinie.

### Anwendungsbeispiel

Positionierverhalten der Maschinenachse Y nach einem Klemmungsvorgang beim Setzen der "Reglerfreigabe". Die Maschinenachse wurde dabei durch den Klemmungsvorgang aus der Istposition  $Y_1$  auf die Klemmungsposition  $Y_k$  gedrückt.

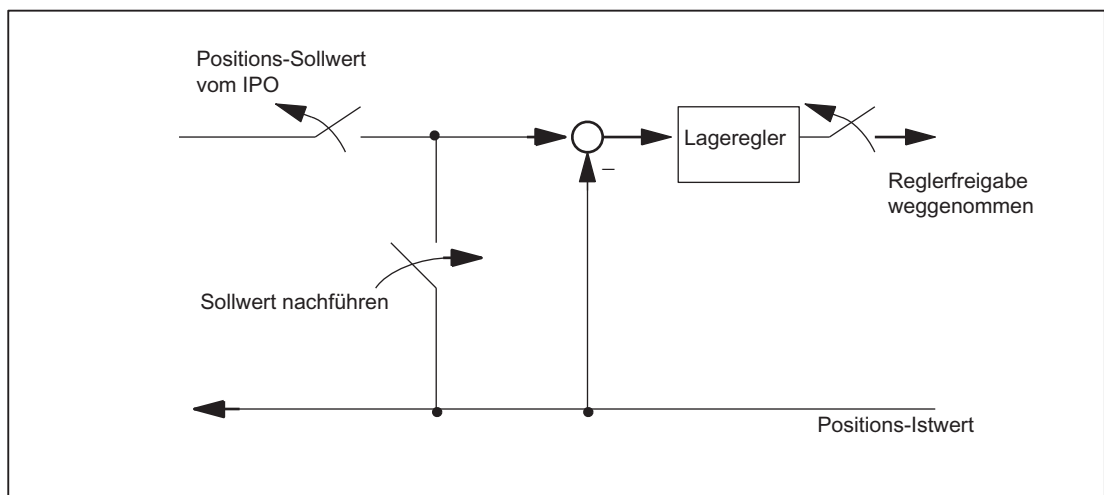


Bild 1-1 Wirkung von Reglerfreigabe und Nachführbetrieb

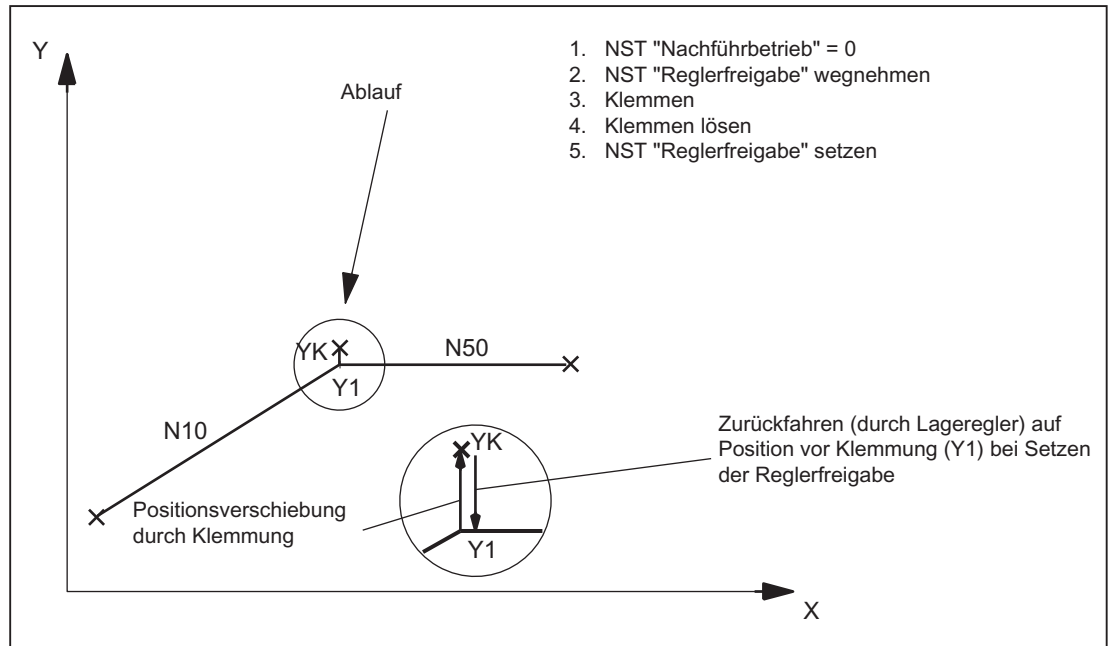


Bild 1-2 Bahnverlauf bei Klemmungsvorgang und "Halten"

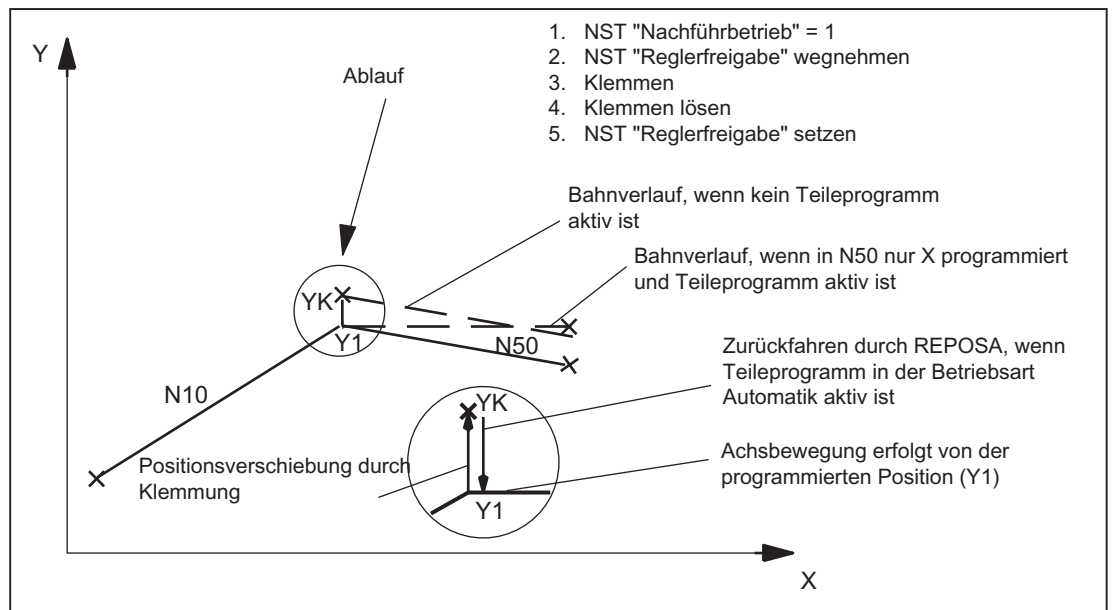


Bild 1-3 Bahnverlauf bei Klemmungsvorgang und "Nachführen"

### Antrieben mit analoger Sollwertschnittstelle

Bei einem Antrieb mit analoger Sollwertschnittstelle besteht die Möglichkeit, die Maschinenachse mit einem externen Sollwert zu verfahren. Wird für die Maschinenachse "Nachführbetrieb" gesetzt, wird die Istposition weiterhin erfasst. Nach Aufheben des Nachführbetriebs ist dann kein Referenzieren erforderlich.

Dabei ist folgendes Vorgehen zu empfehlen:

1. Nachführbetrieb aktivieren:

DB31, ... DBX2.1 = 0 (Reglerfreigabe)

DB31, ... DBX1.4 = 1 (Nachführbetrieb) (im gleichen oder vorhergehenden OB1-Zyklus)

→ Achse/Spindel befindet sich im Nachführbetrieb

2. Externe Reglerfreigabe und externen Drehzahlsollwert aufschalten

→ Achse/Spindel fährt mit externem Sollwert

→ NC erfasst weiterhin die Istposition und führt die Sollposition der Istposition nach

3. Externe Reglerfreigabe und externen Drehzahlsollwert wegnehmen

→ Achse/Spindel steht

4. Nachführbetrieb aufheben

DB31, ... DBX2.1 = 1 (Reglerfreigabe)

DB31, ... DBX1.4 = 0 (Nachführbetrieb)

→ NC synchronisiert auf die aktuelle Istposition. Die nächste Verfahrbewegung beginnt von dieser Position.

---

#### Hinweis

"Nachführbetrieb" kann gesetzt bleiben, da es nur in Verbindung mit "Reglerfreigabe" wirkt.

---

### Nachführbetrieb aufheben

Nach Aufhebung des Nachführbetriebes ist keine erneutes Referenzpunktfahren der Maschinenachse erforderlich, falls die maximal zulässige Gebergrenzfrequenz des aktiven Messsystems während des Nachführbetriebes nicht überschritten wurde. Wird die Gebergrenzfrequenz überschritten, wird dies von der Steuerung erkannt:

- DB31, ... DBX60.4 / 60.5 = 0 (Referiert / Synchronisiert 1 / 2)
- Alarm: "21610 Encoderfrequenz überschritten"

---

#### Hinweis

Wird der "Nachführbetrieb" für eine Maschinenachse ausgeschaltet, die zu einer aktiven Transformation (z.B. TRANSMIT) gehört, kann dies zu Verfahrbewegungen im Rahmen von Rückpositionierbewegungen (REPOS) von andere(n), an der Transformation beteiligte(n) Maschinenachse(n) führen.

---

### Überwachungen

Befindet sich eine Maschinenachse im Nachführbetrieb, sind folgende Überwachungen nicht wirksam:

- Stillstandsüberwachung
- Klemmungsüberwachung
- Positionierüberwachung

Auswirkungen auf andere Nahtstellensignale:

- DB31, ... DBX60.7 = 0 (Position erreicht mit Genauhalt fein)
- DB31, ... DBX60.6 = 0 (Position erreicht mit Genauhalt grob)

### DB31, ... DBX1.5 / 1.6 (Lagemesssystem 1 / 2)

An eine Maschinenachse können 2 Messsysteme angeschlossen werden z.B.:

- Indirektes Motormesssystem
- Direktes Messsystem an der Last

Nur ein Messsystem kann zu einem Zeitpunkt aktiv sein. Alle Regelungs-, Positioniervorgänge etc. der Maschinenachse beziehen sich immer auf das aktive Messsystem.

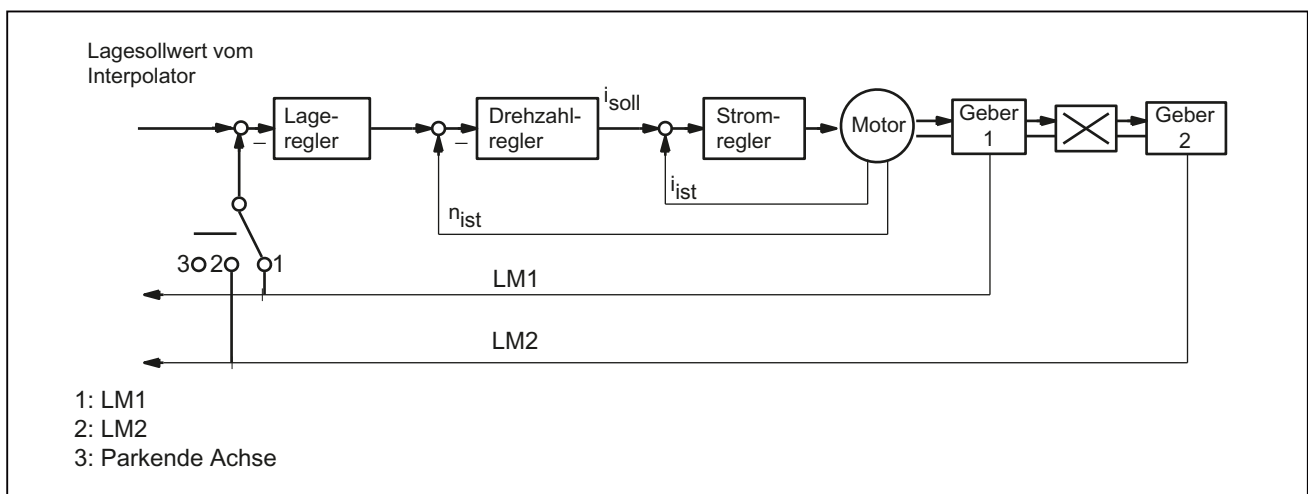


Bild 1-4 Lagemesssystem 1 und 2

Die folgende Tabelle zeigt die Funktionalität der Nahstellensignale in Zusammenhang mit der "Reglerfreigabe":

DB31, ... DBX1.5	DB31, ... DBX1.6	DB31, ... DBX2.1	Funktion
1	0 (oder 1)	1	Lagemesssystem 1 aktiv
0	1	1	Lagemesssystem 2 aktiv
0	0	0	"Parken" aktiv
0	0	1	Spindel ohne Lagemesssystem (drehzahl geregelt)
1 -> 0	0 -> 1	1	Umschaltung: Lagemesssystem 1 → 2
0 -> 1	1 -> 0	1	Umschaltung: Lagemesssystem 2 → 1

### DB31, ... DBX2.1 (Reglerfreigabe)

Mit dem Setzen der Reglerfreigabe wird der Lageregelkreis der Maschinenachse geschlossen. Die Maschinenachse befindet sich in Lageregelung.

DB31, ... DBX2.1 == 1

Mit der Wegnahme der Reglerfreigabe wird der Lageregelkreis und mit zeitlicher Verzögerung der Drehzahlregelkreis der Maschinenachse geöffnet:

DB31, ... DBX2.1 == 0

#### Aktivierungsarten

Die regelungstechnische Reglerfreigabe einer Maschinenachse wird beeinflusst durch:

- NC/PLC-Nahtstellensignal:
  - DB31, ... DBX2.1 (Reglerfreigabe)
  - DB31, ... DBX21.7 (Impulsfreigabe)
  - DB31, ... DBX93.5 (Drive Ready)
  - DB10, DBX56.1 (Not-Halt)
- NCK-intern
 

Alarmer, die als Alarmreaktion die Reglerfreigabe der Maschinenachsen wegnehmen. Die Alarmreaktionen sind beschrieben in:

#### Literatur:

Diagnosehandbuch

#### Wegnahme der Reglerfreigabe bei stehender Maschinenachse:

- Der Lageregelkreis der Maschinenachse wird geöffnet
- DB31, ... DBX61.5 == 0 (Lageregler aktiv)

**Wegnahme der Reglerfreigabe bei fahrender Maschinenachse:**

Ist eine Maschinenachse Teil einer interpolatorischen Bahnbewegung oder Kopplung und wird für diese die Reglerfreigabe weggenommen, werden alle betroffenen Achsen mit Schnellstopp (Drehzahlsollwert = 0) stillgesetzt und ein Alarm angezeigt:

Alarm: "21612 Reglerfreigabe zurückgesetzt während der Bewegung"

- Die Maschinenachse wird, unter Berücksichtigung der parametrisierten Zeitdauer der Bremsrampe, bei Fehlerzuständen mit Schnellstopp (Drehzahlsollwert = 0) abgebremst:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME (max. Zeitdauer der Bremsrampe bei Fehlern)

Ein Alarm wird angezeigt:

Alarm: "21612 Reglerfreigabe zurückgesetzt während der Bewegung"

---

**Hinweis**

Die Reglerfreigabe wird spätestens nach Ablauf der Abschaltverzögerungszeit weggenommen:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME

---

- Der Lageregelkreis der Maschinenachse wird geöffnet. Rückmeldung über Nahtstellensignal:  
DB31, ... DBX61.5 == 0 (Lageregler aktiv).  
Die Zeit für die parametrisierte Abschaltverzögerung der Reglerfreigabe wird gestartet durch das Maschinendatum:  
MD36620 \$MA\_SERVO\_DISABLE\_DELAY\_TIME (Abschaltverzögerung Reglerfreigabe)
- Sobald die Istgeschwindigkeit den Stillstandsbereich erreicht, wird die Reglerfreigabe des Antriebs weggenommen. Rückmeldung über Nahtstellensignal:  
DB31, ... DBX61.6 == 0 (Drehzahlregler aktiv)
- Der Lageistwert der Maschinenachse wird von der Steuerung weiterhin erfasst.
- Am Ende des Bremsvorganges wird die Maschinenachse unabhängig vom entsprechenden NC/PLC-Nahtstellensignal in den Nachführbetrieb geschaltet. Stillstands- und Klemmungsüberwachung sind dabei unwirksam. Siehe oben die Beschreibung zum Nahtstellensignal:  
DB31, ... DBX1.4 (Nachführbetrieb).

**Istwert synchronisieren (Referenzpunktfahren)**

Nach dem Setzen der Reglerfreigabe ist keine erneutes Synchronisieren der Istposition der Maschinenachse (Referenzpunktfahren) erforderlich, falls während der Zeit in der die Maschinenachse nicht in Lageregelung war, die maximal zulässige Grenzfrequenz des Messsystems nicht überschritten wurde.

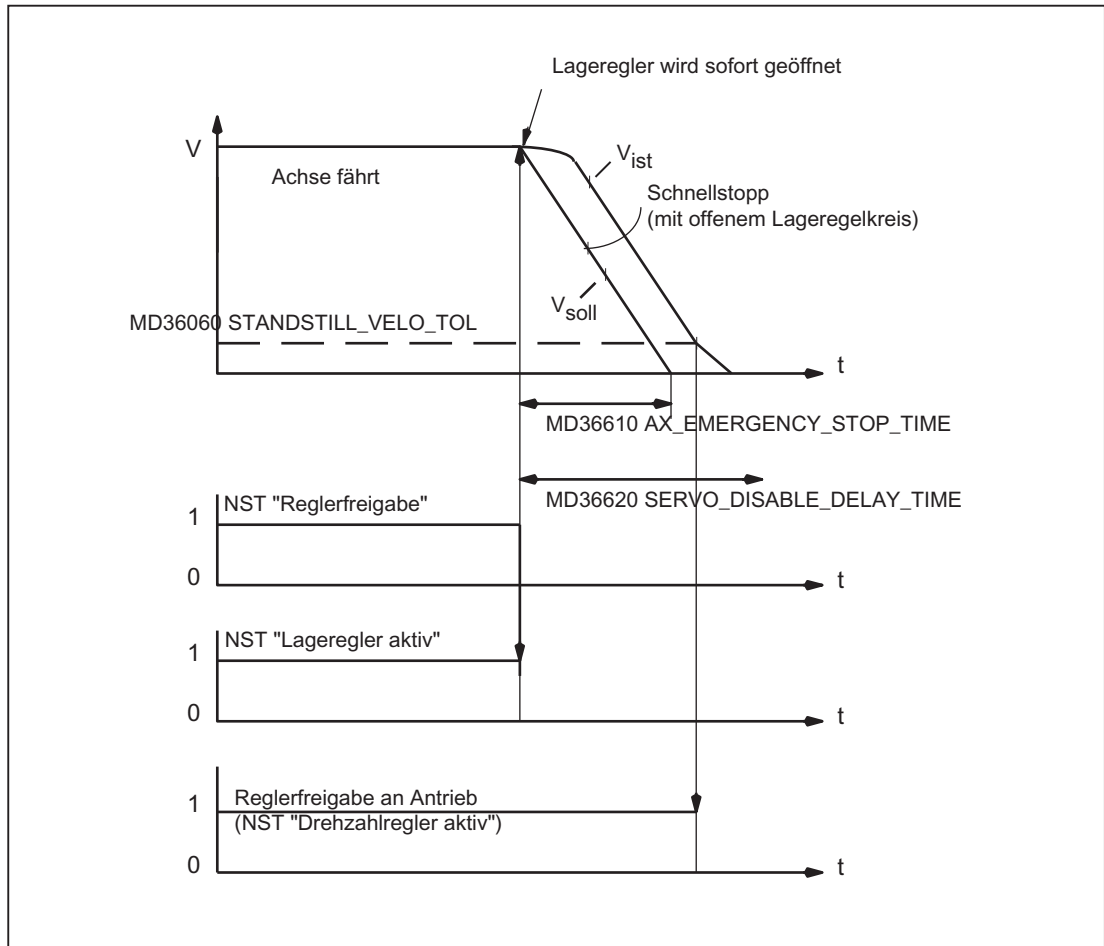


Bild 1-5 Wegnahme der Reglerfreigabe bei fahrender Maschinenachse

**DB31, ... DBX2.2 (Restweg/Spindel-Reset (achs-/spindelspezifisch))**

"Restweg löschen" wirkt in den Betriebsarten: AUTOMATIK bzw. MDA nur im Zusammenhang mit Positionierachsen. Die Positionierachse wird dabei über die aktuelle Bremskennlinie bis zum Stillstand abgebremst. Der noch nicht abgefahrene Restweg der Achse wird gelöscht.

**Spindel-Reset**

Eine ausführliche Beschreibung des Spindel-Reset siehe Kapitel "S1: Spindeln (Seite 1383)".



**DB31, ... DBX9.0 / 9.1 / 9.2 (Regler-Parametersatz)**

Anforderung zur Aktivierung des vorgegebenen Regler-Parametersatzes.

Regler-Parametersatz	DBX9.2	DBX9.1	DBX9.0
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Die Parametersatzumschaltung muss freigegeben sein (nicht erforderlich bei Spindeln) über das Maschinendatum:

MD35590 \$MA\_PARAMSET\_CHANGE\_ENABLE = 1 oder 2

Ausführliche Informationen zur Parametersatzumschaltung siehe Kapitel "Parametersatzanwahl beim Getriebestufenwechsel (Seite 1437)".

**Parametersatzumschaltung bei fahrender Maschinenachse**

Das Verhalten bei Parametersatzumschaltung ist abhängig von der dabei stattfindenden Änderung des Regelkreis-Verstärkungsfaktor Kv:

MD32200 \$MA\_POSCTRL\_GAIN (KV-Faktor) (KV-Faktor)

- "Gleiche Kv-Faktoren" oder "Lageregelung nicht aktiv":

Die NC reagiert sofort auf die Parametersatzumschaltung. Der Parametersatz wird auch während der Bewegung gewechselt.

- "Ungleiche Kv-Faktoren" und "Lageregelung aktiv":

Um eine möglichst ruckfreie Umschaltung durchführen zu können, wird mit der Parametersatzumschaltung so lange gewartet, bis die Achse "steht", d.h. die parametrisierte Stillstandsgeschwindigkeit erreicht bzw. unterschritten wurde:

DB31, ... DBX61.4 = 1 (Achse/Spindel steht)

MD36060 \$MA\_STANDSTILL\_VELO\_TOL (Schwellgeschwindigkeit / Drehzahl 'Achse / Spindel steht')

**Parametersatzumschaltung aus dem Teileprogramm**

Zur Parametersatzumschaltung aus dem Teileprogramm sind vom Anwender (Maschinenhersteller) entsprechende Anwender-spezifische Hilfsfunktionen zu definieren und im PLC-Anwenderprogramm auszuwerten. Das PLC-Anwenderprogramm setzt daraufhin die Anforderung zur Umschaltung auf den entsprechenden Parametersatz.

Ausführliche Informationen zur Hilfsfunktionsausgabe siehe Kapitel "H2: Hilfsfunktionsausgaben an PLC (Seite 395)".

### DB31, ... DBX9.3 (Parametersatzvorgabe von NC gesperrt)

Eine Anforderung zur Parametersatzumschaltung wird ignoriert.

## 1.2.7 Signale von Achse/Spindel

### DB31, ... DBX61.0 (Antriebstest Fahranforderung)

Werden Maschinenachsen von speziellen Testfunktionen wie z. B. "Funktionsgenerator" verfahren, wird für die Verfahrbewegung eine explizite Antriebstest-spezifische Freigabe angefordert:

DB31, ... DBX61.0 == 1 (Antriebstest Fahranforderung)

Die Verfahrbewegung wird ausgeführt, sobald die Fahrfreigabe erfolgt ist:

DB31, ... DBX1.0 == 1 (Antriebstest Fahrfreigabe)

### DB31, ... DBX61.3 (Nachführen aktiv)

Die Maschinenachse befindet sich im Nachführbetrieb.

### DB31, ... DBX61.4 (Achse/Spindel steht ( $n < n_{\min}$ ))

"Achse/Spindel steht" wird von der NC gesetzt, wenn:

- Keine neuen Sollwerte mehr ausgegeben werden **UND**
- Die Istgeschwindigkeit der Maschinenachse kleiner der parametrisierten Stillstandsgeschwindigkeit ist:

MD36060 \$MA\_STANDSTILL\_VELO\_TOL (Schwellgeschwindigkeit Achse steht)

### DB31, ... DBX61.5 (Lageregler aktiv)

Der Lageregelkreis der Maschinenachse ist geschlossen und die Lageregelung ist aktiv.

### DB31, ... DBX61.6 (Drehzahlregler aktiv)

Der Drehzahlregelkreis der Maschinenachse ist geschlossen und die Drehzahlregelung ist aktiv.

### DB31, ... DBX61.7 (Stromregler aktiv)

Der Stromregelkreis der Maschinenachse ist geschlossen und die Stromregelung ist aktiv.

**DB31, ... DBX69.0 / 69.1 / 69.2 (Parametersatz Servo)**

Aktiver Parametersatz. Codierung entsprechend:

DB31, ... DBX9.0 / 9.1 / 9.2 (Reglerparametersatz Anwahl)

**DB31, ... DBX76.0 (Schmierimpuls)**

Nach Power On/Reset der Steuerung ist der Signalzustand 0 (FALSE).

Der "Schmierimpuls" wird **invertiert** (Flankenwechsel), sobald die Maschinenachse die parametrisierte Verfahrestrecke für Schmierung zurückgelegt hat:

MD33050 \$MA\_LUBRICATION\_DIST (Verfahrestrecke für Schmierung von PLC)

**1.2.8 Signale an Achse/Spindel (Digitale Antriebe)****DB31, ... DBX21.0 / 21.1 / 21.2 (Parametersatzanwahl A, B, C)**

Anforderung zur Umschaltung des Antriebsparametersatzes:

DBX 21.2	DBX 21.1	DBX21.0	Parametersatznummer
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

Die Rückmeldung erfolgt über die Nahtstellensignale:

DB31, ... DBX93.0,1 / 93.2 (aktiver Antriebsparametersatz)

### DB31, ... DBX21.3 / 21.4 (Motor-Anwahl A, B)

Anwahl des Motors bzw. der Betriebsart.

DBX 21.4	DBX 21.3	Motornummer	Betriebsart
0	0	1	1
0	1	2	2
1	0	3	3
1	1	4	4

Bei Hauptspindelantrieb sind nur die Betriebsarten 1 und 2 gültig:

- Betriebsarten 1: Sternbetrieb
- Betriebsarten 2: Dreieckbetrieb

### DB31, ... DBX21.5 (Motor-Anwahl erfolgt)

Damit meldet das PLC-Anwenderprogramm an den Antrieb den Abschluss der Motor-Anwahl. Anschließend werden vom Antrieb die Impulse freigegeben.

### DB31, ... DBX21.6 (Integratorsperre n-Regler)

Das PLC-Anwenderprogramm sperrt beim Antrieb den Integrator des Drehzahlreglers. Der Drehzahlregler wird somit von PI- auf P-Regler umgeschaltet.

Hinweis

Bei Aktivierung der Integratorsperre des Drehzahlreglers können je nach Anwendungsfall Ausgleichsvorgänge auftreten (z.B. wenn der Integrator zuvor stationär eine Last hält).

Rückmeldung über das Nahtstellensignal:

DB31, ... DBX93.6 = 1 (Integrator n-Regler gesperrt).

### DB31, ... DBX21.7 (Impulsfreigabe)

Die Impulsfreigabe für das Antriebsmodul wird nur angefordert, wenn alle Freigabesignale (Hardware und Software) anstehen:

- Steuersatzfreigabe
- Regler- und Impulsfreigabe
- Impulsfreigabe (sicherer Betriebshalt)
- Gespeicherter Hardwareeingang
- Sollwertfreigabe
- "Zustand fahrbereit"
  - Kein Antriebsalarm (ZK1-Fehler)
  - Zwischenkreis zugeschaltet
  - Hochlauf abgeschlossen

Siehe auch:

DB31, ... DBX93.7 (Impulse freigegeben)

## 1.2.9 Signale von Achse/Spindel (Digitale Antriebe)

### DB31, ... DBX92.1 (Hochlaufgebersperre aktiv)

Vom Antrieb wird an die PLC zurückgemeldet, dass der Hochlaufgeber-Schnellstop aktiv ist. Damit wird der Antrieb ohne Hochlaufgeberrampe (mit Drehzahlsollwert 0) stillgesetzt.

### DB31, ... DBX93.0, 1, 2 (Aktiver Antriebsparametersatz A, B, C)

Vom Antriebsmodul wird an die PLC zurückgemeldet, welcher Antriebsparametersatz momentan aktiv ist. Mit der Bitkombination A, B, C können von PLC 8 verschiedene Antriebsparametersätze angewählt werden.

### DB31, ... DBX93.3, 4 (Aktiver Motor A, B)

Vom Antriebsmodul (HSA) wird an die PLC zurückgemeldet, welche der 4 Motorarten bzw. Motorbetriebsarten wirksam ist.

Beim Hauptspindelantrieb ist anwählbar:

- Sternbetrieb (A=0, B=0)
- Dreieckbetrieb (A=1, B=0)

### DB31, ... DBX93.5 (DRIVE ready)

Rückmeldung, dass der Antrieb betriebsbereit ist. Damit sind die Voraussetzungen vom Antrieb für ein Verfahren der Achse/Spindel gegeben.

### DB31, ... DBX93.6 (Integrator n-Regler gesperrt)

Der Integrator des Drehzahlreglers ist gesperrt. Der Drehzahlregler wurde somit von PI- auf P-Regler umgeschaltet.

### DB31, ... DBX93.7 (Impulse freigegeben)

Die Impulsfreigabe für das Antriebsmodul ist vorhanden. Damit kann die Achse/Spindel verfahren werden.

### DB31, ... DBX94.0 (Temperaturvorwarnung Motor)

Die Temperatur im Motor ist größer als die eingestellte Motortemperatur-Warnschwelle (Antriebsparameter p0604).

Siehe auch unten Hinweis zu "DB31, ... DBX94.1 (Temperaturvorwarnung Kühlkörper)".

### DB31, ... DBX94.1 (Temperaturvorwarnung Kühlkörper)

Die Temperatur des Kühlkörpers im Leistungsteil ist außerhalb des zulässigen Bereichs. Bleibt die zu hohe Temperatur bestehen, schaltet der Antrieb nach ca. 20 s ab.

---

#### Hinweis

#### Temperaturvorwarnung DB31, ... DBX94.0 und DBX94.1

Die Nahtstellensignale werden von folgenden Signalen des zykl. Antriebstelegramms abgeleitet:

- Fall 1: Temperaturwarnung im Meldungswort
  - DB31, ... DBX94.0 = MELDW, Bit 6 (Keine Warnung Übertemperatur Motor)
  - DB31, ... DBX94.1 = MELDW, Bit 7 (Keine Warnung thermische Überlast Leistungsteil)
- Fall 2: Warnung der Warnungsklasse B (nur im Interface Mode "SIMODRIVE 611u", p2038=1)  
DB31, ... DBX94.0 == 1 und DBX94.1 == 1, wenn gilt:  
zykl. Antriebstelegramm, Zustandswort 1 (ZSW1), Bit 11 / 12 == 2 (Warnungsklasse B)

Die Nahtstellensignale werden aus der Warnung der Warnklasse B abgeleitet, wenn keine spezifische Information aus dem Meldewort vorliegt.

Es wird ein Alarm angezeigt: Alarmnummer = 200.000 + Warnwert (r2124)

Für eine ausführliche Beschreibung zur Einstellung der Motortemperaturüberwachung siehe:

#### Literatur:

- S120 Inbetriebnahmehandbuch, Kapitel "Inbetriebnahme" > "Temperatursensoren bei SINAMICS-Komponenten"
  - S120 Funktionshandbuch, Kapitel "Überwachungs- und Schutzfunktionen"
  - S120 Listenhandbuch
    - MELDW, Bit 6  $\triangleq$  BO: r2135.14 → Funktionsplan: 2548, 8016
    - MELDW, Bit 7  $\triangleq$  BO: r2135.15 → Funktionsplan: 2548, 2452, 2456, 8016
-

### DB31, ... DBX94.2 (Hochlaufvorgang beendet)

Der Drehzahlwert liegt nach einer Änderung des Drehzahlsollwerts wieder innerhalb des parametrisierten Toleranzbandes. Der Hochlaufvorgang ist damit abgeschlossen.

Nachfolgende Drehzahlschwankungen, auch außerhalb des Toleranzbandes z. B. infolge Belastungsänderungen, haben auf das Nahtstellensignal keinen Einfluss.

### DB31, ... DBX94.3 ( $|M_d| < M_{dx}$ )

Der Betrag des aktuellen Moments  $|M_d|$  ist kleiner als das parametrisierte Schwellenmoment  $M_{dx}$  (Drehmomentschwellwert 2, p2194).

Das Schwellenmoment wird in Prozent [%], bezogen auf die aktuelle drehzahlabhängige Momentenbegrenzung, eingestellt.

### DB31, ... DBX94.4 ( $|n_{ist}| < n_{min}$ )

Der Drehzahlwert  $n_{ist}$  ist kleiner als  $n_{min}$  (Drehzahlschwellwert 3, p2161).

### DB31, ... DBX94.5 ( $|n_{ist}| < n_x$ )

Der Drehzahlwert  $n_{ist}$  ist kleiner als  $n_x$  (Drehzahlschwellwert 2, p2155).

### DB31, ... DBX94.6 ( $n_{ist} = n_{soll}$ )

Der Drehzahlwert liegt innerhalb des Toleranzbandes (p2163) um den Drehzahlsollwert.

### DB31, ... DBX95.7 (Warnung der Warnungsklasse C steht an)

Der Antrieb meldet, dass eine Warnung der Warnungsklasse C ansteht.

## 1.3 Funktionen

### 1.3.1 Bildschirmeinstellungen

Folgende Standardeinstellungen Bedientafel-Einstellungen wirken Parameter Mit Hilfe von Maschinendaten der Bedientafelfront können Kontrast, Monitortyp, Vordergrundsprache und Anzeigefinheit eingestellt werden, die jeweils nach dem Systemhochlauf wirken sollen.

#### Kontrast

MD9000 \$MM\_LCD\_CONTRAST (Kontrast)

Der Kontrast kann für Flachbedientafelfronten mit LCD-Display **monochrom** der Kontrast (Helligkeit) eingestellt werden, der nach dem Systemhochlauf wirken soll.

Der Kontrast kann in 16 Stufen (0: Dunkel, 15: Hell) verändert werden.

#### Monitortyp

MD9001 \$MM\_DISPLAY\_TYPE (Monitortyp)

Zur optimalen Farbanpassung ist der jeweilige Monitortyp anzugeben.

#### Vordergrundsprache

MD9003 \$MM\_FIRST\_LANGUAGE (Vordergrundsprache)

Bei SINUMERIK 840D sl sind gleichzeitig 2 Sprachen verfügbar. Über Vordergrundsprache kann die Sprache eingestellt werden, die nach Hochlauf der Steuerung angezeigt wird.

Die Sprache kann im Bedienbereich: DIAGNOSE der HMI-Bedienoberfläche umgeschaltet werden. Nach Hochlauf der Steuerung wirkt wieder die Vordergrundsprache.

#### Anzeigefinheit

MD9004 \$MM\_DISPLAY\_RESOLUTION (Anzeigefinheit)

Über die Anzeigefinheit wird für die Positionsanzeige der Achsen die Anzahl der Nachkommastellen festgelegt. Die Anzeige der Positionsanzeige erfolgt mit maximal 12 Zeichen einschließlich Vorzeichen und Dezimalpunkt. Die Anzahl der Zeichen nach dem Dezimalpunkt kann im Bereich von 0 bis 5 eingestellt werden.

Standardmäßig werden 3 Stellen nach dem Dezimalpunkt angezeigt, entsprechend einer Anzeigefinheit von 10<sup>-3</sup> [mm] oder [Grad].



## REFRESH-Unterdrückung

MD10131 \$MN\_SUPPRESS\_SCREEN\_REFRESH (Verhalten der Bildaktualisierung bei Überlastung)

Vorgabe der Bildaktualisierungs-Strategie bei hoher Auslastung der NC:

- Wert 0: Refresh der aktuellen Werte wird **in allen Kanälen** unterdrückt.
- Wert 1: Refresh der aktuellen Werte **der zeitkritischen Kanäle** wird unterdrückt.
- Wert 2: Refresh der aktuellen Werte wird grundsätzlich **nicht** unterdrückt.

## 1.3.2 Einstellungen für Evolventen-Interpolation

### Einführung

Die Evolvente des Kreises ist eine Kurve, die vom Endpunkt eines fest gespannten, von einem Kreis abgewickelten Fadens beschrieben wird. Die Evolventen-Interpolation ermöglicht Bahnkurven entlang einer Evolvente.

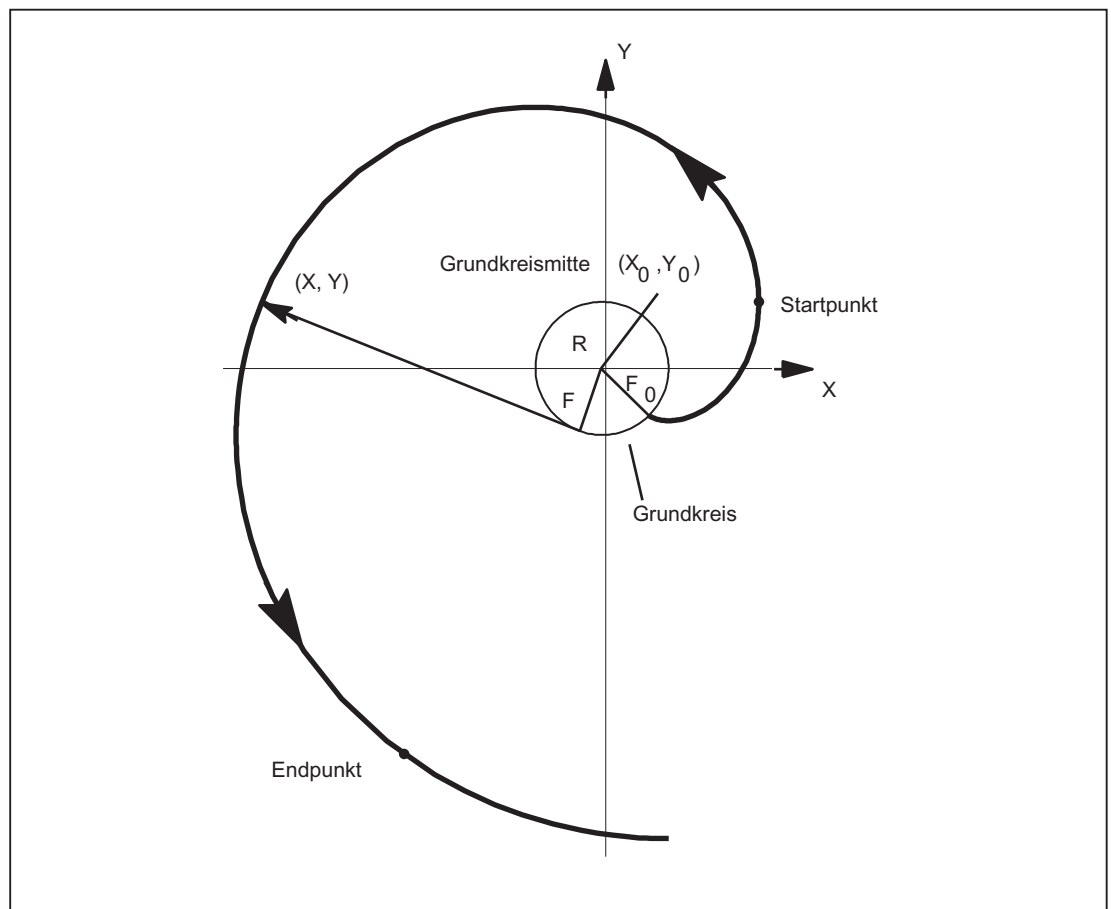


Bild 1-6 Evolvente (vom Grundkreis weg)

## Programmierung

Die Programmierung der Evolventen-Interpolation ist allgemein beschrieben in:

### Literatur:

Programmierhandbuch Grundlagen

Für zwei Fälle der Evolventen-Interpolation haben Maschinendaten zusätzlich zu den programmierten Parametern eine Bedeutung, die ggf. auch durch den Maschinenhersteller/Endbenutzer eingestellt werden müssen.

## Genauigkeit

Falls der programmierte Endpunkt nicht exakt auf der durch den Startpunkt festgelegten Evolventen liegt, wird zwischen den beiden Evolventen, die durch den Startpunkt bzw. den Endpunkt definiert sind, interpoliert (siehe nachfolgende Abbildung).

Die maximale Abweichung des Endpunkts wird festgelegt durch das Maschinendatum:

MD21015 \$MC\_INVOLUTE\_RADIUS\_DELTA(Endpunktüberwachung bei Evolvente)

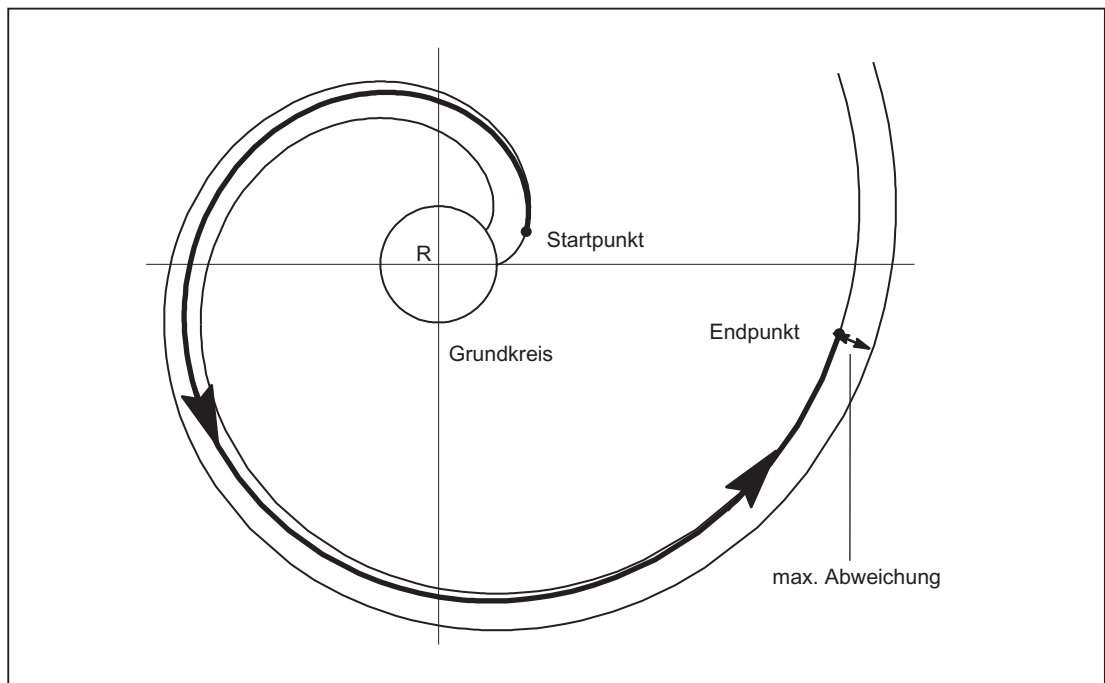


Bild 1-7 MD21015 legt die max. zulässige Abweichung fest

## Grenzwinkel

Wird mit AR eine zum Grundkreis führende Evolvente mit einem Drehwinkel programmiert, der größer als der maximal mögliche Wert ist, wird ein Alarm ausgegeben und die Programmverarbeitung gestoppt.

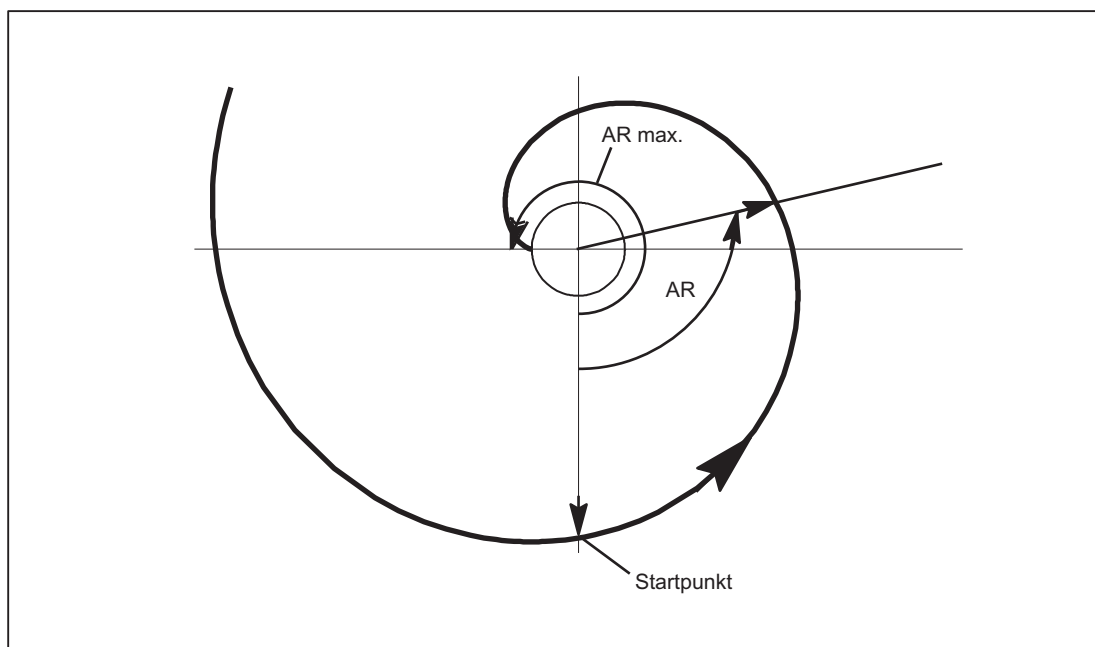


Bild 1-8 Begrenzter Drehwinkel zum Grundkreis hin

Die Anzeige des Alarms kann über folgende Parametrierung unterdrückt werden:

MD21016 \$MC\_INVOLUTE\_AUTO\_ANGLE\_LIMIT = TRUE (automatische Winkelbegrenzung bei Evolventen-Interpolation)

Dann wird der programmierte Drehwinkel gegebenenfalls automatisch begrenzt und die interpolierte Bahn endet an dem Punkt an dem die Evolvente den Grundkreis trifft. Dies erlaubt die einfache Programmierung einer Evolvente, die von einem Punkt außerhalb des Grundkreises startet und direkt auf dem Grundkreis endet.

## Werkzeugradiuskorrektur

Für Evolventen ist nur die 2-1/2D-Werkzeugradiuskorrektur zulässig. Bei aktiver 3D-Werkzeugradiuskorrektur (sowohl Umfangs- als auch Stirnfräsen) wird bei Programmierung einer Evolvente die Bearbeitung mit Alarm 10782 abgebrochen.

Bei 2-1/2-D-Werkzeugradiuskorrektur muss die Ebene der Evolvente in der Korrekturebene liegen. Andernfalls wird der Alarm 10781 erzeugt. Ein zusätzlicher Helixanteil zu einer in der Korrekturebene liegenden Evolvente ist jedoch zulässig.

## Dynamik

Evolventen, die auf dem Grundkreis beginnen oder enden, haben an dieser Stelle eine unendliche Krümmung. Damit die Geschwindigkeit in diesem Punkt bei aktiver Werkzeugradiuskorrektur hinreichend beschränkt wird, ohne diese an anderen Stellen zu stark zu begrenzen, muss die Funktion "Geschwindigkeitsbegrenzungsprofile" aktiviert werden:

MD28530 \$MC\_MM\_PATH\_VELO\_SEGMENTS > 1 (Anzahl Speicherelemente zur Begrenzung der Bahngeschwindigkeit)

Empfohlen wird der Wert 5. Diese Einstellung ist nicht notwendig, wenn nur Evolventenstücke verwendet werden, deren Krümmungsradien sich über einen relativ kleinen Bereich ändern.

### 1.3.3 DEFAULT-Speicher aktivieren

#### GUD Startwerte

Mit den Sprachbefehlen DEF... / REDEF... können GUD Defaultwerte zugewiesen werden. Wenn diese Defaultwerte nach bestimmten Systemzuständen (z. B. RESET) wieder verfügbar sein sollen, müssen sie im System dauerhaft gespeichert werden.

Der Speicherplatz dafür wird dem Speicherbereich entnommen der zugewiesen wurde über das Maschinendatum:

MD18150 \$MM\_GUD\_VALUES\_MEM (Anzahl von zusätzlichen Parametern laut MD18170)

Die Einstellung für die Aktivierung der gespeicherten Defaultwerte erfolgt über das Maschinendatum:

MD11270 \$MN\_DEFAULT\_VALUES\_MEM\_MASK (Aktivierung Defaultwerte für NC-Sprachelemente)

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; S7: "Speicherkonfiguration"

Programmieranleitung Arbeitsvorbereitung

### 1.3.4 PLC-Variable lesen und schreiben

#### Schneller Datenkanal

Für den schnellen Austausch von Informationen zwischen PLC und NC wird in dem Koppelspeicher dieser Baugruppen (DPR) ein Speicherbereich reserviert. In diesem Speicherbereich können beliebige PLC-Variablen (E/A, DB, DW, Merker) ausgetauscht werden.

Auf diesen Speicher wird von der PLC mit 'FunctionCalls' (FC), von der NC mit Systemvariablen zugegriffen.

#### Organisation des Speicherbereichs

Für die Organisation (Struktur) des Speicherbereiches ist der Anwenderprogrammierer (NC und PLC) selbst verantwortlich.

Dabei kann jede beliebige Speicherstelle angesprochen werden, jedoch muss dabei die Grenze entsprechend dem Datenformat gewählt werden (ein DWORD also auf einer 4-Byte-Grenze, ein WORD auf einer 2-Byte-Grenze ...).

Der Zugriff in den Speicherbereich erfolgt über den Datentyp und den Positionsoffset innerhalb des Speicherbereichs.

#### Zugriff von NC

Für den schnellen Zugriff aus einem Teileprogramm oder Synchronaktion auf PLC-Variable, werden in der NC Systemvariablen zur Verfügung gestellt. Das Schreiben / Lesen der Daten durch die NC erfolgt direkt. Der Datentyp ergibt sich aus dem Bezeichner der Systemvariablen. Die Position innerhalb des Speicherbereichs wird als Index in Byte angegeben.

Systemvariable	Datentyp	Wertebereich
\$A_DBB( <Index> )	Byte (8 Bit)	$0 \leq x \leq 255$
\$A_DBW( <Index> )	Wort (16 Bit)	$-32768 \leq x \leq 32767$
\$A_DBD( <Index> )	Doppelwort (32 Bit)	$-2147483648 \leq x \leq 2147483647$
\$A_DBR( <Index> )	Fließkomma (32 Bit)	$\pm(1,5 \cdot 10^{-45} \leq x \leq 3,4 \cdot 10^{38})$

## Zugriff von PLC

Der Zugriff von der PLC erfolgt mittels "FunctionCall" (FC). Im FC werden die Daten sofort und nicht erst bei Zyklusbeginn der PLC, im DPR gelesen bzw. geschrieben. Datentyp und Position im Speicherbereich werden dem FC als Parameter übergeben.

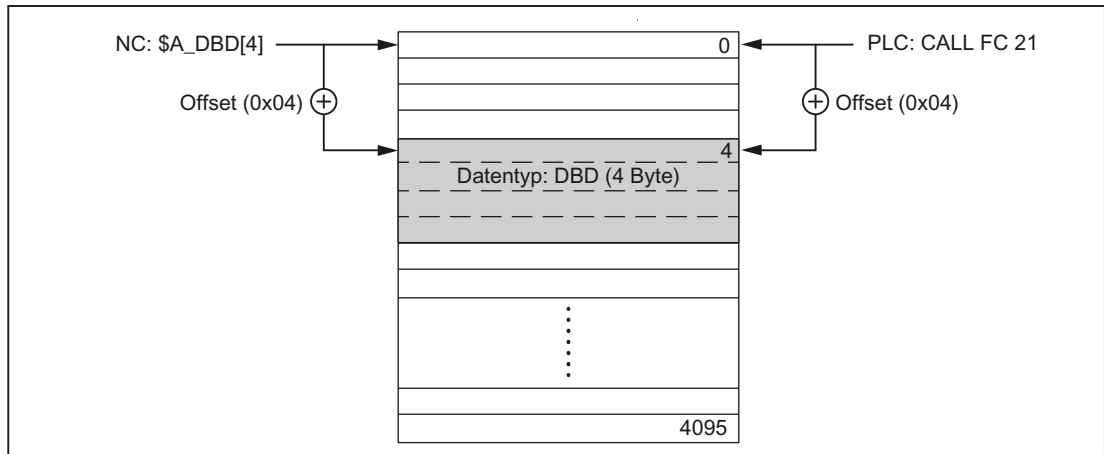


Bild 1-9 Koppelspeicher DPR für die NC/PLC-Kommunikation

## Randbedingungen

- Die Strukturierung des DPR-Speicherbereichs liegt ausschließlich in der Verantwortung der Anwenders. Es werden keine Überprüfungen auf übereinstimmende Projektierung vorgenommen.
- In Ein- und Ausgaberrichtung stehen in Summe 4096 Bytes zur Verfügung.
- Einzelbitoperationen werden nicht unterstützt und müssen vom Anwender auf Byte-Operationen zurückgeführt werden.
- Da die Manipulation der Inhalte der Variablen im Koppelspeicher unmittelbar erfolgt, ist durch den Anwenderprogrammierer darauf zu achten, dass bei mehrfacher Auswertung einer Variablen oder bei der Verknüpfung von Variablen sich die Werte zwischenzeitlich geändert haben (ggf. Werte in lokaler Variable oder R-Parameter zwischenspeichern oder eine Semaphore einrichten).
- Für die Koordination der Zugriffe von verschiedenen Kanälen auf den Koppelspeicher ist der Anwenderprogrammierer verantwortlich.
- Es wird nur für die Datenzugriffe bis 16 Bit (Byte und Word) Datenkonsistenz zugesichert. Für konsistente Übertragung von 32 Bit-Variablen (Double und Real) ist ausschließlich der Anwender verantwortlich. Auf der PLC-Seite steht dazu ein einfacher Semaphore-Mechanismus zur Verfügung.

- Die Daten werden von der PLC im 'Little Endian'-Format im DPR abgelegt.
- Mit \$A\_DBR transferierte Werte unterliegen einer Datenwandlung und damit einem Genauigkeitsverlust. Das Datenformat für Fließkommazahlen ist in der NC DOUBLE (64Bit), in der PLC jedoch nur FLOAT (32Bit). Die Ablage im DPR ist FLOAT. Die Wandlung erfolgt jeweils vor/nach der Ablage im DPR.

Wird z.B. von der NC auf eine Variable im DPR geschrieben und wieder zurückgegeben, wird die Wandlung zweimal durchlaufen. Eine Differenz des geschriebenen/gelesenen Wertes ist aufgrund der Datenablage in den beiden Formaten nicht zu verhindern.

### Beispiel

Umgehung des Problems durch Vergleich auf "EPSILON" (kleine Abweichung)

Programmcode	
N10	DEF REAL DBR
N12	DEF REAL EPSILON = 0.00001
N20	\$A_DBR[0]=145.145
N30	G4 F2
N40	STOPRE
N50	DBR=\$A_DBR[0]
N60	IF ( ABS(DBR/145.145-1.0) < EPSILON ) GOTOF ENDE
N70	MSG ( "Fehler" )
N80	M0
N90	ENDE:
N99	M30

### Aktivierung

Die maximale Anzahl der gleichzeitig schreibbaren Ausgangsvariablen ist einstellbar über: MD28150 \$MC\_MM\_NUM\_VDIVAR\_ELEMENTS (Anzahl Elemente für das schreiben von PLC-Variablen)

**Beispiel**

Es soll eine Variable vom Typ WORD von der PLC an die NC übergeben werden.  
Der Positions-Offset innerhalb des NC-Eingangs (PLC-Ausgangsbereich) soll das 4. Byte sein. Der Positions-Offset muss ein ganzzahliges Vielfaches der Datenbreite sein.

**Schreiben von der PLC:**

Programmcode	Kommentar
. . .	
CALL FC21 (	
Enable :=M10.0,	; wenn TRUE, dann FC21 aktiv
Funct :=B#16#4,	
S7Var :=P#M 104.0 WORD1,	
IVAR1 :=04,	
IVAR2 :=-1,	
Error :=M10.1,	
ErrCode :=MW12);	
. . .	
)	

**Lesen im Teileprogramm**

Programmcode	Kommentar
. . .	
PLCDATA = \$A_DBW[4];	; Lesen eines Wortes
. . .	

**Verhalten bei POWER ON, Satzsuchlauf**

Bei "POWER ON" wird der Koppelspeicher DPR initialisiert.  
Bei "Satzsuchlauf" wird die Ausgabe der PLC-Variablen aufgesammelt und mit dem Anfahrstanz in den Koppelspeicher DPR ausgegeben (analog dem Schreiben von analogen und digitalen Ausgängen).  
Andere Zustandsübergänge haben hier keine Auswirkung.

**Literatur**

Eine ausführliche Beschreibung zum Datenaustausch seitens der PLC mit FC 21 findet sich in:  
SINUMERIK 840D sl: Kapitel "FC 21: Transfer Datenaustausch PLC-NCK (Seite 1138)"



## 1.3.5 Zugriffsschutz über Kennwort und Schlüsselschalter

### 1.3.5.1 Zugriffsschutz über Kennwort und Schlüsselschalter

#### Zugriffsrechte

Der Zugriff auf Funktionen, Programme und Daten und ist benutzerorientiert über 8 hierarchische Schutzstufen geschützt. Diese sind unterteilt in:

- Kennwort-Stufen für Siemens, Maschinenhersteller und Endanwender
- Schlüsselschalter-Stellungen für Endanwender

#### Mehrstufiges Sicherheitskonzept

Über die Kennwort-Stufen und Schlüsselschalter-Stellungen steht ein mehrstufiges Sicherheitskonzept zur Regelung der Zugriffsrechte zur Verfügung.

Schutzstufe	Art	Benutzer	Zugriff auf (Beispiele)
0	Kennwort	Siemens	alle Funktionen, Programme und Daten
1	Kennwort	Maschinenhersteller: Entwicklung	definierte Funktionen, Programme und Daten; z.B.: Optionen eingeben
2	Kennwort	Maschinenhersteller: Inbetriebnehmer	definierte Funktionen, Programme und Daten; z.B.: Großteil der Maschinendaten
3	Kennwort	Endanwender: Service	zugeordnete Funktionen, Programme und Daten
4	Schlüsselschalter-Stellung 3	Endanwender: Programmierer Einrichter	weniger als Schutzstufe 0 bis 3; festgelegt vom Maschinenhersteller oder Endanwender
5	Schlüsselschalter-Stellung 2	Endanwender: Qualifizierter Bediener, der nicht programmiert	weniger als Schutzstufe 0 bis 3; festgelegt vom Endanwender
6	Schlüsselschalter-Stellung 1	Endanwender: Ausgebildeter Bediener, der nicht programmiert	Beispiel: nur Programmanwahl, Werkzeugverschleißeingabe und Eingabe von Nullpunktverschiebungen
7	Schlüsselschalter-Stellung 0	Endanwender: Angelernter Bediener	Beispiel: keine Eingaben und Programmanwahl möglich, nur Maschinensteuertafel bedienbar

### Zugriffsmerkmale

- Die Schutzstufe 0 besitzt die höchsten Zugriffsrechte, die Schutzstufe 7 hat die geringsten Zugriffsrechte.
- Wird einer Schutzstufe ein Zugriffsrecht erteilt, so schließt die höhere Schutzstufe automatisch dieses Zugriffsrecht mit ein.
- Umgekehrt kann ein Zugriffsrecht für eine bestimmte Schutzstufe nur aus einer höheren Schutzstufe heraus geändert werden.
- Die Zugriffsrechte für die Schutzstufen 0 bis 3 werden von Siemens standardmäßig vorgegeben (Default).
- Die Zugriffsberechtigung wird durch Abfrage der aktuellen Schlüsselschalterstellung und durch Vergleich der eingegebenen Kennwörter gesetzt. Dabei überschreibt ein eingegebenes Kennwort die Zugriffsrechte der Schlüsselschalterstellung.
- In jeder Schutzstufe können Optionen gesichert werden. Aber nur in Schutzstufe 0 und 1 können Optionsdaten eingegeben werden.
- Die Zugriffsrechte für die Schutzstufen 4 bis 7 sind Vorschlagswerte und können vom Maschinenhersteller oder Endanwender geändert werden.

#### 1.3.5.2 Kennwort

##### Kennwort setzen

Das Kennwort einer Schutzstufe (0 – 3) wird über die HMI-Bedienoberfläche eingegeben.

Beispiel:

Bedienbereich DIAGNOSE, Softkey: KENNWORT SETZEN

##### Literatur:

Inbetriebnahmehandbuch SINUMERIK 840D sl Basesoftware und HMI sl

##### Kennwort löschen

Die Zugriffsberechtigung durch ein gesetztes Kennwort bleibt so lange wirksam, bis sie explizit durch Löschen des Kennworts wieder zurückgenommen wird.

Beispiel:

Bedienbereich DIAGNOSE, Softkey: KENNWORT LÖSCHEN

##### Literatur:

Inbetriebnahmehandbuch SINUMERIK 840D sl Basesoftware und HMI sl

##### Hinweis

Eine Zugriffsberechtigung bzw. der Status des Kennworts (gesetzt/gelöscht) wird durch POWER OFF/ON nicht beeinflusst!

### Maximale Anzahl von Zeichen

Ein Kennwort kann maximal aus acht Zeichen bestehen. Bei der Wahl des Kennwortes wird empfohlen, sich auf den Zeichenvorrat der Bedientafelfront zu beschränken. Bei einem Kennwort mit weniger als 8 Zeichen werden die restlichen Zeichen als Leerzeichen (Blank) interpretiert.

### Standardeinstellungen

Für die Schutzstufen 1 bis 3 sind als Kennwörter folgende Standardeinstellungen festgelegt:

- Schutzstufe 1: SUNRISE
- Schutzstufe 2: EVENING
- Schutzstufe 3: CUSTOMER

---

#### Hinweis

Nach einem Hochlauf der NC-CPU im Inbetriebnahme-Modus (NCK-Inbetriebnahme-Schalter: Stellung 1) werden die Kennwörter der Schutzstufen 1 – 3 wieder auf die Standardeinstellungen zurückgesetzt. Aus Gründen des Datenschutzes wird dringend empfohlen die Standardeinstellungen zu ändern.

---

### 1.3.5.3 Schlüsselschalter-Stellungen (DB10, DBX56.4 bis 7)

#### Schlüsselschalter

Der Schlüsselschalter verfügt über vier Schalterstellungen, denen die Schutzstufen 4 bis 7 zugeordnet sind. Zum Schlüsselschalter gehören verschiedenfarbige Schlüssel, die in unterschiedlichen Schalterstellungen abgezogen werden können.





Schalterstellung	Abzugsstellung	DB10, DBB56	Schutzstufe
Stellung 0 	-	Bit 4	7
Stellung 1 	0 oder 1 schwarzer Schlüssel	Bit 5	6
Stellung 2 	0 oder 1 oder 2 grüner Schlüssel	Bit 6	5
Stellung 3 	0 oder 1 oder 2 oder 3 roter Schlüssel	Bit 7	4

Bild 1-10 Schalterstellungen 0 bis 3

### Schalterstellungen

Schalterstellung 0 hat die geringsten Zugriffsrechte. Schalterstellung 3 hat die höchsten Zugriffsrechte:

DB10, DBX56.4 / .5 / .6 / .7 (Schalterstellungen 0 / 1 / 2 / 3)

Den Schalterstellungen können maschinenspezifische Freigaben für den Zugriff auf Programme, Daten und Funktionen zugeordnet werden. Ausführliche Informationen dazu finden sich in:

#### Literatur

- Inbetriebnahmehandbuch CNC: NCK, PLC, Antriebe; Grundlagen, Kapitel: Prinzipielles zu den Schutzstufen
- Inbetriebnahmehandbuch SINUMERIK Operate (IM9); Allgemeine Einstellungen, Kapitel: Zugriffsstufen

### Vorgaben durch das PLC-Anwenderprogramm

Die Schalterstellungen des Schlüsselschalters werden durch PLC-Grundprogramm in die NC/PLC-Nahtstelle übertragen. Die entsprechenden Nahtstellensignale können durch das PLC- Anwenderprogramm verändert werden. Dabei sollte aus Sicht der NC immer nur eine Schalterstellung aktiv, d.h. das entsprechende Nahtstellensignal auf den Wert 1 gesetzt sein. Sind aus Sicht der NC gleichzeitig mehrere Schalterstellungen aktiv, wird NC-intern die Schalterstellung 3, d.h. die Schlüsselschalterstellung mit den höchsten Zugriffsrechten aktiv.

#### 1.3.5.4 Parametrierbare Schutzstufen

##### Parametrierbare Schutzstufe

Für verschiedene Funktionen und Datenbereiche kann die Schutzstufe frei parametrierbar werden. Die Einstellung der Schutzstufe erfolgt über Bedientafel-Maschinendaten mit folgender Bezeichnungssystematik: \$MM\_USER\_CLASS\_<Funktion\_Datenbereich>

Beispiele:

\$MM_USER_CLASS_READ_TOA	Werkzeugkorrekturen lesen
\$MM_USER_CLASS_WRITE_TOA	Werkzeugkorrekturen schreiben
\$MM_USER_CLASS_READ_PROGRAM	Teileprogramme lesen
\$MM_USER_CLASS_WRITE_PROGRAM	Teileprogramme schreiben/editieren

##### Standardwerte

Bei Auslieferung oder nach einer Standard-Inbetriebnahme ist der Standardwert für die Schutzstufe bis auf wenige Ausnahmen auf 7, d.h. niedrigste Schutzstufe eingestellt.

### 1.3.6 "Parken" einer Maschinenachse

Im Zustand "Parken" kann eine Maschinenachse, ohne einen Alarm auszulösen, mechanisch bewegt oder gewartet (z. B. Gebertausch) werden. Dazu sind die achsspezifischen NC/PLC-Nahtstellensignale für das aktive Lagemesssystem und der Reglerfreigabe der Maschinenachse zurückzusetzen:

- DB31, ... DBX1.5 / .6 = 0 (Lagemesssystem 1 / 2)
- DB31, ... DBX2.1 = 0 (Reglerfreigabe)

Der Geberstatus des aktiven Messsystems der Achse wird daraufhin als "Nicht referenziert" angezeigt:

- DB31, ... DBX60.4 / .5 == 0 (Referenziert/Synchronisiert, Geber 1 / 2)

### "Parken" aufheben

Der Zustand "Parken" wird durch Setzen der NC/PLC-Nahtstellensignale wieder aufgehoben:

- DB31, ... DBX1.5 / .6 = 1 (Lagemesssystem 1 / 2)
- DB31, ... DBX2.1 = 1 (Reglerfreigabe)

Die Lageregelung der Maschinenachse wird daraufhin an der aktuellen Position wieder aktiv.

Der Geberstatus des Messsystems ist abhängig vom Messsystemtyp:

- Inkrementelles Messsystem: Zustand "Nicht referenziert"  
DB31, ... DBX60.4 / .5 == 0 (Referenziert/Synchronisiert, Geber 1 / 2)
- Absolutes Messsystem: Zustand "Referenziert"  
DB31, ... DBX60.4 / .5 == 1 (Referenziert/Synchronisiert, Geber 1 / 2)

#### Inkrementelle Messsysteme: Geberstatus "Referenziert"

Zum Erreichen des Geberstatus "Referenziert", müssen inkrementelle Messsysteme neu referenziert werden.

#### **WARNUNG**

Wurden während des Parkens Veränderungen am Messsystem vorgenommen, die eine Änderung an den parametrisierten Maschinendaten erforderlich machen, z.B. Anbau eines anderen Gebers, muss das Messsystem vollständig neu vermessen und referenziert werden. Siehe Kapitel "R1: Referenzieren (Seite 1319)".

## 1.3.7 Umschaltung von Motor- / Antriebsparameterdatensätzen

### 1.3.7.1 Allgemeine Informationen

#### Verhalten ab SW 4.5 SP1

Bis einschließlich SW 4.5 war das Format der Schnittstellen zur Motor- und Antriebsparameterdatensatzumschaltung in der NC/PLC-Nahtstelle fest vorgegeben.

Ab SW 4.5 SP1 wurde das Mengengerüst der parametrierbaren Motordatensätze und Antriebsparameterdatensätze pro Motordatensatz erhöht und damit einhergehend die starre Zuordnung der Signale zur Motor- und Antriebsparameterdatensatzumschaltung in der NC/PLC-Nahtstellensignale flexibilisiert.

#### Motor- und Antriebsparameterdatensätzen

Zur optimalen Anpassung an die jeweilige Bearbeitungssituation oder aufgrund unterschiedlicher Maschinenkonfigurationen kann es erforderlich sein, dass in einem Antrieb für Motoren, Antriebsparameter und Geber jeweils mehrere unterschiedliche Datensätze vorliegen. Die Erstellung der grundlegenden Datensätze der Antriebsobjekte erfolgt während der Inbetriebnahme mit Hilfe des "Antriebsassistenten".

---

#### Hinweis

#### Literatur

Inbetriebnahmehandbuch: CNC: NCK, PLC- Antrieb, Kapitel "Inbetriebnahme NC-gesteuerte Antriebe"

---

Die nachfolgende Duplizierung und Verwaltung der Datensätze erfolgt über die Bedienoberfläche:

SINUMERIK Operate: Bedienbereich "Inbetriebnahme" > "Antriebssystem" > "Antriebe" > "Datensätze"

Die Aktivierung des für eine Maschinenachse in der konkreten Bearbeitungssituation erforderlichen Motor- (MDS) bzw. Antriebsparameterdatensatzes (DDS), muss vom PLC-Anwenderprogramm über die im Folgenden beschriebenen Schnittstellen vorgenommen werden.

#### Axiale NC/PLC-Nahtstelle

Die Schnittstellen in der axiale NC/PLC-Nahtstelle zur Umschaltung der Motor- und Antriebsparameterdatensätze ist in drei Bereiche unterteilt:

- Gültigkeit und Format der Anforderungs- / Anzeige-Schnittstellen (Seite 71)
- Anforderung eines neuen Motor- und/oder Antriebsparameterdatensatzes (Seite 72)
- Anzeige des aktiven Motor- und/oder Antriebsparameterdatensatzes (Seite 72)

### 1.3.7.2 Gültigkeit und Format der Anforderungs- / Anzeige-Schnittstellen

#### Gültigkeit

Sobald im Hochlauf der Steuerung alle erforderlichen Informationen vom Antrieb übermittelt und von der NC ausgewertet wurden, werden die Anforderungs- und Anzeige-Schnittstellen als gültig angezeigt:

DB31, ... DBX130.7 == 1 (Schnittstelle ist gültig)

Werden vom Antrieb keine oder inkompatible Informationen übertragen, bleiben die Anforderungs- und Anzeige-Schnittstellen ungültig.

---

#### Hinweis

Bei ungültigen Anforderungs- und Anzeige-Schnittstellen liegt es in der alleinigen Verantwortung des Anwenders / Maschinenherstellers über die Anforderungs- und Anzeige-Schnittstellen eine Datensatzumschaltung vorzunehmen.

---

#### Format

Das aktuelle Format der Schnittstellen, d.h. welche der fünf Bit der Anforderungs- und Anzeige-Schnittstellen für die Adressierung des Motor- und welche für den Antriebsparameterdatensatz verwendet werden, ist abhängig von der Anzahl der im Antrieb vorhandenen Motor- und Antriebsparameterdatensätze. Die Formatangabe erfolgt über:

DB31, ... DBX130.0-4, mit Bit x = <Wert>

<Wert>	Bedeutung
0	Bit-Stelle für Motordatensatz-Umschaltung (MDS) bzw. Ungültige Bit-Stelle
1	Bit-Stelle für Antriebsparameterdatensatz-Umschaltung (DDS)

#### Siehe auch

Beispiel (Seite 73)

Übersicht der Schnittstellen (Seite 74)

### 1.3.7.3 Anforderung eines neuen Motor- und/oder Antriebsparameterdatensatzes

Die Anforderung zum Aktivieren eines bestimmten Motor- und Antriebsparameterdatensatzes erfolgt über :

DB31, ... DBX21.0 - .4 = <MDS / DDS-Index>

#### Wertebereich

Die Adressierung eines Datensatz erfolgt über den Index i, mit i = 0, 1, 2, ...

Datensatz	Wertebereich, Index i
MDS[ i ]	$0 \leq i \leq 32$
DDS[ i ]	$0 \leq i \leq 32$

#### Formatierung

Die Formatierung der Schnittstelle, d.h. welche Bits für den Motor- und welche für die des Antriebsparameterdatensatz-Index zur Verfügung stehen, wird angezeigt über:

DB31, ... DBX130.0 - .4 (siehe Kapitel "Gültigkeit und Format der Anforderungs- / Anzeige-Schnittstellen (Seite 71)")

#### Konkrete Anzahl von Datensätzen

Die konkrete Anzahl von im Antrieb vorhandenen Motor- und Antriebsparameterdatensätzen kann über folgende Antriebsparameter ermittelt werden:

- p0130 (Anzahl Motordatensätze)
- p0180 (Anzahl Antriebsparameterdatensätze)

### 1.3.7.4 Anzeige des aktiven Motor- und/oder Antriebsparameterdatensatzes

#### Anzeige des aktiven Motor- und/oder Antriebsparameterdatensatzes

Der Index eines aktiven Datensatzes kann gelesen werden über:

DB31, ... DBX93.0 - .4

Wertebereich und Formatierung sind identisch zur Anforderungsschnittstelle. Siehe Kapitel "Anforderung eines neuen Motor- und/oder Antriebsparameterdatensatzes (Seite 72)"



### 1.3.7.5 Beispiel

Im Antrieb sind zwei Motordatensätze (MDS) und zwei Antriebsparameterdatensätze (DDS) vorhanden. Dies entspricht "Nr.": 9 der im Bild 1-11 Motor- / Antriebsparameterdatensatz-Umschaltung (Seite 74) dargestellten möglichen Datensatzkombinationen.

#### Format

Bit-Stellen für Antriebsdatensatz-Umschaltung (DDS):

- DB31, ... DBX130.0 == 1

Bit-Stellen für Motordatensatz -Umschaltung (MDS):

- DB31, ... DBX130.1 == 0

Ungültige Bit-Stellen:

- DB31, ... DBX130.2 == 0
- DB31, ... DBX130.3 == 0
- DB31, ... DBX130.4 == 0

#### Schnittstellen der Antriebsparameterdatensätze (DDS)

Relevante Bit-Stellen der Anforderungs- und Anzeige-Schnittstellen:

- DB31, ... DBX21.0 / DBX93.0
  - DB31, ... DBX21.0 / DBX93.0 == 0  $\Rightarrow$  1. Antriebsparameterdatensatz DDS[0]
  - DB31, ... DBX21.0 / DBX93.0 == 1  $\Rightarrow$  2. Antriebsparameterdatensatz DDS[1]

#### Schnittstellen der Motordatensätze (MDS)

Relevante Bit-Stellen der Anforderungs- und Anzeige-Schnittstellen:

- DB31, ... DBX21.1 / DBX93.1
  - DB31, ... DBX21.1 / DBX93.1 == 0  $\Rightarrow$  1. Motordatensatz MDS[0]
  - DB31, ... DBX21.1 / DBX93.1 == 1  $\Rightarrow$  2. Motordatensatz MDS[1]

#### Ungültige Bit-Stellen (MDS / DDS)

Ungültige Bit-Stellen der Anforderungs- und Anzeige-Schnittstellen:

- DB31, ... DBX21.1 / DBX93.2
- DB31, ... DBX21.1 / DBX93.3
- DB31, ... DBX21.1 / DBX93.4

#### Siehe auch

Übersicht der Schnittstellen (Seite 74)

1.3.7.6 Übersicht der Schnittstellen

Nr.	DDS pro MDS		DB31, ... DBX21.x / 93.x					DB31, ... DBX130.x				
	MDS		4	3	2	1	0	4	3	2	1	0
1	1	1						0	0	0	0	0
2	2	1						0	0	0	0	0
3	3	1						0	0	0	0	0
4	4	1						0	0	0	0	0
5	8	1						0	0	0	0	0
6	16	1						0	0	0	0	0
7	32	1						0	0	0	0	0
8	1	2						0	0	0	0	1
9	2	2						0	0	0	0	1
10	3	2						0	0	0	0	1
11	4	2						0	0	0	0	1
12	8	2						0	0	0	0	1
13	16	2						0	0	0	0	1
14	1	4						0	0	0	1	1
15	2	4						0	0	0	1	1
16	3	4						0	0	0	1	1
17	4	4						0	0	0	1	1
18	8	4						0	0	0	1	1
19	1	8						0	0	1	1	1
20	2	8						0	0	1	1	1
21	3	8						0	0	1	1	1
22	4	8						0	0	1	1	1
23	1	16						0	1	1	1	1
24	2	16						0	1	1	1	1
25	1	32						1	1	1	1	1

Unterstützte Kombination

Relevante Bit-Stelle bezüglich:

Motor Datensatz (MDS)

Antriebsparameter Datensatz (DDS)

Ungültige Bit-Stelle

MDS                                    Anzahl Motordatensätze  
 DDS pro MDS                        Anzahl Antriebsparameterdatensätze pro Motordatensatz  
 DB31, ... DBX21.x                Anforderungsschnittstelle der zu aktivierenden Datensätze  
 DB31, ... DBX93.x                Anzeigeschnittstelle der aktiven Datensätze  
 DB31, ... DBX130.x              Schnittstelle zur Anzeige der Formatierung der Anforderungs- und Anzeigeschnittstellen

Bild 1-11      Motor- / Antriebsparameterdatensatz-Umschaltung

### 1.3.7.7 Randbedingungen

#### Variable Anzahl Antriebsparameterdatensätze für den "letzten" Motordatensatz

Der "letzte" Motordatensatz ist der Motordatensatz mit der höchsten Nummer bzw. Index.

Im Allgemeinen gilt, dass im Antrieb für jeden Motordatensatz die gleiche Anzahl von Antriebsparameterdatensätzen (Anzahl "DDS pro MDS") angelegt wird. Nur für den "letzten" Motordatensatz kann davon abweichend, eine beliebige Anzahl von Antriebsparameterdatensätzen  $a$  parametrierbar sein:

$$1 \leq a \leq (\text{Anzahl "DDS pro MDS"})$$

#### Beispiel

Es sollen 4 Motordatensätze (MDS) und 8 Antriebsparameterdatensätze (DDS) pro Motordatensatz (DDS pro MDS) parametrierbar sein. Dies entspricht "Nr.": 22 der im Bild 1-11 Motor- / Antriebsparameterdatensatz-Umschaltung (Seite 74) dargestellten möglichen Datensatzkombinationen:

- Motordatensätze: MDS[ 0 ], MDS[ 1 ], ... MDS[ 3 ] ("letzter" Motordatensatz)
- Antriebsparameterdatensätzen pro Motordatensatz: DDS[ 0 ] ... DDS[ 7 ]

Die Anzahl Antriebsparameterdatensätze für die einzelnen Motordatensätze ist somit:

Motordatensatz	Anzahl Antriebsparameterdatensätze pro Motordatensatz
MDS[ 0 ] ... MDS[ 2 ]	8 DDS
MDS[ 3 ]	1 - 8 DDS

#### Siehe auch

Übersicht der Schnittstellen (Seite 74)

## 1.4 Beispiele

### Parametersatzumschaltung

Über eine Parametersatzumschaltung wird für die Maschinenachse X1 der Verstärkungsfaktor der Lageregelung (Kv-Faktor) von  $K_v = 4.0$  auf  $K_v = 0.5$  umgeschaltet.

### Voraussetzungen

Die Parametersatzumschaltung muss freigegeben sein durch das Maschinendatum:

MD35590 \$MA\_PARAMSET\_CHANGE\_ENABLE [AX1] = 1 oder 2 (Parametersatzwechsel möglich)

Angewählt ist der 1. Parametersatz der Maschinenachse X1, entsprechend den Maschinendaten mit dem Index "0". NC/PLC-Nahtstelle:

DB31, ... DBX9.0 - DBX9.2 = 0 (Regler Parametersatz)

### Parametersatz-abhängige Maschinendaten

Die Parametersatz-abhängigen Maschinendaten sind wie folgt eingestellt:

Maschinendatum	Bemerkung
MD32200 \$MA_POSCTRL_GAIN [0, AX1] = 4.0	Kv -Einstellung für Parametersatz 1
MD32200 \$MA_POSCTRL_GAIN [1, AX1] = 2.0	Kv -Einstellung für Parametersatz 2
MD32200 \$MA_POSCTRL_GAIN [2, AX1] = 1.0	Kv -Einstellung für Parametersatz 3
MD32200 \$MA_POSCTRL_GAIN [3, AX1] = 0.5	Kv -Einstellung für Parametersatz 4
MD32200 \$MA_POSCTRL_GAIN [4, AX1] = 0.25	Kv -Einstellung für Parametersatz 5
MD32200 \$MA_POSCTRL_GAIN [5, AX1] = 0.125	Kv -Einstellung für Parametersatz 6
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [0, AX1] = 3	Nenner Lastgetriebe für Parametersatz 1
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [1, AX1] = 3	Nenner Lastgetriebe für Parametersatz 2
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [2, AX1] = 3	Nenner Lastgetriebe für Parametersatz 3
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [3, AX1] = 3	Nenner Lastgetriebe für Parametersatz 4
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [4, AX1] = 3	Nenner Lastgetriebe für Parametersatz 5
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [5, AX1] = 3	Nenner Lastgetriebe für Parametersatz 6

Maschinendatum	Bemerkung
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [0, AX1] = 5	Zähler Lastgetriebe für Parametersatz 1
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [1, AX1] = 5	Zähler Lastgetriebe für Parametersatz 2
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [2, AX1] = 5	Zähler Lastgetriebe für Parametersatz 3
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [3, AX1] = 5	Zähler Lastgetriebe für Parametersatz 4
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [4, AX1] = 5	Zähler Lastgetriebe für Parametersatz 5
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [5, AX1] = 5	Zähler Lastgetriebe für Parametersatz 6
MD35130 \$MA_AX_VELO_LIMIT [0..5, AX1]	Einstellung für jeden Parametersatz*)
MD32800 \$MA_EQUIV_CURRCTRL_TIME [0..5, AX1]	Einstellung für jeden Parametersatz*)
MD32810 \$MA_EQUIV_SPEEDCTRL_TIME [0..5, AX1]	Einstellung für jeden Parametersatz*)
MD32910 \$MA_DYN_MATCH_TIME [0..5, AX1]	Einstellung für jeden Parametersatz*)

\*) Die entsprechende Zeile ist für jeden Parametersatz nach den Syntaxregeln gesondert anzugeben.

## Umschaltung

Zur Umschaltung des Verstärkungsfaktor der Lageregelung wird vom PLC-Anwenderprogramm für die Maschinenachse X1 der 4. Parametersatz angewählt.

- Anforderung durch PLC-Anwenderprogramm:
  - DB31, ... DBX9.0 – DBX9.2 = 3 (Parametersatz Servo)
    - Für die Maschinenachse AX1 wird die Umschaltung auf den 4. Parametersatz angefordert.
    - Nach einer Verzögerungszeit wird der Parametersatz umgeschaltet.
    - Aktiv ist jetzt der Parametersatz 4, entsprechend den Maschinendaten mit dem Index "3"
- Rückmeldung durch NC:
  - DB31, ... DBX69.0 – DBX69.2 = 3 (Parametersatz Servo)
    - Die Umschaltung des Parametersatzes wird von der NC bestätigt / quittiert.

## 1.5 Datenlisten

### 1.5.1 Maschinendaten

#### 1.5.1.1 Anzeige-Maschinendaten

Nummer	Bezeichner: \$MM_	Beschreibung
<b>SINUMERIK Operate</b>		
9000	LCD_CONTRAST	Kontrast
9001	DISPLAY_TYPE	Monitortyp
9004	DISPLAY_RESOLUTION	Anzeigefeinheit

#### 1.5.1.2 NC-spezifischen Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10350	FASTIO_DIG_NUM_INPUTS	Anzahl der aktiven digitalen NCK-Eingangsbytes
10360	FASTIO_DIG_NUM_OUTPUTS	Anzahl der aktiven digitalen NCK-Ausgangsbytes
10361	FASTIO_DIG_SHORT_CIRCUIT	Kurzschluss digitaler ein- und Ausgänge
11120	LUD_EXTENDED_SCOPE	Programmglobale Variablen aktivieren (PUD)
11270	DEFAULT_VALUES_MEM_MSK	Aktiv. Funktion: DEFAULT-Werte von GUD speichern.
18150	MM_GUD_VALUES_MEM	Speicherplatz für GUD reservieren

#### 1.5.1.3 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
21015	INVOLUTE_RADIUS_DELTA	NC-Startsperre ohne Referenzpunkt
21016	INVOLUTE_AUTO_ANGLE_LIMIT	Automatische Winkelbegrenzung bei Evolventen-Interpolation
27800	TECHNOLOGY_MODE	Technologie im Kanal
28150	MM_NUM_VDIVAR_ELEMENTS	Anzahl Schreibelemente für PLC-Variable
28530	MM_PATH_VELO_SEGMENTS	Anzahl von Speicherelementen zur Begrenzung der Bahngeschwindigkeit im Satz

#### 1.5.1.4 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30350	SIMU_AX_VDI_OUTPUT	Ausgabe der Achssignale bei Simulationsachsen
33050	LUBRICATION_DIST	Schmierimpulsdistanz
35590	PARAMSET_CHANGE_ENABLE	Parametersatzvorgabe durch PLC möglich
36060	STANDSTILL_VELO_TOL	Maximale Geschwindigkeit/Drehzahl bei Achse/Spindel steht
36610	AX_EMERGENCY_STOP_TIME	Zeitdauer der Bremsrampe bei Fehlerzuständen
36620	SERVO_DISABLE_DELAY_TIME	Abschaltverzögerung Reglerfreigabe

## 1.5.2 Systemvariablen

Bezeichner	Beschreibung
\$P_FUMB	freier Teileprogrammspeicher (Free User Memory Buffer)
\$A_DBB[n]	Datum auf der PLC (Daten vom Type BYTE)
\$A_DBW[n]	Datum auf der PLC (Daten vom Type WORD)
\$A_DBD[n]	Datum auf der PLC (Daten vom Type DWORD)
\$A_DBR[n]	Datum auf der PLC (Daten vom Type REAL)

## 1.5.3 Signale

### 1.5.3.1 Signale an NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Schlüsselschalter-Stellung 0 bis 3	DB10.DBX56.4-7	DB2600.DBX0.4-7

### 1.5.3.2 Signale von NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Ferndiagnose aktiv (HMI-Alarm steht an)	DB10.DBX103.0	-
AT-Box ready	DB10.DBX103.5	-
HMI Temperaturgrenze	DB10.DBX103.6	-
HMI Batteriealarm	DB10.DBX103.7	-
NCK--Ready	DB10.DBX104.7	-
HMI2-CPU-Ready E_MMC2 Ready	DB10.DBX108.1	-
HMI-CPU1-Ready (HMI an MPI)	DB10.DBX108.2	-
HMI1-CPU an BTSS ready	DB10.DBX108.3	DB2700.DBX2.3
Antriebe im zyklischen Betrieb	DB10.DBX108.5	DB2700.DBX2.5
Antriebe-Ready	DB10.DBX108.6	DB2700.DBX2.6
NC-Ready	DB10.DBX108.7	DB2700.DBX2.7
NCK-Alarm steht an	DB10.DBX109.0	DB2700.DBX3.0
Kühlk.-Temperatur-Alarm NCU	DB10.DBX109.5	-
Luft-Temperatur-Alarm	DB10.DBX109.6	DB2700.DBX3.6
NCK-Batterie-Alarm	DB10.DBX109.7	-



### 1.5.3.3 Signale an Bedientafelfront

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Bildschirm hell	DB19.DBX0.0	-
Bildschirm dunkel	DB19.DBX0.1	-
Tastensperre	DB19.DBX0.2	DB1900.DBX5000.2
Cancelalarme löschen (nur HMI Advanced)	DB19.DBX0.3	-
Recallalarme löschen (nur HMI Advanced)	DB19.DBX0.4	-
Istwert in WKS	DB19.DBX0.7	DB1900.DBX5000.7
Teileprogramm entladen	DB19.DBX13.5	-
Teileprogramm laden	DB19.DBX13.6	-
Teileprogramm Anwahl	DB19.DBX13.7	DB1700.DBX1000.7
Filesystem aktiv/passiv	DB19.DBX14.7	-
Teileprogrammhandling: Nummer der Steuerdaten	DB19.DBX16.7	DB1700.DBX1001.7 == 1
BA-Wechselsperre	DB19.DBX44.0	-

### 1.5.3.4 Signale von Bedientafelfront

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Bildschirm ist dunkel	DB19.DBX20.1	-
MKS/WKS umschalten	DB19.DBX20.7	DB1900.DBX0.7
Error (Teileprogramm-Handling-Status)	DB19.DBX26.0	DB1700.DBX2000.2
O.K. (Teileprogramm-Handling-Status)	DB19.DBX26.1	DB1700.DBX2000.1
Aktiv (Teileprogramm-Handling-Status)	DB19.DBX26.3	DB1700.DBX2000.3
Entladen (Teileprogramm-Handling-Status)	DB19.DBX26.5	DB1700.DBX2000.5
Laden (Teileprogramm-Handling-Status)	DB19.DBX26.6	DB1700.DBX2000.6
Anwahl (Teileprogramm-Handling-Status)	DB19.DBX26.7	DB1700.DBX2000.7
FC9: Start Messen im Jog	DB19.DBX42.0	-
FC9 Out: Active	DB19.DBX45.0	-
FC9 Out: Done	DB19.DBX45.1	-
FC9 Out: Error	DB19.DBX45.2	-
FC9 Out: StartErr	DB19.DBX45.3	-

### 1.5.3.5 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Restweg löschen (kanalspezifisch)	DB21, ... .DBX6.2	DB3200.DBX6.2

### 1.5.3.6 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Kanalspezifischer NCK-Alarm steht an	DB21, ... .DBX36.6	DB3300.DBX4.6
NCK-Alarm mit Bearbeitungsstillstand steht an	DB21, ... .DBX36.7	DB3300.DBX4.7
Überspeichern aktiv	DB21, ... .DBX318.7	-

### 1.5.3.7 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Achsen-/Spindelsperre	DB31, ... .DBX1.3	DB380x.DBX1.3
Nachführbetrieb	DB31, ... .DBX1.4	DB380x.DBX1.4
Lagemesssystem 1	DB31, ... .DBX1.5	DB380x.DBX1.5
Lagemesssystem 2	DB31, ... .DBX1.6	DB380x.DBX1.6
Reglerfreigabe	DB31, ... .DBX2.1	DB380x.DBX2.1
Restweg löschen (achsspezifisch)/Spindel-Reset	DB31, ... .DBX2.2	DB380x.DBX2.2
Motor- / Antriebsparameterdatensatz: Anwahl	DB31, ... .DBX21.0-4	DB380x.DBX4001.0-4
Motoranwahl erfolgt	DB31, ... .DBX21.5	-
Integratorsperre n-Regler	DB31, ... .DBX21.6	DB380x.DBX4001.6
Impulsfreigabe	DB31, ... .DBX21.7	DB380x.DBX4001.7

## 1.5.3.8 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referenziert / synchronisiert, Geber 1 / 2	DB31, ... .DBX60.4 / 5	DB390x.DBX0.4 / 5
Fahrbefehl minus / plus	DB31, ... .DBX64.6 / 7	DB390x.DBX4.6 / 7
Nachführen aktiv	DB31, ... .DBX61.3	DB390x.DBX1.3
Achse/Spindel steht ( $n < n_{\min}$ )	DB31, ... .DBX61.4	DB390x.DBX1.4
Lageregler aktiv	DB31, ... .DBX61.5	DB390x.DBX1.5
Drehzahlregler aktiv	DB31, ... .DBX61.6	DB390x.DBX1.6
Stromregler aktiv	DB31, ... .DBX61.7	DB390x.DBX1.7
Schmierimpuls	DB31, ... .DBX76.0	DB390x.DBX1002.0
Aktiver Motor- / Antriebsparameterdatensatz	DB31, ... .DBX93.0-4	DB390x.DBX4001.0-4
Drive Ready	DB31, ... .DBX93.5	DB390x.DBX4001.5
Integrator n-Regler gesperrt	DB31, ... .DBX93.6	DB390x.DBX4001.6
Impulse freigegeben	DB31, ... .DBX93.7	DB390x.DBX4001.7
Motor-Temperatur-Vorwarnung	DB31, ... .DBX94.0	DB390x.DBX4002.0
Kühlkörper-Temperatur-Vorwarnung	DB31, ... .DBX94.1	DB390x.DBX4002.1
Hochlaufvorgang beendet	DB31, ... .DBX94.2	DB390x.DBX4002.2
$ M_d  < M_{dx}$	DB31, ... .DBX94.3	DB390x.DBX4002.3
$ n_{\text{ist}}  < n_{\min}$	DB31, ... .DBX94.4	DB390x.DBX4002.4
$ n_{\text{ist}}  < n_x$	DB31, ... .DBX94.5	DB390x.DBX4002.5
$n_{\text{ist}} = n_{\text{soll}}$	DB31, ... .DBX94.6	DB390x.DBX4002.6
Schnittstelle Motor- / Antriebsparameterdatensatz: Formatierung	DB31, ... .DBX130.0-4	DB390x.DBX4008.0-4
Schnittstelle Motor- / Antriebsparameterdatensatz: Gültig	DB31, ... .DBX130.7	DB390x.DBX4008.7



## A3: Achsüberwachungen, Schutzbereiche

### 2.1 Kurzbeschreibung

#### 2.1.1 Achsüberwachungen

Zum Schutz von Mensch und Maschine sind in der Steuerung umfangreiche Überwachungsfunktionen vorhanden:

- Konturüberwachung
- Positionierüberwachung
- Stillstandsüberwachung
- Klemmungsüberwachung
- Drehzahlsollwertüberwachung
- Istgeschwindigkeitsüberwachung
- Messsystem-Überwachung
- Endschalte-Überwachung
- Überwachung der Arbeitsfeldbegrenzung

#### 2.1.2 Schutzbereiche

Mit Hilfe von Schutzbereichen können Elemente der Maschine (z. B. Spindelfutter, Werkzeugwechsler, Werkzeugträger, Reitstock, einschwenkbarer Messtaster etc.) und das Werkstück vor Kollisionen geschützt werden.

Beim automatischen Abarbeiten von Teileprogrammen in der Betriebsart AUTOMATIK bzw. MDA prüft die NC am Anfang jedes Teileprogrammsatzes, ob es beim Verfahren der programmierten Bahn zu Kollisionen von Schutzbereichen kommen würde.

Nach einer manuellen Deaktivierung eines aktiven Schutzbereichs kann in diesen gefahren werden. Nach Verlassen des Schutzbereichs wird der Schutzbereich automatisch wieder aktiv.

Die Definition, Aktivierung und Deaktivierung von Schutzbereichen erfolgt über Teileprogrammanweisungen.

## 2.2 Achsüberwachungen

### 2.2.1 Konturüberwachung

#### 2.2.1.1 Konturfehler

Konturfehler entstehen durch Signalverzerrungen im Lageregelkreis.

Man unterscheidet lineare und nichtlineare Signalverzerrungen.

#### Lineare Signalverzerrungen

Lineare Signalverzerrungen entstehen durch:

- Nicht optimal eingestellte Drehzahl- bzw. Lageregler
- Ungleiche  $K_v$ -Faktoren der an der Bahnerzeugung beteiligten Vorschubachsen

Bei gleichem  $K_v$ -Faktor zweier linear interpolierender Achsen folgt der Istpunkt dem Sollpunkt auf gleicher Bahn, jedoch zeitlich verzögert. Bei ungleichem  $K_v$ -Faktor entsteht ein Parallelversatz zwischen Soll- und Istbahn.

- Ungleiche Dynamik der Vorschubantriebe

Eine ungleiche Antriebsdynamik führt insbesondere an Konturänderungen zu Bahnabweichungen. Kreise werden durch eine ungleiche Dynamik der beiden Vorschubantriebe zu Ellipsen verzerrt.

#### Nichtlineare Signalverzerrungen

Nichtlineare Signalverzerrungen entstehen durch:

- Wirksamwerden der Strombegrenzung innerhalb des Bearbeitungsbereichs
- Wirksamwerden der Begrenzung des Drehzahlsollwerts
- Umkehrspanne innerhalb und/oder außerhalb des Lageregelkreises

Beim Durchfahren einer Kreisbahn treten vor allem Konturfehler durch die Umkehrspanne und durch Reibung auf.

Beim Fahren von Geraden entsteht ein Konturfehler durch eine Umkehrspanne außerhalb des Lageregelkreises z. B. durch eine kippende Frässpindel. Sie führt zu einem Parallelversatz zwischen Ist- und Sollbahn. Dieser ist umso größer, je flacher die Steigung der Geraden verläuft.

- Nichtlineares Reibungsverhalten der Schlittenführungen

## 2.2.1.2 Schleppabstandsüberwachung

### Funktion

Regelungstechnisch entsteht beim Verfahren einer Maschinenachse immer ein gewisser Schleppabstand, d. h. eine Differenz zwischen Soll- und Istposition.

Der sich einstellende Schleppabstand ist abhängig von:

- Lageregelkreisverstärkung  
MD32200 \$MA\_POSCTRL\_GAIN (K<sub>v</sub>-Faktor)
- Maximale Beschleunigung  
MD32300 \$MA\_MAX\_AX\_ACCEL (Maximale Achsbeschleunigung)
- Maximale Geschwindigkeit  
MD32000 \$MA\_MAX\_AX\_VELO (Maximale Achsgeschwindigkeit)
- Bei aktiver Vorsteuerung:  
Genauigkeit des Streckenmodells und den Parametern:  
MD32610 \$MA\_VELO\_FFW\_WEIGHT (Faktor für Geschwindigkeits-Vorsteuerung)  
MD32800 \$MA\_EQUIV\_CURRCTRL\_TIME (Ersatzzeitkonstante Stromregelkreis für Vorsteuerung)  
MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung)

In der Beschleunigungsphase wird der Schleppabstand beim Verfahren einer Maschinenachse zunächst immer größer. Nach einer von der Parametrierung des Lageregelkreises abhängigen Zeit bleibt er dann im Idealfall konstant. Aufgrund äußerer Einflüsse kommt es während eines Bearbeitungsvorgangs immer zu mehr oder weniger großen Schwankungen des Schleppabstandes. Damit diese Schwankungen des Schleppabstandes keinen Alarm auslösen, ist für die Schleppabstandsüberwachung ein Toleranzband vorzugeben, innerhalb dessen sich der Schleppabstand bewegen darf:

MD36400 \$MA\_CONTOUR\_TOL (Toleranzband Konturüberwachung)

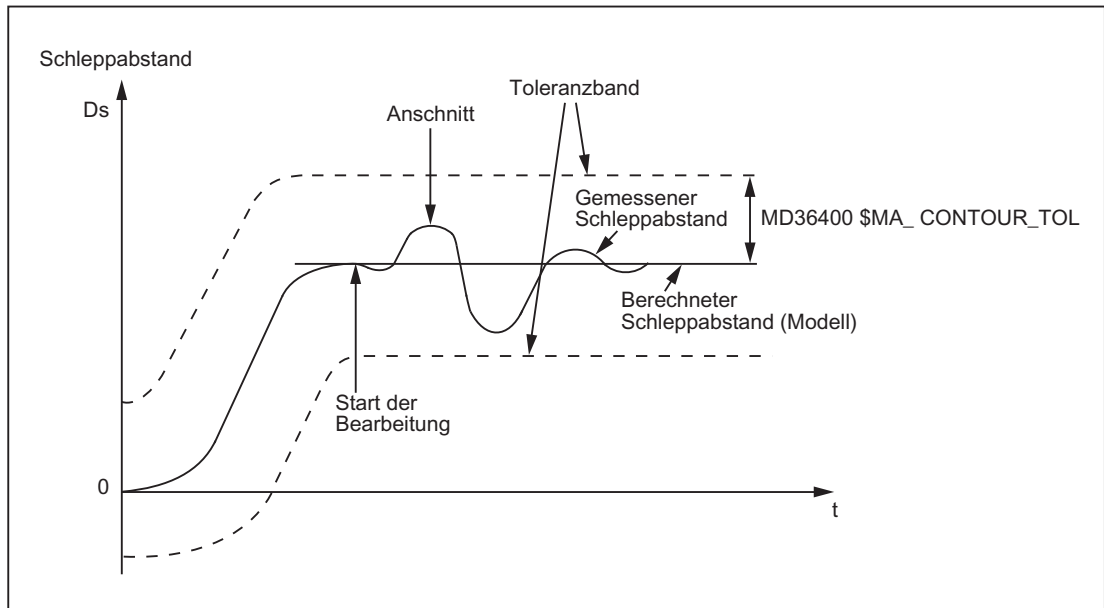


Bild 2-1 Schleppabstandsüberwachung

### Wirksamkeit

Die Schleppabstandsüberwachung ist nur wirksam bei aktiver Lageregelung und folgenden Achstypen:

- Linearachsen mit und ohne Vorsteuerung
- Rundachsen mit und ohne Vorsteuerung
- Lagegeregelten Spindeln

### Fehlerfall

Bei Überschreiten der parametrisierten Toleranzgrenze wird folgender Alarm angezeigt:

25050 "Achse <Achsbezeichner> Konturüberwachung"

Die betroffene Achse / Spindel wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

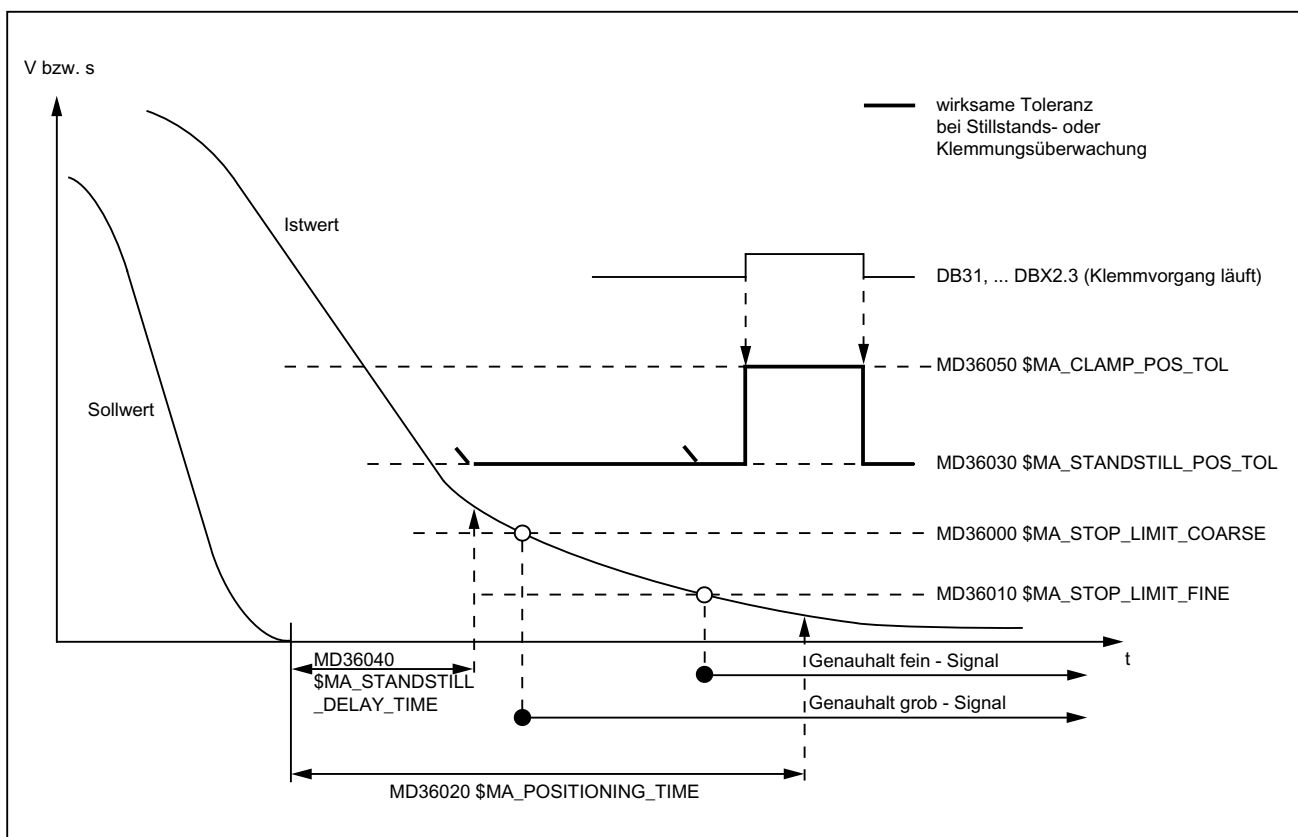


## 2.2.2 Positionier-, Stillstands- und Klemmungsüberwachung

### 2.2.2.1 Zusammenhang zwischen Positionier-, Stillstands- und Klemmungsüberwachung

#### Übersicht

Die folgende Übersicht zeigt den Zusammenhang zwischen Positionier-, Stillstands- und Klemmungsüberwachung:



### 2.2.2.2 Positionierüberwachung

#### Funktion

Zum Abschluss eines Positioniervorgangs:

- Sollgeschwindigkeit = 0 **UND**
- DB31, ... DBX64.6/7 (Fahrbefehl minus/plus) = 0

überwacht die Positionierüberwachung, dass der Schleppabstand jeder beteiligten Maschinenachse innerhalb der Verzögerungszeit kleiner der Genauhalttoleranz fein wird:

MD36010 \$MA\_STOP\_LIMIT\_FINE (Genauhalt fein)

MD36020 \$MA\_POSITIONING\_TIME (Verzögerungszeit Genauhalt fein)

Nach dem Erreichen von "Genauhalt fein" wird die Positionierüberwachung abgeschaltet.

---

#### Hinweis

Je kleiner die Genauhalttoleranz fein gewählt wird, desto länger dauert der Positioniervorgang und damit die Zeit bis zum Satzwechsel.

---

#### Regeln zur MD-Einstellung

MD36010 \$MA_STOP_LIMIT_FINE	MD36020 \$MA_POSITIONING_TIME
groß	kann relativ kurz gewählt werden
klein	muss relativ lang gewählt werden

MD32200 \$MA_POSCTRL_GAIN (Kv-Faktor)	MD36020 \$MA_POSITIONING_TIME
klein	muss relativ lang gewählt werden
groß	kann relativ kurz gewählt werden

## Wirksamkeit

Die Positionierüberwachung ist nur wirksam bei aktiver Lageregelung und folgenden Achstypen:

- Linearachsen
- Rundachsen
- Lagegeregelten Spindeln

## Fehlerfall

Bei Überschreiten der parametrisierten Positionierüberwachungszeit wird folgender Alarm angezeigt:

25080 "Achse <Achsbezeichner> Positionierüberwachung"

Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

### 2.2.2.3 Stillstandsüberwachung

## Funktion

Zum Abschluss eines Positioniervorgangs:

- Sollgeschwindigkeit = 0 **UND**
- DB31, ... DBX64.6/7 (Fahrbefehl minus/plus) = 0

überwacht die Stillstandsüberwachung, dass der Schleppabstand jeder beteiligten Maschinenachse innerhalb der Verzögerungszeit kleiner der Stillstandstoleranz wird:

MD36040 \$MA\_STANDSTILL\_DELAY\_TIME (Verzögerungszeit Stillstandsüberwachung)

MD36030 \$MA\_STANDSTILL\_POS\_TOL (Stillstandstoleranz)

Nach Erreichen des erforderlichen Genauhalt-Zustandes ist der Positioniervorgang abgeschlossen:

DB31, ... DBX60.6/7 (Position erreicht mit Genauhalt grob/fein) = 1

Die Positionierüberwachung wird abgeschaltet und von der Stillstandsüberwachung abgelöst.

Die Stillstandsüberwachung überwacht die Einhaltung der Stillstandstoleranz. Solange keine neue Fahranforderung ansteht, darf die Maschinenachse die Stillstandstoleranz nicht verlassen.

### Wirksamkeit

Die Stillstandsüberwachung ist nur wirksam bei aktiver Lageregelung und folgenden Achstypen:

- Linearachsen
- Rundachsen
- Lagegeregelten Spindeln

### Fehlerfall

Bei Überschreiten der Verzögerungszeit und / oder der Stillstandstoleranz wird folgender Alarm angezeigt:

25040 "Achse <Achsbezeichner> Stillstandsüberwachung"

Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

#### 2.2.2.4 Parametersatzabhängige Genauhalt- und Stillstandstoleranz

Zur Anpassung an unterschiedliche Bearbeitungssituationen und / oder Achsdynamiken, z. B.:

- Betriebszustand A: Hohe Genauigkeit, lange Bearbeitungszeit
- Betriebszustand B: Geringere Genauigkeit, kürzere Bearbeitungszeit
- Änderung der Massenverhältnisse nach Getriebeumschaltung

können die Positionstoleranzen:

- MD36000 \$MA\_STOP\_LIMIT\_COARSE (Genauhalt grob)
- MD36010 \$MA\_STOP\_LIMIT\_FINE (Genauhalt fein)
- MD36030 \$MA\_STANDSTILL\_POS\_TOL (Stillstandstoleranz)

parametersatzabhängig mit einem gemeinsamen Faktor bewertet werden:

MD36012 \$MA\_STOP\_LIMIT\_FACTOR (Faktor Genauhalt grob / fein und Stillstand)

Da der Faktor für alle drei Positionstoleranzen gemeinsam gilt, bleibt das Verhältnis der Werte untereinander konstant.

### 2.2.2.5 Klemmungsüberwachung

#### Funktion

Bei Maschinenachsen, die nach Abschluss eines Positioniervorgangs mechanisch geklemmt werden, kann es aufgrund des Klemmvorgangs zu größeren Bewegungen kommen (> Stillstandstoleranz). Daher wird während des Klemmvorgangs die Stillstandsüberwachung von der Klemmungsüberwachung ersetzt.

Die Klemmungsüberwachung überwacht die Einhaltung der parametrisierten Klemmungstoleranz:

MD36050 \$MA\_CLAMP\_POS\_TOL (Klemmungstoleranz)

#### Aktivierung

Die Klemmungsüberwachung wird durch folgendes Nahtstellensignal aktiviert:

DB31, ... DBX2.3 (Klemmvorgang läuft)

---

#### Hinweis

Die Klemmungsüberwachung ist im "Nachführbetrieb" (DB31, ... DBX1.4 = 1) nicht aktiv.

---

#### Fehlerfall

Bei Überschreitung der Klemmungstoleranz wird folgender Alarm angezeigt:

26000 "Klemmungsüberwachung"

Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

#### Automatisches Anhalten zum Lösen der Klemmung

Soll bei Bahnsteuerbetrieb eine geklemmte Achse wieder verfahren werden, hält die NC die Bahnbewegung vorausschauend zum Satzanfang des Bewegungssatzes der geklemmten Achse so lange an, bis die geklemmte Achse wieder verfahren kann. Wird die Klemmung vor dem Anhalten aufgehoben, erfolgt kein Stopp der Bahnbewegung.

**Parametrierung:**

MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H01' (Sonderfunktionen bei geklemmter Achse)

**Hinweis**

Ob eine Achse geklemmt ist, erkennt die NC anhand der "Reglerfreigabe" der Achse:

DB31, ... DBX2.2 = 0: keine Reglerfreigabe ⇒ Achse ist geklemmt

DB31, ... DBX2.2 = 1: Reglerfreigabe ⇒ Achse ist nicht geklemmt

**Voraussetzungen bezüglich des PLC-Anwenderprogramms**

- Die Achse wird aus der Klemmung genommen, wenn ein Fahrbefehl ansteht.
- Für die Achse gilt immer:  
 DB31, ... DBX2.2 (Reglerfreigabe) = 0: Achse ist geklemmt.  
 DB31, ... DBX2.2 (Reglerfreigabe) = 1: Achse ist nicht geklemmt.

Das folgende Bild zeigt beispielhaft die Nahtstellensignale und Zustände beim Lösen der Achsklemmung:

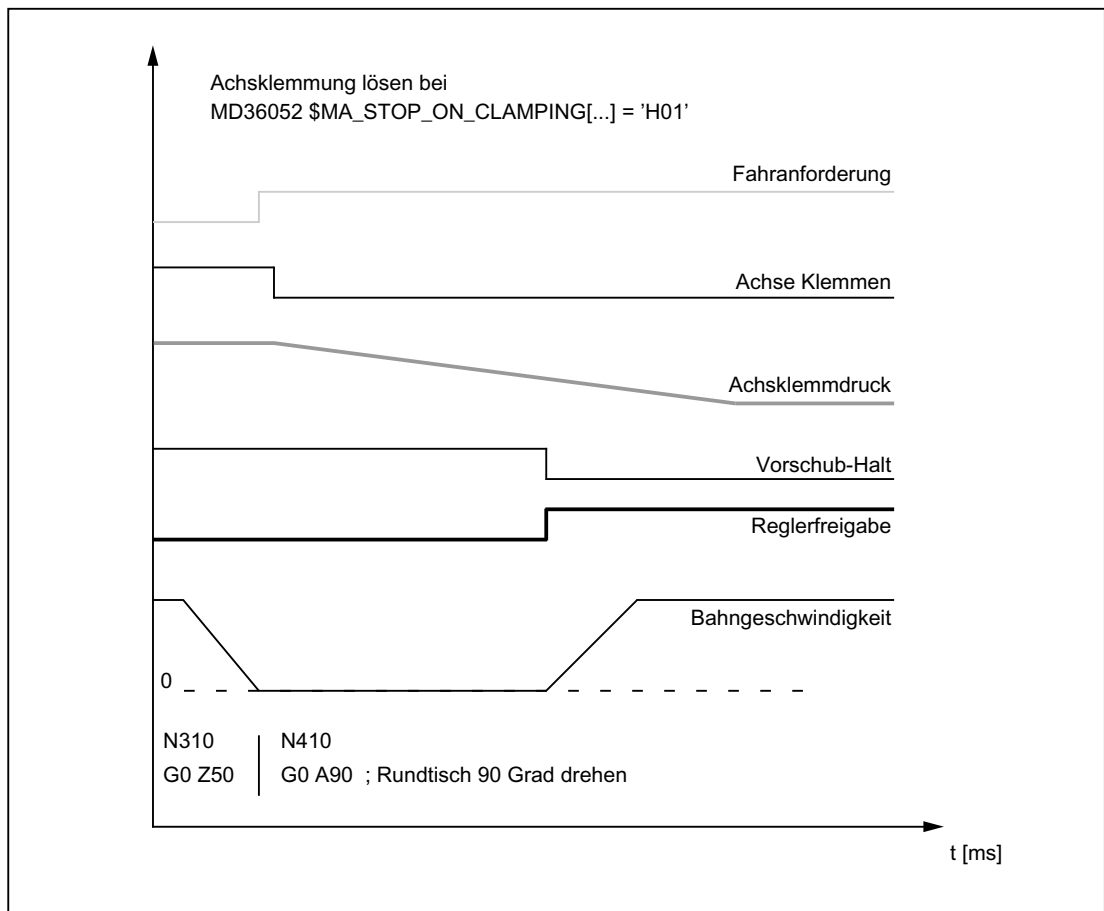


Bild 2-2 Achsklemmung lösen bei MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H01'

Die Teileprogrammätze N310 und N410 beziehen sich auf folgendes Programmierbeispiel:

Programmcode	Kommentar
N100	G0 X0 Y0 Z0 A0 G90 G54 F500
N101	G641 ADIS=.1 ADISPOS=5
N210	G1 X10 ; Bearbeiten
N220	G1 X5 Y20
N310	G0 Z50 ; Freifahren
N410	G0 A90 ; Rundtisch drehen
N510	G0 X100 ; Anfahren
N520	G0 Z2
N610	G1 Z-4 ; Bearbeiten
N620	G1 X0 Y-20

### Optimiertes Lösen der Achsklemmung über Fahrbefehl

Soll bei Bahnsteuerbetrieb eine geklemmte Achse verfahren werden, wird in den unmittelbar vor dem Verfahrtsatz der geklemmten Achse stehenden Eilgang-Sätzen (G0) ein Fahrbefehl für die geklemmte Achse ausgegeben. Damit kann das PLC-Anwenderprogramm die Achsklemmung wieder rechtzeitig lösen.

#### Hinweis

Der Fahrbefehl wird maximal 2 Eilgang-Sätze (inklusive Zwischensätze) vorher gesetzt, um den Bezug zum veranlassenden Teileprogrammatz zu erhalten.

#### Parametrierung:

MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H03' (Sonderfunktionen bei geklemmter Achse)

#### Voraussetzungen bezüglich des PLC-Anwenderprogramms

- Die Achse wird aus der Klemmung genommen, sobald ein Fahrbefehl ansteht.
- Die Achse darf aus der Klemmung genommen werden, auch wenn nur positioniert (G0) wird.

Das folgende Bild zeigt beispielhaft die Nahtstellensignale und Zustände beim Lösen der Achsklemmung:

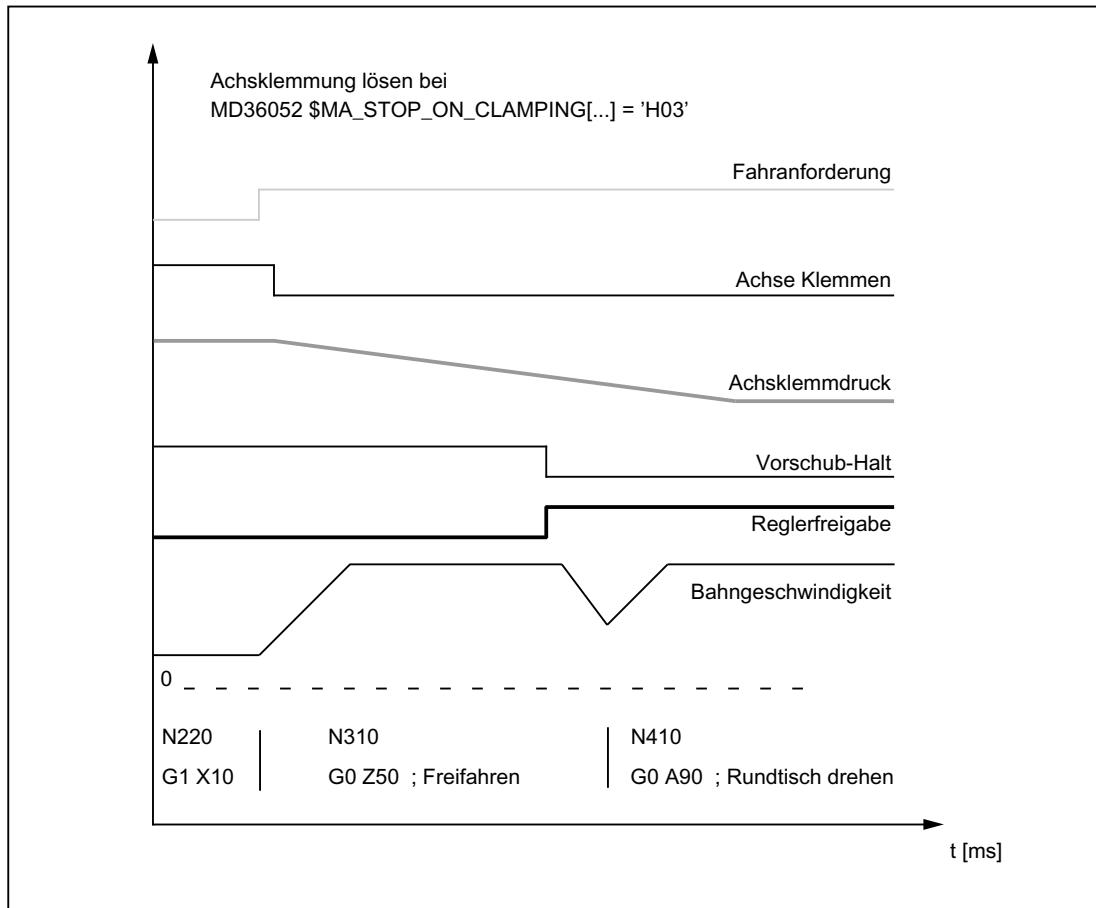


Bild 2-3 Achsklemmung lösen bei MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H03'



## Automatisches Anhalten zum Setzen der Klemmung

Soll bei Bahnsteuerbetrieb eine Achse geklemmt werden, hält die NC die Bahnbewegung vor dem nächsten "Nicht-Eilgangssatz" an, falls die Achse bis dahin noch nicht geklemmt ist, d. h. die PLC die Vorschubkorrektur noch auf den Wert Null gesetzt hat.

### Parametrierung:

MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H04' (Sonderfunktionen bei geklemmter Achse)

### Voraussetzungen bezüglich des PLC-Anwenderprogramms

- Die Achse wird immer geklemmt, wenn kein Fahrbefehl ansteht.
- Die Achse muss während des Positionierens der anderen Achsen nicht geklemmt sein.  
Ob die Achsen positionieren ist daraus ersichtlich, ob Eilgang (G0) programmiert ist.  
Der Stopp wird deshalb nicht unmittelbar auf den Satzbeginn gesetzt, in dem die Achse steht, sondern erst auf den Satzbeginn des nächsten Bearbeitungssatzes (Verfahrsatz, der nicht mit Eilgang verfahren wird).
- Die Achse ist geklemmt, wenn die Vorschubkorrektur eines Bearbeitungssatzes ungleich 0 ist.  
Ist die Achse vor dem nächsten Bearbeitungssatz geklemmt, d. h. die Vorschubkorrektur wieder ungleich 0, wird kein Stopp erzeugt.

Das folgende Bild zeigt beispielhaft die Nahtstellensignale und Zustände beim Setzen der Achsklemmung. Die Teileprogrammsätze N410, N510, N520 und N610 beziehen sich auf das schematische Beispiel unter Randbedingungen.

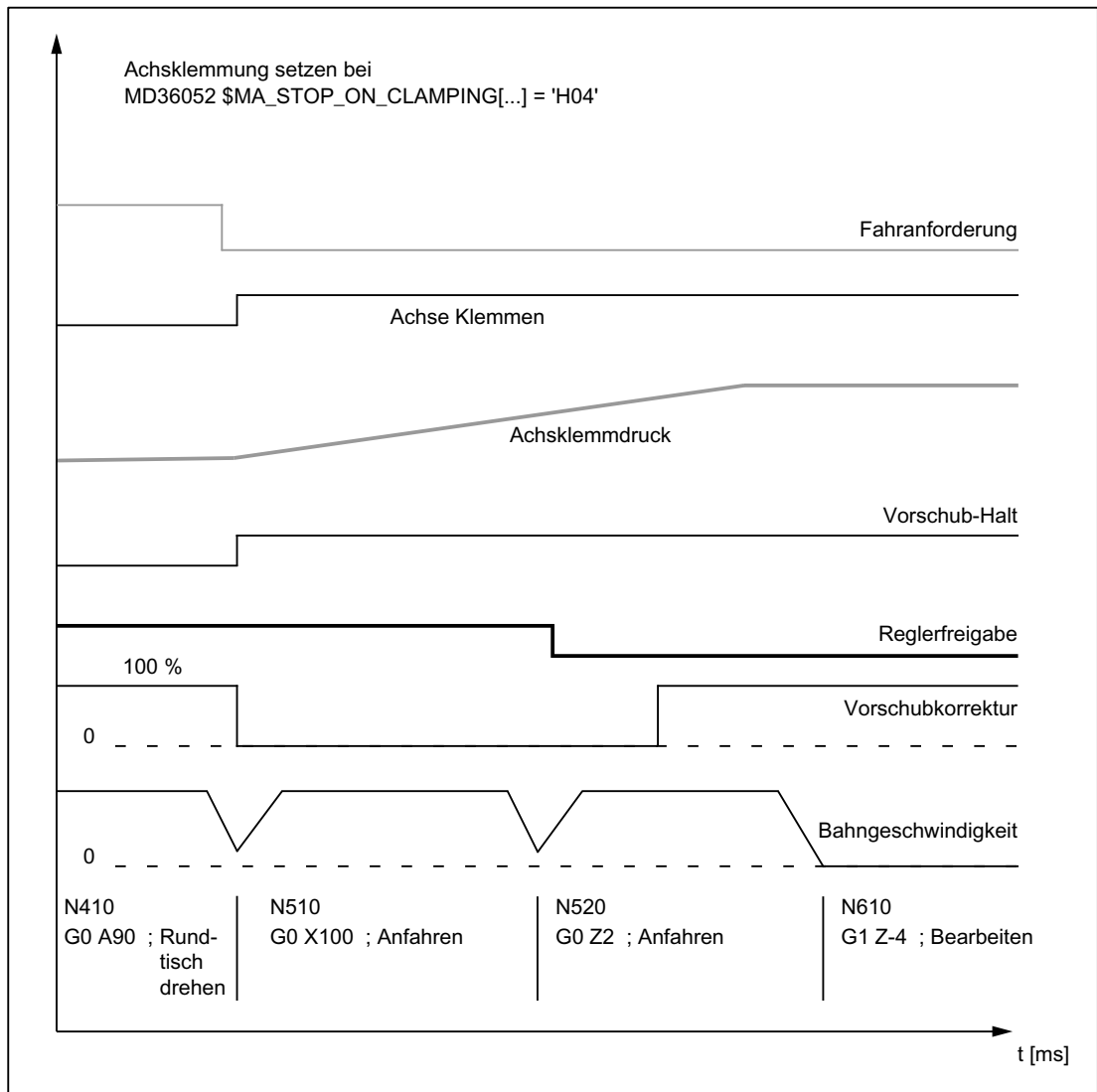


Bild 2-4 Achsklemmung setzen bei MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H04'

## Randbedingungen

### Bahnsteuerbetrieb

Für die oben genannten Funktionen:

- Automatisches Anhalten zum Lösen der Klemmung
- Optimiertes Lösen der Achsklemmung über Fahrbefehl
- Automatisches Anhalten zum Setzen der Klemmung

muss die Funktion "Look Ahead" aktiv sein.

Teileprogrammsätze ohne Bahnbewegung (z. B. M82/M83) unterbrechen den Bahnsteuerbetrieb und damit auch "Look Ahead".

Beispiel:

In das verwendete Programmierbeispiel werden die Teileprogrammätze N320 und N420 eingefügt:

Programmcode		Kommentar
N100	G0 X0 Y0 Z0 A0 G90 G54 F500	
N101	G641 ADIS=.1 ADISPOS=5	
N210	G1 X10	; Bearbeiten
N220	G1 X5 Y20	
N310	G0 Z50	; Freifahren
N320	M82	; keine Bahnbewegung
N410	G0 A90	; Rundtisch drehen
N420	M83	; keine Bahnbewegung
N510	G0 X100	; Anfahren
N520	G0 Z2	
N610	G1 Z-4	; Bearbeiten
N620	G1 X0 Y-20	

Die Funktion verhält sich daraufhin wie folgt:

- MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H03'

Wirkt sich nicht mehr aus.

Das vorausschauende Setzen des Fahrbefehls wird nur für Sätze mit aktivem Bahnsteuerbetrieb durchgeführt. M82 erzeugt einen Stopp und unterbricht damit den Bahnsteuerbetrieb. Das vorausschauende Anhalten auf N410 wäre nicht erforderlich, da sowieso angehalten wird.

- MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H04'

Erzeugt unabhängig von M83 einen Stopp, der abhängig von "Vorschubkorrektur 0%" durchgeführt wird. Das Anhalten vor dem ersten Bearbeitungssatz ist somit gegeben.

#### Hinweis

#### MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H01' oder 'H04'

Beide Funktionen können unabhängig vom Klemmen von Achsen verwendet werden:

- MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H01'  
Erzeugt vorausschauend einen Stopp der Bahnbewegung, wenn die Reglerfreigabe für die betreffende Achse fehlt.
- MD36052 \$MA\_STOP\_ON\_CLAMPING = 'H04'  
Erzeugt vorausschauend einen Stopp der Bahnbewegung, wenn die Vorschubkorrektur = 0% am Übergang von Teileprogrammätzen mit Eilgang auf Teileprogrammätze ohne Eilgang ist.

Beide Funktionen stellen sicher, dass die Bahnbewegung im Bahnsteuerbetrieb bereits vor dem Beginn des betreffenden Teileprogrammatzes gestoppt ist und nicht erst im Teileprogrammatz angehalten wird.

### Satzwechselkriterium: Klemmungstoleranz

Nach Aktivierung der Klemmungsüberwachung (DB31, ... DBX2.3 = 1) wirkt als Satzwechselkriterium bei Verfahrssätzen, bei denen am Satzende angehalten wird, nicht mehr die entsprechende Genauhaltbedingung, sondern die parametrisierte Klemmungstoleranz:

MD36050 \$MA\_CLAMP\_POS\_TOL (Klemmungstoleranz bei Nahtstellensignal "Klemmung aktiv")

### Verhalten bei Aufheben der Klemmung

Ist die Achse durch den Klemmungsvorgang bewegt worden, wird sie nach dem Lösen der Klemmung und dem Setzen der Reglerfreigabe von der NC wieder auf die Sollposition zurück positioniert. Die Rückpositionierung erfolgt abhängig davon, ob für die Achse "Nachführbetrieb" aktiviert wurde:

- Ohne Nachführbetrieb: Rückpositionierung durch Lageregler
- Mit Nachführbetrieb: Rückpositionierung durch Interpolator

Siehe dazu auch Nahtstellensignal DB31, ... DBX1.4 (Nachführbetrieb).

---

#### Hinweis

Als Kriterium zur Aktivierung von "Nachführbetrieb" können vom PLC-Anwenderprogramm folgende Nahtstellensignale ausgewertet werden:

DB31, ... DBX60.6/7 (Position erreicht mit Genauhalt grob/fein)

---

## 2.2.3 Drehzahlsollwertüberwachung

### Funktion

Der Drehzahlsollwert setzt sich zusammen aus:

- Drehzahlsollwert des Lagereglers
- Drehzahlsollwertanteil der Vorsteuerung (nur bei aktiver Vorsteuerung)
- Driftkompensation (nur bei Antrieben mit analoger Sollwertschnittstelle)

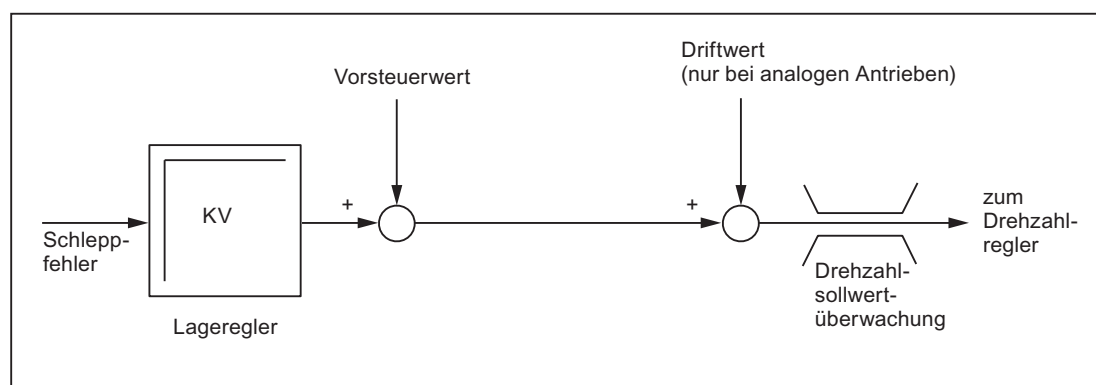


Bild 2-5 Drehzahlsollwertberechnung

Die Drehzahlsollwertüberwachung stellt durch Begrenzung der Stell- bzw. Ausgangsgröße (10 V bei analoger Sollwertschnittstelle oder Nenndrehzahl bei digitalen Antrieben) sicher, dass die physikalischen Begrenzungen der Antriebe nicht überschritten werden:

MD36210 \$MA\_CTRL0UT\_LIMIT (Maximaler Drehzahlsollwert)

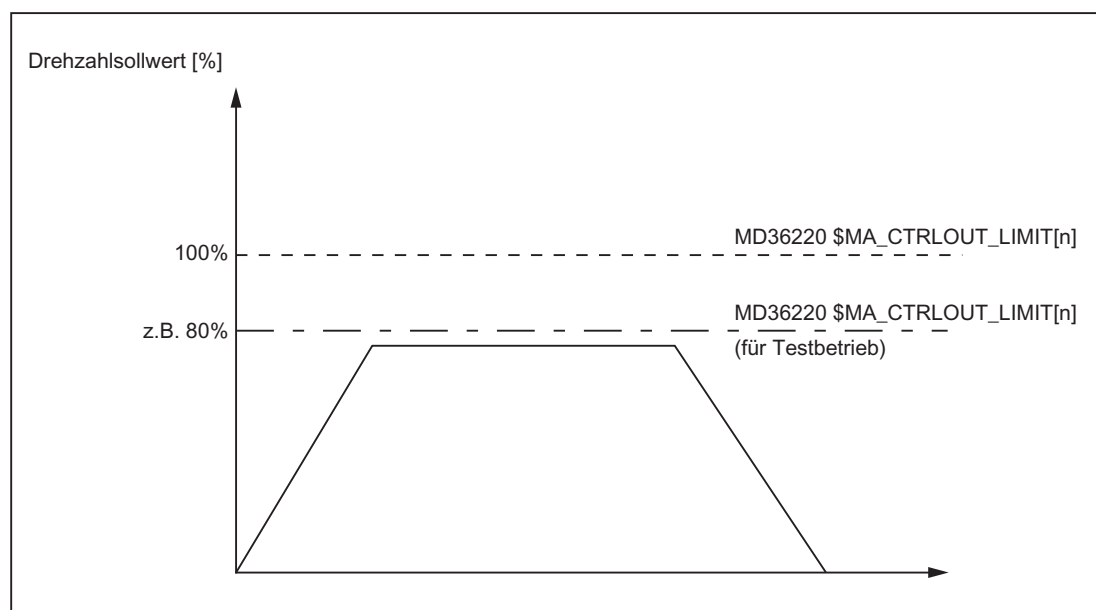


Bild 2-6 Drehzahlsollwertbegrenzung

### Verzögerung Drehzahlsollwertüberwachung

Damit es nicht in jedem Fall einer Drehzahlbegrenzung zu einer Fehlerreaktion kommt, kann eine Verzögerungszeit parametrierbar werden:

MD36220 \$MA\_CTRL\_OUT\_LIMIT\_TIME (Verzögerung Drehzahlsollwertüberwachung)

Erst wenn eine Drehzahlbegrenzung länger als die eingestellte Zeit erforderlich wird, erfolgt die entsprechende Fehlerreaktion.

### Wirksamkeit

Die Drehzahlsollwertüberwachung ist nur für lagegeregelt Achsen aktiv und kann nicht ausgeschaltet werden.

### Fehlerfall

Bei Überschreiten der parametrierbaren Verzögerungszeit wird folgender Alarm angezeigt:

25060 "Achse <Achsenbezeichner> Drehzahlsollwertbegrenzung"

Die betroffene Achse wird im Nachführbetrieb über die parametrierbare Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

---

### Hinweis

Mit dem Erreichen der Drehzahlsollwertüberwachung wird der Lageregelkreis der Achse durch die Begrenzung nichtlinear. Hieraus resultieren Konturfehler, sofern die Achse an der Konturerzeugung beteiligt ist.

---

## 2.2.4 Istgeschwindigkeitsüberwachung

### Funktion

Die Istgeschwindigkeitsüberwachung überwacht, dass die aktuelle Istgeschwindigkeit einer Achse/Spindel den parametrisierten Schwellwert nicht überschreitet:

MD36200 \$MA\_AX\_VELO\_LIMIT (Schwellwert Geschwindigkeitsüberwachung)

Der Schwellwert sollte 10-15% über der parametrisierten Maximalgeschwindigkeit liegen:

- Bei Achsen:  
MD32000 \$MA\_MAX\_AX\_VELO (Maximale Achsgeschwindigkeit)
- Bei Spindeln:  
MD35110 \$MA\_GEAR\_STEP\_MAX\_VELO\_LIMIT[n] (Maximaldrehzahl der Getriebestufe)

Mit dieser Einstellung tritt eine Überschreitung des Schwellwerts der Geschwindigkeitsüberwachung normalerweise nicht auf (Ausnahme: Antriebsfehler).

### Aktivierung

Die Istgeschwindigkeitsüberwachung wird aktiv, sobald das aktive Messsystem gültige Istwerte (Gebergrenzfrequenz nicht überschritten) liefert.

### Wirksamkeit

Die Istgeschwindigkeitsüberwachung ist nur wirksam bei aktiver Lageregelung und folgenden Achstypen:

- Linearachsen
- Rundachsen
- Gesteuerten und lagegeregelten Spindeln

### Fehlerfall

Bei Überschreitung des Schwellwerts wird folgender Alarm angezeigt:

25030 "Achse <Achsbezeichner> Istgeschwindigkeit Alarmgrenze"

Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

## 2.2.5 Messsystem-Überwachung

Die NC hat keinen direkten Zugriff auf die Messsystem-Hardware, die Messsystem-Überwachungen werden deshalb hauptsächlich von der Antriebs-Software durchgeführt.

### Überwachungsfunktionen im Antrieb

- Überwachung von Hardware-Fehlern (z. B. Messsystem-Ausfall, Leitungsbruch)
- Nullmarkenüberwachung

#### Literatur:

Antriebsfunktionen SINAMICS S120

Im Antrieb durchgeführte Messsystem-Überwachungen werden auf NCK-Alarme (Alarm 25000 und folgende) oder NC-Reaktionen (z. B. Abbruch von Referenzieren oder fliegendem Messen) abgebildet. Das genaue Verhalten der NC ist abhängig von der Einstellung im Maschinendatum:

MD36310 \$MA\_ENC\_ZERO\_MONITORING

Wert	Bedeutung	
= 0	Überwachung von HW-Fehlern:	<p>EIN</p> <p>Bei detektierten Hardware-Fehlern im aktiven Messsystem wird der POWER ON-Alarm 25000 angezeigt:</p> <p>"Achse &lt;Achsbezeichner&gt; Hardwarefehler aktiver Geber"</p> <p>Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:</p> <p>MD36610 \$MA_AX_EMERGENCY_STOP_TIME (Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)</p> <p>Bei detektierten Hardware-Fehlern im passiven Messsystem wird der Alarm 25001 angezeigt:</p> <p>"Achse &lt;Achsbezeichner&gt; Hardwarefehler passiver Geber"</p> <p>Es erfolgt keine weitere Alarmreaktion.</p>
	Nullmarkenüberwachung:	<p>AUS</p> <p>Die Alarme 25020 und 25021 (s. u.) werden unterdrückt.</p>
= 100	Keine Nullmarkenüberwachung sowie Ausblenden sämtlicher Geberüberwachungen (d. h. neben Alarm 25020 (25021) werden auch die Alarme 25000 (25001) und 25010 (25011) unterdrückt.	



Wert	Bedeutung	
> 0 aber < 100	Überwachung von HW-Fehlern:	EIN (s. o.)
	Nullmarkenüberwachung:	<p>EIN</p> <p>Bei Ansprechen der Nullmarkenüberwachung im aktiven Messsystem wird der Alarm 25020 angezeigt: "Achse &lt;Achsbezeichner&gt; Nullmarkenüberwachung aktiver Geber"</p> <p>Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt: MD36610 \$MA_AX_EMERGENCY_STOP_TIME (Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)</p> <p>Bei Ansprechen der Nullmarkenüberwachung im passiven Messsystem wird der Alarm 25021 angezeigt: "Achse &lt;Achsbezeichner&gt; Nullmarkenueberwachung passiver Geber"</p> <p>Es erfolgt keine weitere Alarmreaktion.</p>
> 100	Überwachung von HW-Fehlern:	<p>EIN mit abgeschwächter Fehlermeldung:</p> <p>Der POWER ON-Alarm 25000 wird durch den Reset-Alarm 25010 ersetzt, der Reset-Alarm 25001 durch den Cancel-Alarm 25011.</p>
	Nullmarkenüberwachung:	EIN (s. o.)

Details zu den Alarmen siehe:

**Literatur:**

Diagnosehandbuch

**ACHTUNG**

Bei Hardware-Fehlern wird der Referenzierstatus der Maschinenachse zurückgesetzt:  
DB31, ... DBX60.4/5 (Referenziert/Synchronisiert 1/2) = 0

### Überwachungsfunktionen im NCK

- Gebergrenzfrequenzüberwachung
- Plausibilitätskontrolle bei Absolutwertgebern

### 2.2.5.1 Gebergrenzfrequenzüberwachung

#### Funktion

Die NC-seitige Gebergrenzfrequenzüberwachung basiert auf den Projektier- und Telegramm-Informationen des Antriebs. Sie überwacht, dass die Geberfrequenz die projektierte Gebergrenzfrequenz nicht überschreitet:

MD36300 \$MA\_ENC\_FREQ\_LIMIT (Gebergrenzfrequenz)

Die Gebergrenzfrequenzüberwachung bezieht sich immer auf das in der NC/PLC-Nahtstelle angewählte, aktive Messsystem:

DB31, ... DBX1.5 / 1.6 (Lagemesssystem 1/2)

#### Wirksamkeit

Die Gebergrenzfrequenzüberwachung ist wirksam bei:

- Linearachsen
- Rundachsen
- Gesteuerten und lagegeregelten Spindeln

#### Fehlerfall

Bei Überschreiten der Gebergrenzfrequenz erfolgt:

- Meldung an die PLC:  
DB31, ... DBX60.2 bzw. 60.3 = 1 (Gebergrenzfrequenz überschritten 1 bzw. 2)
- Spindeln

Spindeln werden nicht stillgesetzt, sondern drehen drehzahl geregelt weiter.

Wird die Spindeldrehzahl soweit reduziert, dass die Geberfrequenz die Gebergrenzfrequenz wieder unterschreitet, wird das Istwertsystem der Spindel automatisch neu synchronisiert.

- Achsen

Es wird folgender Alarm angezeigt:

21610 "Kanal <Kanalnummer> Achse <Achsbezeichner> Geber <Gebernummer> - Frequenz überschritten"

Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

---

#### Hinweis

Nach Überschreitung der Gebergrenzfrequenz muss eine lagegeregelte Maschinenachse neu referenziert werden (siehe Kapitel "R1: Referenzieren (Seite 1319)").

---

## 2.2.5.2 Plausibilitätskontrolle bei Absolutwertgebern

### Funktion

Bei Absolutwertgebern (MD30240 \$MA\_ENC\_TYPE = 4) werden vom Messsystem gelieferte Absolutwerte zur Plausibilitätskontrolle des Istwerts genutzt.

Dazu vergleicht die NC den softwaretechnisch zyklisch im Lageregler-Takt anhand der Inkrementalinformationen des Gebers mitgeführten Positionswert mit einem unmittelbar aus den Absolut- und Inkrementalinformationen des Gebers neu gebildeten Positionswert und überwacht, dass die ermittelte Positionsdifferenz die zulässige Abweichung nicht überschreitet:

MD36310 \$MA\_ENC\_ZERO\_MONITORING (zulässige Abweichung in 1/2 Grobstrichen zwischen der absoluten und der inkrementellen Geberspur)

---

### Hinweis

Die Plausibilitätskontrolle bei Absolutwertgebern deckt vor allem Abweichungen aufgrund von Verschmutzungen der Absolutspur bzw. Störungen der Absolutwert-Übertragung auf. Kleine Verfälschungen der Inkrementalspur (Störimpulse, Pulsfehler) werden dagegen nicht erkannt, die Plausibilitätskontrolle reagiert in solchen Fällen erst bei Abweichungen im Millimeterbereich. Sie ist deshalb als Zusatzüberwachung gedacht, die vornehmlich Störungen der Absolutlage diagnostizieren hilft.

---

### Hinweis

#### Rotatorische Absolutwertgeber

Soll die Plausibilitätskontrolle bei einem rotatorischen Absolutwertgeber eingesetzt werden, muss bei der Einstellung des Modulobereichs (MD34220 \$MA\_ENC\_ABS\_TURNS\_MODULO) der SINAMICS-Parameter p0979 berücksichtigt werden.

---

<b>ACHTUNG</b>
----------------

<b>Hochrüstung der NCK-Software</b>
-------------------------------------

Ist bei Absolutwertgebern die Plausibilitätskontrolle aktiv (MD36310 > 0), müssen bei einer Hochrüstung der NCK-Software die bisherigen Einstellwerte von MD36310 überprüft und ggf. erhöht werden.
---

### Nullmarken-Diagnose

Bei Absolutwertgebern muss bei der Inbetriebnahme die zulässige Abweichung für die Plausibilitätskontrolle ermittelt werden. Dies kann erfolgen über das Maschinendatum:

MD36312 \$MA\_ENC\_ABS\_ZEROMON\_WARNING (Nullmarkenüberwachung Warnschwelle)

Wert	Bedeutung
0	keine Nullmarken-Diagnose
> 0	zulässige Abweichung in 1/2 Grobstrichen zwischen der absoluten und der inkrementellen Geberspur

#### Vorgehensweise bei der Inbetriebnahme:

1. Nullmarkenüberwachung deaktivieren:  
MD36310 \$MA\_ENC\_ZERO\_MONITORING = 0
2. Nullmarken-Diagnose aktivieren:  
MD36312 \$MA\_ENC\_ABS\_ZEROMON\_WARNING = 1
3. Achse bewegen und Systemvariable \$VA\_ENC\_ZERO\_MON\_ERR\_CNT (Anzahl der erkannten Grenzwert-Überschreitungen) beobachten.
4. Falls \$VA\_ENC\_ZERO\_MON\_ERR\_CNT ≠ 0:  
MD36312-Wert erhöhen und Schritt 3 wiederholen.
5. Falls \$VA\_ENC\_ZERO\_MON\_ERR\_CNT = 0 (über einen längeren Zeitraum!):  
Der richtige Wert für MD36310 ist gefunden! Wert aus MD36312 in MD36310 übernehmen und anschließend MD36312 auf "0" setzen.

---

#### Hinweis

Abhängig von der Steifigkeit der Maschine (optimal sind möglichst kleine Last-Massen/Trägheitsmomente) und der Regler-Einstellungen wird das Regelspiel unterschiedlich stark "pendeln". Dies muss durch maschinenspezifische Grenzwert-Eingabe in MD36310 berücksichtigt werden.

---

## Fehlerfall

### Alarm 25020

Bei Ansprechen der Plausibilitätskontrolle im **aktiven** Messsystem wird Alarm 25020 angezeigt:

"Achse <Achsbezeichner> Nullmarkenueberwachung aktiver Geber"

Die betroffene Achse wird im Nachführbetrieb über die parametrisierte Bremsrampe stillgesetzt:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME  
(Maximale Zeitdauer der Bremsrampe bei Fehlerzuständen)

### Alarm 25021

Bei Ansprechen der Plausibilitätskontrolle im **passiven** Messsystem wird Alarm 25021 angezeigt:

"Achse <Achsbezeichner> Nullmarkenueberwachung passiver Geber"

Es erfolgt keine weitere Alarmreaktion.

---

### Hinweis

Im Fehlerfall geht die Justage des Absolutwertgebers verloren und die Achse ist nicht mehr referenziert. Eine Neujustage des Absolutwertgebers muss durchgeführt werden (siehe Kapitel "Referenzieren bei Absolutwertgebern (Seite 1358)").

ACHTUNG
Fehler in der Inkrementalspur, die nicht durch die Amplitudenüberwachung erfasst werden können, können zu Positionsabweichungen im Millimeter-Bereich führen. Die Abweichung ist dabei abhängig von der Gitterteilung/Strichzahl und der Verfahrgeschwindigkeit der Achse bei Auftreten des Fehlers.  Eine vollständige Positionsüberwachung ist nur durch Redundanz, d. h. durch Vergleich mit einem unabhängigen zweiten Messsystem möglich.

### 2.2.5.3 Anwenderspezifische Fehlerreaktionen

#### Anwenderspezifische Nullmarkenüberwachung

Das standardmäßige Alarm- und Reaktionsverhalten der Nullmarkenüberwachung kann bei absoluten Messsystemen (MD30240 \$MA\_ENC\_TYPE = 4) mit Hilfe von Systemvariablen anwenderspezifisch angepasst werden. Damit hat der Anwender die Möglichkeit, per Synchronaktion oder OEM-Applikation eine eigene Überwachung zu realisieren und alle in diesem Applikations-Zusammenhang vorhandenen Reaktionsmöglichkeiten zu nutzen, z. B.:

- Alarm absetzen
- Zyklen nutzen (z. B. Werkzeugwechsel-Position anfahren)
- ...

Beispiel:

Der Anwender kann das Alarm- und Reaktionsverhalten so anpassen, dass bei einem teuren Werkstück, bei dem das Stillsetzen der Achse infolge eines Alarms zur Beschädigung führen würde, die Bearbeitung erst beendet wird, bevor mit entsprechenden Synchronaktions-Befehlen die Bearbeitungsgüte des Werkstücks bewertet wird.

#### Wirksamkeit

Die anwenderspezifische Überwachung kann parallel oder alternativ zur standardmäßigen Nullmarkenüberwachung wirksam sein, abhängig von der Einstellung im Maschinendatum:

MD36310 \$MA\_ENC\_ZERO\_MONITORING

Wert	Bedeutung
0	Soll ausschließlich eine anwenderspezifische Überwachung realisiert werden, muss die standardmäßige Nullmarkenüberwachung deaktiviert werden: MD36310 = 0 <b>und</b> MD36312 = 0
> 0	Anwenderspezifische Überwachung und Standard-Nullmarkenüberwachung wirken parallel.
100	Alle Geberüberwachungen sind ausgeschaltet.

Wenn beide Überwachungen wirksam sind (MD36310 > 0), kann eine **kaskadierte Überwachung** realisiert werden.

Beispiel:

Bei einem Wert unterhalb des im MD36310 angegebenen Schwellenwerts setzt die anwenderspezifische Überwachung eine Vorwarnung ab, erst bei Überschreiten des Schwellenwerts erkennt dann die Standard-Nullmarkenüberwachung auf Störung und schaltet ab.

## Systemvariablen

Zur Realisierung der anwenderspezifischen Fehlerreaktionen stehen folgende Systemvariablen zur Verfügung:

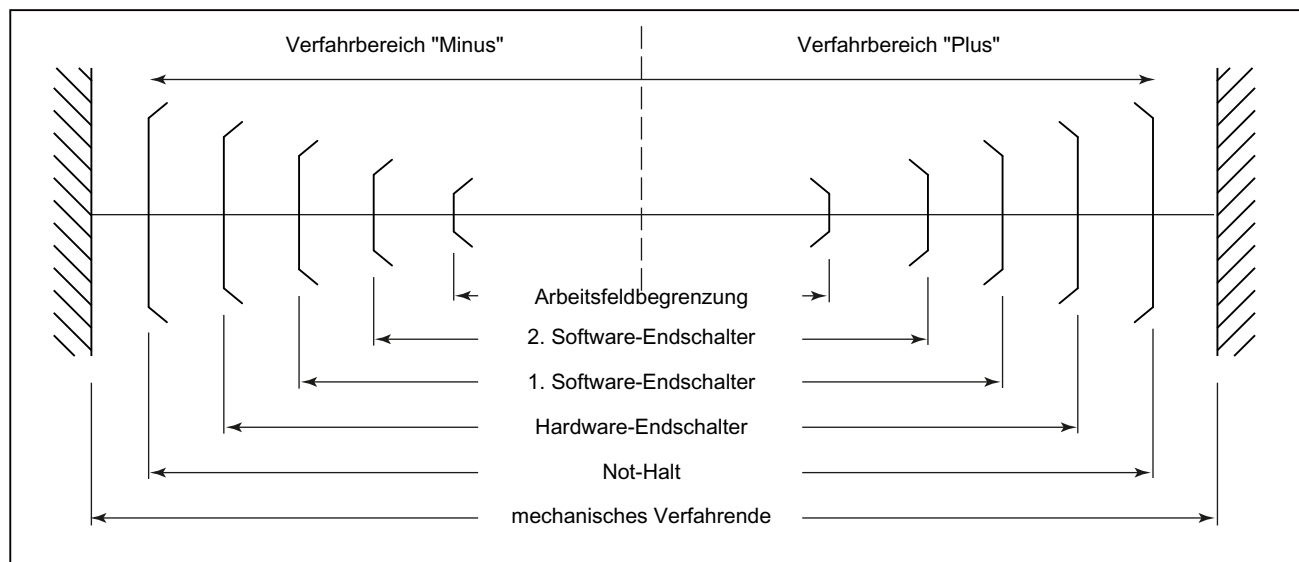
Systemvariable	Bedeutung
\$VA_ENC_ZERO_MON_ERR_CNT[<n>,<Achse>]	<p>Anzahl der erkannten Grenzwert-Überschreitungen</p> <p>Enthält die aktuelle Anzahl der erkannten Grenzwert-Überschreitungen beim Vergleich zwischen absoluter und inkrementeller Geberspur</p> <p>Der Wert wird auf 0 zurückgesetzt bei:</p> <ul style="list-style-type: none"> <li>• POWER ON</li> <li>• An- bzw. Abwahl von Parken</li> </ul> <p>Reset bewirkt kein Zurücksetzen.</p>
\$VA_ABSOLUTE_ENC_DELTA_INIT[<n>,<Achse>]	<p>Anfangs-Differenz bei Absolutgeber</p> <p>Enthält den Anfangs-Differenzwert zwischen der letzten im statischen NC-Speicher gepufferten Absolut-Position und der aktuellen Absolut-Position.</p> <p>Format des Differenzwerts: Zahl der internen Inkremente (siehe MD10200 \$MN_INT_INCR_PER_MM bzw. MD10210 \$MN_INT_INCR_PER_DEG)</p> <p>Der Wert wird aktualisiert bei:</p> <ul style="list-style-type: none"> <li>• POWER ON</li> <li>• Warmstart</li> <li>• Abwahl von Parken</li> <li>• Rückkehr unter die Gebergrenzfrequenz</li> </ul> <p>Es erfolgt kein Rücksetzen bei Reset.</p>

<n>: Gebernummer

<Achse>: Achsbezeichner

## 2.2.6 Endschalter-Überwachung

Übersicht der Endbegrenzungen und möglichen Endschalter-Überwachungen:



### 2.2.6.1 Hardware-Endschalter

#### Funktion

Ein Hardware-Endschalter wird normalerweise am Verfahrbereichsende einer Maschinenachse angebracht. Er dient zum Schutz vor einem versehentlichen Überfahren des maximalen Verfahrbereichs der Maschinenachse, während die Maschinenachse noch nicht referenziert ist.

Wird der Hardware-Endschalter ausgelöst, setzt das vom Maschinenhersteller erstellte PLC-Anwenderprogramm das entsprechende Nahtstellensignal:

DB31, ... DBX12.0/1 = 1 (Hardware-Endschalter minus/plus)

#### Parametrierung

Das Bremsverhalten der Maschinenachse bei Erreichen des Hardware-Endschalters ist parametrierbar über das Maschinendatum:

MD36600 \$MA\_BRAKE\_MODE\_CHOICE (Bremsverhalten bei Hardware-Endschalter)

Wert	Bedeutung
0	Bremsen mit der parametrierten axialen Beschleunigung
1	Schnellstopp (Sollgeschwindigkeit = 0)



### Wirksamkeit

Die Hardware-Endschalterüberwachung ist nach dem Hochlauf der Steuerung in allen Betriebsarten aktiv.

### Auswirkung

Bei Erreichen des Hardware-Endschalters erfolgt:

- Alarm 21614 "Kanal <Kanalnummer> Achse <Achsbezeichner> Hardwareendschalter <Richtung>"
- Abbremsen der Maschinenachse entsprechend dem parametrisierten Bremsverhalten.
- Steht die Achse/Spindel in einem interpolatorischen Zusammenhang mit anderen Achsen/Spindeln, so werden auch diese gemäß ihrem parametrisierten Bremsverhalten abgebremst.
- Die Verfahrtasten der betroffenen Maschinenachse werden richtungsabhängig gesperrt.

## 2.2.6.2 Software-Endschalter

### Funktion

Software-Endschalter dienen zur Begrenzung des Verfahrbereiches einer Maschinenachse. Pro Maschinenachse stehen pro Verfahrrichtung zwei (1. und 2.) Software-Endschalter zur Verfügung:

MD36100 POS\_LIMIT\_MINUS (1. Software-Endschalter minus)

MD36110 POS\_LIMIT\_PLUS (1. Software-Endschalter plus)

MD36120 POS\_LIMIT\_MINUS2 (2. Software-Endschalter minus)

MD36130 POS\_LIMIT\_PLUS2 (2. Software-Endschalter plus)

Standardmäßig ist der 1. Software-Endschalter aktiv. Durch das PLC-Anwenderprogramm kann richtungsspezifisch der 2. Software-Endschalter aktiviert werden:

DB31, ... DBX12.2 / 12.3 (2. Software-Endschalter minus / plus)

### Wirksamkeit

Die Software-Endschalter sind wirksam:

- Sofort nach dem erfolgreichem Referenzieren der Maschinenachse.
- In allen Betriebsarten.

## Randbedingungen

- Die Software-Endschalter beziehen sich auf das Maschinenkoordinatensystem.
- Die Software-Endschalter müssen innerhalb des Bereichs der Hardware-Endschalter liegen.
- Die Maschinenachse kann auf die Position des aktiven Software-Endschalters gefahren werden.
- PRESET  
Nach Verwendung der Funktion PRESET erfolgt keine Software-Endschalterüberwachung mehr. Die Maschinenachse muss erst erneut referenziert werden.
- Endlos drehende Rundachsen  
Bei endlos drehenden Rundachsen erfolgt keine Software-Endschalterüberwachung:  
MD30310 \$MA\_ROT\_IS\_MODULO == 1 (Modulowandlung für Rundachse und Spindel)  
Ausnahme: Aufbau-Rundachsen

## Auswirkungen

### Automatik-Betriebsarten (AUTOMATIK, MDA)

- Ohne Transformation, ohne überlagerte Bewegung, unveränderter Software-Endschalter:  
Ein Teileprogrammsatz, dessen programmierte Verfahrbewegung zu einem Überfahren des Software-Endschalters führen würde, wird nicht begonnen.
- Mit Transformation:  
Abhängig von der Transformationsart ergeben sich unterschiedliche Reaktionen:
  - Verhalten wie oben.  
oder
  - Der Teileprogrammsatz, dessen programmierte Verfahrbewegung zu einem Überfahren des Software-Endschalters führen würde, wird begonnen. Die betroffene Maschinenachse kommt auf dem aktiven Software-Endschalter zum Stehen. Die anderen an der Verfahrbewegung beteiligten Maschinenachsen werden abgebremst. Die programmierte Kontur wird dabei verlassen.
- Mit überlagelter Bewegung  
Der Teileprogrammsatz, dessen programmierte Verfahrbewegung zu einem Überfahren des Software-Endschalters führen würde, wird begonnen. Maschinenachsen, die mit überlagelter Bewegung verfahren werden oder worden sind, kommen auf dem jeweiligen aktiven Software-Endschalter zum Stehen. Die anderen an der Verfahrbewegung beteiligten Maschinenachsen werden abgebremst. Die programmierte Kontur wird dabei verlassen.

**Manuelle Betriebsarten**

- JOG ohne Transformation

Die Maschinenachse kommt auf der Software-Endschalterposition zum Stehen.

- JOG mit Transformation

Die Maschinenachse kommt auf der Software-Endschalterposition zum Stehen. Andere an der Verfahrbewegung beteiligte Maschinenachsen werden abgebremst. Die vorgegebene Bahn wird dabei verlassen.

**Allgemein**

- Umschalten des Software-Endschalters (1. ↔ 2. Software-Endschalter)

Liegt die Istposition der Maschinenachse nach dem Umschalten hinter dem Software-Endschalter, wird die Maschinenachse mit der maximal zulässigen Beschleunigung angehalten.

- Überfahren des Software-Endschalters in der Betriebsart JOG

Ist die Position eines Software-Endschalters erreicht und soll per erneuter Betätigung der Verfahrtaste weiter in dieser Richtung verfahren werden, wird ein Alarm angezeigt und die Achse nicht weiter verfahren:

Alarm 10621 "Kanal <Kanalnummer> Achse <Achsbezeichner> steht auf Softwareendschalter <Richtung>"

**2.2.7 Überwachung der Arbeitsfeldbegrenzung****2.2.7.1 Allgemein****Funktion**

Mit der Funktion "Arbeitsfeldbegrenzung" kann der Verfahrbereich von Geometrie- und Zusatzachsen eines Kanals auf einen zulässigen Arbeitsbereich eingegrenzt werden. Die Funktion überwacht das Einhalten der Arbeitsfeldgrenzen sowohl in der Betriebsart AUTOMATIK als auch in der Betriebsart JOG.

Folgende Varianten stehen zur Verfügung:

- Arbeitsfeldbegrenzung im Basiskoordinatensystem (BKS)

Die Angabe der Verfahrbereichsgrenzen erfolgt bezogen auf das Basiskoordinatensystem.

- Arbeitsfeldbegrenzung im Werkstück-Koordinatensystem (WKS) oder im einstellbaren Nullpunktsystem (ENS)

Die Angabe der Verfahrbereichsgrenzen erfolgt bezogen auf das Werkstück-Koordinatensystem oder das Einstellbare Nullpunktsystem.

Die beiden Überwachungsarten sind unabhängig voneinander. Sind beide gleichzeitig aktiv, wirkt richtungsabhängig diejenige Verfahrbereichsgrenze, welche den Verfahrbereich der Achse am weitesten einschränkt.

## Bezugspunkt am Werkzeug

Die Berücksichtigung der Werkzeugdaten (Werkzeuglänge und Werkzeugradius) und damit der Bezugspunkt am Werkzeug bei der Überwachung der Arbeitsfeldbegrenzung ist abhängig vom Status der Transformation im Kanal:

- **Transformation nicht aktiv**

Ohne Transformation wird bei Verfahrbewegungen mit einem aktiven Werkzeug die Position der Werkzeugschneidspitze P überwacht, d. h. bei der Überwachung wird die Werkzeuglänge automatisch berücksichtigt.

Die Berücksichtigung des Werkzeugradius muss separat aktiviert werden:

MD21020 \$MC\_WORKAREA\_WITH\_TOOL\_RADIUS (Berücksichtigung des Werkzeugradius bei Arbeitsfeldbegrenzung)

- **Transformation aktiv**

Bei bestimmten Transformationen kann die Überwachung der Arbeitsfeldbegrenzung vom Verhalten ohne Transformation abweichen:

- Die Werkzeuglänge ist Bestandteil der Transformation (\$MC\_TRAFO\_INCLUDES\_TOOL\_X = TRUE):

In diesem Fall wird die Werkzeuglänge nicht berücksichtigt, d. h. die Überwachung bezieht sich auf den Werkzeugträgerbezugspunkt.

- Transformation mit Orientierungsänderung:

Bei Transformationen mit Orientierungsänderung bezieht sich die Überwachung immer auf den Werkzeugmittelpunkt. MD21020 hat keinen Einfluss.

---

### Hinweis

Das Maschinendatum \$MC\_TRAFO\_INCLUDES\_TOOL\_... wird nur bei bestimmten Transformationen ausgewertet. Bedingung für eine mögliche Auswertung ist, dass die Orientierung des Werkzeugs in Bezug auf das Basiskoordinatensystem durch die Transformation nicht verändert werden kann. Bei den Standardtransformationen ist diese Bedingung nur für die Transformationsart "Schräge Achse" erfüllt.

---

## Verhalten

### Automatikbetriebsarten

- Mit / ohne Transformation

Der Teileprogrammsatz, dessen programmierte Verfahrbewegung zu einem Überfahren der Arbeitsfeldbegrenzung führen würde, wird nicht ausgeführt.

- Mit überlagerter Bewegung

Die Achse, die durch eine überlagerte Bewegung die Arbeitsfeldbegrenzung verletzen würde, wird mit maximaler Bremsbeschleunigung ohne Ruckbegrenzung (*BRISK*) abgebremst und kommt auf der Position der Arbeitsfeldbegrenzung zum Stehen. Andere an der Bewegung beteiligte Achsen werden entsprechend des aktuellen Beschleunigungsverhaltens (z.B. *SOFT*) abgebremst. Durch unterschiedliche Bremsbeschleunigungen kann dabei der Bahnzusammenhang verloren gehen (Konturverletzung).

**Manuelle Betriebsarten**

- JOG mit / ohne Transformation

Die Achse kommt auf der Position der Arbeitsfeldbegrenzung zum Stehen.

**Einschaltverhalten**

Bewegt sich eine Achse beim Einschalten der Arbeitsfeldbegrenzung außerhalb des zulässigen Arbeitsfeldes, wird sie sofort mit maximaler Beschleunigung angehalten.

**Überfahren der Arbeitsfeldbegrenzung in der Betriebsart JOG**

In der Betriebsart JOG wird eine Achse von der Steuerung maximal bis zu ihrer Arbeitsfeldgrenze verfahren. Nach einer erneuten Betätigung der Verfahrtaste wird ein Alarm angezeigt und die Achse nicht weiter verfahren.

**Geometrieachstausch**

Über das folgende Maschinendatum ist einstellbar, ob beim Geometrieachstausch die aktive Arbeitsfeldbegrenzung erhalten bleibt oder deaktiviert wird:

MD10604 \$MN\_WALIM\_GEOAX\_CHANGE\_MODE = <Wert>

<Wert>	Bedeutung
0	Die Arbeitsfeldbegrenzung wird beim Geometrieachstausch deaktiviert.
1	Die Arbeitsfeldbegrenzung bleibt beim Geometrieachstausch aktiviert.

**2.2.7.2 Arbeitsfeldbegrenzung im BKS****Anwendung**

Durch die "Arbeitsfeldbegrenzung im BKS" wird der Arbeitsraum einer Werkzeugmaschine so begrenzt, dass die umliegenden Anlageteile (z. B. Werkzeugrevolver, Messstationen) vor Beschädigung geschützt sind.

**Arbeitsfeldgrenzen**

Die unteren und oberen Arbeitsfeldgrenzen der einzelnen Achsen werden über Settingdaten eingestellt oder über Teileprogrammanweisungen programmiert:

**Arbeitsfeldbegrenzung über Settingdaten**

Die Einstellungen erfolgen über die sofort wirksamen achsspezifischen Settingdaten:

SD43420 \$SA\_WORKAREA\_LIMIT\_PLUS (Arbeitsfeldbegrenzung plus)

SD43430 \$SA\_WORKAREA\_LIMIT\_MINUS (Arbeitsfeldbegrenzung minus)

### Programmierte Arbeitsfeldbegrenzung

Die Programmierung erfolgt über die G-Befehle:

- G25 X...Y...Z...      untere Arbeitsfeldbegrenzung
- G26 X...Y...Z...      obere Arbeitsfeldbegrenzung

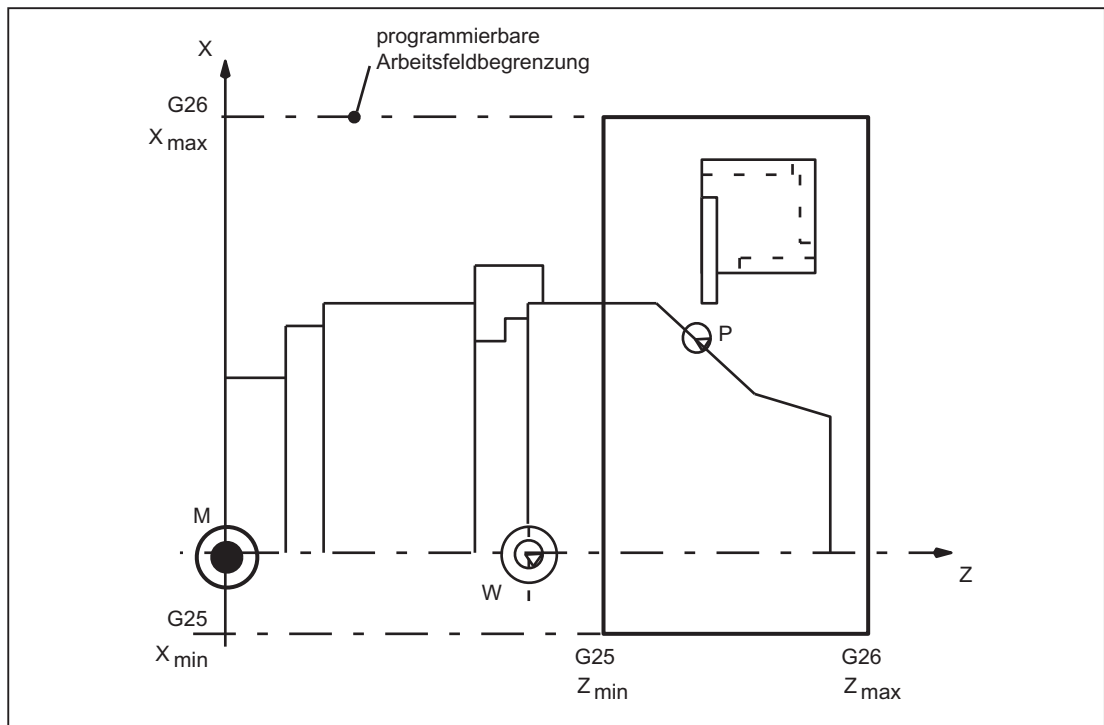


Bild 2-7      Programmierte Arbeitsfeldbegrenzung

Die programmierte Arbeitsfeldbegrenzung hat Vorrang und überschreibt die in SD43420 und SD43430 eingetragenen Werte.

### Aktivierung/Deaktivierung

#### Arbeitsfeldbegrenzung über Settingdaten

Die Aktivierung bzw. Deaktivierung der Arbeitsfeldbegrenzung für einzelne Achsen erfolgt richtungsspezifisch über die sofort wirksamen Settingdaten:

SD43400 \$SA\_WORKAREA\_PLUS\_ENABLE (Arbeitsfeldbegrenzung in positiver Richtung aktiv)

SD43410 \$SA\_WORKAREA\_MINUS\_ENABLE (Arbeitsfeldbegrenzung in negativer Richtung aktiv)

Wert	Bedeutung
0	Die Arbeitsfeldbegrenzung in positiver bzw. negativer Richtung ist <b>ausgeschaltet</b> .
1	Die Arbeitsfeldbegrenzung in positiver bzw. negativer Richtung ist <b>aktiv</b> .

### Programmierte Arbeitsfeldbegrenzung

Die Aktivierung bzw. Deaktivierung der gesamten "Arbeitsfeldbegrenzung im BKS" erfolgt durch die Teileprogrammanweisungen:

WALIMON            Arbeitsfeldbegrenzung EIN

bzw.

WALIMOF            Arbeitsfeldbegrenzung AUS

## Ändern der Arbeitsfeldbegrenzung

### Arbeitsfeldbegrenzung über Settingdaten

HMI-Bedienoberfläche: Bedienbereich "Parameter"

- Automatikbetriebsarten:
  - Änderungen: nur im RESET-Zustand möglich
  - Wirksamkeit: sofort
- Manuelle Betriebsarten:
  - Änderungen: immer möglich
  - Wirksamkeit: mit Start der nächsten Verfahrbewegung

### Programmierte Arbeitsfeldbegrenzung

Die Arbeitsfeldbegrenzung kann im Teileprogramm über  $G_{25}$  bzw.  $G_{26}$  <Achsbezeichner> <Wert> geändert werden. Die Änderung ist sofort wirksam.

Der neue Wert der Arbeitsfeldbegrenzung bleibt auch nach NC-RESET und POWER ON erhalten, wenn für SD43420 und SD43430 die Sicherung in die remanente Datenhaltung des NCK aktiviert wurde:

```
MD10710 $MN_PROG_SD_RESET_SAVE_TAB[0] = 43420
```

```
MD10710 $MN_PROG_SD_RESET_SAVE_TAB[1] = 43430
```

## Löschstellung

Die Löschstellung der Arbeitsfeldbegrenzung (WALIMON oder WALIMOF) ist einstellbar über:

```
MD20150 $MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen)
```

### 2.2.7.3 Arbeitsfeldbegrenzung im WKS/ENS

#### Anwendung

Die Arbeitsfeldbegrenzung ermöglicht kanalspezifische eine flexible werkstückspezifische Begrenzung des Verfahrbereichs der Kanalachsen im Werkstückkoordinatensystem (WKS) oder Einstellbaren Nullpunktsystem (ENS). Sie ist hauptsächlich für den Einsatz im Bereich konventioneller Drehmaschinen gedacht.

#### Arbeitsfeldbegrenzungsgruppe

Damit beim Umschalten von Achszuordnungen, z.B. beim Ein/Ausschalten von Transformationen oder des aktiven Frames, die achsspezifischen Arbeitsfeldgrenzen nicht für alle Kanalachsen neu geschrieben werden müssen, stehen Arbeitsfeldbegrenzungsgruppen zur Verfügung.

Eine Arbeitsfeldbegrenzungsgruppe umfasst folgende Daten:

- Arbeitsfeldgrenzen für alle Kanalachsen
- Bezugssystem der Arbeitsfeldbegrenzung

Die Anzahl der Arbeitsfeldbegrenzungsgruppen wird eingestellt im Maschinendatum:

MD28600 \$MC\_MM\_NUM\_WORKAREA\_CS\_GROUPS

#### Arbeitsfeldgrenzen einstellen

Die Arbeitsfeldgrenzen werden kanalspezifisch für jede Kanalachse über folgende Systemvariablen eingestellt:

- \$P\_WORKAREA\_CS\_LIMIT\_PLUS[<Gruppe>, <Kanalachsname>]
- \$P\_WORKAREA\_CS\_LIMIT\_MINUS[<Gruppe>, <Kanalachsname>]

mit <Gruppe> = Nummer der Arbeitsfeldbegrenzungsgruppe

#### Arbeitsfeldgrenzen freigeben

Die Arbeitsfeldgrenzen werden kanalspezifisch für jede Kanalachse über folgende Systemvariablen freigegeben:

- \$P\_WORKAREA\_CS\_PLUS\_ENABLE[<Gruppe>, <Kanalachsname>]
- \$P\_WORKAREA\_CS\_MINUS\_ENABLE[<Gruppe>, <Kanalachsname>]

mit <Gruppe> = Nummer der Arbeitsfeldbegrenzungsgruppe

Durch die Freigabe erfolgt keine Aktivierung.



### Bezugssystem auswählen

Das Bezugssystem für eine Arbeitsfeldbegrenzungsgruppe wird kanalspezifisch über folgende Systemvariable eingestellt:

`$P_WORKAREA_CS_COORD_SYSTEM[<Gruppe>] = <Wert>`

<Wert>	Bedeutung
1	Arbeitsfeldbegrenzung bezogen auf WKS
3	Arbeitsfeldbegrenzung bezogen auf ENS

mit <Gruppe> = Nummer der Arbeitsfeldbegrenzungsgruppe

### Arbeitsfeldgrenzen einschalten

Das Einschalten der Arbeitsfeldbegrenzungen einer Arbeitsfeldbegrenzungsgruppe erfolgt im Teileprogramm oder Synchronaktion über den Befehl:

`WALCSn` (Einschalten der Arbeitsfeldbegrenzungen der Gruppe n, mit n = 1, 2, ...)

### Arbeitsfeldgrenzen ausschalten

Das Ausschalten der Arbeitsfeldbegrenzungen einer Arbeitsfeldbegrenzungsgruppe erfolgt im Teileprogramm oder Synchronaktion über den Befehl:

`WALCS0` (Ausschalten der im Kanal aktiven Arbeitsfeldbegrenzungen)

### Arbeitsfeldgrenzen ändern

Über die oben genannten Systemvariablen können die Arbeitsfeldgrenzen jederzeit geändert werden. Änderungen werden mit dem nächsten Einschalten der Arbeitsfeldbegrenzungsgruppe (`WALCSn`) wirksam.

### Datenablage

Die Systemvariablen der Arbeitsfeldbegrenzungen werden remanent im statischen Speicher der NC abgelegt.

---

#### Hinweis

Bei Linearachsen wird bei der Ablage der Begrenzungswerte die Grundeinstellung für das Maßsystem (MD10240 `$MN_SCALING_SYSTEM_IS_METRIC`) berücksichtigt.

---

## Datensicherung

Die Systemvariablen der Arbeitsfeldbegrenzungen können in eigenen Dateien gesichert werden:

Sicherungsdatei	Zur Sicherung von:
_N_CHx_WAL	Werte der Systemvariablen für den Kanal x.
_N_COMPLETE_WAL	Werte der Systemvariablen für alle Kanäle.

### Hinweis

Die Systemvariablen der Arbeitsfeldbegrenzungen sind Bestandteil der Datei "\_N\_INITIAL\_INI".

## Verhalten in Betriebsart JOG

Ausgangssituation:

- In der Betriebsart JOG verfahren gleichzeitig **mehrere** Geometrieachsen (z. B. durch mehrere Handräder)
- Zwischen dem Basiskoordinatensystem (BKS) und dem Bezugskoordinatensystem der Arbeitsfeldbegrenzung (WKS oder ENS) ist ein **drehender** Frame aktiv

Verhalten bei Ansprechen einer Arbeitsfeldbegrenzung:

- Die Verfahrbewegungen der nicht betroffenen Geometrieachsen werden fortgesetzt
- Die betroffene Geometrieachse wird auf der Arbeitsfeldgrenze gestoppt

## Löschstellung einstellen

Die Festlegung der Arbeitsfeldbegrenzungsgruppe, die bei Hochlauf, Reset bzw. Teileprogrammende und Teileprogrammstart wirksam werden soll, erfolgt kanalspezifisch über das Maschinendatum:

MD20150 \$MC\_GCODE\_RESET\_VALUE[59] = <Gruppe>

Die Arbeitsfeldbegrenzungsgruppe, die wirksam wird, ist aber noch abhängig von der Einstellung im Maschinendatum:

MD20152 \$MC\_GCODE\_RESET\_MODE[59] = <Mode>

<Mode>	Bedeutung
0	Die Arbeitsfeldgruppe gemäß MD20150 wird wirksam
1	Die letzte aktive Arbeitsfeldgruppe bleibt aktiv

## 2.2.8 Ausschalten aller Überwachungen: "Parken"

Wird eine Maschinenachse in den Zustand "Parken" versetzt, werden bezüglich dieser Achse keine Geberistwerte mehr erfasst, sowie alle in den vorherigen Kapiteln beschriebenen Überwachungen (Messsystem-, Stillstands-, Klemmungsüberwachung etc.) ausgeschaltet.

### Maschinenachse mit Messsystem

Bei einer Maschinenachse mit Messsystem wird "Parken" durch Abwahl aller Messsysteme aktiviert:

- DB31, ... DBX1.5 = 0 (Lagemesssystem 1)
- DB31, ... DBX1.6 = 0 (Lagemesssystem 2)

Mit dem Abschalten der Messsysteme wird die Achse als nicht mehr referenziert gekennzeichnet:

- DB31, ... DBX60.4 = 0 (Referenziert/Synchronisiert 1)
- DB31, ... DBX60.5 = 0 (Referenziert/Synchronisiert 2)

---

#### Hinweis

Nach dem Aufheben des Zustands "Parken" muss die Achse neu referenziert werden.

---

### Maschinenachse ohne Messsystem

Bei einer Maschinenachse ohne Messsystem (drehzahlgesteuerte Spindel) wird ein dem "Parken" entsprechender Zustand durch Wegnahme der Reglerfreigabe aktiviert:

- DB31, ... DBX2.1 = 0 (Reglerfreigabe)

## 2.3 Schutzbereiche

### 2.3.1 Allgemeines

#### Funktion

Schutzbereiche sind statische oder bewegliche 2- bzw. 3-dimensionale Bereiche innerhalb einer Maschine zum Schutz von Maschinenelementen vor Kollisionen.

Folgende Elemente können geschützt werden:

- Feste Bestandteile der Maschine und Anbauten (z. B. Werkzeugmagazin, einschwenkbarer Messtaster). Von Interesse sind hier nur die Elemente, die durch mögliche Achskonstellationen erreicht werden können.
- Bewegliche Teile, die zum Werkzeug gehören (z. B. Werkzeug, Werkzeugträger).
- Bewegliche Teile, die zum Werkstück gehören (z. B. Teile des Werkstücks, Aufspanntisch, Spannpratzen, Spindelfutter, Reitstock).

Schutzbereiche werden über Teileprogrammanweisungen oder Systemvariablen so definiert, dass sie das zu schützende Element vollständig umschließen. Die Aktivierung bzw. Deaktivierung von Schutzbereichen erfolgt ebenfalls über Teileprogrammanweisungen.

Die Überwachung der Schutzbereiche erfolgt durch die NC kanalbezogen, d. h. es werden alle aktiven Schutzbereiche eines Kanals gegenseitig auf Kollision überwacht.

#### Definition eines Schutzbereichs

Es können 2- oder 3-dimensionale Schutzbereiche als Polygonzüge mit maximal 10 Eckpunkten definiert werden. Die Schutzbereiche können auch Kreisbögen als Konturelemente enthalten.

Die Polygonzüge werden in einer vorher festgelegten Ebene definiert.

Die Ausdehnung in der 3. Dimension kann zwischen  $-\infty$  bis  $+\infty$  begrenzt werden.

Dabei sind 4 Fälle möglich:

- Dimension des Schutzbereiches von  $-\infty$  bis  $+\infty$
- Dimension des Schutzbereiches von  $-\infty$  bis obere Grenze
- Dimension des Schutzbereiches von untere Grenze bis  $+\infty$
- Dimension des Schutzbereiches von untere Grenze bis obere Grenze

#### Koordinatensystem

Die Definition eines Schutzbereiches erfolgt bezogen auf die Geometrieachsen eines Kanals im Basiskoordinatensystem.

## Bezug

- Werkzeugbezogene Schutzbereiche  
Koordinaten für werkzeugbezogene Schutzbereiche sind absolut, bezogen auf den Werkzeugträgerbezugspunkt F, anzugeben.
- Werkstückbezogene Schutzbereiche  
Koordinaten für werkstückbezogene Schutzbereiche sind absolut, bezogen auf den Nullpunkt des Basiskoordinatensystems, anzugeben.

---

### Hinweis

Ist kein werkzeugbezogener Schutzbereich aktiv, wird die Werkzeugbahn gegen die Werkstück-bezogenen Schutzbereiche geprüft.

Ist kein werkstückbezogener Schutzbereich aktiv, so findet keine Schutzbereichsüberwachung statt.

---

## Orientierung

Die Orientierung der Schutzbereiche wird bestimmt durch die Festlegung der Ebene (Abszisse/Ordinate), in der die Konturbeschreibung erfolgt und die senkrecht auf der Kontur des Schutzbereiches stehende Achse (Applikate).

Die Orientierung der Schutzbereiche muss für werkzeug- und werkstückbezogene Schutzbereiche gleich sein.

## 2.3.2 Schutzbereichsarten

### Maschinen- und kanaldefinierte Schutzbereiche

- Maschinendefinierter Schutzbereich  
Daten für maschinenbezogene Schutzbereiche sind einmal in der Steuerung definiert. Diese Schutzbereiche können von allen Kanälen aktiviert werden.
- Kanaldefinierte Schutzbereiche  
Daten für kanalbezogene Schutzbereiche sind in einem Kanal definiert. Diese Schutzbereiche können nur von diesem Kanal aktiviert werden.

### Beispiel: Doppelschlittendrehmaschine

- Die werkzeugbezogenen Schutzbereiche werden Kanal 1 bzw. 2 zugeordnet.
- Die werkstückbezogenen Schutzbereiche werden der Maschine zugeordnet.
- Das Koordinatensystem muss für beide Kanäle gleich sein.

### Maximale Anzahl von Schutzbereichen

Die Einstellung für die maximal definierbare Anzahl von maschinen- und kanalbezogenen Schutzbereichen erfolgt über:

MD18190 \$MN\_MM\_NUM\_PROTECT\_AREA\_NCK (Anzahl der Dateien für maschinenbezogene Schutzbereiche)

MD28200 \$MC\_MM\_NUM\_PROTECT\_AREA\_CHAN (Anzahl der Dateien für kanalspezifische Schutzbereiche)

### Koordinaten

Die Koordinaten eines Schutzbereiches sind immer absolut zum Bezugspunkt des Schutzbereiches anzugeben. Bei der Aktivierung des Schutzbereiches über das Teileprogramm ist es möglich, den Bezugspunkt des Schutzbereiches relativ zu verschieben.

### Beispiele

In den folgenden Abbildungen sind einige Beispiele für Schutzbereiche dargestellt:

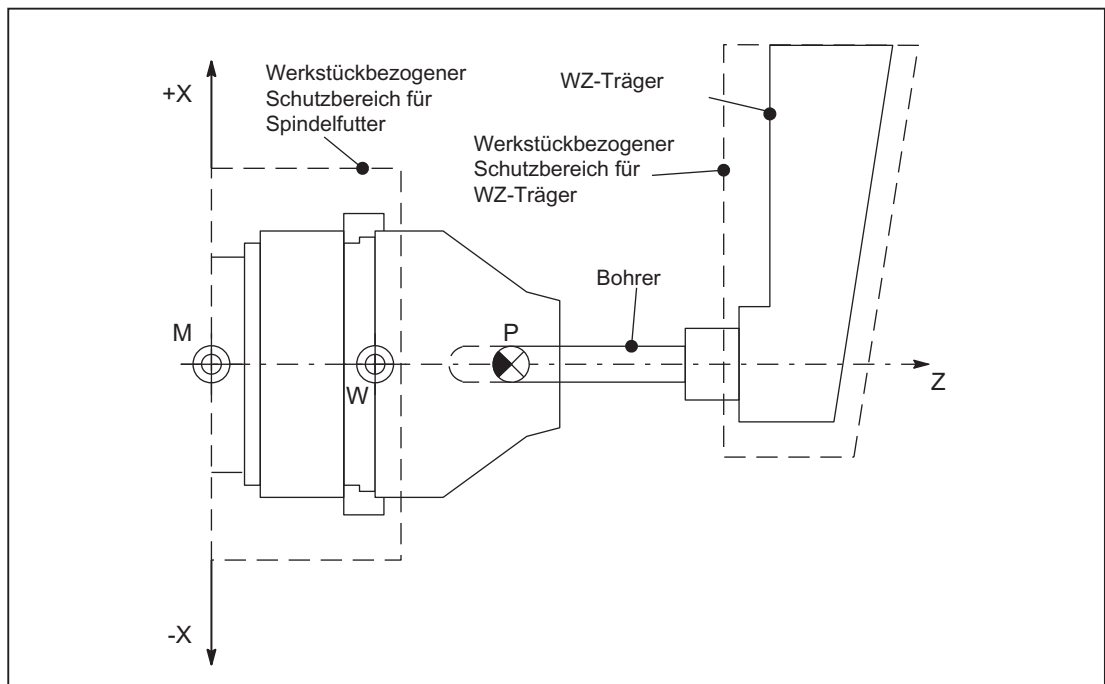


Bild 2-8 Beispiel für Drehmaschine

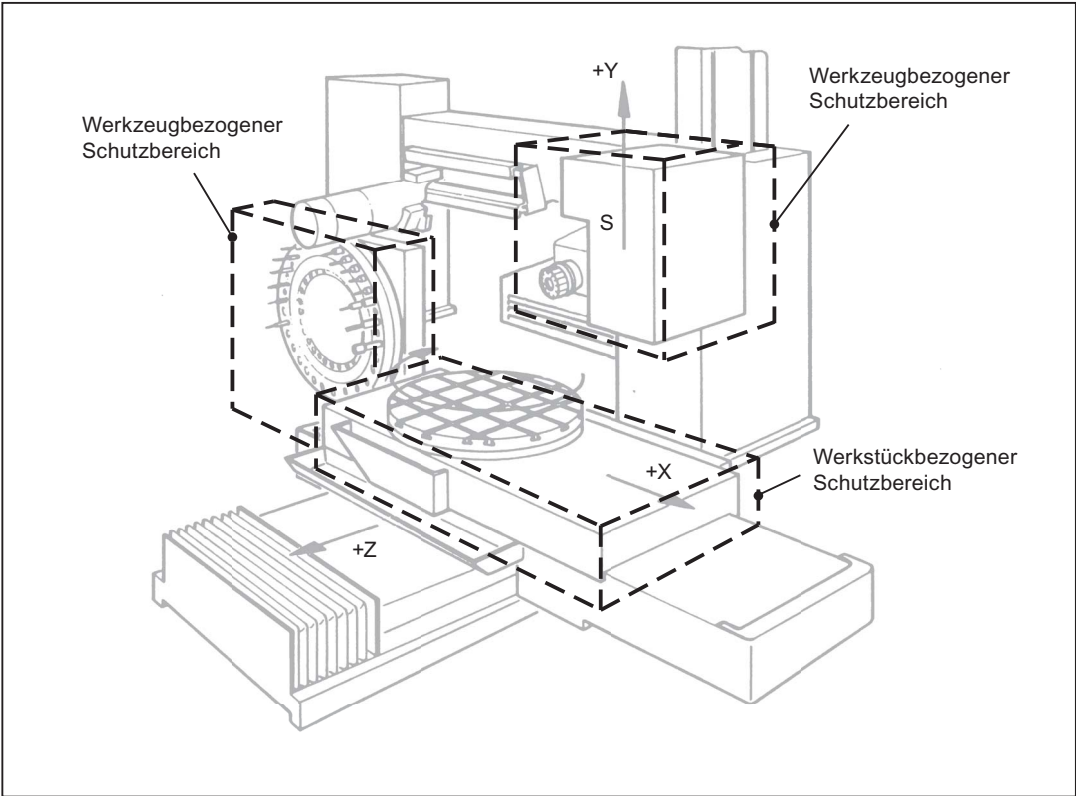


Bild 2-9 Beispiel für Fräsmaschine

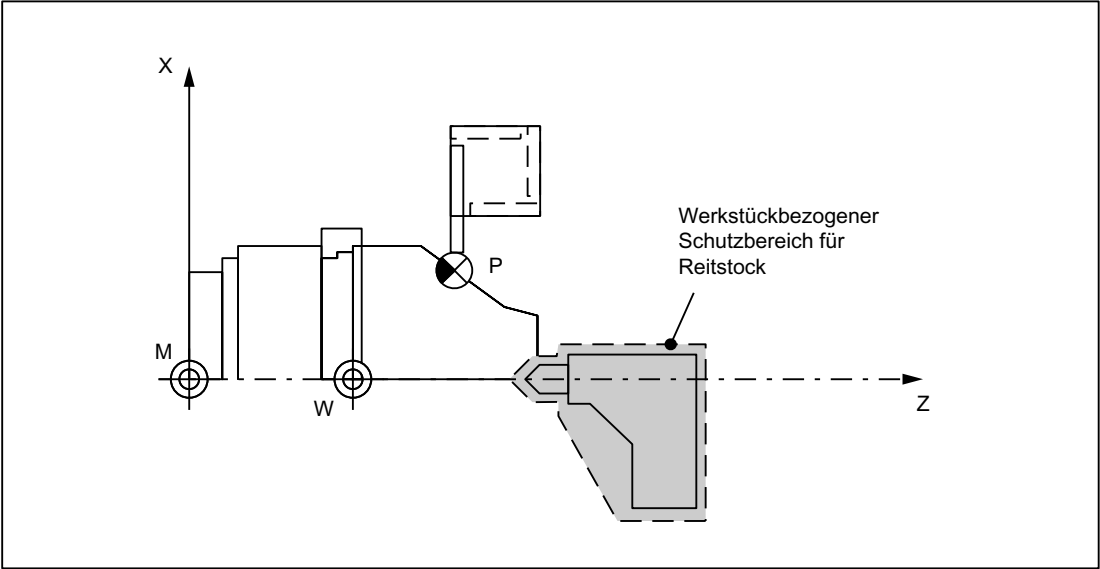


Bild 2-10 Beispiel für Drehmaschine mit Schutzbereich für Reitstock

### 2.3.3 Definition per Teileprogrammanweisung

#### Allgemeines

Eine Schutzbereichsdefinition umfasst folgende Informationen:

- Schutzbereichstyp (werkstück- oder werkzeugbezogen)
- Orientierung des Schutzbereichs
- Art der Begrenzung in der 3. Dimension
- Obere und untere Grenze des Schutzbereichs in der 3. Dimension
- Aktivierungstyp ("Schutzbereich sofort aktiv": nur über Systemvariable möglich)
- Konturelemente

#### Definition Schutzbereiche

Bei der Definition von Schutzbereichen ist folgende Systematik einzuhalten:

- Festlegung der Arbeitsebene: G17, G18 oder G19
- Definitionsbeginn
  - Kanalspezifischer Schutzbereiche: CPROTDEF(...)
  - Maschinen- bzw. NC-spezifischer Schutzbereich: NPROTDEF(...)
- Konturbeschreibung des Schutzbereichs
- Definitionsende: EXECUTE(...)

#### Festlegung der Arbeitsebene

Die gewünschte Arbeitsebene, auf die sich die Konturbeschreibung des Schutzbereichs bezieht, muss vor Definitionsbeginn mit G17, G18, G19 angewählt werden. Sie darf vor dem Definitionsende nicht geändert werden. Eine Programmierung der Applikate zwischen Definitionsbeginn und -ende ist nicht zulässig.



## Definitionsbeginn

Der Definitionsbeginn wird durch das entsprechende Unterprogramm definiert:

- CPROTDEF(n, t, applim, appplus, appminus)
- NPROTDEF(n, t, applim, appplus, appminus)

Parameter	Typ	Beschreibung	
n	INT	Nummer des definierten Schutzbereichs	
t	BOOL	Schutzbereichstyp	
		TRUE	Werkzeug-orientierter Schutzbereich
		FALSE	Werkstück-orientierter Schutzbereich
applim	INT	Art der Begrenzung in der 3. Dimension	
		0	Keine Begrenzung
		1	Begrenzung in Plus-Richtung
		2	Begrenzung in Minus-Richtung
		3	Begrenzung in Plus- und Minus-Richtung
appminus	REAL	Wert der Begrenzung in Minus-Richtung in der 3. Dimension <sup>1)</sup>	
appplus	REAL	Wert der Begrenzung in Plus-Richtung in der 3. Dimension <sup>1)</sup>	
1) Es muss gelten: appplus > appminus			

## Konturbeschreibung des Schutzbereichs

Die Kontur eines Schutzbereichs wird mittels Verfahrbewegungen beschrieben. Diese werden nicht ausgeführt und haben keine Verbindung zu vorhergehenden oder nachfolgenden Geometriebeschreibungen. Sie definieren ausschließlich den Schutzbereich.

Die Kontur eines Schutzbereichs wird mit maximal 11 Verfahrbewegungen in der angewählten Arbeitsebene beschrieben. Dabei ist die erste Verfahrbewegung die Bewegung an die Kontur. Der letzte Punkt der Konturbeschreibung muss immer mit dem ersten Punkt der Konturbeschreibung zusammenfallen. Bei rotationssymmetrischen Konturen (z. B. Spindelfutter) ist die Gesamtkontur zu beschreiben, nicht nur die Kontur bis zur Drehmitte.

2.3 Schutzbereiche

Als Schutzbereich gilt der Bereich links von der Kontur:

- Innenschutzbereich  
Die Kontur für einen Innenschutzbereich ist gegen den Uhrzeigersinn zu beschreiben.
- Außenschutzbereiche (Nur zulässig für werkstückbezogene Schutzbereiche)  
Die Kontur für einen Außenschutzbereich ist im Uhrzeigersinn zu beschreiben.

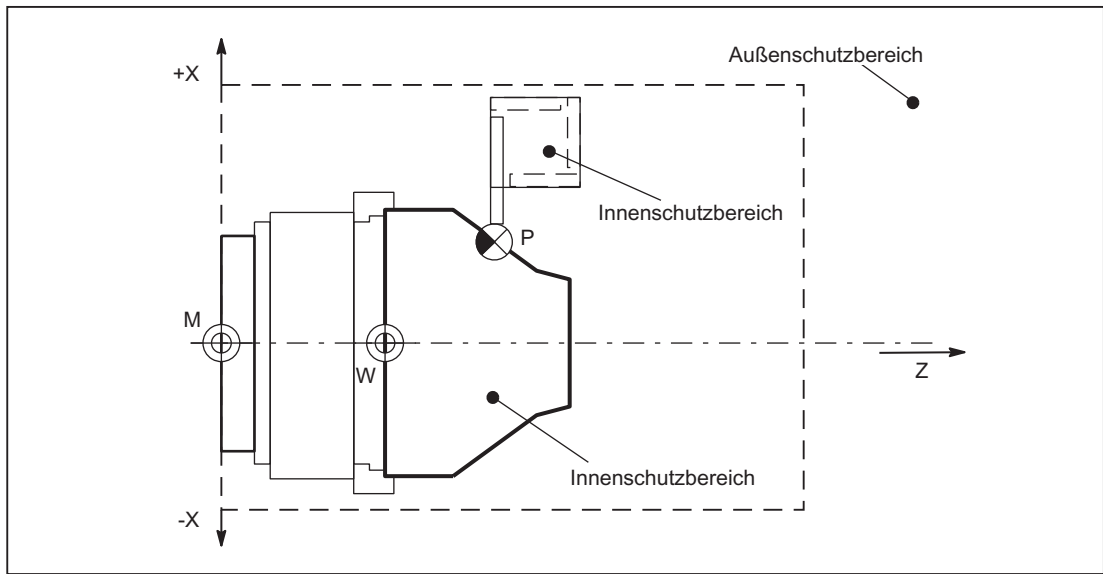


Bild 2-11 Beispiele: Außen- und Innenschutzbereich

Werkzeugbezogene Schutzbereiche müssen konvex sein. Wird ein konkaver Schutzbereich gewünscht, so ist der Schutzbereich in mehrere konvexe Schutzbereiche zu zerlegen.

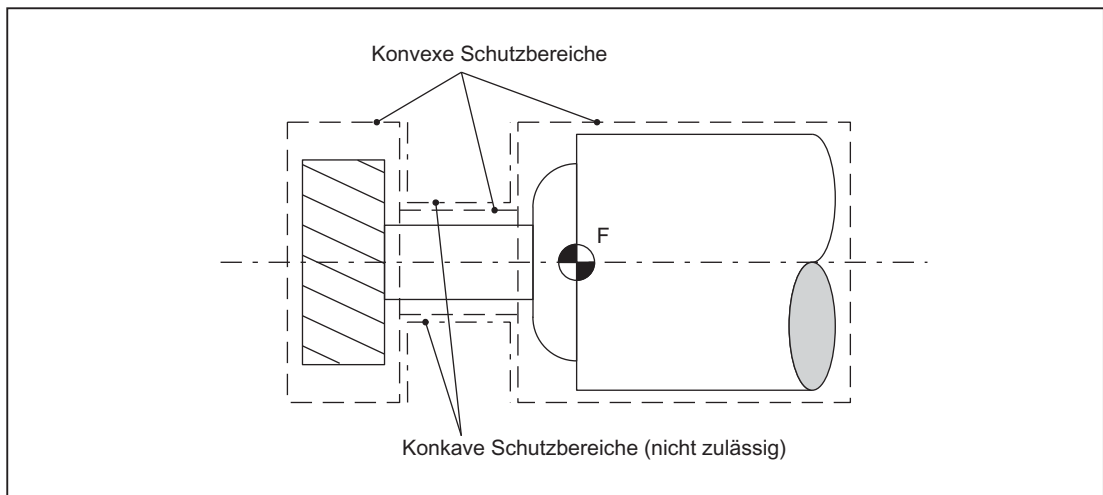


Bild 2-12 Beispiele: konvexe und konkave werkzeugbezogene Schutzbereiche

## Konturelemente

Folgende Konturelemente sind zulässig:

- G<sub>0</sub>, G<sub>1</sub> für gerade Konturelemente
- G<sub>2</sub> für Kreisabschnitte im Uhrzeigersinn

Nur zulässig bei werkstückbezogenen Schutzbereichen.

Nicht zulässig bei werkzeugbezogenen Schutzbereichen, da sie nur konvex sein dürfen.

- G<sub>3</sub> für Kreisabschnitte gegen den Uhrzeigersinn

Ein Schutzbereich kann nicht durch einen Vollkreis beschrieben werden. Ein Vollkreis muss in zwei Teilkreise aufgeteilt werden.

Die Folge G<sub>2</sub>, G<sub>3</sub> bzw. G<sub>3</sub>, G<sub>2</sub> ist nicht zulässig. Zwischen die beiden Kreissätze muss ein kurzer G<sub>1</sub>-Satz eingefügt werden.

## Randbedingungen

Während der Definition eines Schutzbereichs dürfen folgende Funktionen nicht aktiv sein bzw. verwendet werden:

- Werkzeugradiuskorrektur (Fräserradiuskorrektur, Schneidradiuskorrektur)
- Transformation
- Referenzpunktfahren (G<sub>74</sub>)
- Festpunktfahren (G<sub>75</sub>)
- Verweilzeit (G<sub>4</sub>)
- Satzvorlauf-Stopp (STOPRE)
- Programmende (M<sub>17</sub>, M<sub>30</sub>)
- M-Funktionen: M<sub>0</sub>, M<sub>1</sub>, M<sub>2</sub>

Programmierbare Frames (TRANS, ROT, SCALE, MIRROR) und einstellbare Frames (G<sub>54</sub> bis G<sub>57</sub>) sind unwirksam.

Inch-/Metrisch-Umschaltungen mit G<sub>70</sub>/G<sub>71</sub> bzw. G<sub>700</sub>/G<sub>710</sub> sind wirksam.

## Definitionsende

Das Definitionsende wird durch folgendes Unterprogramm definiert:

```
EXECUTE (NOT_USED)
```

Parameter	Typ	Beschreibung
NOT_USED	INT	Die Fehlervariable ist bei Schutzbereichen mit EXECUTE wirkungslos.

Die Definition eines maschinen- oder kanalspezifischen Schutzbereichs endet mit dem Unterprogramm EXECUTE (n).

### 2.3.4 Definition per Systemvariable

#### Allgemeines

Bei der Definition der Schutzbereiche über Teileprogrammanweisungen (siehe Kapitel "Definition per Teileprogrammanweisung (Seite 128)") werden die Schutzbereichsdaten in Systemvariablen abgelegt. Die Systemvariablen können auch direkt geschrieben werden, d. h. die Definition von Schutzbereichen kann auch direkt in den Systemvariablen vorgenommen werden.

Für die Beschreibung der Kontur eines Schutzbereichs gelten dabei die gleichen Randbedingungen wie bei der Schutzbereichsdefinition über Teileprogrammanweisungen.

#### Systemvariablen

Die Schutzbereichsdefinitionen umfassen folgende Systemvariablen:

Systemvariable	Typ	Bedeutung
\$SN_PA_ACTIV_IMMED[n] \$SC_PA_ACTIV_IMMED[n]	BOOL	Aktivierungstyp Der Schutzbereich ist nach dem Hochlaufen der Steuerung und dem Referieren der Achsen sofort aktiv/nicht aktiv. FALSE   nicht sofort aktiv TRUE   sofort aktiv
\$SN_PA_T_W[n] \$SC_PA_T_W[n]	INT	Schutzbereichstyp 0   Werkstück-bezogener Schutzbereich 1   reserviert 2   reserviert 3   Werkzeug-bezogener Schutzbereich
\$SN_PA_ORI[n] \$SC_PA_ORI[n]	INT	Orientierung des Schutzbereichs, d. h. Polygonzug in der Ebene aus: 0   1. und 2. Geometrieachse 1   3. und 1. Geometrieachse 2   2. und 3. Geometrieachse
\$SN_PA_LIM_3DIM[n] \$SC_PA_LIM_3DIM[n]	INT	Art der Begrenzung in der 3. Dimension 0   keine Begrenzung 1   Begrenzung in Plus-Richtung 2   Begrenzung in Minus-Richtung 3   Begrenzung in Plus- und Minus-Richtung
\$SN_PA_PLUS_LIM[n] \$SC_PA_PLUS_LIM[n]	REAL	Wert der Begrenzung in Plus-Richtung in der 3. Dimension
\$SN_PA_MINUS_LIM[n] \$SC_PA_MINUS_LIM[n]	REAL	Wert der Begrenzung in Minus-Richtung in der 3. Dimension
\$SN_PA_CONT_NUM[n] \$SC_PA_CONT_NUM[n]	INT	Anzahl der gültigen Konturelemente
\$SN_PA_CONT_TYP[n, i] \$SC_PA_CONT_TYP[n, i]	INT	Konturtyp[i], Konturtyp (G1, G2, G3) des i-ten Konturelements

Systemvariable	Typ	Bedeutung
\$SN_PA_CONT_ABS[n, i] \$SC_PA_CONT_ABS[n, i]	REAL	Endpunkt der Kontur[i], Abszissenwert
\$SN_PA_CONT_ORD[n, i] \$SC_PA_CONT_ORD[n, i]	REAL	Endpunkt der Kontur[i], Ordinatenwert
\$SN_PA_CENT_ABS[n, i] \$SC_PA_CENT_ABS[n, i]	REAL	Mittelpunkt der Kreiskontur[i], absoluter Abszissenwert
\$SN_PA_CENT_ORD[n, i] \$SC_PA_CENT_ORD[n, i]	REAL	Mittelpunkt der Kreiskontur[i], absoluter Ordinatenwert
\$SN... sind Systemvariable für NC- bzw. Maschinen-spezifische Schutzbereiche. \$SC... sind Systemvariable für kanalspezifische Schutzbereiche. Der Index "n" entspricht der Nummer des Schutzbereichs: 0 = 1. Schutzbereich Der Index "i" entspricht der Nummer des Konturelements: 0 = 1. Konturelement Die Konturelemente sind in aufsteigender Reihenfolge zu definieren.		

### Hinweis

Die Systemvariablen der Schutzbereichsdefinitionen werden bei REORG nicht restauriert.

## Daten der Schutzbereichsdefinitionen

### Datenablage

Die Schutzbereichsdefinitionen werden in den folgenden Dateien abgelegt:

Datei	Bausteine
_N_NCK_PRO	Datenbaustein für NC-spezifische Schutzbereiche
_N_CHAN1_PRO	Datenbaustein für kanalspezifische Schutzbereiche im Kanal 1
_N_CHAN2_PRO	Datenbaustein für kanalspezifische Schutzbereiche im Kanal 2

### Datensicherung

Die Schutzbereichsdefinitionen werden in den folgenden Dateien gesichert:

Datei	Bausteine
_N_INITIAL_INI	Alle Datenbausteine der Schutzbereiche
_N_COMPLETE_PRO	Alle Datenbausteine der Schutzbereiche
_N_CHAN_PRO	Alle Datenbausteine der kanalspezifischen Schutzbereiche

### 2.3.5 Aktivieren und Deaktivieren von Schutzbereichen

Der Aktivierungsstatus eines Schutzbereichs kann folgende Werte annehmen:

- Aktiviert
- Voraktiviert
- Voraktiviert mit bedingtem Stopp
- Deaktiviert

#### Aktivierung, Voraktivierung und Deaktivierung über Teileprogramm

Der Aktivierungsstatus eines Schutzbereichs kann im Teileprogramm jederzeit über folgende Funktionen verändert werden:

- Kanalspezifischer Schutzbereich:

CPROT(<n>, <state>[, <xMov>, <yMov>, <zMov>])

- Maschinenspezifischer Schutzbereich:

NPROT(<n>, <state>[, <xMov>, <yMov>, <zMov>])

Parameter	Typ	Beschreibung
<n>:	INT	Schutzbereichsnummer
<state>:	INT	Aktivierungsstatus
		0   deaktiviert
		1   voraktiviert
		2   aktiviert
		3   voraktiviert mit bedingtem Stopp
<xMov>, <yMov>, <zMov>:	REAL	Additive Verschiebungswerte abhängig vom Bezugssystem: <ul style="list-style-type: none"> <li>• Werkstück-bezogene Schutzbereich: Maschinennullpunkt</li> <li>• Werkzeug- bezogene Schutzbereich: Werkzeugträgerbezugspunkt</li> </ul>

#### ACHTUNG

Ein Schutzbereich wird erst nach dem Referenzieren aller Geometrieachsen des Kanals in dem er aktiviert wurde, berücksichtigt.

### Aktivierung über NC/PLC-Nahtstellensignale

Nur Schutzbereiche, die über das Teileprogramm **voraktiviert** (siehe unten Absatz "Voraktivierung über Teileprogramm") wurden, können im PLC-Anwenderprogramm über die NC/PLC-Nahtstellensignale aktiviert werden:

- DB21, ... DBX8.0 - 9.1 = 1 (Maschinenbezogenen Schutzbereich 1 - 10 aktivieren)
- DB21, ... DBX10.0 - 11.1 = 1 (Kanalspezifischen Schutzbereich 1 - 10 aktivieren)

Zu Voraktivierung siehe unten Absatz "Voraktivierung über Teileprogramm"

Die Aktivierung voraktivierter Schutzbereiche muss vor der Verfahrbewegung der Geometrieachsen erfolgen werden. Erfolgt die Aktivierung während der Verfahrbewegung, werden diese Schutzbereiche für die aktuelle Verfahrbewegung nicht mehr berücksichtigt.  
Reaktion:

- Alarm "10704 Schutzbereichsüberwachung ist nicht gewährleistet"
- DB21, ... DBX39.0 = 1(Schutzbereichsüberwachung nicht gewährleistet)

<b>ACHTUNG</b>
Die Aktivierung voraktivierter Schutzbereiche muss vor der Verfahrbewegung der Geometrieachsen erfolgen werden.

### Automatische Aktivierung nach Hochlauf der NC

Schutzbereiche, die sofort nach Hochlauf der NC aktiv sein sollen, können über folgende Systemvariablen festgelegt werden:

- Kanalspezifische Schutzbereiche:  
\$SC\_PA\_ACTIV\_IMMED[<Schutzbereichsnummer>]
- Maschinenspezifische Schutzbereiche:  
\$SN\_PA\_ACTIV\_IMMED[<Schutzbereichsnummer>]

---

#### Hinweis

Bei automatischer Aktivierung ist keine relative Verschiebung eines Schutzbereichs möglich.

<b>ACHTUNG</b>
Ein Schutzbereich wird erst nach dem Referenzieren aller Geometrieachsen des Kanals in dem er aktiviert wurde, berücksichtigt.

### Voraktivierung über Teileprogramm

Schutzbereiche die zu einem späteren Zeitpunkt vom PLC-Anwenderprogramm aus aktiviert werden sollen, müssen im Teileprogramm voraktiviert werden:

CPROT bzw. NPROT (<Schutzbereichsnummer>, 1)

Voraktivierte Schutzbereiche werden über folgende NC/PLC-Nahtstellensignale angezeigt:

- DB21, ... DBX272.0 - 273.1 == 1 (Maschinenbezogener Schutzbereich 1 - 10 voraktiviert)
- DB21, ... DBX274.0 - 275.1 == 1 (Kanalspezifischer Schutzbereich 1 - 10 voraktiviert)

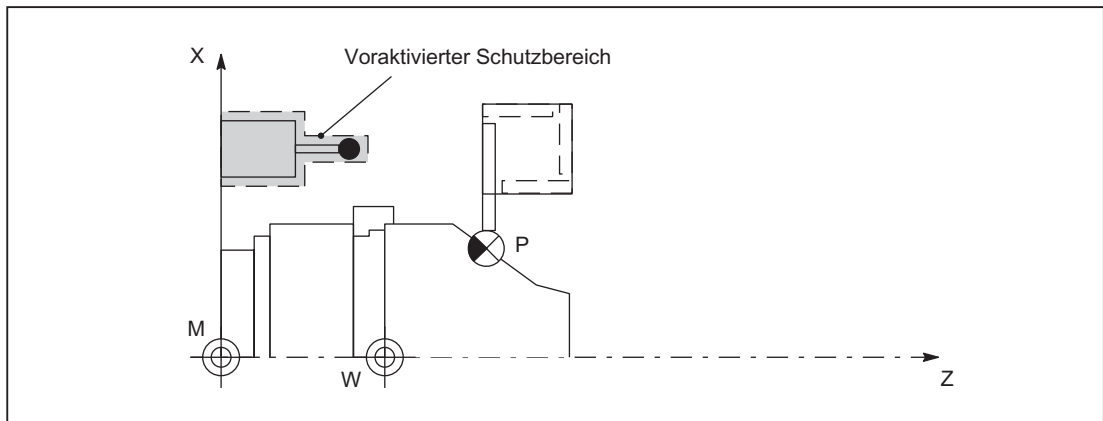


Bild 2-13 Beispiel: Drehmaschine mit voraktiviertem Schutzbereich für einen Messtaster

### Voraktivierung mit bedingtem Stopp

Die Voraktivierung von Schutzbereichen mit bedingtem Stopp erfolgt im Teileprogramm über:

CPROT bzw. NPROT (<Schutzbereichsnummer>, 3)

Bei einem voraktivierten Schutzbereich mit bedingtem Stopp wird eine Verfahrbewegung **nicht** vor diesem angehalten, wenn die Verfahrbewegung in den Schutzraum hinein führt. Ein Stopp erfolgt nur, wenn der Schutzbereich aktiviert wurde. Dieses Verhalten soll eine vom Anwender steuerbare unterbrechungsfreie Bearbeitung für den Fall ermöglichen, dass der Schutzbereich nur zeitweise benötigt wird.

---

#### Hinweis

#### Mögliche Schutzraumverletzung

Wird ein mit bedingtem Stopp voraktivierter Schutzraum nicht rechtzeitig aktiviert, kann die NC aufgrund des bis zum Aktivierungszeitpunkt nicht berücksichtigten Bremswegs unter Umständen nicht mehr rechtzeitig vor dem Schutzraum anhalten.

---



### Deaktivierung über Teileprogramm

Ein Schutzbereich, der über Teileprogramm oder NC/PLC-Nahtstellensignal aktiviert wurde, kann jederzeit aus dem Teilprogramm heraus mit Status = 0 wieder deaktiviert werden:

- Kanalspezifischer Schutzbereich:

`CROT(n, 0)`

- Maschinenspezifischer Schutzbereich:

`NROT(n, 0)`

### Deaktivierung über NC/PLC-Nahtstellensignal

Nur Schutzbereiche, die über ein Teileprogramm **voraktiviert** und über die NC/PLC-Nahtstellensignale aktiviert wurden, können über die NC/PLC-Nahtstellensignale wieder deaktiviert werden:

- DB21, ... DBX8.0 bis DBX9.1 = 0 (Maschinenbezogenen Schutzbereich 1 - 10 aktivieren)
- DB21, ... DBX10.0 bis DBX11.1 = 0 (Kanalspezifischen Schutzbereich 1 - 10 aktivieren)

Schutzbereiche, die über ein Teileprogramm direkt **aktiviert** wurden, können vom PLC-Anwenderprogramm aus **nicht** deaktiviert werden.

---

#### Hinweis

Es wird empfohlen, Schutzbereiche, die über das PLC-Anwenderprogramm aktiviert werden sollen, speziell dafür vorzusehen sind. Nur für diese Schutzbereiche ist eine Voraktivierung im Teileprogramm sinnvoll.

---

### Automatisches Deaktivieren über Maschinendatenparametrierung

Beim Ausführen der Funktionen **Geometrieachstausch** und **Transformationswechsel** können die aktiven Schutzbereiche automatisch deaktiviert werden. Die Einstellung erfolgt NC-spezifisch über das Maschinendatum:

MD10618 \$MN\_PROTAREA\_GEOAX\_CHANGE\_MODE

Bit	Wert	Bedeutung
0	0	Die aktiven Schutzbereiche werden beim Transformationswechsel deaktiviert.
	1	Die aktiven Schutzbereiche bleiben beim Transformationswechsel aktiv.
1	0	Die aktiven Schutzbereiche werden beim Geometrieachstausch deaktiviert.
	1	Die aktiven Schutzbereiche bleiben beim Geometrieachstausch aktiv.

### Schutzbereichsverletzung anzeigen

Verletzungen von aktivierten Schutzbereichen oder mögliche Verletzungen von voraktivierten Schutzbereichen, falls diese aktiviert würden, werden über folgende NC/PLC-Nahtstellensignale angezeigt:

- DB21, ... DBX276.0 - 277.1 == 1 (Maschinenbezogener Schutzbereich 1 - 10 verletzt)
- DB21, ... DBX278.0 - 279.1 == 1 (Kanalspezifischer Schutzbereich 1 - 10 verletzt)

### Verhalten in besonderen Systemzuständen

#### Satzsuchlauf mit Berechnung

Bei Satzsuchlauf mit Berechnung wird immer der zuletzt programmierte Aktivierungszustand eines Schutzbereichs berücksichtigt.

#### Programmtest

In den Betriebsarten AUTOMATIK und MDA werden aktivierte und voraktivierte Schutzbereiche auch im Zustand "Programmtest" überwacht.

#### NC-RESET und Programmende

Der Aktivierungsstatus eines Schutzbereichs bleibt über `NC-RESET` und Programmende hinaus erhalten.

### Speicherbedarf

Der für Schutzbereiche benötigte Speicherbedarf wird über folgende Maschinendaten parametrisiert:

- Persistenten Speicher
  - MD18190 \$MN\_MM\_NUM\_PROTECT\_AREA\_NCK (Anzahl der verfügbaren maschinendefinierten Schutzbereiche)
  - MD28200 \$MC\_MM\_NUM\_PROTECT\_AREA\_CHAN (Anzahl der verfügbaren kanaldefinierten Schutzbereiche)
- Dynamischer Speicher
  - MD28210 \$MC\_MM\_NUM\_PROTECT\_AREA\_ACTIVE (Maximale Anzahl der gleichzeitig im Kanal aktivierbaren Schutzbereiche)
  - MD28212 \$MC\_MM\_NUM\_PROTECT\_AREA\_CONTUR (Maximale Anzahl definierbarer Konturelemente pro Schutzbereich)

### Siehe auch

Definition per Teileprogrammanweisung (Seite 128)

Definition per Systemvariable (Seite 132)

## 2.3.6 Schutzbereichsverletzung und zeitweise Freigabe einzelner Schutzbereiche

### Zeitweise Freigabe von Schutzbereichen

Tritt beim Start oder während einer Verfahrbewegung eine Schutzbereichsverletzung auf, kann unter bestimmten Voraussetzungen der Schutzbereich vorübergehend d.h. zeitweise zum Durchfahren freigegeben werden. Sowohl in den Betriebsarten AUTOMATIK und MDA als auch in der Betriebsart JOG erfolgt die zeitweise Freigabe von Schutzbereichen über Bedienhandlungen.

Eine zeitweise Freigabe ist **nur** bei **werkstückbezogenen** Schutzbereichen möglich.

**Werkzeugbezogene** Schutzbereiche müssen entweder im Teileprogramm deaktiviert oder über die NC/PLC-Nahtstelle in den Zustand "voraktiviert" versetzt werden.

#### Beenden der zeitweisen Freigabe

Die zeitweise Freigabe eines Schutzbereichs ist nach folgenden Ereignissen beendet:

- NC-RESET
- Betriebsarten AUTOMATIK oder MDA: Das Satzende liegt außerhalb des Schutzbereichs
- Betriebsart JOG: Das Ende der Verfahrbewegung liegt außerhalb des Schutzbereichs
- Der Schutzbereich wird aktiviert

### Verhalten in den Betriebsarten AUTOMATIK und MDA

In den Betriebsarten AUTOMATIK und MDA werden keine Verfahrbewegungen in oder durch aktive Schutzbereiche freigegeben:

- Eine Verfahrbewegung, die von außerhalb in einen aktiven Schutzbereich hinein führen würde, wird am Satzendpunkt des letzten, außerhalb des Schutzbereichs liegenden Satzes angehalten.
- Eine Verfahrbewegung, die innerhalb eines aktiven Schutzbereichs beginnt, wird nicht gestartet.

#### Zeitweises Freigeben von Schutzbereichen

Wird in der Betriebsart AUTOMATIK oder MDA eine Verfahrbewegung aufgrund einer Schutzbereichsverletzung angehalten, wird dies dem Bediener durch einen Alarm angezeigt. Entscheidet der Bediener, dass die Verfahrbewegung fortgesetzt werden kann, kann er das Durchfahren von Schutzbereichen freigeben.

Die Freigabe ist nur temporär und erfolgt über das Auslösen von NC-START:

DB21, ... DBX7.1 = 1 (NC-START)

Pro verletzten Schutzraum wird ein Alarm angezeigt. Für jeden freizugebenden Schutzraum muss vom Bediener ein NC-START Signal ausgelöst werden.

Die Verfahrbewegung wird fortgesetzt, wenn alle Schutzbereiche die zum Anhalten der Verfahrbewegung geführt haben, freigegeben sind.

#### **Fortsetzen einer Verfahrbewegung ohne zeitweises Freigeben**

Eine Verfahrbewegung wurde aufgrund einer Schutzbereichsverletzung mit Alarm angehalten. Wird der betroffene Schutzbereich über die NC/PLC-Nahtstelle in den Zustand "voraktiviert" versetzt, kann die Verfahrbewegung mit NC-START fortgesetzt werden, ohne dass der Schutzbereich zeitweise freigegeben wird.

#### **Stärkere Absicherung der Freigabe von Schutzbereichen**

Soll die Freigabe eines Schutzbereiches stärker als nur durch NC-START abgesichert werden, ist dies vom Maschinenhersteller bzw. Anwender im PLC-Anwenderprogramm auszuführen.

### **Verhalten in der Betriebsart JOG**

#### **Gleichzeitiges Verfahren mehrerer Geometrieachsen**

In der Betriebsart JOG können Verfahrbewegungen in mehreren Geometrieachsen gleichzeitig ausgeführt werden. Der Verfahrbereich jeder beteiligten Geometrieachse wird achsspezifisch zum Startzeitpunkt der Verfahrbewegung bezüglich der Verfahrbereichsgrenzen (Softwareendschalter, Arbeitsfeldbegrenzung, etc.) und aktiven Schutzbereiche begrenzt. Eine sichere Überwachung aller aktiven Schutzbereiche kann dabei nicht gewährleistet werden. Als Rückmeldung an den Anwender erfolgt:

- Alarm: "10704 Schutzbereichsüberwachung nicht gewährleistet"
- DB31, ... DBX39.0 = 1 (Schutzbereichsüberwachung nicht gewährleistet)

Nach dem Ende der Verfahrbewegungen wird der Alarm automatisch gelöscht. Liegt die aktuelle Position innerhalb eines aktivierten oder voraktivierten Schutzbereichs erfolgt:

- Alarm 10702 bzw. 10703 wird angezeigt
- Weitere Verfahrbewegungen werden gesperrt
- Das NC/PLC-Nahtstellensignal für den betroffenen Schutzbereich wird gesetzt  
DB21, ... DBX276.0 – 277.1 bzw. DBX278.0 – 279.1 (Schutzbereich verletzt)

Zum Fortsetzen siehe unten Absatz "Zeitweises Freigeben von Schutzbereichen" und Kapitel "Aktivieren und Deaktivieren von Schutzbereichen (Seite 134)".

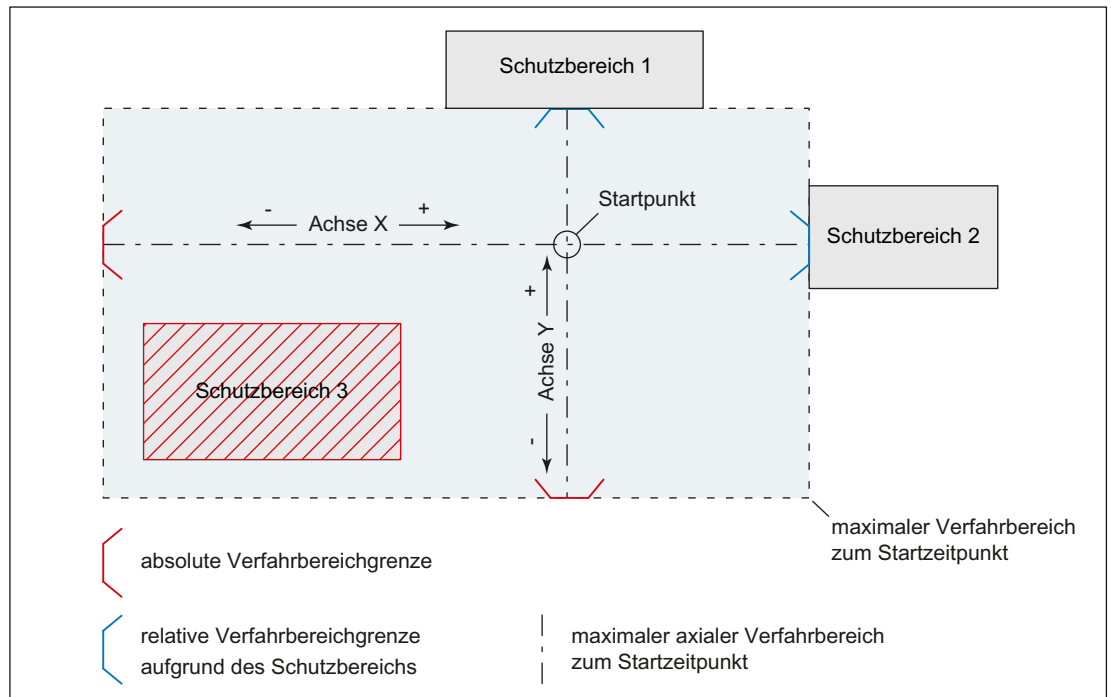
**Beispiel: 3 aktivierte Schutzbereiche und gleichzeitiges Verfahren von 2 Geometrieachsen**

Bild 2-14 Bewegungsbereich der Geometrieachsen zum Startzeitpunkt

Zum Startzeitpunkt der Verfahrbewegungen der Achsen X und Y werden, ausgehend vom Startpunkt, die axialen Verfahrbereichsgrenzen ermittelt:

- X Achse
  - positive Verfahrriichtung: Schutzbereich 2
  - negative Verfahrriichtung: absolute Verfahrbereichsgrenze z.B. Softwareendschalter
- Y Achse
  - positive Verfahrriichtung: Schutzbereich 1
  - negative Verfahrriichtung: absolute Verfahrbereichsgrenze z.B. Arbeitsfeldbegrenzung

Der sich daraus ergebende maximale Verfahrbereich zum Startzeitpunkt berücksichtigt dabei nicht den Schutzbereich 3. Somit wäre eine Schutzbereichsverletzung im Schutzbereich 3 möglich.

**Hinweis**

Aktivierte und voraktivierte Schutzbereiche werden auch in den manuellen Betriebsarten JOG, INC und DRF überwacht.

### **Begrenzung der Verfahrbewegung einer Achse**

Wird die Verfahrbewegung einer Achse durch das Erreichen eines Schutzbereiches begrenzt, so wird ein selbstlöschender Alarm "Schutzbereich beim JOG erreicht" erzeugt mit Angabe des verletzten Schutzbereiches und der verfahrenen Achse. Es ist beim Verfahren einer Achse mit JOG sicher gestellt, dass kein Schutzbereich verletzt ist. (Dieses Verhalten ist analog dem Fahren auf Software-Endschalter bzw. Arbeitsfeldbegrenzung.)

Der Alarm wird gelöscht:

- Beim Fahren einer Achse, die nicht in den Schutzbereich führt
- Beim Freigeben des Schutzbereiches
- Bei NC-RESET

Wird an einer Schutzbereichsgrenze eine Bewegung Richtung des Schutzbereichs gestartet, so kommt ein selbstlöschender Alarm "Schutzbereich beim JOG erreicht" und die Bewegung wird nicht gestartet.

### **Zeitweises Freigeben von Schutzbereichen**

Wird innerhalb oder auf der Grenze eines aktivierten Schutzbereichs eine Verfahrbewegung gestartet, wird Alarm 10702 bzw. 10703 angezeigt und die Verfahrbewegung nicht gestartet. Die Verfahrbewegung kann ausgeführt werden, wenn der betroffene Schutzbereich dafür zeitweise freigegeben wird. Folgende Aktionen sind dazu auszuführen:

- Erzeugen einer positiven Flanke auf dem NC/PLC-Nahtstellensignal: DB21, ... DBX1.1 (Schutzbereich freigeben)
- Erneuter Start der gleichen Verfahrbewegung

---

#### **Hinweis**

Für den zeitweilig freigegebenen Schutzbereich wird dennoch beim Durchfahren das NC/PLC-Nahtstellensignal "Schutzbereich verletzt" gesetzt:

DB21, ... DBX276.0 – 277.1 bzw. DBX278.0 – 279.1

---

Wird eine Verfahrbewegung gestartet, die nicht in den freigegebenen Schutzbereich führt, wird die Freigabe zurückgesetzt.

Sind durch die Verfahrbewegung weitere Schutzbereiche betroffen, werden pro Schutzbereich weitere Alarme 10702 bzw. 10703 angezeigt. Die in den Alarmen angezeigten Schutzbereiche können durch das Erzeugen weiterer positiver Flanken auf dem NC/PLC-Nahtstellensignal freigegeben werden:

DB21, ... DBX1.1 (Schutzbereich freigeben)

### **Verhalten bei Betriebsartenwechsel**

Die in der Betriebsart JOG gegebenen zeitweise Freigabe von Schutzbereichen bleiben nach einem Wechsel in die Betriebsart AUTOMATIK bzw. MDA erhalten. Ebenso bleiben die zeitweiligen Freigaben die in der Betriebsart AUTOMATIK bzw. MDA gegebenen wurden bei einem Wechsel in die Betriebsart JOG erhalten.

### Rücksetzen einer Freigabe

Mit dem nächsten Stillstand einer Geometrieachse bei dem der zeitweilig freigegebene Schutzbereich vollständig verlassen wurde, wird die Freigabe intern und an der NC/PLC-Nahtstelle rückgesetzt:

DB21, ... DBX276.0 – 277.1 bzw. DBX278.0 – 279.1 = 0 (Schutzbereich verletzt)

### Verhalten bei Achstausch

Bezüglich der Schutzbereiche wird nach einem Achstausch im abgebenden Kanal von der letzten im Kanal angefahrenen Position ausgegangen. Verfahrbewegungen im übernehmenden Kanal werden nicht berücksichtigt. Es ist daher darauf zu achten, dass ein Achstausch nicht aus einer Position mit Schutzraumverletzung erfolgt.

### Verhalten bei überlagerten Bewegungen

Überlagerte Bewegungen, die im Hauptlauf eingerechnet werden, können von der Satzaufbereitung bezüglich der aktiven Schutzbereiche nicht berücksichtigt werden.

Es erfolgen folgende Reaktionen:

- Alarm: "10704 Schutzbereichsüberwachung nicht gewährleistet"
- DB31, ... DBX39.0 = 1 (Schutzbereichsüberwachung nicht gewährleistet)

## 2.3.7 Einschränkungen bei Schutzbereichen

### Einschränkungen der Schutzbereichsüberwachung

Unter folgenden Bedingungen ist keine Schutzbereichsüberwachung möglich:

- Orientierungsachsen
- Schutzbereichsüberwachung für feststehende Maschinen-bezogene Schutzräume bei TRANSMIT oder Mantelflächentransformation.  
Ausnahme: Schutzbereiche, die rotationssymmetrisch um die Spindelachse definiert sind. Dabei darf keine DRF-Verschiebung aktiv sein.
- Gegenseitiges Überwachen von Werkzeug-bezogenen Schutzräumen

### Positionierachsen

Bei Positionierachsen wird nur der programmierte Satzpunkt überwacht.

Während der Verfahrbewegung der Positionierachsen wird ein Alarm angezeigt:

Alarm: " 10704 Schutzbereichsüberwachung nicht gewährleistet".

### **Achstausch**

Ist eine für den Achstausch vorgesehene Achse in einem Kanal nicht aktiv, wird die zuletzt im Kanal angefahrne Position der Achse als aktuelle Position angenommen. Wurde diese Achse noch nicht im Kanal verfahren, wird die Position Null angenommen.

### **Maschinenbezogene Schutzbereiche**

Ein Maschinen-bezogener Schutzbereich bzw. dessen Kontur wird mittels der Geometrieachsen d.h. bezogen auf das Basiskoordinatensystem (BKS) eines Kanals definiert. Damit eine korrekte Schutzbereichsüberwachung in allen Kanälen in denen der Maschinen-bezogene Schutzbereich aktiv ist, stattfinden kann, müssen die Basiskoordinatensystem (BKS) aller betroffenen Kanäle identisch (Lage des Koordinatenursprungs bezogen auf den Maschinennullpunkt und Orientierung der Koordinatenachsen) sein.



## 2.4 Randbedingungen

### 2.4.1 Achsüberwachungen

#### Einstellungen

Für das korrekte Arbeiten der Überwachungen sind neben den genannten Maschinendaten folgende Einstellungen vorzunehmen bzw. zu prüfen:

##### Allgemein

- MD31030 \$MA\_LEADSCREW\_PITCH (Steigung der Kugelrollspindel)
- MD31050 \$MA\_DRIVE\_AX\_RATIO\_DENOM (Nenner Lastgetriebe)
- MD31060 \$MA\_DRIVE\_AX\_RATIO\_NUMERA (Zähler Lastgetriebe)
- MD31070 \$MA\_DRIVE\_ENC\_RATIO\_DENOM (Nenner Messgetriebe)
- MD31080 \$MA\_DRIVE\_ENC\_RATIO\_NUMERA (Zähler Messgetriebe)
- MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung)
- Geberauflösung

Die entsprechenden Maschinendaten siehe Kapitel "G2: Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (Seite 331)".

##### Nur Antriebe mit analoger Drehzahlsollwertschnittstelle

- MD32260 \$MA\_RATED\_VELO (Nenn-Motordrehzahl)
- MD32250 \$MA\_RATED\_OUTVAL (Nenn-Ausgangsspannung)

## 2.5 Beispiele

### 2.5.1 Achsüberwachungen

#### 2.5.1.1 Arbeitsfeldbegrenzung im WKS/ENS

##### Vorhandene Kanalachsen

Im Kanal sind 4 Achsen definiert: X, Y, Z und A

Die A-Achse ist eine Rundachse (nicht modulo).

##### Anzahl der Arbeitsfeldbegrenzungsgruppen parametrieren

Es sollen 3 Arbeitsfeldbegrenzungsgruppen bereitgestellt werden:

MD28600 \$MC\_MM\_NUM\_WORKAREA\_CS\_GROUP = 3

##### Arbeitsfeldbegrenzungsgruppen definieren

Ferner sollen 2 Arbeitsfeldbegrenzungsgruppen definiert werden:

###### Arbeitsfeldbegrenzungsgruppe 1

In der ersten Arbeitsfeldbegrenzungsgruppe sollen die Achsen im ENS-Koordinatensystem begrenzt werden:

- X-Achse in Plus-Richtung: 10 mm
- X-Achse in Minus-Richtung: keine Begrenzung
- Y-Achse in Plus-Richtung: keine Begrenzung
- Y-Achse in Minus-Richtung: 25 mm
- Z-Achse in Plus-Richtung: keine Begrenzung
- Z-Achse in Minus-Richtung: keine Begrenzung
- A-Achse in Plus-Richtung: 10 Grad
- A-Achse in Minus-Richtung: -40 Grad

Die Systemvariablen werden wie folgt belegt:

Programmcode	Kommentar
N1 \$P_WORKKAREA_CS_COORD_SYSTEM[1]=3	; Die Arbeitsfeldbegrenzung der Arbeitsfeldbegrenzungsgruppe 1 gilt im ENS.
N10 \$P_WORKKAREA_CS_PLUS_ENABLE[1,X]=TRUE	
N11 \$P_WORKKAREA_CS_LIMIT_PLUS[1,X]=10	
N12 \$P_WORKKAREA_CS_MINUS_ENABLE[1,X]=FALSE	
N20 \$P_WORKKAREA_CS_PLUS_ENABLE[1,Y]=FALSE	
N22 \$P_WORKKAREA_CS_MINUS_ENABLE[1,Y]=TRUE	
N23 \$P_WORKKAREA_CS_LIMIT_MINUS[1,Y]=25	
N30 \$P_WORKKAREA_CS_PLUS_ENABLE[1,Z]=FALSE	
N32 \$P_WORKKAREA_CS_MINUS_ENABLE[1,Z]=FALSE	
N40 \$P_WORKKAREA_CS_PLUS_ENABLE[1,A]=TRUE	
N41 \$P_WORKKAREA_CS_LIMIT_PLUS[1,A]=10	
N42 \$P_WORKKAREA_CS_MINUS_ENABLE[1,A]=TRUE	
N43 \$P_WORKKAREA_CS_LIMIT_MINUS[1,A]=-40	

### Arbeitsfeldbegrenzungsgruppe 2

In der zweiten Arbeitsfeldbegrenzungsgruppe sollen die Achsen im WKS-Koordinatensystem begrenzt werden:

- X-Achse in Plus-Richtung: 10 mm
- X-Achse in Minus-Richtung: keine Begrenzung
- Y-Achse in Plus-Richtung: 34 mm
- Y-Achse in Minus-Richtung: -25 mm
- Z-Achse in Plus-Richtung: keine Begrenzung
- Z-Achse in Minus-Richtung: -600 mm
- A-Achse in Plus-Richtung: keine Begrenzung
- A-Achse in Minus-Richtung: keine Begrenzung

Die Systemvariablen werden wie folgt belegt:

Programmcode	Kommentar
N51 \$P_WORKAREA_CS_COORD_SYSTEM[2]=1	; Die Arbeitsfeldbegrenzung der Arbeitsfeldbegrenzungsgruppe 2 gilt im WKS.
N60 \$P_WORKAREA_CS_PLUS_ENABLE[2,X]=TRUE	
N61 \$P_WORKAREA_CS_LIMIT_PLUS[2,X]=10	
N62	
\$P_WORKAREA_CS_MINUS_ENABLE[2,X]=FALSE	
N70 \$P_WORKAREA_CS_PLUS_ENABLE[2,Y]=TRUE	
N73 \$P_WORKAREA_CS_LIMIT_PLUS[2,Y]=34	
N72 \$P_WORKAREA_CS_MINUS_ENABLE[2,Y]=TRUE	
N73 \$P_WORKAREA_CS_LIMIT_MINUS[2,Y]=-25	
N80 \$P_WORKAREA_CS_PLUS_ENABLE[2,Z]=FALSE	
N82 \$P_WORKAREA_CS_MINUS_ENABLE[2,Z]=TRUE	
N83 \$P_WORKAREA_CS_LIMIT_PLUS[2,Z]=-600	
N90 \$P_WORKAREA_CS_PLUS_ENABLE[2,A]=FALSE	
N92	
\$P_WORKAREA_CS_MINUS_ENABLE[2,A]=FALSE	

### Arbeitsfeldbegrenzungsgruppe 2 aktivieren

Um die Arbeitsfeldbegrenzungsgruppe 2 zu aktivieren, muss im Teileprogramm folgende Anweisung stehen:

```
...  
N100 WALCS2 ...  
...
```

## 2.5.2 Schutzbereiche

### 2.5.2.1 Schutzbereichsdefinition und Aktivierung

#### Anforderung

Für eine Drehmaschine sollen folgende Innenschutzbereiche definiert werden:

- 1 maschinen- und werkstückbezogener Schutzbereich für das Spindelfutter, ohne Begrenzung in der 3. Dimension
- 1 kanalspezifischer Schutzbereich für das Werkstück, ohne Begrenzung in der 3. Dimension
- 1 kanalspezifischer, werkzeugbezogener Schutzbereich für den Werkzeugträger, ohne Begrenzung in der 3. Dimension

Zur Definition des Schutzbereiches für das Werkstück wird der Werkstücknullpunkt in den Maschinennullpunkt gelegt.

Bei der Aktivierung wird der Schutzbereich um 100 mm in der Z-Achse in positiver Richtung verschoben.

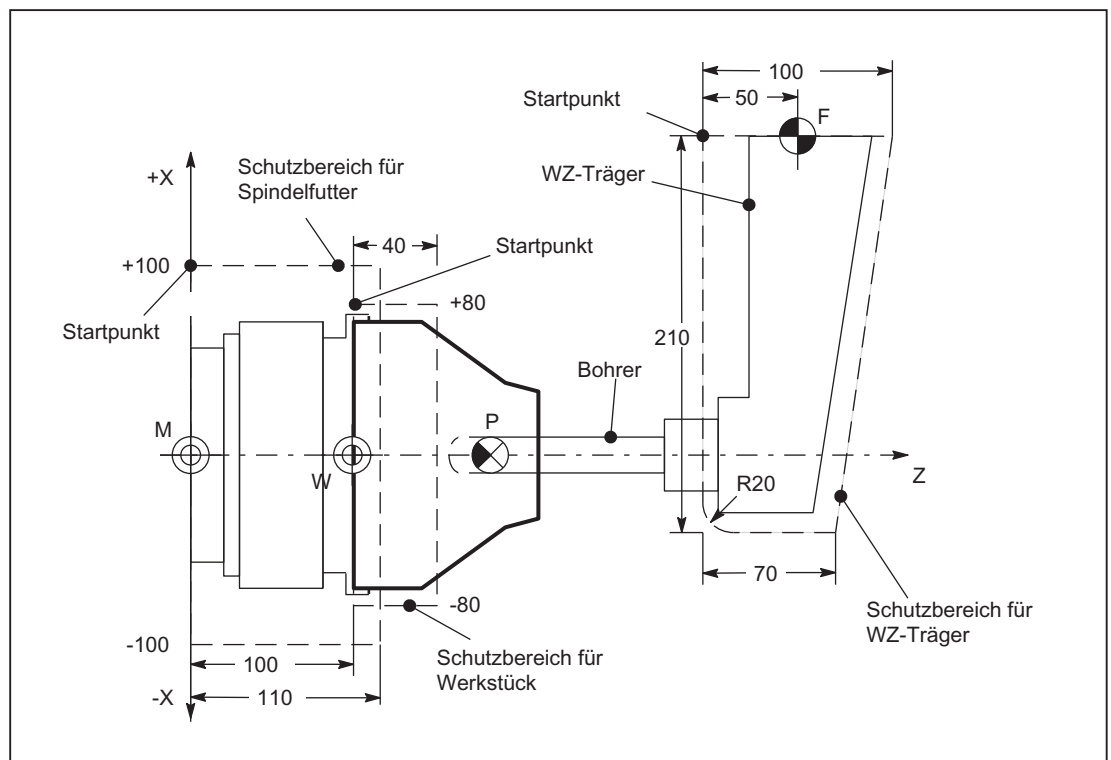


Bild 2-15 Beispiel für Schutzbereiche an einer Drehmaschine

## Schutzbereichsdefinition im Teileprogramm

Tabelle 2- 1 Teileprogramm Ausschnitt zur Schutzbereichsdefinition:

Programmcode	Kommentar
DEF INT AB	
G18	; Festlegung der Arbeitsebene
NPROTDEF (1, FALSE, 0, 0, 0)	; Definitionsbeginn: Schutzbereich für Spindelfutter
G01 X100 Z0	; Konturbeschreibung: 1. Konturelement
G01 X-100 Z0	; Konturbeschreibung: 2. Konturelement
G01 X-100 Z110	; Konturbeschreibung: 3. Konturelement
G01 X100 Z110	; Konturbeschreibung: 4. Konturelement
G01 X100 Z0	; Konturbeschreibung: 5. Konturelement
EXECUTE (AB)	; Definitionsende: Schutzbereich für Spindelfutter
CPROTDEF (1, FALSE, 0, 0, 0)	; Definitionsbeginn: Schutzbereich für Werkstück
G01 X80 Z0	; Konturbeschreibung: 1. Konturelement
G01 X-80 Z0	; Konturbeschreibung: 2. Konturelement
G01 X-80 Z40	; Konturbeschreibung: 3. Konturelement
G01 X80 Z40	; Konturbeschreibung: 4. Konturelement
G01 X80 Z0	; Konturbeschreibung: 5. Konturelement
EXECUTE (AB)	; Definitionsende: Schutzbereich für Werkstück
CPROTDEF (2, TRUE, 0, 0, 0)	; Definitionsbeginn: Schutzbereich für Werkzeugträger
G01 X0 Z-50	; Konturbeschreibung: 1. Konturelement
G01 X-190 Z-50	; Konturbeschreibung: 2. Konturelement
G03 X-210 Z-30 I-20	; Konturbeschreibung: 3. Konturelement
G01 X-210 Z20	; Konturbeschreibung: 4. Konturelement
G01 X0 Z50	; Konturbeschreibung: 5. Konturelement
G01 X0 Z-50	; Konturbeschreibung: 6. Konturelement
EXECUTE (AB)	; Definitionsende: Schutzbereich für Werkzeugträger

## Schutzbereichsdefinition mit Systemvariablen

Tabelle 2- 2 Schutzbereich: Spindelfutter

Systemvariable	Wert	Anmerkung
\$\$SN_PA_ACTIV_IMMED[0]	0	; Schutzbereich für Spindelfutter nicht sofort aktiv
\$\$SN_PA_T_W[0]	" "	; Maschinen-bezogener Schutzbereich für Spindelfutter
\$\$SN_PA_ORI[0]	1	; Orientierung des Schutzbereichs: 1= 3. und 1. Geometrieachse
\$\$SN_PA_LIM_3DIM[0]	0	; Art der Begrenzung in der 3. Dimension: 0 = keine Begrenzung
\$\$SN_PA_PLUS_LIM[0]	0	; Wert der Begrenzung in Plus-Richtung in der 3. Dimension
\$\$SN_PA_MINUS_LIM[0]	0	; Wert der Begrenzung in Minus-Richtung in der 3. Dimension
\$\$SN_PA_CONT_NUM[0]	4	; Anzahl der gültigen Konturelemente
\$\$SN_PA_CONT_TYP[0,0]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Spindelfutter, Konturelement 0
\$\$SN_PA_CONT_TYP[0,1]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Spindelfutter, Konturelement 1
\$\$SN_PA_CONT_TYP[0,2]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Spindelfutter, Konturelement 2
\$\$SN_PA_CONT_TYP[0,3]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Spindelfutter, Konturelement 3
\$\$SN_PA_CONT_TYP[0,4]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Spindelfutter, Konturelement 4
\$\$SN_PA_CONT_TYP[0,5]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Spindelfutter, Konturelement 5
\$\$SN_PA_CONT_TYP[0,6]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Spindelfutter, Konturelement 6
\$\$SN_PA_CONT_TYP[0,7]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Spindelfutter, Konturelement 7
\$\$SN_PA_CONT_TYP[0,8]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Spindelfutter, Konturelement 8
\$\$SN_PA_CONT_TYP[0,9]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Spindelfutter, Konturelement 9
\$\$SN_PA_CONT_ORD[0,0]	-100	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 0
\$\$SN_PA_CONT_ORD[0,1]	-100	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 1
\$\$SN_PA_CONT_ORD[0,2]	100	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 2
\$\$SN_PA_CONT_ORD[0,3]	100	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 3
\$\$SN_PA_CONT_ORD[0,4]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 4
\$\$SN_PA_CONT_ORD[0,5]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 5
\$\$SN_PA_CONT_ORD[0,6]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 6

2.5 Beispiele

\$SN_PA_CONT_ORD[0,7]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 7
\$SN_PA_CONT_ORD[0,8]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 8
\$SN_PA_CONT_ORD[0,9]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 9
\$SN_PA_CONT_ABS[0,0]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 0
\$SN_PA_CONT_ABS[0,1]	110	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 1
\$SN_PA_CONT_ABS[0,2]	110	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 2
\$SN_PA_CONT_ABS[0,3]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 3
\$SN_PA_CONT_ABS[0,4]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 4
\$SN_PA_CONT_ABS[0,5]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 5
\$SN_PA_CONT_ABS[0,6]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 6
\$SN_PA_CONT_ABS[0,7]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 7
\$SN_PA_CONT_ABS[0,8]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 8
\$SN_PA_CONT_ABS[0,9]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 9
\$SN_PA_CENT_ORD[0,0]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 0
\$SN_PA_CENT_ORD[0,1]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 1
\$SN_PA_CENT_ORD[0,2]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 2
\$SN_PA_CENT_ORD[0,3]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 3
\$SN_PA_CENT_ORD[0,4]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 4
\$SN_PA_CENT_ORD[0,5]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 5
\$SN_PA_CENT_ORD[0,6]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 6
\$SN_PA_CENT_ORD[0,7]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 7
\$SN_PA_CENT_ORD[0,8]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 8
\$SN_PA_CENT_ORD[0,9]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Spindelfutter, Konturelement 9
\$SN_PA_CENT_ABS[0,0]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 0



\$SN_PA_CENT_ABS[0,1]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 1
\$SN_PA_CENT_ABS[0,2]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 2
\$SN_PA_CENT_ABS[0,3]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 3
\$SN_PA_CENT_ABS[0,4]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 4
\$SN_PA_CENT_ABS[0,5]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 5
\$SN_PA_CENT_ABS[0,6]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 6
\$SN_PA_CENT_ABS[0,7]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 7
\$SN_PA_CENT_ABS[0,8]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 8
\$SN_PA_CENT_ABS[0,9]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Spindelfutter, Konturelement 9

Tabelle 2- 3 Schutzbereich: Werkstück und Werkzeugträger

Systemvariable	Wert	Anmerkung
\$SN_PA_ACTIV_IMMED[0]	0	; Schutzbereich für Werkstück nicht sofort aktiv
\$SN_PA_ACTIV_IMMED[1]	0	; Schutzbereich für Werkzeugträger nicht sofort aktiv
\$SC_PA_TW[0]	" "	; Schutzbereich für Werkstück, Kanal-spezifisch
\$SC_PA_TW[1]	'H01'	; Schutzbereich für Werkzeugträger, Kanal-spezifisch
\$SC_PA_ORI[0]	1	; Orientierung des Schutzbereichs: 1= 3. und 1. Geometrieachse ; Schutzbereich Werkstück
\$SC_PA_ORI[1]	1	; Orientierung des Schutzbereichs: 1= 3. und 1. Geometrieachse ; Schutzbereich Werkzeugträger
\$SC_PA_LIM_3DIM[0]	0	; Art der Begrenzung in der 3. Dimension: 0 = keine Begrenzung ; Schutzbereich Werkstück Werkzeugträger 0
\$SC_PA_LIM_3DIM[1]	0	; Art der Begrenzung in der 3. Dimension: 0 = keine Begrenzung ; Schutzbereich Werkzeugträger
\$SC_PA_PLUS_LIM[0]	0	; Wert der Begrenzung in Plus-Richtung in der 3. Dimension ; Schutzbereich Werkstück
\$SC_PA_PLUS_LIM[1]	0	; Wert der Begrenzung in Plus-Richtung in der 3. Dimension ; Schutzbereich Werkzeugträger
\$SC_PA_MINUS_LIM[0]	0	; Wert der Begrenzung in Minus-Richtung in der 3. Dimension ; Schutzbereich Werkstück
\$SC_PA_MINUS_LIM[1]	0	; Wert der Begrenzung in Minus-Richtung in der 3. Dimension ; Schutzbereich Werkzeugträger
\$SC_PA_CONT_NUM[0]	4	; Anzahl der gültigen Konturelemente, ; Schutzbereich Werkstück

2.5 Beispiele

\$SC_PA_CONT_NUM[1]	5	; Anzahl der gültigen Konturelemente, ; Schutzbereich Werkzeugträger 1
\$SN_PA_CONT_TYP[0,0]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkstück, Konturelement 0
\$SN_PA_CONT_TYP[0,1]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkstück, Konturelement 1
\$SN_PA_CONT_TYP[0,2]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkstück, Konturelement 2
\$SN_PA_CONT_TYP[0,3]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkstück, Konturelement 3
\$SN_PA_CONT_TYP[0,4]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkstück, Konturelement 4
\$SN_PA_CONT_TYP[0,5]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkstück, Konturelement 5
\$SN_PA_CONT_TYP[0,6]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkstück, Konturelement 6
\$SN_PA_CONT_TYP[0,7]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkstück, Konturelement 7
\$SN_PA_CONT_TYP[0,8]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkstück, Konturelement 8
\$SN_PA_CONT_TYP[0,9]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkstück, Konturelement 9
\$SN_PA_CONT_TYP[1,0]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkzeugträger, Konturelement 0
\$SN_PA_CONT_TYP[1,1]	3	; Konturtyp[i] : 3 = G3 f. Kreiselement gegen den Uhrzeigersinn, ; Schutzbereich Werkzeugträger, Konturelement 1
\$SN_PA_CONT_TYP[1,2]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkzeugträger, Konturelement 2
\$SN_PA_CONT_TYP[1,3]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkzeugträger, Konturelement 3
\$SN_PA_CONT_TYP[1,4]	1	; Konturtyp[i] : 1 = G1 für Gerade, ; Schutzbereich Werkzeugträger, Konturelement 4
\$SN_PA_CONT_TYP[1,5]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkzeugträger, Konturelement 5
\$SN_PA_CONT_TYP[1,6]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkzeugträger, Konturelement 6
\$SN_PA_CONT_TYP[1,7]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkzeugträger, Konturelement 7
\$SN_PA_CONT_TYP[1,8]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkzeugträger, Konturelement 8
\$SN_PA_CONT_TYP[1,9]	0	; Konturtyp[i] : 0 = nicht definiert, ; Schutzbereich Werkzeugträger, Konturelement 9
\$SN_PA_CONT_ORD[0,0]	-80	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 0
\$SN_PA_CONT_ORD[0,1]	-80	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 1
\$SN_PA_CONT_ORD[0,2]	80	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 2

\$SN_PA_CONT_ORD[0,3]	80	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 3
\$SN_PA_CONT_ORD[0,4]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 4
\$SN_PA_CONT_ORD[0,5]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 5
\$SN_PA_CONT_ORD[0,6]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 6
\$SN_PA_CONT_ORD[0,7]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 7
\$SN_PA_CONT_ORD[0,8]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 8
\$SN_PA_CONT_ORD[0,9]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 9
\$SN_PA_CONT_ORD[1,0]	-190	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 0
\$SN_PA_CONT_ORD[1,1]	-210	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 1
\$SN_PA_CONT_ORD[1,2]	-210	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 2
\$SN_PA_CONT_ORD[1,3]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 3
\$SN_PA_CONT_ORD[1,4]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 4
\$SN_PA_CONT_ORD[1,5]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 5
\$SN_PA_CONT_ORD[1,6]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 6
\$SN_PA_CONT_ORD[1,7]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 7
\$SN_PA_CONT_ORD[1,8]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 8
\$SN_PA_CONT_ORD[1,9]	0	; Endpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 9
\$SN_PA_CONT_ABS[0,0]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 0
\$SN_PA_CONT_ABS[0,1]	40	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 1
\$SN_PA_CONT_ABS[0,2]	40	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 2
\$SN_PA_CONT_ABS[0,3]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 3
\$SN_PA_CONT_ABS[0,4]	-50	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 4
\$SN_PA_CONT_ABS[0,5]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 5
\$SN_PA_CONT_ABS[0,6]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 6

2.5 Beispiele

\$SN_PA_CONT_ABS[0,7]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 7
\$SN_PA_CONT_ABS[0,8]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 8
\$SN_PA_CONT_ABS[0,9]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 9
\$SN_PA_CONT_ABS[1,0]	-50	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 0
\$SN_PA_CONT_ABS[1,1]	-30	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 1
\$SN_PA_CONT_ABS[1,2]	20	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 2
\$SN_PA_CONT_ABS[1,3]	50	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 3
\$SN_PA_CONT_ABS[1,4]	-50	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 4
\$SN_PA_CONT_ABS[1,5]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 5
\$SN_PA_CONT_ABS[1,6]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 6
\$SN_PA_CONT_ABS[1,7]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 7
\$SN_PA_CONT_ABS[1,8]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 8
\$SN_PA_CONT_ABS[1,9]	0	; Endpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 9
\$SN_PA_CENT_ORD[0,0]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 0
\$SN_PA_CENT_ORD[0,1]	-190	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 1
\$SN_PA_CENT_ORD[0,2]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 2
\$SN_PA_CENT_ORD[0,3]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 3
\$SN_PA_CENT_ORD[0,4]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 4
\$SN_PA_CENT_ORD[0,5]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 5
\$SN_PA_CENT_ORD[0,6]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 6
\$SN_PA_CENT_ORD[0,7]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 7
\$SN_PA_CENT_ORD[0,8]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkstück, Konturelement 8
\$SN_PA_CENT_ORD[0,9]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 9
\$SN_PA_CENT_ORD[1,0]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 0

\$SN_PA_CENT_ORD[1,1]	-190	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 1
\$SN_PA_CENT_ORD[1,2]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 2
\$SN_PA_CENT_ORD[1,3]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 3
\$SN_PA_CENT_ORD[1,4]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 4
\$SN_PA_CENT_ORD[1,5]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 5
\$SN_PA_CENT_ORD[1,6]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 6
\$SN_PA_CENT_ORD[1,7]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 7
\$SN_PA_CENT_ORD[1,8]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 8
\$SN_PA_CENT_ORD[1,9]	0	; Mittelpunkt der Kontur[i], Ordinatenwert ; Schutzbereich Werkzeugträger, Konturelement 9
\$SN_PA_CENT_ABS[0,0]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 0
\$SN_PA_CENT_ABS[0,1]	-30	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 1
\$SN_PA_CENT_ABS[0,2]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 2
\$SN_PA_CENT_ABS[0,3]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 3
\$SN_PA_CENT_ABS[0,4]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 4
\$SN_PA_CENT_ABS[0,5]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 5
\$SN_PA_CENT_ABS[0,6]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 6
\$SN_PA_CENT_ABS[0,7]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 7
\$SN_PA_CENT_ABS[0,8]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 8
\$SN_PA_CENT_ABS[0,9]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkstück, Konturelement 9
\$SN_PA_CENT_ABS[1,0]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 0
\$SN_PA_CENT_ABS[1,1]	-30	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 1
\$SN_PA_CENT_ABS[1,2]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 2
\$SN_PA_CENT_ABS[1,3]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 3
\$SN_PA_CENT_ABS[1,4]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 4

2.5 Beispiele

\$SN_PA_CENT_ABS[1,5]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 5
\$SN_PA_CENT_ABS[1,6]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 6
\$SN_PA_CENT_ABS[1,7]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 7
\$SN_PA_CENT_ABS[1,8]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 8
\$SN_PA_CENT_ABS[1,9]	0	; Mittelpunkt der Kontur[i], Abszissenwert ; Schutzbereich Werkzeugträger, Konturelement 9

**Aktivierung**

Tabelle 2-4 Teileprogramm Ausschnitt zur Aktivierung der drei Schutzbereiche für Spindelfutter, Werkstück und Werkzeugträger:

Programmcode	Kommentar
NPROT(1, 2, 0, 0, 0)	; Schutzbereich: Spindelfutter
CPROT(1, 2, 0, 0, 100)	; Schutzbereich: Werkstück mit Verschiebung um 100 mm in der Z-Achse.
CPROT(2, 2, 0, 0, 0)	; Schutzbereich: Werkzeugträger

## 2.6 Datenlisten

### 2.6.1 Maschinendaten

#### 2.6.1.1 NC-spezifische Maschinendaten

#### Achsüberwachungen

Nummer	Bezeichner: \$MN_	Beschreibung
10604	WALIM_GEOAX_CHANGE_MODE	Arbeitsfeldbegrenzung beim Umschalten von Geometrieachsen
10710	PROG_SD_RESET_SAVE_TAB	Zu aktualisierende Settingdaten

#### Schutzbereiche

Nummer	Bezeichner: \$MN_	Beschreibung
10618	PROTAREA_GEOAX_CHANGE_MODE	Schutzbereich beim Umschalten von Geoachsen
18190	MM_NUM_PROTECT_AREA_NCK	Anzahl der Dateien für maschinenbezogene Schutzbereiche

## 2.6.1.2 Kanal-spezifische Maschinendaten

## Achsüberwachungen

Nummer	Bezeichner: \$MC_	Beschreibung
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen
21020	WORKAREA_WITH_TOOL_RADIUS	Berücksichtigung des Werkzeugradius bei Arbeitsfeldbegrenzung
24130	TRAFO_INCLUDES_TOOL_1	Werkzeugbehandlung bei aktiver 1. Transformation
24230	TRAFO_INCLUDES_TOOL_2	Werkzeugbehandlung bei aktiver 2. Transformation
24330	TRAFO_INCLUDES_TOOL_3	Werkzeugbehandlung bei aktiver 3. Transformation
24426	TRAFO_INCLUDES_TOOL_4	Werkzeugbehandlung bei aktiver 4. Transformation
24436	TRAFO_INCLUDES_TOOL_5	Werkzeugbehandlung bei aktiver 5. Transformation
24446	TRAFO_INCLUDES_TOOL_6	Werkzeugbehandlung bei aktiver 6. Transformation
24456	TRAFO_INCLUDES_TOOL_7	Werkzeugbehandlung bei aktiver 7. Transformation
24466	TRAFO_INCLUDES_TOOL_8	Werkzeugbehandlung bei aktiver 8. Transformation
24476	TRAFO_INCLUDES_TOOL_9	Werkzeugbehandlung bei aktiver 9. Transformation
24486	TRAFO_INCLUDES_TOOL_10	Werkzeugbehandlung bei aktiver 10. Transformation
25106	TRAFO_INCLUDES_TOOL_11	Werkzeugbehandlung bei aktiver 11. Transformation
25116	TRAFO_INCLUDES_TOOL_12	Werkzeugbehandlung bei aktiver 12. Transformation
25126	TRAFO_INCLUDES_TOOL_13	Werkzeugbehandlung bei aktiver 13. Transformation
25136	TRAFO_INCLUDES_TOOL_14	Werkzeugbehandlung bei aktiver 14. Transformation
25146	TRAFO_INCLUDES_TOOL_15	Werkzeugbehandlung bei aktiver 15. Transformation
25156	TRAFO_INCLUDES_TOOL_16	Werkzeugbehandlung bei aktiver 16. Transformation
25166	TRAFO_INCLUDES_TOOL_17	Werkzeugbehandlung bei aktiver 17. Transformation
25176	TRAFO_INCLUDES_TOOL_18	Werkzeugbehandlung bei aktiver 18. Transformation
25186	TRAFO_INCLUDES_TOOL_19	Werkzeugbehandlung bei aktiver 19. Transformation
25196	TRAFO_INCLUDES_TOOL_20	Werkzeugbehandlung bei aktiver 20. Transformation
28600	MM_NUM_WORKAREA_CS_GROUPS	Anzahl an Koordinatensystem-spezifischen Arbeitsfeldbegrenzungen

## Schutzbereiche

Nummer	Bezeichner: \$MC_	Beschreibung
28200	MM_NUM_PROTECT_AREA_CHAN (SRAM)	Anzahl der Dateien für kanalspezifische Schutzbereiche
28210	MM_NUM_PROTECT_AREA_ACTIVE	Anzahl der gleichzeitig aktiven Schutzbereiche in einem Kanal
28212	MM_NUM_PROTECT_AREA_CONTUR	Elements für aktive Schutzbereiche (DRAM)



## 2.6.1.3 Achs-/Spindel-spezifische Maschinendaten

## Achsüberwachungen

Nummer	Bezeichner: \$MA_	Beschreibung
30240	ENC_TYPE	Geber-Typ der Istwerterfassung (Lageistwert)
30310	ROT_IS_MODULO	Modulowandlung für Rundachse/Spindel
30800	WORK_AREA_CHECK_TYPE	Art der Prüfung der Arbeitsfeldgrenzen
32200	POSCTRL_GAIN[n]	KV-Faktor
32250	RATED_OUTVAL	Nennausgangsspannung
32260	RATED_VELO	Motorenndrehzahl
32300	MAX_AX_ACCEL	Maximale Achsbeschleunigung
32800	EQUIV_CURRCTRL_TIME	Ersatzzeitkonstante Stromregelkreis für Vorsteuerung
32810	EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung
32910	DYN_MATCH_TIME[n]	Zeitkonstante der Dynamikanpassung
35160	SPIND_EXTERN_VELO_LIMIT	Spindeldrehzahlbegrenzung von PLCC
36000	STOP_LIMIT_COARSE	Genauhalt grob
36010	STOP_LIMIT_FINE	Genauhalt fein
36020	POSITIONING_TIME	Verzögerungszeit Genauhalt fein
36030	STANDSTILL_POS_TOL	Stillstandstoleranz
36040	STANDSTILL_DELAY_TIME	Verzögerungszeit Stillstandsüberwachung
36050	CLAMP_POS_TOL	Klemmungstoleranz bei NST "Klemmung aktiv"
36052	STOP_ON_CLAMPING	Sonderfunktionen bei geklemmter Achse
36060	STANDSTILL_VELO_TOL	Maximale Geschwindigkeit/Drehzahl "Achse/Spindel steht"
36100	POS_LIMIT_MINUS	1. Softwareendschalter minus
36110	POS_LIMIT_PLUS	1. Softwareendschalter plus
36120	POS_LIMIT_MINUS2	2. Softwareendschalter minus
36130	POS_LIMIT_PLUS2	2. Softwareendschalter plus
36200	AX_VELO_LIMIT	Schwellwert für Geschwindigkeitsüberwachung
36210	CTRLOUT_LIMIT	Maximaler Drehzahlsollwert
36220	CTRLOUT_LIMIT_TIME	Verzögerungszeit für Drehzahlsollwertüberwachung
36300	ENC_FREQ_LIMIT	Gebergrenzfrequenz
36302	ENC_FREQ_LIMIT_LOW	Gebergrenzfrequenz für Geber-Neusynchronisation
36310	ENC_ZERO_MONITORING	Nullmarkenüberwachung
36312	ENC_ABS_ZEROMON_WARNING	Nullmarkenüberwachung Warnschwelle
36400	CONTOUR_TOL	Toleranzband Konturüberwachung
36500	ENC_CHANGE_TOL	Maximale Toleranz bei Lageistwertumschaltung
36510	ENC_DIFF_TOL	Toleranz Messsystem-Gleichlauf
36600	BRAKE_MODE_CHOICE	Bremsverhalten bei Hardwareendschalter
36610	AX_EMERGENCY_STOP_TIME	Maximale Zeitdauer der Bremsrampe bei Fehlern
36620	SERVO_DISABLE_DELAY_TIME	Abschaltverzögerung Reglerfreigabe

## Schutzbereiche

Keine

## 2.6.2 Settingdaten

### 2.6.2.1 Achs-/Spindel-spezifische Setindaten

## Achsüberwachungen

Nummer	Bezeichner: \$SA_	Beschreibung
43400	WORKAREA_PLUS_ENABLE	Arbeitsfeldbegrenzung in positiver Richtung aktiv
43410	WORKAREA_MINUS_ENABLE	Arbeitsfeldbegrenzung in negativer Richtung aktiv
43420	WORKAREA_LIMIT_PLUS	Arbeitsfeldbegrenzung plus
43430	WORKAREA_LIMIT_MINUS	Arbeitsfeldbegrenzung minus

## Schutzbereiche

Keine

## 2.6.3 Signale

### 2.6.3.1 Signale an Kanal

#### Achsüberwachungen

Keine

#### Schutzbereiche

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Schutzbereiche freigeben	DB21, ... .DBX1.1	DB3200.DBX1.1
Vorschubsperr	DB21, ... .DBX6.0	DB3200.DBX6.0
maschinenbezogenen Schutzbereich 1-8 aktivieren	DB21, ... .DBX8.0-7	DB3200.DBX8.0-7
maschinenbezogenen Schutzbereich 9 aktivieren	DB21, ... .DBX9.0	DB3200.DBX9.0
maschinenbezogenen Schutzbereich 10 aktivieren	DB21, ... .DBX9.1	DB3200.DBX9.1
kanalspezifischen Schutzbereich 1-8 aktivieren	DB21, ... .DBX10.0-7	DB3200.DBX10.0-7
kanalspezifischen Schutzbereich 9 aktivieren	DB21, ... .DBX11.0	DB3200.DBX11.0
kanalspezifischen Schutzbereich 10 aktivieren	DB21, ... .DBX11.1	DB3200.DBX11.1

### 2.6.3.2 Signale von Kanal

#### Achsüberwachungen

Keine

#### Schutzbereiche

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
maschinenbezogener Schutzbereich 1-8 voraktiviert	DB21, ... .DBX272.0-7	DB3300.DBX8.0-7
maschinenbezogener Schutzbereich 9 voraktiviert	DB21, ... .DBX273.0	DB3300.DBX9.0
maschinenbezogener Schutzbereich 10 voraktiviert	DB21, ... .DBX273.1	DB3300.DBX9.1
kanalspezifischer Schutzbereich 1-8 voraktiviert	DB21, ... .DBX274.0-7	DB3300.DBX10.0-7
kanalspezifischer Schutzbereich 9 voraktiviert	DB21, ... .DBX275.0	DB3300.DBX11.0
kanalspezifischer Schutzbereich 10 voraktiviert	DB21, ... .DBX275.1	DB3300.DBX11.1
maschinenbezogener Schutzbereich 1-8 verletzt	DB21, ... .DBX276.0-7	DB3300.DBX12.0-7
maschinenbezogener Schutzbereich 9 verletzt	DB21, ... .DBX277.0	DB3300.DBX13.0
maschinenbezogener Schutzbereich 10 verletzt	DB21, ... .DBX277.1	DB3300.DBX13.1
kanalspezifischer Schutzbereich 1-8 verletzt	DB21, ... .DBX278.0-7	DB3300.DBX14.0-7
kanalspezifischer Schutzbereich 9 verletzt	DB21, ... .DBX279.0	DB3300.DBX15.0
kanalspezifischer Schutzbereich 10 verletzt	DB21, ... .DBX279.1	DB3300.DBX15.1

### 2.6.3.3 Signale an Achse/Spindel

#### Achsüberwachungen

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Nachführbetrieb	DB31, ... .DBX1.4	DB380x.DBX1.4
Lagemesssystem 1 / 2	DB31, ... .DBX1.5/6	DB380x.DBX1.5/6
Reglerfreigabe	DB31, ... .DBX2.1	DB380x.DBX2.1
Klemmvorgang läuft	DB31, ... .DBX2.3	DB380x.DBX2.3
Geschwindigkeits-/Spindeldrehzahlbegrenzung	DB31, ... .DBX3.6	DB380x.DBX3.6
Vorschub-Halt / Spindel-Halt	DB31, ... .DBX4.3	DB380x.DBX4.3
Hardwareendschalter minus / Hardwareendschalter plus	DB31, ... .DBX12.0/1	DB380x.DBX1000.0/1
Softwareendschalter minus / 2. Softwareendschalter plus	DB31, ... .DBX12.2/3	DB380x.DBX1000.2/3
Gebergrenzfrequenz überschritten 1 / 2	DB31, ... .DBX60.2/3	DB390x.DBX0.2
Referiert/Synchronisiert 1 / 2	DB31, ... .DBX60.4/5	DB390x.DBX0.4/5
Fahrbefehl minus / plus	DB31, ... .DBX64.6/7	DB390x.DBX4.6/7

#### Schutzbereiche

Keine



## B1: Bahnsteuerbetrieb, Genauhalt, LookAhead

### 3.1 Kurzbeschreibung

#### Genauhalt oder Genauhaltbetrieb

Genauhaltbetrieb ist ein Verfahrensmodus, bei dem am Ende eines jeden Verfahrstrahles alle an der Verfahrbewegung beteiligten Achsen (außer Achsen von strahlübergreifenden Verfahrbewegungen) bis zum Stillstand abgebremst werden. Der Strahlwechsel zum nachfolgenden Verfahrstrahl erfolgt erst, wenn alle an der Verfahrbewegung beteiligten Achsen ihre programmierte Zielposition in Abhängigkeit des gewählten Genauhaltkriteriums erreicht haben.

#### Bahnsteuerbetrieb

Bahnsteuerbetrieb ist ein Verfahrensmodus, bei dem die NC versucht, die programmierte Bahngeschwindigkeit möglichst konstant beizubehalten. Insbesondere soll dabei das Abbremsen der Bahnachsen an den Strahlgrenzen des Teilerfahrens vermieden werden.

#### LookAhead

LookAhead ist eine Funktion zur Optimierung des Bahnsteuerbetriebs.

Um an Werkstücken hohe Oberflächengüten zu erzielen, ist eine gleichmäßige Bearbeitung notwendig. Deshalb sollte es während der Bearbeitung möglichst nicht zu Schwankungen der Bahngeschwindigkeit kommen. Ohne LookAhead betrachtet die NC zur Ermittlung der möglichen Bahngeschwindigkeit nur den unmittelbar auf den aktuellen Verfahrstrahl folgenden Verfahrstrahl. Beinhaltet der folgende Verfahrstrahl nur einen kurzen Verfahrweg, muss die NC die Bahngeschwindigkeit vermindern (Bremsen im aktuellen Verfahrstrahl), um eventuell rechtzeitig am Strahlende des Folgestrahles anhalten zu können.

Durch "Vorausschauen" über eine parametrierbare Anzahl von dem aktuellen Verfahrstrahl folgenden Verfahrstrahlen kann mit LookAhead eine unter Umständen wesentlich höhere Bahngeschwindigkeit erzielt werden, da der NC jetzt wesentlich mehr Verfahrstrahlen bzw. Verfahrweg zur Berechnung zur Verfügung steht.

Dadurch ergeben sich folgende Vorteile:

- Bearbeitung mit durchschnittlich höherer Bahngeschwindigkeit
- Verbesserung der Oberflächengüte durch Vermeidung von Brems- und Beschleunigungsvorgängen

### 3.1 Kurzbeschreibung

#### Glättung der Bahngeschwindigkeit

"Glättung der Bahngeschwindigkeit" ist eine Funktion speziell für Anwendungen, die eine möglichst gleichmäßige Bahngeschwindigkeit erfordern (z. B. Hochgeschwindigkeitsfräsen im Formenbau). Dazu wird bei der Glättung der Bahngeschwindigkeit auf Brems- und Beschleunigungsvorgänge verzichtet, die zu hochfrequenten Anregungen von Maschinenresonanzen führen würden.

Dadurch ergeben sich folgende Vorteile:

- Verbesserung der Oberflächengüte und der Stückzeit durch Vermeidung von Anregungen von Maschinenresonanzen
- Konstanter Verlauf der Bahngeschwindigkeit bzw. Schnittgeschwindigkeiten durch Vermeidung von "überflüssigen" Beschleunigungsvorgängen, d. h. Beschleunigungsvorgänge, die keinen großen Gewinn hinsichtlich der Programmlaufzeit bewirken.

#### Anpassung der Bahndynamik

Die "Anpassung der Bahndynamik" ist neben der "Glättung der Bahngeschwindigkeit" eine weitere Funktion zur Vermeidung hochfrequenter Anregungen von Maschinenresonanzen bei gleichzeitiger Optimierung der Bahndynamik. Dazu werden hochfrequente Änderungen der Bahngeschwindigkeit automatisch mit geringeren Ruck- oder Beschleunigungswerten als den in den Maschinendaten parametrisierten Dynamikgrenzwerten durchgeführt.

Somit wirken bei niederfrequenten Änderungen der Bahngeschwindigkeit die vollen Dynamikgrenzwerte, während bei hochfrequenten Änderungen durch die automatische Dynamikanpassung nur die reduzierten Dynamikgrenzwerte wirken.

#### Dynamikmodus für Bahninterpolation

Zur Optimierung der Bahndynamik zählen auch die Technologie-spezifischen Dynamikeinstellungen, die für unterschiedliche Bearbeitungstechnologien (u. a. Gewindebohren, Schruppen, Schlichten) voreingestellt sind und im Teileprogramm durch Aufruf des jeweiligen Dynamikmodus aktiviert werden können.

#### Freiformflächenmodus

Jegliche Schwankung der Krümmung bzw. Torsion führt zu einer Veränderung der Bahngeschwindigkeit. Dadurch erhält man in der Regel bei der Bearbeitung von Freiformflächen-Werkstücken unnötige Brems- und Beschleunigungsvorgänge, die sich negativ auf die Qualität der Werkstückoberflächen auswirken können.



Für die Freiformflächenbearbeitung stehen daher folgende Funktionen zur Verfügung:

- Funktion "Freiformflächenmodus: Grundfunktionen"

Damit wird die Festlegung des Verlaufs der Bahngeschwindigkeit "unempfindlicher" gegenüber Schwankungen der Krümmung und Torsion.

- Funktion "Freiformflächenmodus: Erweiterungsfunktion"

Diese Erweiterung der LookAhead-Standardfunktionalität dient zur Berechnung des Bahngeschwindigkeitsprofils bei der Freiformflächenbearbeitung.

Die Vorteile des Freiformflächenmodus liegen in einer gleichmäßigeren Werkstückoberfläche und einer geringeren Belastung der Maschine.

## NC-Satz-Kompression

Nach Abschluss der Konstruktion eines Werkstückes mit einem CAD/CAM-System übernimmt dieses gewöhnlich auch die Generierung des entsprechenden Teileprogramms zur Erzeugung der Werkstückoberfläche. Dabei verwenden die meisten CAD/CAM-Systeme zur Beschreibung auch gekrümmter Abschnitte der Werkstückoberfläche Linearsätze. Zur Einhaltung der erforderlichen Konturgenauigkeit sind dabei gewöhnlich sehr viele Stützstellen notwendig. Demzufolge ergeben sich viele lineare Verfahrssätze mit meist sehr kurzen Verfahrwegen.

Durch die Funktion "NC-Satz-Kompression" erfolgt eine nachträgliche Annäherung an die durch die Linearsätze vorgegebene Kontur durch Polynom-Sätze. Dabei wird eine parametrierbare Anzahl von Linearsätzen durch einen Polynom-Satz ersetzt. Die Anzahl von Linearsätzen, die durch einen Polynom-Satz ersetzt werden können, ist darüber hinaus noch abhängig von der vorgegebenen maximal erlaubten Konturabweichung und dem Konturverlauf.

Die Vorteile, die sich durch die Verwendung von Polynom-Sätzen ergeben, sind:

- Reduzierung der Anzahl von benötigten Teileprogrammsätzen zur Beschreibung der Werkstückkontur
- Erhöhung der maximal möglichen Bahngeschwindigkeiten

## Zusammenfassung kurzer Spline-Sätze

Ein Spline definiert eine Kurve, die aus Polynomen 2. oder 3. Grades zusammengesetzt wird. Mit der Spline-Interpolation kann die Steuerung aus nur wenigen vorgegebenen Stützpunkten einer Sollkontur einen glatten Kurvenverlauf erzeugen.

### Literatur:

Programmierhandbuch Arbeitsvorbereitung; Spezielle Wegbefehle,  
Kapitel: Spline-Interpolation

Die Vorteile der Spline-Interpolation gegenüber der Linear-Interpolation sind:

- Reduzierung der Anzahl von benötigten Teileprogrammsätzen zur Beschreibung einer gekrümmten Kontur.
- Weicher, Mechanik-schonender Kurvenverlauf auch beim Übergang zwischen den Teileprogrammsätzen.

### 3.1 Kurzbeschreibung

Die Nachteile der Spline-Interpolation gegenüber der Linear-Interpolation sind:

- Für eine Spline-Kurve kann kein exakter Kurvenverlauf, sondern nur ein Toleranzband vorgegeben werden, innerhalb dessen die Spline-Kurve liegen soll.

Wie bei der Linear-Interpolation können auch bei der Aufbereitung von Splines so kurze Sätze entstehen, dass zur Interpolation dieser Spline-Sätze die Bahngeschwindigkeit reduziert werden muss. Dies ist auch dann der Fall, wenn der Spline eigentlich eine lange, glatte Kurve darstellt. Mit der Funktion "Zusammenfassung kurzer Spline-Sätze" können die Spline-Sätze so zusammengefasst werden, dass die resultierende Satzlänge ausreichend groß ist und nicht zu einer Verringerung der Bahngeschwindigkeit führt.

---

#### **Hinweis**

##### **NC-Satz-Kompressor**

Der NC-Satz-Kompressor (COMPON, COMPCURV oder COMPCAD) ist bei der Komprimierung von Spline-Sätzen nicht anwendbar, da mit diesem nur Linear-Sätze komprimiert werden können.

---

## 3.2 Genauhaltbetrieb

### Genauhalt oder Genauhaltbetrieb

Genauhalt oder Genauhaltbetrieb ist ein Verfahrmodus, bei dem am Ende eines jeden Verfahrsatzes alle an der Verfahrbewegung beteiligten Bahnachsen und Zusatzachsen, die nicht satzübergreifend verfahren, zum Satzende bis zum Stillstand abgebremst werden. Der Satzwechsel zum nachfolgenden Verfahrersatz erfolgt erst, wenn alle an der Verfahrbewegung beteiligten Achsen ihre programmierte Zielposition in Abhängigkeit des gewählten Genauhaltkriteriums erreicht haben.

Daraus ergibt sich folgendes Verhalten:

- Durch das Abbremsen der Achsen und die Wartezeit bis zum Erreichen des Zustandes "Genauhalt" für alle beteiligten Maschinenachsen verlängert sich die Programmlaufzeit gegenüber dem Bahnsteuerbetrieb erheblich.
- Bei Genauhaltbetrieb während der Bearbeitung kann es zu Freischneidmarken auf der Werkstückoberfläche kommen.

### Zustand "Genauhalt"

Als Genauhalt wird auch ein Zustand der Maschinenachse bezogen auf die Positionsdifferenz bezüglich ihrer Sollposition am Ende einer Verfahrbewegung bezeichnet. Die Maschinenachse erreicht den Zustand "Genauhalt", sobald ihr Schleppabstand kleiner der vorgegebenen Positionsdifferenz (Genauhaltkriterium) wird.

### Anwendung

Der Genauhaltbetrieb ist immer dann zu verwenden, wenn die programmierte Kontur exakt abgefahren werden muss.

### Aktivierung

Der Genauhaltbetrieb kann im Teileprogramm modal oder satzweise aktiviert werden:

G-Befehl	Bedeutung
G60	<b>modal</b> wirksamer Genauhalt
G9	<b>satzweise</b> wirksamer Genauhalt

### Genauhaltkriterien "Genauhalt grob" und "Genauhalt fein"

Über die Genauhaltkriterien "Genauhalt grob" und "Genauhalt fein" werden Toleranzfenster vorgegeben, die für das Erreichen des Zustandes "Genauhalt" einer Maschinenachse beachtet werden sollen:

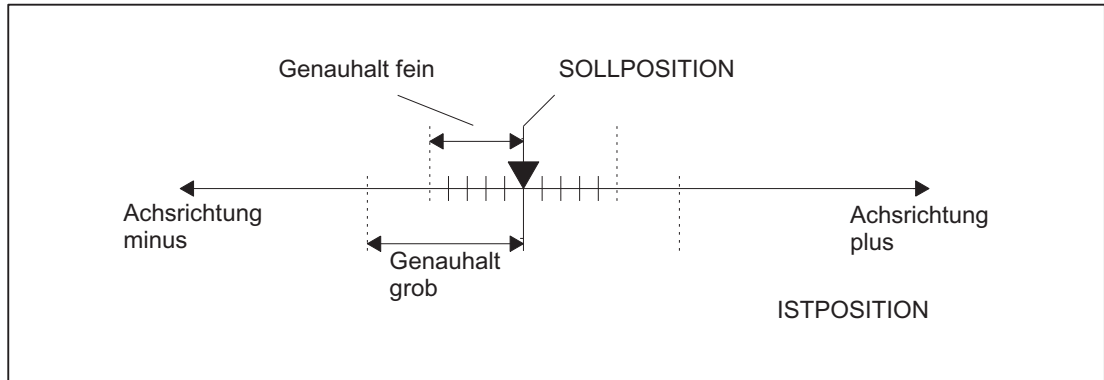


Bild 3-1 Toleranzfenster der Genauhaltkriterien

Die Parametrierung der beiden Genauhaltkriterien erfolgt über die Maschinendaten:

MD36010 \$MA\_STOP\_LIMIT\_FINE (Genauhalt fein)

MD36000 \$MA\_STOP\_LIMIT\_COARSE (Genauhalt grob)

#### Hinweis

Die Toleranzfenster der Genauhaltkriterien "Genauhalt grob" und "Genauhalt fein" sollten so parametrieren, dass folgende Forderung erfüllt ist:

"Genauhalt grob" > "Genauhalt fein"

### Genauhaltkriterium "Interpolator-Ende"

Bei Genauhaltkriterium "Interpolator-Ende" erfolgt der Satzwechsel zum nachfolgenden Verfahrersatz, sobald alle an der Verfahrbewegung beteiligten Bahnachsen und Zusatzachsen, die nicht satzübergreifend verfahren, sollwertbezogen ihre im Verfahrersatz programmierte Position erreicht haben. D. h. der Interpolator hat den Verfahrersatz abgefahren.

Die Istposition bzw. der Schleppabstand der beteiligten Maschinenachsen wird beim Genauhaltkriterium "Interpolator-Ende" nicht betrachtet. Dadurch kann es, abhängig von der Dynamik der Maschinenachsen, zu einem im Vergleich zu den Genauhaltkriterien "Genauhalt grob" und "Genauhalt fein" größeren Verschleifen der Kontur an den Satzübergängen kommen.

## Aktivierung eines Genauhaltkriteriums

Ein Genauhaltkriterium wird im Teileprogramm durch Programmierung des entsprechenden G-Befehls aktiviert:

G-Befehl	Genauhaltkriterium
G601	Genauhalt fein
G602	Genauhalt grob
G603	Interpolator-Ende

## Satzwechsel in Abhängigkeit der Genauhaltkriterien

Das nachfolgende Bild veranschaulicht den Zeitpunkt des Satzwechsels in Abhängigkeit vom gewählten Genauhaltkriterium.

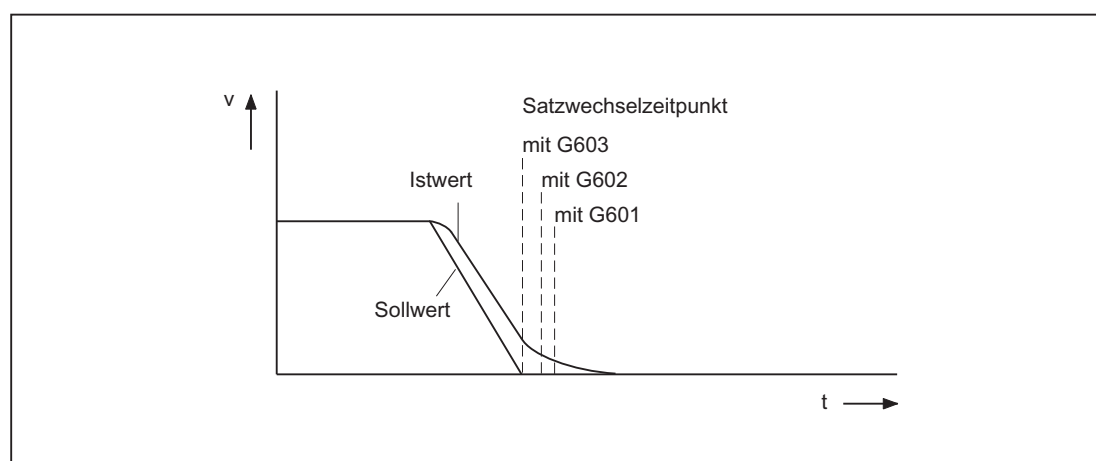


Bild 3-2 Satzwechsel in Abhängigkeit vom gewählten Genauhaltkriterium

## Bewertungsfaktor für Genauhaltkriterien

Eine Parametersatz-abhängige Bewertung der Genauhaltkriterien kann über das folgende achsspezifische Maschinendatum vorgegeben werden:

MD36012 \$MA\_STOP\_LIMIT\_FACTOR (Faktor Genauhalt grob / fein und Stillstand)

### Anwendungen:

- Anpassen des Positionierverhaltens an veränderte Massenverhältnisse, z. B. nach Getriebeumschaltung.
- Verkürzen der Positionierzeit in Abhängigkeit von verschiedenen Bearbeitungszuständen, z. B. Schruppen, Schlichten.

### Parametrierbare Vorgabe des wirksamen Genauhaltkriteriums

Das wirksame Genauhaltkriterium kann für die Teileprogrammbefehle der 1. G-Funktionsgruppe unabhängig vom im Teileprogramm programmierten Genauhaltkriterium fest vorgegeben werden. Die Vorgabe kann unabhängig voneinander für folgende Teileprogrammbefehle erfolgen:

- Eilgang: G0
- Bearbeitungsbefehle: G1, G2, G3, CIP, ASPLINE, BSPLINE, CSPLINE, POLY, G33, G34, G35, G331, G332, OEMIPO01, OEMIPO02, CT

Die Einstellung erfolgt kanalspezifisch über das Maschinendatum:

MD20550 \$MC\_EXACT\_POS\_MODE (Genauhalt-Bedingungen bei G0 und G1)

#### Codierung

Die Vorgabe des jeweiligen Genauhaltkriteriums erfolgt platzkodiert:

MD20550 \$MC\_EXACT\_POS\_MODE = <ZE>

- Einerstelle E: Eilgang
- Zehnerstelle Z: alle anderen Teileprogrammbefehle der 1. G-Funktionsgruppe

Z bzw. E	Wirksames Genauhaltkriterium
0	Programmiertes Genauhaltkriterium
1	G601 (Genauhaltfenster fein)
2	G602 (Genauhaltfenster grob)
3	G603 (Interpolator-Ende)

#### Beispiel:

MD20550 \$MC\_EXACT\_POS\_MODE = 02

Einerstelle = 2:

Beim Verfahren mit Eilgang wirkt unabhängig von jeglicher Programmierung im Teileprogramm immer Genauhaltkriterium G602 (Genauhaltfenster grob).

Zehnerstelle = 0:

Beim Verfahren mit allen anderen Teileprogrammbefehlen der 1. G-Funktionsgruppe wirkt das im Teileprogramm programmierte Genauhaltkriterium.

### Parametrierbares Genauhaltkriterium bei Eilgangübergängen im Bahnsteuerbetrieb

Das Verhalten am Satzübergang von Teileprogrammsätzen vor und nach Eilgangsätzen lässt sich wie folgt parametrieren:

MD20552 \$MC\_EXACT\_POS\_MODE\_G0\_TO\_G1 = <Wert>

Wert	Bedeutung
0	Kein zusätzlicher Stopp am Satzübergang.
1	Stopp am Satzübergang: Verhalten wie bei G601 (Genauhaltfenster fein)
2	Stopp am Satzübergang: Verhalten wie bei G602 (Genauhaltfenster grob)
3	Stopp am Satzübergang: Verhalten wie bei G603 (Interpolator-Ende)
4	Wie 0, zusätzlich wird beim Wechsel von G0 nach Nicht-G0 vorausschauend im G0-Satz der Override des nachfolgenden Nicht-G0-Satzes berücksichtigt.
5	Wie 0, zusätzlich wird beim Wechsel von G0 nach Nicht-G0 und Nicht-G0 nach G0 vorausschauend der Override des nachfolgenden Satzes berücksichtigt.

## 3.3 Bahnsteuerbetrieb

### 3.3.1 Allgemeine Funktionalität

#### Bahnsteuerbetrieb

Im Bahnsteuerbetrieb wird die Bahngeschwindigkeit am Satzende zum Satzwechsel **nicht** auf eine Geschwindigkeit abgebremst, die ein Erreichen des Genauhaltkriteriums ermöglicht. Ziel ist dagegen, ein größeres Abbremsen der Bahnachsen am Satzwechsellpunkt zu vermeiden, um mit möglichst gleicher Bahngeschwindigkeit in den nächsten Satz zu wechseln. Um dieses Ziel zu erreichen, wird mit Anwahl des Bahnsteuerbetriebs zusätzlich die Funktion "LookAhead" aktiviert.

Bahnsteuerbetrieb bewirkt, dass knickförmige Satzübergänge durch lokale Änderungen des programmierten Verlaufs tangential gestaltet bzw. geglättet werden. Das Ausmaß der Änderung relativ zum programmierten Verlauf kann durch die Kriterien des Überlastfaktors oder Überschleifens eingegrenzt werden.

Bahnsteuerbetrieb bewirkt:

- eine Verrundung der Kontur.
- kürzere Bearbeitungszeiten durch fehlende Brems- und Beschleunigungsvorgänge, die für das Erreichen des Genauhaltkriteriums benötigt werden.
- bessere Schnittbedingungen durch den gleichmäßigeren Geschwindigkeitsverlauf.

Bahnsteuerbetrieb ist sinnvoll, wenn:

- eine Kontur möglichst zügig abgefahren werden soll (z. B. mit Eilgang).
- der exakte Verlauf im Rahmen eines Fehlerkriteriums vom programmierten abweichen darf, um einen durchgehend stetigen Verlauf zu erzeugen.

Bahnsteuerbetrieb ist nicht sinnvoll, wenn:

- eine Kontur exakt abgefahren werden soll.
- absolute Geschwindigkeitskonstanz erforderlich ist.



## Impliziter Genauhalt

In einigen Fällen muss im Bahnsteuerbetrieb ein Genauhalt erzeugt werden, um Folgeaktionen ausführen zu können. In diesen Situationen wird die Bahngeschwindigkeit auf Null abgebremst.

- Werden Hilfsfunktionen vor der Verfahrbewegung ausgegeben, so wird der vorhergehende Satz erst mit dem Erreichen des angewählten Genauhaltkriteriums beendet.
- Werden Hilfsfunktionen nach der Verfahrbewegung ausgegeben, so werden diese nach Interpolator-Ende des Satzes ausgegeben.
- Enthält ein ausführbarer Satz (z. B. Starten einer Positionierachse) keine Verfahrinformation für die Bahnachsen, so wird der vorhergehende Satz mit dem Erreichen des angewählten Genauhaltkriteriums beendet.
- Wird eine Positionierachse zur Geometrieachse deklariert, so wird mit Programmierung der Geometrieachse der vorhergehende Satz mit Interpolator-Ende abgeschlossen.
- Wird eine Synchronachse programmiert, die zuletzt als Positionierachse oder als Spindel programmiert war (Grundstellung der Zusatzachse ist Positionierachse), so wird der vorhergehende Satz mit Interpolator-Ende beendet.
- Wird die Transformation gewechselt, so wird der zuvor bearbeitete Satz mit dem aktiven Genauhaltkriterium abgeschlossen.
- Ein Satz wird mit Interpolator-Ende beendet, falls im Folgesatz die Umschaltung des Beschleunigungsprofils BRISK/SOFT erfolgt (siehe Kapitel "B2: Beschleunigung (Seite 245)").
- Wird im Teileprogramm die Funktion "Zwischenspeicher leeren" programmiert, so wird der vorhergehende Satz mit dem Erreichen des angewählten Genauhaltkriteriums beendet.

## Geschwindigkeit = 0 im Bahnsteuerbetrieb

Unabhängig vom impliziten Genauhalt wird die Bahnbewegung am Satzende auf Geschwindigkeit Null abgebremst, wenn:

- Positionierachsen mit der Anweisung `POS` programmiert sind und deren Verfahrzeit länger als die Verfahrzeit der Bahnachsen ist. Der Satzwechsel erfolgt mit Erreichen des "Genauhalt fein" der Positionierachse.
- die Zeit für die Positionierung einer Spindel, die mit der Anweisung `SPOS` programmiert wurde, länger dauert als die Verfahrzeit der Bahnachsen. Der Satzwechsel erfolgt mit Erreichen des "Genauhalt fein" der positionierenden Spindel.
- im aktuellen Satz Geometrieachsen verfahren und im folgenden Satz stattdessen Synchronachsen verfahren, bzw. im aktuellen Satz Synchronachsen verfahren und im nächsten Satz stattdessen Geometrieachsen verfahren werden.
- eine Synchronisation erforderlich wird.

### 3.3.2 Geschwindigkeitsabsenkung gemäß Überlastfaktor

#### Funktion

Die Funktion senkt im Bahnsteuerbetrieb die Bahngeschwindigkeit soweit ab, dass unter Wahrung der Beschleunigungsgrenze und unter Berücksichtigung eines Überlastfaktors der nichttangente Satzübergang in einem Interpolatortakt überfahren werden kann.

Mit der Geschwindigkeitsabsenkung werden bei nichttangente Konturverlauf am Satzübergang axiale Geschwindigkeitssprünge erzeugt. Diese werden auch von den mitfahrenden Synchronachsen nachvollzogen. Mit dem Geschwindigkeitssprung wird vermieden, dass die Bahngeschwindigkeit zu null abgesenkt wird. Der Sprung wird dann ausgeführt, wenn die axiale Geschwindigkeit mit der Achsbeschleunigung auf eine Geschwindigkeit reduziert wurde, ab der dann mit dem Sprung auf den neuen Sollwert gelangt werden kann. Die Sprunghöhe des Sollwerts kann mittels des Kriteriums Überlastfaktor begrenzt werden. Da die Sprunghöhe achsbezogen ist, wird am Satzübergang die kleinste Sprunghöhe der beim Satzwechsel aktiven Bahnachsen berücksichtigt.

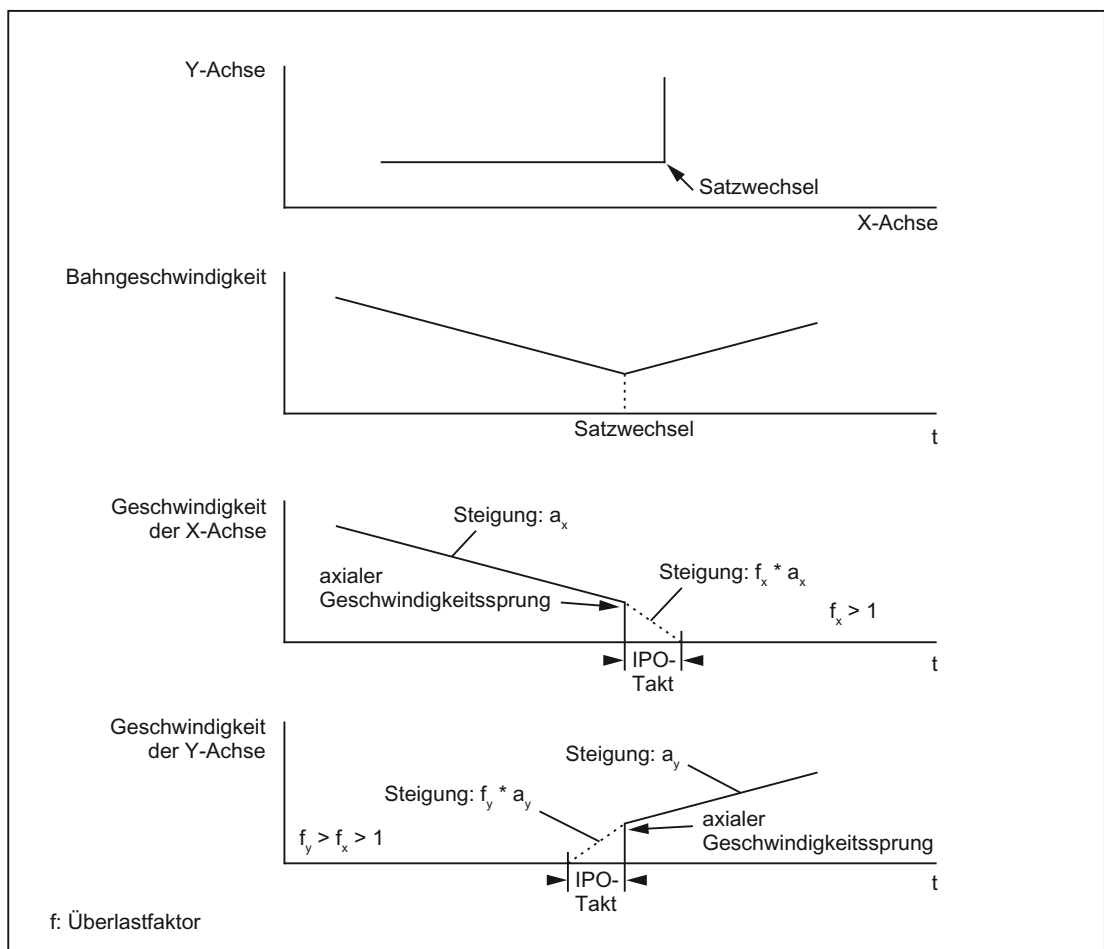


Bild 3-3 Axiale Geschwindigkeitsänderung am Satzübergang

Bei nahezu tangenalem Satzübergang wird die Bahngeschwindigkeit nicht abgesenkt, falls die zulässigen Achsbeschleunigungen nicht überschritten werden. Damit wird erreicht, dass sehr kleine Knicke in der Kontur (z. B. 0,5°) direkt überfahren werden.

## Überlastfaktor

Der Überlastfaktor begrenzt den Geschwindigkeitssprung der Maschinenachse am Satzübergang. Damit der Geschwindigkeitssprung die Achsbelastbarkeit nicht überschreitet, wird der Sprung aus der Beschleunigung der Achse abgeleitet.

Der Überlastfaktor gibt an, um welches Maß die Beschleunigung der Maschinenachse (MD32300 \$MA\_MAX\_AX\_ACCEL) für einen IPO-Takt überschritten werden darf.

Der Geschwindigkeitshub ergibt sich wie folgt:

Geschwindigkeitshub = Achsbeschleunigung \* (Überlastfaktor - 1) \* Interpolator-Takt

Der Überlastfaktor ist hinterlegt im Maschinendatum:

MD32310 \$MA\_MAX\_ACCEL\_OVL\_FACTOR (Überlastfaktor für axiale Geschwindigkeitssprünge)

Faktor 1.0 bedeutet, dass nur tangenale Übergänge mit endlicher Geschwindigkeit überfahren werden können. Bei allen anderen Übergängen wird sollwertseitig auf Geschwindigkeit 0 abgebremst. Das Verhalten entspricht der Funktionalität "Genauhalt mit Interpolator-Ende". Da dies für den Bahnsteuerbetrieb nicht erwünscht ist, ist der Faktor größer 1.0 einzustellen.

---

### Hinweis

Für die Inbetriebnahme ist zu beachten, dass der Faktor dann herabzusetzen ist, wenn die Maschine bei knickförmigem Satzübergang zu Schwingungen angeregt wird und Überschleifen nicht verwendet werden soll.

---

Durch Setzen des folgenden Maschinendatums werden Satzübergänge unabhängig vom eingestelltem Überlastfaktor mit G641 / G642 überschleifen:

MD20490 \$MC\_IGNORE\_OVL\_FACTOR\_FOR\_ADIS

## Aktivierung / Deaktivierung

Der Bahnsteuerbetrieb mit Geschwindigkeitsabsenkung gemäß Überlastfaktor kann in jedem NC-Teileprogramm durch den modal wirksamen Befehl G64 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb G64 kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt G60
- Überschleifen G641, G642, G643, G644 oder G645

### Impliziter Bahnsteuerbetrieb

Können im Bahnsteuerbetrieb mit Überschleifen G641 wegen sehr kurzer Weglängen der Sätze (z. B. nulltaktige Sätze) keine Überschleifsätze eingefügt werden, wird auf Bahnsteuerbetrieb G64 umgeschaltet.

### 3.3.3 Überschleifen

#### Funktion

Überschleifen bedeutet, dass ein knickförmiger Satzübergang durch eine lokale Änderung des programmierten Verlaufs in einen tangentialen Satzübergang geändert wird. Der Bereich nahe dem ursprünglichen knickförmigen Satzübergang (auch die Übergänge der von der CNC eingefügten Zwischensätze) erhält dadurch einen stetigen Verlauf.

Beim Überschleifen werden nicht nur die Geometrieachsen berücksichtigt, sondern alle miteinander synchron verfahrenen Maschinenachsen. Überschleifen glättet somit auch den Verlauf von Orientierungsachsen, sowie allgemein Geschwindigkeitssprünge bei Synchronachsen.

---

#### Hinweis

Überschleifen kann und soll die Funktionen für definiertes Glätten RND, RNDM, ASPLINE, BSPLINE, CSPLINE nicht ersetzen.

Wird eine durch G641, G642, G643, G644 oder G645 erzeugte Überschleifbewegung unterbrochen, wird beim nachfolgenden Repositionieren nicht der Unterbrechungspunkt angefahren, sondern der Eckpunkt der Originalkontur.

---

#### Synchronisation

Beim Überschleifen werden die unstetig angrenzenden Sätze verkürzt und dafür ein oder zwei Zwischensätze eingefügt. Die ursprüngliche Satzgrenze ist damit aufgehoben und für etwaige Synchronisierungsbedingungen (z. B. Hilfsfunktionsausgabe parallel zur Bewegung, Stopp am Satzende) nicht mehr verfügbar.

Zweckmäßigerweise werden beim Überschleifen alle Synchronisierungsbedingungen auf das Satzende des verkürzten ersten Satzes bezogen und nicht etwa auf das Ende des Überschleifzwischensatzes. Der Folgesatz ist damit nicht begonnen worden und bei einem Stopp auf Satzende könnte die Kontur des Folgesatzes noch geändert werden.

#### Ausführung

Überschleifen wird nur dann ausgeführt, wenn der Satzübergang mit endlicher Geschwindigkeit überfahren werden soll. Die maximale Bahngeschwindigkeit wird durch die Krümmung der Kurve mitbestimmt. Dabei werden die maximalen Beschleunigungswerte der Achsen nicht überschritten. Ein Satz ohne Verfahrinformation für die Bahnachsen erfordert die Geschwindigkeit "Null" und somit kein Überschleifen.

Überschleifen wird ferner dann eingesetzt, wenn das Überfahren des Satzübergangs eine Geschwindigkeit erfordert, die die zulässige Geschwindigkeit am Satzübergang gemäß dem Verfahren bei G64 (siehe Kapitel "Geschwindigkeitsabsenkung gemäß Überlastfaktor (Seite 178)" Absatz "Überlastfaktor") unterschreitet.

### Keine Überschleifzwischensätze

In den folgenden Fällen wird kein Überschleifzwischensatz eingefügt:

- Zwischen beiden Sätzen wird angehalten.

Dies tritt auf, wenn:

- eine Hilfsfunktionsausgabe vor Bewegung im Folgesatz steht.
- der Folgesatz keine Bahnbewegung enthält.
- für den Folgesatz zum ersten Mal eine Achse als Bahnachse verfährt, die zuvor Positionierachse war.
- für den Folgesatz zum ersten Mal eine Achse als Positionierachse verfährt, die zuvor Bahnachse war.
- der Vorgängersatz Geometrieachsen verfährt und der Folgesatz nicht.
- der Folgesatz Geometrieachsen verfährt und der Vorgängersatz nicht.
- vor Gewindeschneiden der Folgesatz  $G_{33}$  als Wegbedingung hat und der Vorgängersatz nicht.
- zwischen BRISK und SOFT gewechselt wird.
- transformationsbedeutsame Achsen nicht vollständig der Bahnbewegung zugeordnet sind (z. B. bei Pendeln, Positionierachsen).

- Der Überschleifssatz würde die Teileprogrammbearbeitung verlangsamen.

Dies tritt auf:

- zwischen sehr kurzen Sätzen.

Da jeder Satz mindestens einen Interpolationstakt benötigt, würde der eingefügte Zwischensatz die Bearbeitungszeit verdoppeln.

- wenn ein Satzübergang mit G64 (Bahnsteuerbetrieb ohne Überschleifen) ohne Geschwindigkeitsreduzierung überfahren werden darf.

Überschleifen würde die Bearbeitungszeit erhöhen. D. h. der Wert des erlaubten Überlastfaktors (MD32310 \$MA\_MAX\_ACCEL\_OVL\_FACTOR) hat Einfluss darauf, ob ein Satzübergang überschleift wird oder nicht. Der Überlastfaktor wird nur beim Überschleifen mit G641/G642 berücksichtigt. Beim Überschleifen mit G643 hat der Überlastfaktor keinen Einfluss (dieses Verhalten kann auch für G641 und G642 eingestellt werden, indem MD20490 \$MC\_IGNORE\_OVL\_FACTOR\_FOR\_ADIS = TRUE gesetzt wird).

3.3 Bahnsteuerbetrieb

- Das Überschleifen ist nicht parametrierbar.
  - Dies tritt auf, wenn:
    - bei G641 in G0-Sätzen ADISPOS = 0 ist (Vorbelegung!).
    - bei G641 in Nicht-G0-Sätzen ADIS = 0 ist (Vorbelegung!).
    - bei G641 beim Übergang zwischen G0 und Nicht-G0 bzw. Nicht-G0 und G0 der kleinere Wert aus ADISPOS und ADIS gilt.
    - bei G642/G643 alle achsspezifischen Toleranzen gleich Null sind.
- Der Satz enthält keine Verfahrbewegung (Nullsatz).
  - Dies tritt auf, wenn:
    - Synchronaktionen aktiv sind.
 

Normalerweise werden Nullsätze vom Interpreter eliminiert. Wenn aber Synchronaktionen aktiv sind, wird dieser Nullsatz eingekettet und ausgeführt. Hierbei wird ein Genauhalt entsprechend aktiver Programmierung ausgelöst. Damit soll die Synchronaktion die Möglichkeit bekommen, gegebenenfalls zu schalten.
    - durch Programmsprünge Nullsätze erzeugt werden.

**Synchronachsen**

Sind mehrere Verläufe synchron zu führen (z. B. Kontur, Zusatzachse), so hat jeder Verlauf grundsätzlich seinen eigenen Überschleifbereich.

Dies genau zu berücksichtigen ist nicht praktikabel. Aufgrund der besonderen Bedeutung der Kontur (Geometrieachse) wird deshalb wie folgt vorgegangen:

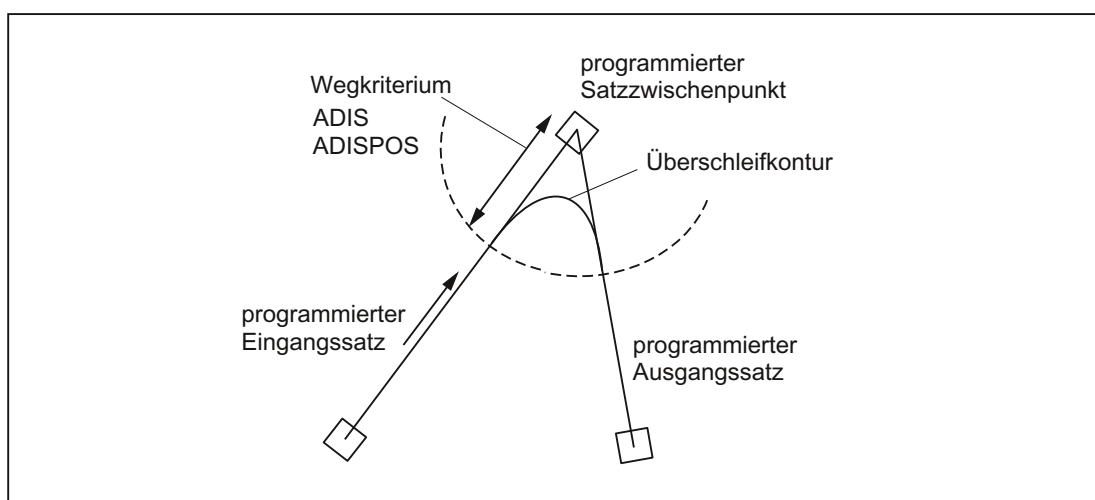
Überschleifverhalten mit synchronen Verläufen		
Ausgangsverlauf bezüglich:		Ergebnis bezüglich Überschleifbahn:
Geometrieachsen	Orientierungs-/Synchronachse	
glatt	glatt	Vorgegebener Verlauf wird exakt abgefahren.
glatt	knickförmig	Zwischensätze, die Geometrieachsen fahren exakt, alle Orientierungs-/ Synchronachsverläufe werden verschliffen.
knickförmig	glatt	Zwischensatz, die Geometrieachsen überschleifen, alle Orientierungs-/Synchronachsverläufe werden verschliffen.
knickförmig	knickförmig	Zwischensatz, die Geometrieachsen überschleifen, alle Orientierungs-/ Synchronachsverläufe werden verschliffen.

### 3.3.3.1 Überschleifen nach Wegkriterium (G641)

#### Funktion

Beim Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium wird die Größe des Verrundungsbereichs durch die Wegkriterien ADIS und ADISPOS beeinflusst.

Die Wegkriterien ADIS und ADISPOS beschreiben die Strecke, die der Überschleifensatz vor dem Satzende frühestens beginnen darf, bzw. die Strecke nach Satzende, in der der Überschleifensatz beendet sein muss.



#### Hinweis

Spitze Ecken erzeugen stark gekrümmte Überschleifkurven und haben damit auch eine entsprechende Geschwindigkeitsreduzierung zur Folge.

#### Hinweis

ADISPOS wird wie ADIS gehandhabt, ist aber speziell nur für die Bewegungsart Eilgang G0 einzusetzen.

### Wirksamkeit des Wegkriteriums

- ADIS bzw. ADISPOS müssen programmiert werden. Ist die Voreinstellung "Null" verhält sich G641 wie G64.
- Sind nicht beide aufeinanderfolgende Sätze Eilgang G0, so gilt der kleinere Überschleifabstand.
- Wird ein sehr kleiner Wert für ADIS verwendet, so ist zu beachten, dass die Steuerung sicherstellt, dass jeder interpolierte Satz - auch ein Überschleifzwischenatz - mindestens einen Interpolationspunkt enthält. Die maximale Bahngeschwindigkeit wird damit auf ADIS/Interpolationstakt begrenzt.
- Unabhängig von ADIS und ADISPOS wird der Überschleifbereich durch die Satz-Weglänge begrenzt.

Bei Sätzen mit kurzen Wegstrecken (Strecke  $< 4 \cdot \text{ADIS}$  bzw.  $4 \cdot \text{ADISPOS}$ ) wird der Überschleifabstand reduziert, damit ein abfahrbarer Teil des ursprünglichen Satzes noch erhalten bleibt. Die verbleibende Länge ist vom Achsverlauf abhängig und beträgt ca. 60% des Wegs, welcher in dem Satz noch abgefahren werden soll. ADIS bzw. ADISPOS werden somit auf die verbleibenden 40% des zu verfahrenen Wegs reduziert. Durch diesen Algorithmus wird weiterhin verhindert, dass bei einer sehr kleinen Konturänderung ein Überschleifsatz eingefügt wird. In diesem Fall wird solange auf Bahnsteuerbetrieb G64 umgeschaltet, bis Überschleifsätze wieder eingefügt werden können.

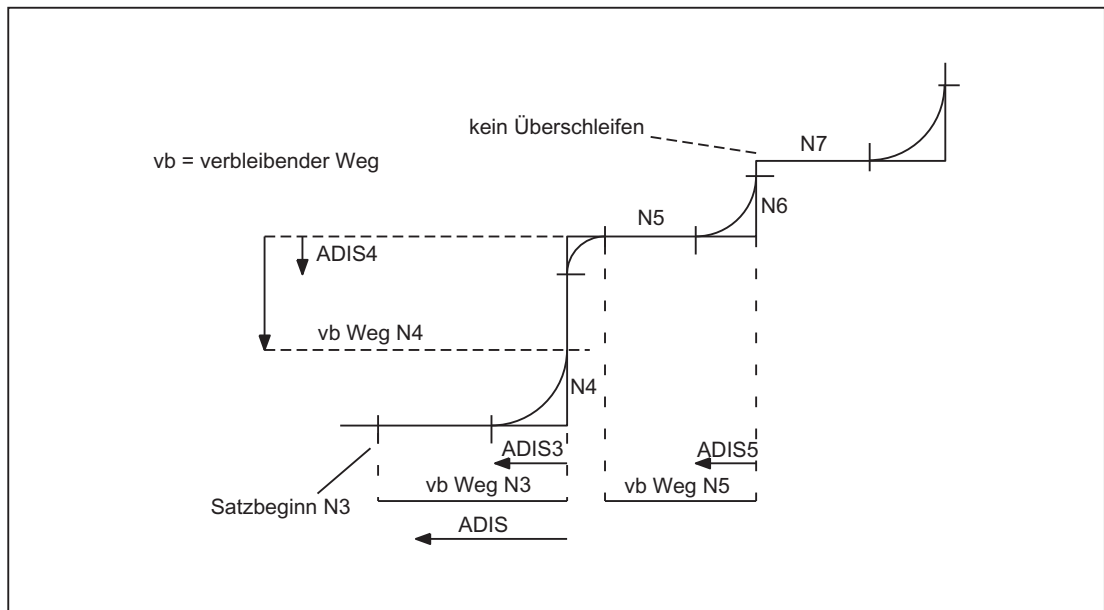


Bild 3-4 Bahnverlauf mit Begrenzung von ADIS



## Aktivierung / Deaktivierung

Der Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G641 aktiviert werden. Vor oder mit der Aktivierung sollten die Wegkriterien ADIS/ADISPOS angegeben werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium (G641) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G642, G643, G644 oder G645

## Programmbeispiel

Programmcode	Kommentar
N1 G641 Y50 F10 ADIS=0.5	; Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium aktivieren (Überschleifabstand: 0,5 mm).
N2 X50	
N3 X50.7	
N4 Y50.7	
N5 Y51.4	
N6 Y51.0	
N7 X52.1	

### 3.3.3.2 Überschleifen unter Einhaltung definierter Toleranzen (G642/G643)

#### Funktion

Beim Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen erfolgt das Überschleifen im Normalfall unter Einhaltung der maximal erlaubten Bahnabweichung. Anstelle dieser achsspezifischen Toleranz kann aber auch die Einhaltung der maximalen Konturabweichung (Konturtoleranz) oder der maximalen Winkelabweichung der Werkzeugorientierung (Orientierungstoleranz) konfiguriert werden.

#### Aktivierung

Der Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G642 bzw. G643 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen (G642/G643) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G641, G644 oder G645

### Unterschiede G642 - G643

Die Funktionen G642 und G643 weisen im Überschleifverhalten folgende Unterschiede auf:

G642	G643
Bei G642 wird der Überschleifweg aus dem kürzesten Überschleifweg aller Achsen bestimmt. Dieser Wert wird bei der Erzeugung eines Überschleifsatzes berücksichtigt.	Bei G643 kann der Überschleifweg jeder Achse unterschiedlich sein. Die Überschleifwege werden achsspezifisch satzintern berücksichtigt (=> kein eigener Überschleifsatz).
Bei G642 bestimmt sich der Überschleifbereich aus der kleinsten Toleranzvorgabe.	Sehr unterschiedliche Vorgaben für die Konturtoleranz und die Orientierungstoleranz können sich nur bei G643 auswirken.

### Parametrierung

#### Maximale Bahnabweichung

Die beim Überschleifen mit G642/G643 maximal erlaubte Bahnabweichung wird für jede Achse eingestellt im Maschinendatum:

MD33100 \$MA\_COMPRESS\_POS\_TOL

#### Konturtoleranz und Orientierungstoleranz

Die Konturtoleranz und die Orientierungstoleranz werden eingestellt in den kanalspezifischen Settingdaten:

SD42465 \$SC\_SMOOTH\_CONTUR\_TOL (Maximale Konturabweichung)

SD42466 \$SC\_SMOOTH\_ORI\_TOL (Maximale Winkelabweichung der Werkzeugorientierung)

Die Settingdaten können im NC-Programm programmiert und dadurch für jeden Satzübergang anders vorgegeben werden.

---

#### Hinweis

Das Settingdatum SD42466 \$SC\_SMOOTH\_ORI\_TOL ist nur bei aktiver Orientierungstransformation wirksam.

---

**Überschleifverhalten**

Das Überschleifverhalten mit G642 und G643 wird konfiguriert über das Maschinendatum:

MD20480 \$MC\_SMOOTHING\_MODE (Verhalten des Überschleifens mit G64x)

Die Einerstellen (**E**) definieren das Verhalten bei G643, die Zehnerstellen (**Z**) das Verhalten bei G642:

Wert E bzw. Z	Bedeutung
0	<b>Alle Achsen:</b> Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL
1	<b>Geometrieachsen:</b> Überschleifen unter Einhaltung der Konturtoleranz: SD42465 \$SC_SMOOTH_CONTUR_TOL <b>Restliche Achsen:</b> Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL
2	<b>Geometrieachsen:</b> Überschleifen unter Einhaltung der Orientierungstoleranz: SD42466 \$SC_SMOOTH_ORI_TOL <b>Restliche Achsen:</b> Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL
3	<b>Geometrieachsen:</b> Überschleifen unter Einhaltung der Konturtoleranz und Orientierungstoleranz: SD42465 \$SC_SMOOTH_CONTUR_TOL SD42466 \$SC_SMOOTH_ORI_TOL <b>Restliche Achsen:</b> Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL
4	<b>Alle Achsen:</b> Es wird die mit ADIS bzw. ADISPOS programmierte Überschleiflänge verwendet (wie bei G641). Die Vorgaben der achsspezifischen Toleranz bzw. der Kontur- und Orientierungstoleranz werden ignoriert.

**Profil für die Grenzgeschwindigkeit**

Die Benutzung eines Geschwindigkeitsprofils beim Überschleifen unter Einhaltung definierter Toleranzen wird über die Hunderterstelle von MD20480 gesteuert:

Wert	Bedeutung
< 100:	Innerhalb des Überschleifbereichs wird ein Profil der Grenzgeschwindigkeit berechnet, wie es sich aus den vorgegebenen maximalen Werten für Beschleunigung und Ruck der beteiligten Achsen bzw. der Bahn ergibt. Dies kann zu einem Ansteigen der Bahngeschwindigkeit in dem Überschleifbereich führen, und damit zu einem Beschleunigen der beteiligten Achsen.
≥100:	Für Überschleifsätze mit G641/G642 wird kein Profil der Grenzgeschwindigkeit berechnet. Es wird nur eine konstante Grenzgeschwindigkeit festgelegt. Damit wird verhindert, dass beim Überschleifen mit G641/G642 die beteiligten Achsen im Überschleifbereich eventuell beschleunigt werden. Diese Einstellung kann jedoch unter Umständen, insbesondere bei großen Überschleifbereichen, dazu führen, dass in Überschleifsätzen mit zu kleiner Geschwindigkeit gefahren wird.
1xx:	Kein Geschwindigkeitsprofil für G641
2xx:	Kein Geschwindigkeitsprofil für G642

**Hinweis**

Siehe auch MD28530 \$MC\_MM\_PATH\_VELO\_SEGMENTS (Anzahl der Speicherelemente z. Begrenzung der Bahngeschwindigkeit)

**Randbedingungen**

Einschränkung für Schutzbereiche bei aktiver Radius-Korrektur und einer Werkzeugorientierung:

Eine Werkzeugorientierung, die nicht senkrecht zu einer der drei Grundebenen des Basis-Koordinatensystems steht, berücksichtigt zwar die Radius-Korrektur, die Schutzbereiche werden jedoch nicht in die entsprechende Ebene gedreht.

Für G643 hat unbedingt zu gelten:

MD28530 \$MC\_MM\_PATH\_VELO\_SEGMENTS > 0 (Anzahl der Speicherelemente zur Begrenzung der Bahngeschwindigkeit)

Ist diese Bedingung erfüllt, so muss für alle Achsen gelten:

MD35240 \$MC\_ACCEL\_TYPE\_DRIVE = FALSE (Beschleunigungskennlinie DRIVE für Achsen Ein / Aus)

### 3.3.3.3 Überschleifen mit maximal möglicher Achsdynamik (G644)

#### Funktion

Bei diesem Modus des Bahnsteuerbetriebs mit Überschleifen steht die maximal mögliche Dynamik der Achsen im Vordergrund.

---

#### Hinweis

Das Überschleifen mit G644 ist nur möglich, wenn:

- alle beteiligten Achsen nur eine lineare Bewegung in den beiden betrachteten Sätzen enthalten.
- **keine** kinematische Transformation aktiv ist.

Falls eine beteiligte Achse ein Polynom enthält (Polynom programmiert, Spline aktiv, Kompressor aktiv) oder eine kinematische Transformation aktiv ist, wird der Satzübergang mit G642 überschliften.

---

#### Aktivierung

Der Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Achsdynamik kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G644 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Achsdynamik (G644) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G641, G642, G643 oder G645

**Parametrierung**

Das Überschleifverhalten mit G644 wird konfiguriert über die Tausender- und Zehntausenderstelle im Maschinendatum:

MD20480 \$MC\_SMOOTHING\_MODE (Verhalten des Überschleifens mit G64x)

Wert	Bedeutung
<b>Tausenderstelle:</b>	
0xxx:	Beim Überschleifen mit G644 werden die mit dem folgenden Maschinendatum angegebenen maximalen Abweichungen jeder Achse eingehalten: MD33100 \$MA_COMPRESS_POS_TOL Falls die Dynamik der Achse es zulässt, wird dabei evtl. die vorgegebene Toleranz nicht ausgenutzt.
1xxx:	Vorgabe des maximalen Überschleifwegs durch Programmierung von ADIS=... bzw. ADISPOS=... (wie bei G641).
2xxx:	Vorgabe der maximal auftretenden Frequenzen jeder Achse im Überschleifbereich mit dem Maschinendatum: MD32440 \$MA_LOOKAH_FREQUENCY (Glättungsfrequenz bei Look Ahead) Der Überschleifbereich wird so festgelegt, dass bei der Überschleifbewegung keine Frequenzen auftreten, welche die vorgegebene maximale Frequenz überschreiten.
3xxx:	Jede Achse, die einen Geschwindigkeitssprung an einer Ecke hat, fährt mit maximal möglicher Dynamik (maximale Beschleunigung und maximaler Ruck) um die Ecke. SOFT: Wenn SOFT aktiv ist, wird die maximale Beschleunigung <b>und</b> der maximale Ruck jeder Achse eingehalten. BRISK: Wenn BRISK aktiv ist, wird nur die maximale Beschleunigung eingehalten, der Ruck wird <b>nicht</b> begrenzt. Bei dieser Einstellung werden also weder die maximal auftretenden Abweichungen noch der maximale Überschleifweg kontrolliert. Die sich ergebenden Abweichungen bzw. Überschleifwege bestimmen sich allein aus den dynamischen Grenzwerten der jeweiligen Achse und der aktuellen Bahngeschwindigkeit.
4xxx:	Wie bei 0xxx werden die mit dem folgenden Maschinendatum angegebenen maximalen Abweichungen jeder Achse eingehalten: MD33100 \$MA_COMPRESS_POS_TOL Im Unterschied zu 0xxx wird die vorgegebene Toleranz nach Möglichkeit auch ausgenutzt. Die Achse erreicht daher nicht ihre maximal mögliche Dynamik.
5xxx:	Wie bei 1xxx wird der maximale Überschleifweg durch Programmierung von ADIS=... bzw. ADISPOS=... vorgegeben. Im Unterschied zu 1xxx wird der vorgegebene Überschleifweg nach Möglichkeit auch ausgenutzt. Die beteiligten Achsen erreichen dann eventuell nicht ihre maximal mögliche Dynamik.
<b>Zehntausenderstelle</b>	
0xxxx	Die Geschwindigkeitsprofile der Achsen werden im Überschleifbereich bei BRISK ohne Ruckbegrenzung und bei SOFT mit Ruckbegrenzung bestimmt.
1xxxx	Die Geschwindigkeitsprofile der Achsen werden im Überschleifbereich immer mit Ruckbegrenzung bestimmt, unabhängig davon ob BRISK oder SOFT aktiv ist.

Bei der Vorgabe der maximalen axialen Abweichungen (MD33100 \$MA\_COMPRESS\_POS\_TOL) bzw. des maximalen Überschleifabstandes (ADIS/ADISPOS) wird der zur Verfügung stehende Überschleifweg normalerweise nicht ausgenutzt, wenn es die Dynamik der beteiligten Achsen erlauben. Dadurch hängt die Länge des Überschleifwegs vom aktiven Bahnvorschub ab. Bei kleineren Bahngeschwindigkeiten erhält man geringere Abweichungen von der programmierten Kontur. Es kann jedoch eingestellt werden, dass in diesen Fällen nach Möglichkeit die vorgegebenen maximalen axialen Abweichungen bzw. der vorgegebene Überschleifabstand ausgenutzt wird. In diesem Fall sind dann die Abweichungen von der programmierten Kontur unabhängig vom programmierten Bahnvorschub.

---

#### Hinweis

Außer den bisher genannten kann zusätzlich folgende Begrenzung wirksam werden:

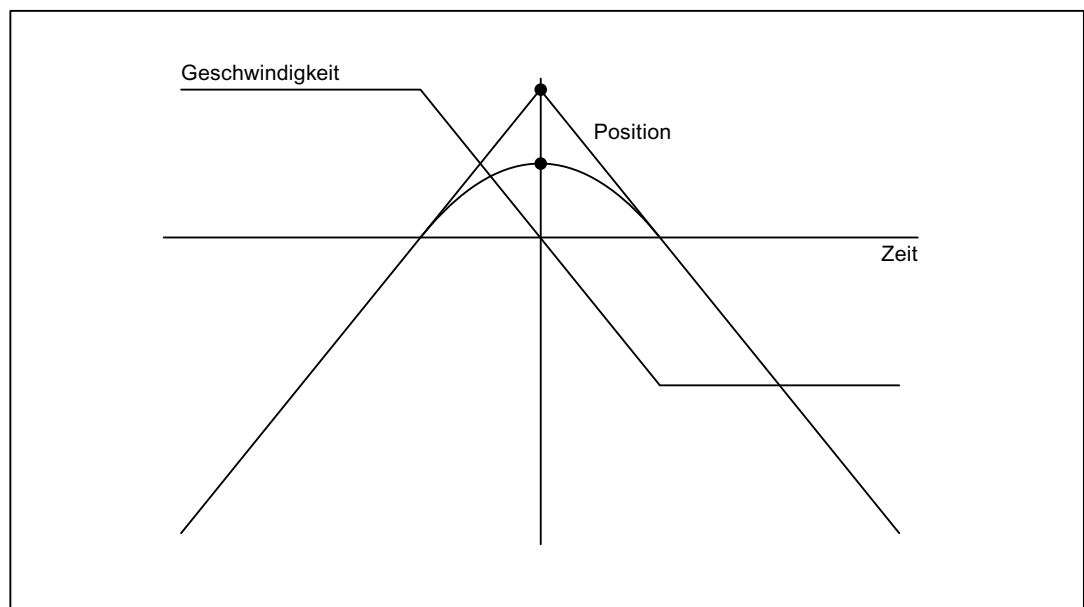
Der Überschleifabstand kann maximal die Hälfte der Länge der beteiligten Originalsätze betragen.

---

### Ruckbegrenzung

Die Glättung des Geschwindigkeitssprungs jeder Achse und damit die Form des Überschleifwegs hängt davon ab, ob eine Interpolation mit oder ohne Ruckbegrenzung durchgeführt wird.

Ohne Ruckbegrenzung erreicht die Beschleunigung jeder Achse im gesamten Überschleifbereich ihren Maximalwert:



Mit Ruckbegrenzung wird der Ruck jeder Achse im Überschleifbereich auf ihren jeweiligen Maximalwert begrenzt. Demzufolge besteht die Überschleifbewegung im Allgemeinen aus 3 Phasen:

- **1. Phase**

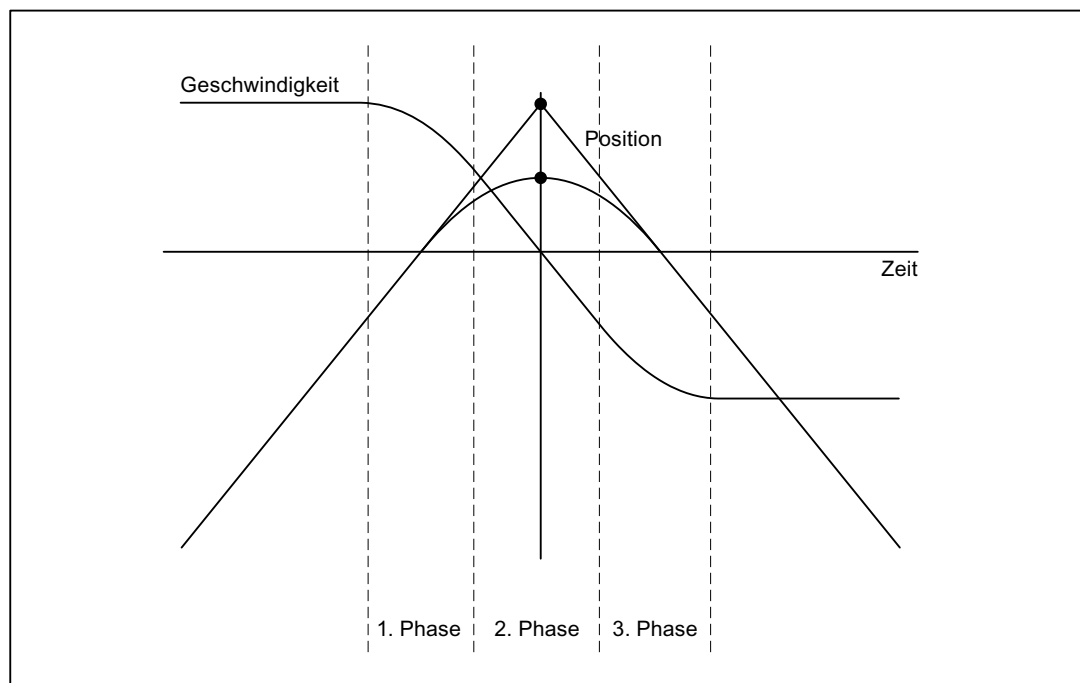
In der 1. Phase wird die maximale Beschleunigung jeder Achse aufgebaut. Dabei ist der Ruck konstant und gleich dem maximal möglichen Ruck der jeweiligen Achse.

- **2. Phase**

Die 2. Phase wird mit der maximal erlaubten Beschleunigung durchfahren.

- **3. Phase**

In der 3. Phase schließlich wird die Beschleunigung jeder Achse mit dem maximal erlaubten Ruck wieder auf Null abgebaut.





### 3.3.3.4 Überschleifen tangentialer Satzübergänge (G645)

#### Funktion

Bei diesem Modus des Bahnsteuerbetriebs mit Überschleifen werden auch bei tangentialen Satzübergängen Überschleifsätze gebildet, wenn der Krümmungsverlauf der Originalkontur in mindestens einer Achse einen Sprung aufweist.

Die Überschleifbewegung wird hierbei so festgelegt, dass alle beteiligten Achsen keinen Sprung in der Beschleunigung erfahren und die parametrisierten maximalen Abweichungen zur Originalkontur (MD33120 \$MA\_PATH\_TRANS\_POS\_TOL) nicht überschritten werden.

Bei knickförmigen, nicht-tangentialen Satzübergängen ist das Überschleifverhalten wie bei G642 (siehe Kapitel "Überschleifen unter Einhaltung definierter Toleranzen (G642/G643) (Seite 185)").

#### Aktivierung / Deaktivierung

Der Bahnsteuerbetrieb mit Überschleifen tangentialer Satzübergänge kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G645 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen tangentialer Satzübergänge (G645) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G641, G642, G643 oder G644

#### Vergleich G642 - G645

Beim Überschleifen mit G642 werden nur Satzübergänge überschleift, die eine Ecke darstellen, d. h. die Geschwindigkeit mindestens einer Achse hat einen Sprung. Ist ein Satzübergang jedoch tangential, aber die Krümmung macht einen Sprung, wird mit G642 kein Überschleifsatz eingefügt. Wird dieser Satzübergang mit endlicher Geschwindigkeit überfahren, so erfahren die Achsen einen mehr oder weniger großen Sprung der Beschleunigung, der (bei aktiver Ruckbegrenzung!) den parametrisierten Grenzwert (MD32432 \$MA\_PATH\_TRANS\_JERK\_LIM) nicht überschreiten darf. Abhängig von der Höhe des Grenzwerts wird die Bahngeschwindigkeit am Satzübergang dadurch evtl. stark verringert. Diese Einschränkung wird durch die Verwendung von G645 vermieden, da hier die Überschleifbewegung so festgelegt wird, dass keine Beschleunigungssprünge auftreten.

## Parametrierung

Mit dem folgenden Maschinendatum wird für jede Achse die maximal erlaubte Bahnabweichung beim Überschleifen mit G645 angegeben:

MD33120 \$MA\_PATH\_TRANS\_POS\_TOL

Dieser Wert ist nur relevant für tangentielle Satzübergänge, die nicht beschleunigungsstetig sind. Beim Überschleifen von knickförmigen, nicht-tangentialen Satzübergängen wird (wie bei G642) der Toleranzwert aus MD33100 \$MA\_COMPRESS\_POS\_TOL wirksam.

## Siehe auch

Freiformflächenmodus: Grundfunktionen (Seite 222)

### 3.3.4 LookAhead

#### 3.3.4.1 Standardfunktionalität

## Funktion

LookAhead ist eine im Bahnsteuerbetrieb (G64, G64x) aktive Funktion, die über den aktuellen Satz hinaus für mehrere NC-Teileprogrammätze eine vorausschauende Geschwindigkeitsführung ermittelt.

---

### Hinweis

LookAhead ist nur für Bahnachsen verfügbar, **nicht** für Spindeln und Positionierachsen.

---

Beinhaltet ein Teileprogramm aufeinanderfolgende Sätze mit sehr kleinen Bahnwegen, dann wird ohne LookAhead pro Satz nur eine Geschwindigkeit erreicht, die zum Satzendpunkt ein Abbremsen der Achsen unter Wahrung der Beschleunigungsgrenzen ermöglicht. Dies bedeutet, dass die programmierte Geschwindigkeit überhaupt nicht erreicht wird. Mit LookAhead ist es dagegen möglich, bei annähernd tangentialen Satzübergängen die Beschleunigungs- und Bremsphase über mehrere Sätze zu realisieren und somit bei kleinen Wegen einen höheren Vorschub zu erzielen.

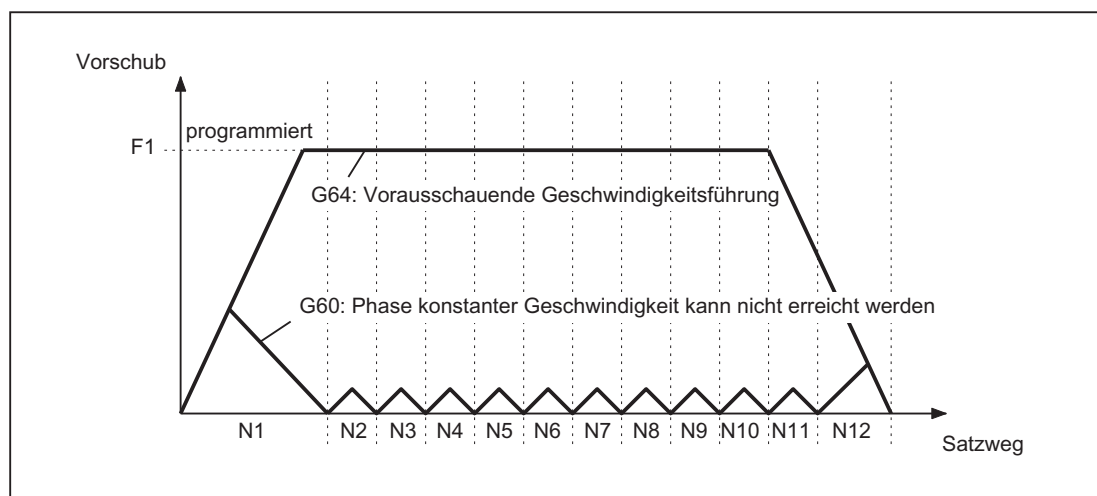


Bild 3-5 Geschwindigkeitsführung bei kurzen Wegstrecken und Genauhalt G60 bzw. Bahnsteuerbetrieb G64 mit LookAhead

Auf Geschwindigkeitsbeschränkungen wird vorausschauend so abgebremst, dass eine Verletzung der Beschleunigungs- und der Geschwindigkeitsgrenze vermieden wird.

LookAhead berücksichtigt planbare Geschwindigkeitsbeschränkungen wie:

- Genauhalt am Satzende
- Geschwindigkeitsbegrenzung im Satz
- Beschleunigungsbegrenzung im Satz
- Geschwindigkeitsbegrenzung am Satzübergang
- Synchronisieren mit Satzwechsel am Satzübergang

## Funktionsweise

LookAhead analysiert satzbezogen die planbaren Geschwindigkeitsbeschränkungen und legt dementsprechend die benötigten Bremsrampenprofile fest. Die Vorausschau wird automatisch an Satzlänge, Bremsvermögen und zulässige Bahngeschwindigkeit angepasst.

Aus Sicherheitsgründen wird die Geschwindigkeit am Satzende des letzten vorbereiteten Satzes zunächst zu 0 angenommen, da der anschließende Satz sehr klein oder ein Genauhaltsatz sein könnte und die Achsen zum Satzende zum Stillstand erreicht haben sollen.

Bei einer Folge von Sätzen mit hoher Sollgeschwindigkeit und sehr kurzen Wegstrecken kann in den einzelnen Sätzen die Geschwindigkeit, abhängig vom aktuellen vorausgeschauten Geschwindigkeitswert, erhöht werden, um die geforderte Sollgeschwindigkeit zu erreichen und anschließend wieder reduziert werden, damit die Geschwindigkeit am Satzende des letzten vorausschauenden Folgesatz 0 werden kann. Man erhält dadurch ein sägezahnförmiges Geschwindigkeitsprofil (siehe folgendes Bild), welches durch Herabsetzen der Sollgeschwindigkeit oder durch Vergrößern der vorausschauenden Satzanzahl vermieden werden kann.

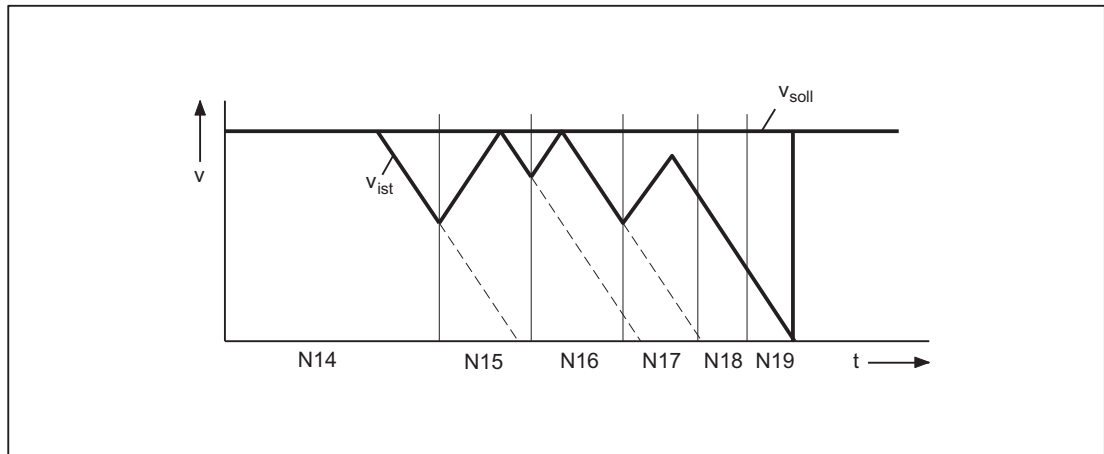


Bild 3-6 Beispiel für satzübergreifende Geschwindigkeitsführung (Zahl der vorausschauenden Sätze = 2)

### Aktivierung / Deaktivierung

LookAhead wird aktiviert durch Anwahl des Bahnsteuerbetriebs G64, G641, G642, G643, G644 oder G645.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G09.

LookAhead wird deaktiviert durch Anwahl des modalen Genauhalt (G60).

### Parametrierung

#### Satzanzahl

Um im Bahnsteuerbetrieb sicher zu fahren, muss der Vorschub über mehrere Sätze angepasst werden. Die Anzahl der vorausschauenden Sätze wird steuerungsintern automatisch ermittelt, wird optional aber über ein Maschinendatum begrenzt. Die Standardeinstellung ist "1" und bedeutet, dass LookAhead die Geschwindigkeitsführung nur für den Folgesatz berücksichtigt.

Da LookAhead vor allem bei (relativ zum Bremsweg) kurzen Sätzen wichtig ist, ist für das vorausschauende Bremsen die benötigte Satzanzahl von Interesse. Es ist ausreichend, eine Weglänge gleich dem Bremsweg zu betrachten, der nötig ist, um aus der maximalen Geschwindigkeit Stillstand zu erreichen.

Für eine Maschine mit einer niedrigen Achsbeschleunigung von  $a = 1 \text{ m/s}^2$  und einem dazu hohen Vorschub von  $v_{\text{Bahn}} = 10 \text{ m/min}$  erhält man mit einer erreichbaren Blockzykluszeit der Steuerung von  $T_B = 10 \text{ ms}$  eine für die Steuerung folgende Satzanzahl  $n_{\text{LookAhead}}$  von:

$$n_{\text{LookAhead}} = \text{Bremsweg} / \text{Satzlänge} = (v_{\text{Bahn}}^2 / (2a)) / (v_{\text{Bahn}} * T_B) = 9$$

Unter den angegebenen Bedingungen ist es nicht sinnvoll, den Vorschub über 10 Sätze anzupassen. Die angegebene Satzanzahl für die Vorausschau der LookAhead-Funktion verändert den LookAhead-Algorithmus und den Speicheraufwand nicht.

Da in einem Programm sehr oft die Bearbeitungsgeschwindigkeit kleiner der Maximalgeschwindigkeit ist, würde man mehr Sätze vorausschauen wie nötig ist, was zusätzliche unnötige Rechenleistung beanspruchen würde. Daher wird die benötigte Satzanzahl aus der Geschwindigkeit abgeleitet, die sich aus folgender Multiplikation ergibt:

- Programmierte Geschwindigkeit \* MD12100 \$MN\_OVR\_FACTOR\_LIMIT\_BIN  
(bei Verwendung eines **binär**kodierten Vorschub-Korrekturschalters)
- Programmierte Geschwindigkeit \* MD12030 \$MN\_OVR\_FACTOR\_FEEDRATE[30]  
(bei Verwendung eines **gray**kodierten Vorschub-Korrekturschalters)

Der Wert von MD12100 bzw. der 31. Override-Wert von MD12030 legt dabei die Dynamik-Reserven fest, die die Geschwindigkeitsführung für eine Überhöhung des Bahnvorschubs bereithält.

---

#### Hinweis

Der 31. Override-Wert von MD12030 sollte dem höchsten tatsächlich verwendeten Override-Faktor entsprechen.

---

#### Hinweis

Die vorausschauende Satzanzahl wird durch die mögliche Anzahl der NC-Sätze im IPO-Puffer begrenzt.

---

#### Geschwindigkeitsprofile

Neben den festen planbaren Geschwindigkeitsbeschränkungen kann LookAhead zusätzlich auch die programmierte Geschwindigkeit miteinbeziehen. Damit ist es möglich, über den aktuellen Satz hinaus vorausschauend die geringere Geschwindigkeit zu erreichen.

- **Ermittlung der Folgesatzgeschwindigkeit**

Ein mögliches Geschwindigkeitsprofil enthält die Ermittlung der Folgesatzgeschwindigkeit.

Anhand von Informationen aus dem aktuellen und dem folgenden NC-Satz wird ein Geschwindigkeitsprofil berechnet, aus dem wiederum die erforderlichen Geschwindigkeitsreduzierungen für den aktuellen Override abgeleitet werden.

Der ermittelte Maximalwert des Geschwindigkeitsprofils wird durch die maximale Bahngeschwindigkeit begrenzt.

Mit der Funktion ist es möglich, Override-berücksichtigend im aktuellen Satz eine Geschwindigkeitsreduzierung zu starten, so dass zu Beginn des Folgesatzes dessen geringere Geschwindigkeit erreicht sein kann. Dauert die Reduzierung der Geschwindigkeit länger als die Fahrzeit des aktuellen Satzes, so wird im Folgesatz die Geschwindigkeit weiter verringert. Die Geschwindigkeitsführung wird immer nur für den Folgesatz berücksichtigt.

Aktiviert wird die Funktion über das Maschinendatum:

MD20400 \$MC\_LOOKAH\_USE\_VELO\_NEXT\_BLOCK

Wert	Bedeutung
TRUE	Funktion aktiv
FALSE	Funktion <b>nicht</b> aktiv

- **Festlegung von Override-Eckwerten**

Ist das Geschwindigkeitsprofil der Folgesatzgeschwindigkeit nicht ausreichend, weil z. B. sehr hohe Override-Werte (z. B. 200 %) verwendet werden bzw. konstante Schnittgeschwindigkeit G96/G961 aktiv ist und somit die Geschwindigkeit im Folgesatz immer noch reduziert werden muss, so bietet LookAhead eine Möglichkeit an, die programmierte Geschwindigkeit über mehrere NC-Sätze vorausschauend zu reduzieren:

Mittels Festlegung von Override-Eckwerten berechnet sich LookAhead für jeden Eckwert ein begrenzendes Geschwindigkeitsprofil. Von diesen Profilen werden die erforderlichen Geschwindigkeitsreduzierungen für den aktuellen Override abgeleitet.

Der ermittelte Maximalwert des Geschwindigkeitsprofils wird durch die maximale Bahngeschwindigkeit begrenzt.

Der obere Eckwert sollte den Geschwindigkeitsbereich abdecken, der durch den Maximalwert des folgenden Maschinendatums erreicht wird:

MD12030 \$MN\_OVR\_FACTOR\_FEEDRATE[n] (Bewertung des Bahnvorschub-Korrekturschalters)

Er kann auch erreicht werden durch den Wert des Maschinendatums:

MD12100 \$MN\_OVR\_FACTOR\_LIMIT\_BIN (Begrenzung bei binärkodiertem Korrektorschalter)

Damit kann eine Reduzierung der Geschwindigkeit in den Satz hinein, in dem sie programmiert ist, vermieden werden.

Sind auch schon bei 100 %-Override deutliche satzübergreifende Geschwindigkeitsreduzierungen erforderlich, so sollte auch im unteren Override-Bereich ein Eckwert gesetzt werden.

Die Anzahl der verwendeten Override-Eckwerte pro Kanal werden angegeben im Maschinendatum:

MD20430 \$MC\_LOOKAH\_NUM\_OVR\_POINTS (Anzahl der Korrektorschalter-Eckwerte bei LookAhead)

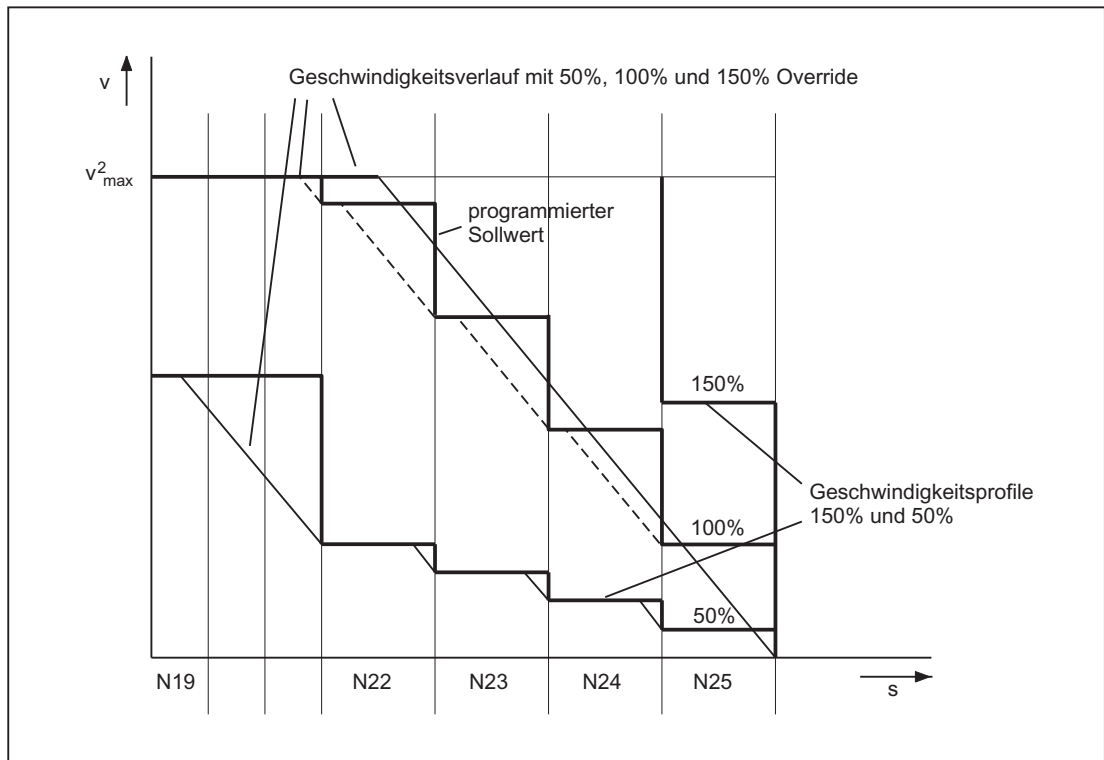
Die dazugehörigen Eckwerte werden hinterlegt im Maschinendatum:

MD20440 \$MC\_LOOKAH\_OVR\_POINTS (Korrektorschalter-Eckwerte bei LookAhead)

Beispiel:

Begrenzende Geschwindigkeitsverläufe mit:

- Override = 50 %, 100 % bzw. 150 %
- vorausschauender Satzanzahl = 4
- MD20430 \$MC\_LOOKAH\_NUM\_OVR\_POINTS = 2
- MD20440 \$MC\_LOOKAH\_OVR\_POINTS = 1.5, 0.5
- MD20400 \$MC\_LOOKAH\_USE\_VELO\_NEXT\_BLOCK = 1



Eine Kombination beider Verfahren (Ermittlung der Folgesatzgeschwindigkeit und Festlegung von Override-Eckwerten) zur Ermittlung der Geschwindigkeitsprofile ist möglich und in der Regel auch sinnvoll, weil bereits mit den vorgegebenen Maschinendaten für diese Funktionen der größte Bereich der Override-abhängigen Geschwindigkeitsbeschränkungen abgedeckt ist.

---

#### Hinweis

Ist keines der beiden Verfahren aktiviert, so wird die Sollgeschwindigkeit immer erst im aktuellen Satz angefahren.

---

#### Hinweis

Planbare Geschwindigkeitsbeschränkungen begrenzen die Override-abhängigen Geschwindigkeitsbeschränkungen.

---

#### Entlastungsfaktor bei Blockzyklus-Problemen

Blockzyklusprobleme treten auf, wenn die Verfahrnlänge der abzuarbeitenden NC-Sätze so kurz ist, dass die LookAhead-Funktion die Maschinengeschwindigkeit reduzieren muss, um der Satzaufbereitung genügend Zeit zur Verfügung zu stellen. In dieser Situation kann ein ständiges Abbremsen und Beschleunigen der Bahnbewegung auftreten.

Derartige Geschwindigkeits-Schwankungen können durch Angabe eines Entlastungsfaktors gedämpft werden:

MD20450 \$MC\_LOOKAH\_RELIEVE\_BLOCK\_CYCLE (Entlastungsfaktor für die Blockzykluszeit)



## Randbedingungen

### Achsspezifischer Vorschub-Halt / Achssperre

Achsspezifischer Vorschub-Halt und achsspezifische Achssperre werden von LookAhead nicht berücksichtigt.

Soll eine Achse interpoliert werden, die aber andererseits per achsspezifischem Vorschub-Halt oder Achsen-Sperre stehenbleiben soll, so hält LookAhead die Bahnbewegung nicht vor dem betreffenden Satz an, sondern bremst **im** Satz ab.

Stört dieses Verhalten, so kann über die PLC ein **achsspezifischer** Vorschub-Halt auf eine **kanalspezifische** Vorschubsperre umgelegt und damit die Bahn sofort angehalten werden (siehe auch Kapitel "Klemmungsüberwachung (Seite 93)").

### 3.3.4.2 Freiformflächenmodus: Erweiterungsfunktion

#### Funktion

Die Funktion "Freiformflächenmodus: Erweiterungsfunktion" ist eine Erweiterung der LookAhead-Standardfunktionalität und dient zur Berechnung des Bahngeschwindigkeitsprofils bei der Freiformflächenbearbeitung (siehe auch Kapitel "Freiformflächenmodus: Grundfunktionen (Seite 222)").

Ihr Einsatz optimiert den Bahnsteuerbetrieb wie folgt:

- Symmetrie zwischen Beschleunigungs- und Bremsprofil
- Gleichmäßiger Beschleunigungsvorgang auch bei sich ändernder Ruck- bzw. Beschleunigungsbegrenzung
- Gleichmäßiger Beschleunigungsvorgang von Sollgeschwindigkeitsprofilen unabhängig davon, inwieweit sie mit der vorgegebenen Dynamikbegrenzung anfahrbar sind oder nicht
- Vorausschauendes Bremsen auf niedrigere Sollgeschwindigkeiten

Die Gleichmäßigkeit und die Einhaltung der Dynamikbegrenzung gewährleisten, dass die Sollgeschwindigkeitsprofile zu einem homogenen Geschwindigkeitsprofil auf dem Teil geglättet werden. Damit wird sichergestellt, dass sich Schleppfehler möglichst gering auf die Oberflächengüte auswirken.

Die Funktion bietet daher folgende Vorteile:

- Gleichmäßigere Oberfläche des Werkstücks
- Geringere Belastung der Maschine

## Anwendungen

Die Funktion "Freiformflächenmodus: Erweiterungsfunktion" wird zur Bearbeitung von Werkstücken verwendet, die vorwiegend aus Freiformflächen bestehen.

---

### Hinweis

Bei Standardbearbeitungen werden keine besseren Ergebnisse erzielt, weshalb in solchen Fällen die Standardfunktionalität von LookAhead verwendet werden sollte.

---

## Voraussetzungen

- Die Funktion ist nur wirksam:
  - in der Betriebsart: AUTOMATIK
  - im Beschleunigungsmodus: Beschleunigung mit Ruckbegrenzung (SOFT)
- Eine Aktivierung ist nur möglich, wenn der notwendige Speicher konfiguriert ist:

MD28533 \$MC\_MM\_LOOKAH\_FFORM\_UNITS = <Wert>

Eine sinnvolle Wertzuweisung hängt ab vom Teileprogramm, den Satzlängen, der Achsdynamik sowie einer aktiven kinematischen Transformation.

Als Richtwert für die Freiformflächenbearbeitung gilt folgende Einstellung: 18

---

### Hinweis

MD28533 sollte wegen dem zusätzlichen Speicherbedarf nur für die Kanäle gesetzt werden, in denen auch Freiformflächen bearbeitet werden.

---

## Aktivierung / Deaktivierung

Die Funktion kann für jeden Dynamikmodus (siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 219)") unabhängig ein- oder ausgeschaltet werden:

MD20443 \$MC\_LOOKAH\_FFORM[<n>]= <Wert>

Index <n>	Dynamikmodus	<Wert>	Freiformflächenmodus: Erweiterungsfunktion
0	Standard-Dynamikeinstellungen (DYNNORM)	0	aus
		1	ein
1	Positionierbetrieb, Gewindebohren (DYNPOS)	0	aus
		1	ein
2	Schruppen (DYNROUGH)	0	aus
		1	ein
3	Schlichten (DYNSEMIFIN)	0	aus
		1	ein
4	Feinschlichten (DYNFINISH)	0	aus
		1	ein

Typischerweise ist die Funktion "Freiformflächenmodus: Erweiterungsfunktion" nur aktiv, wenn auch die Funktion "Freiformflächenmodus: Grundfunktionen" aktiv ist. Die Einstellungen in MD20443 \$MC\_LOOKAH\_FFORM[<n>] sollten daher mit den Einstellungen in MD20606 \$MC\_PREPDYN\_SMOOTHING\_ON[<n>] übereinstimmen.

In den Dynamikmodi, in denen die Funktion "Freiformflächenmodus: Erweiterungsfunktion" ausgeschaltet ist, ist die LookAhead-Standardfunktionalität aktiv.

## Programmierung

In der Regel wird die Funktion "Freiformflächenmodus: Erweiterungsfunktion" durch einen Wechsel des Dynamikmodus im Teileprogramm wirksam.

## Beispiel

Folgende Parametrierung ist gegeben:

MD20443 \$MC\_LOOKAH\_FFORM[0] = 0

MD20443 \$MC\_LOOKAH\_FFORM[1] = 0

MD20443 \$MC\_LOOKAH\_FFORM[2] = 1

MD20443 \$MC\_LOOKAH\_FFORM[3] = 1

MD20443 \$MC\_LOOKAH\_FFORM[4] = 1

3.3 Bahnsteuerbetrieb

Programmcode	Kommentar
N10 DYNPOS	; Dynamikmodus DYNPOS einschalten. Im Dynamikmodus DYNPOS ist die LookAhead-Standardfunktionalität aktiv.
...	
N100 G17 G54 F10000	
N101 DYNFINISH	; Dynamikmodus DYNFINISH einschalten. Im Dynamikmodus DYNFINISH ist die Funktion "Freiformflächenmodus: Erweiterungsfunktionen" aktiv.
N102 SOFT G642	
N103 X-0.274 Y149.679 Z100.000 G0	
N104 COMPCAD	
...	
N1009 Z4.994 G01	
N10010 X.520 Y149.679 Z5.000	
N10011 X10.841 Y149.679 Z5.000	
N10012 X11.635 Y149.679 Z5.010	
N10013 X12.032 Y149.679 Z5.031	
M30	

**Hinweis**

Beim Wechsel zwischen der LookAhead-Standardfunktionalität und der Funktion "Freiformflächenmodus: Erweiterungsfunktion" bzw. umgekehrt wird der Bahnsteuerbetrieb durch einen Interpolator-Stopp unterbrochen.

**Randbedingungen**

**Automatische Funktionsumschaltung**

Die Anwendung folgender Funktionen bei aktiver Funktion "Freiformflächenmodus: Erweiterungsfunktion" führt zu einer automatischen Umschaltung auf die LookAhead-Standardfunktionalität:

- Gewindeschneiden/Gewindebohren (G33, G34, G35, G331, G332, G63)
- Bahn-Leitwertkopplung
- Stanzen, Nibbeln
- Kartesisches PTP-Fahren

Danach wird die Funktion "Freiformflächenmodus: Erweiterungsfunktion" automatisch wieder eingeschaltet.

### **Verwendung der Befehle der G-Funktionsgruppe 15 (Vorschubtypen)**

Die Verwendung folgender Vorschubtypen ist bei aktiver Funktion "Freiformflächenmodus: Erweiterungsfunktion" nicht zu empfehlen:

- Umdrehungsvorschub (G95, G96, G97, ...)
- Zeitreziproker Vorschub (G93)

### **Zeitverhalten bei Vorschubkorrekturen**

Vorschubkorrekturen (über Maschinensteuertafel, \$AC\_OVR, ...) können die Verfahrzeit gegenüber der LookAhead-Standardfunktionalität merklich verlängern.

### **Eilgangbewegung (G0)**

In die Freiformflächenbearbeitung eingestreute G0-Sätze schalten die LookAhead-Funktionalität nicht um (von der Funktion "Freiformflächenmodus: Erweiterungsfunktion" auf die LookAhead-Standardfunktionalität oder umgekehrt).

D. h. obwohl mit G0 die Standard-Dynamikeinstellung (DYNNORM) wirksam ist, wird dadurch nicht automatisch auch die für DYNNORM voreingestellte LookAhead-Standardfunktionalität (→ MD20443 \$MC\_LOOKAH\_FFORM[0]) wirksam.

Durch das Beibehalten der aktuell aktiven LookAhead-Funktionalität wird ein homogeneres Geschwindigkeitsprofil erreicht, zumal in der Regel die G0- und Polynomsätze durch Überschleifen glatt verbunden sind.

### **Anzahl der NC-Sätze im IPO-Puffer**

In der Regel empfiehlt es sich, für die Verwendung der Funktion "Freiformflächenmodus: Erweiterungsfunktion" die konfigurierte Anzahl der NC-Sätze im Interpolationspuffer deutlich zu erhöhen:

MD28060 \$MC\_MM\_IPO\_BUFFER\_SIZE > 100

Ein zu geringer Satzspeicher verringert ggf. die Gleichmäßigkeit des Bahngeschwindigkeitsprofils.

## 3.4 Dynamikanpassungen

### 3.4.1 Glättung der Bahngeschwindigkeit

#### Einleitung

Die Geschwindigkeitsführung nutzt die vorgegebene Achsdynamik aus. Wenn der programmierte Vorschub nicht erreicht werden kann, wird die Bahngeschwindigkeit an den parametrisierten axialen Grenzwerten und den Grenzwerten der Bahn (Geschwindigkeit, Beschleunigung, Ruck) geführt. Damit kann es zu häufigen Brems- und Beschleunigungsvorgängen auf der Bahn kommen.

Findet bei einer Bearbeitung mit hoher Bahngeschwindigkeit ein kurzzeitiger Beschleunigungsvorgang statt, der nach sehr kurzer Zeit wieder zu einem Bremsvorgang führt, so wird die Bearbeitungszeit hierdurch nicht deutlich reduziert. Diese Beschleunigungsvorgänge können jedoch zu unerwünschten Erscheinungen führen, z. B. wenn Maschinenresonanzen angeregt werden.

Bei einigen Anwendungen im Formenbau, speziell beim Hochgeschwindigkeitsfräsen, ist dagegen eine gleichmäßige Bahngeschwindigkeit wünschenswert. In diesen Fällen kann es daher sinnvoll sein, zugunsten einer ruhigeren Bahngeschwindigkeit auf kurzzeitige Beschleunigungsvorgänge zu verzichten.

#### Funktion

Bei aktiver Funktion "Glättung der Bahngeschwindigkeit" ist zugunsten einer ruhigeren Bahngeschwindigkeitsführung ein Glättungsfaktor wirksam, der den maximal zulässigen Produktivitätsverlust bestimmt: Beschleunigungsvorgänge, die weniger als dieser Faktor zu einer kürzeren Programmlaufzeit beitragen, werden nicht durchgeführt. Betrachtet werden dabei nur Beschleunigungsvorgänge, deren Frequenzen oberhalb der parametrierbaren Grenzfrequenzen der beteiligten Achsen liegen.

Vorteile:

- Vermeidung von Anregungen möglicher Maschinenresonanzen aufgrund von ständigen, kurzzeitigen Brems- und Beschleunigungsvorgängen (im Bereich weniger IPO-Takte).
- Vermeidung ständig variierender Schnittgeschwindigkeiten durch Beschleunigungsvorgänge, die keinen großen Gewinn für die Programmlaufzeit bewirken.

---

### Hinweis

Die Glättung der Bahngeschwindigkeit bewirkt keinen Konturfehler.

Schwankungen der Achsgeschwindigkeit aufgrund von Krümmungen in der Kontur bei konstanter Bahngeschwindigkeit können weiterhin auftreten und werden mit dieser Funktion nicht reduziert.

Schwankungen der Bahngeschwindigkeit aufgrund der Vorgabe eines neuen Vorschubs werden ebenfalls nicht verändert. Dies liegt in der Verantwortung des Teileprogrammerstellers.

---

### Voraussetzungen

- Die Glättung der Bahngeschwindigkeit ist nur wirksam im Bahnsteuerbetrieb mit LookAhead über mehrere Sätze bei SOFT und BRISK. Die Glättung ist **nicht** wirksam bei GO.
- Die Taktzeiten der Steuerung müssen so parametrierbar sein, dass der Vorlauf genügend Sätze aufbereiten kann, um einen Beschleunigungsvorgang analysieren zu können.

### Aktivierung / Deaktivierung

Die Funktion "Glättung der Bahngeschwindigkeit" wird aktiviert/deaktiviert mit dem Maschinendatum:

MD20460 \$MC\_LOOKAH\_SMOOTH\_FACTOR (Glättungsfaktor bei LookAhead)

Wert	Bedeutung
0.0	Glättung der Bahngeschwindigkeit <b>nicht</b> aktiv (Standardeinstellung)
> 0	Glättung der Bahngeschwindigkeit aktiv

Eine Änderung der MD-Einstellung wird erst durch NEW CONF wirksam.

### Parametrierung

#### Glättungsfaktor

Der Glättungsfaktor wird eingestellt über das kanalspezifische Maschinendatum:

MD20460 \$MC\_LOOKAH\_SMOOTH\_FACTOR (Glättungsfaktor bei LookAhead)

Der Prozentwert definiert, wieviel ein Bearbeitungsschritt ohne Beschleunigungen/Verzögerungen länger sein darf als der entsprechende Schritt mit Ausführung der Beschleunigungen/Verzögerungen.

Dies wäre ein "worst case"-Wert, wenn alle Beschleunigungsvorgänge innerhalb des Teileprogramms, außer der ersten Anfahrbewegung, weggeglättet werden würden. Die tatsächliche Verlängerung wird in jedem Fall kleiner sein, möglicherweise sogar 0, falls das Kriterium für keinen Beschleunigungsvorgang anspricht. Es können also durchaus Werte von 50 bis 100 % eingetragen werden, ohne eine deutliche Verlängerung der Bearbeitungszeit zu erhalten.

### Berücksichtigung des programmierten Vorschubs

Die Glättung der Bahngeschwindigkeit kann mit oder ohne Berücksichtigung des programmierten Vorschubs durchgeführt werden. Die Auswahl erfolgt über das Maschinendatum:

MD20462 \$MC\_LOOKAH\_SMOOTH\_WITH\_FEED (Bahnglättung mit programmiertem Vorschub)

Wert	Bedeutung
0	Programmierter Vorschub wird <b>nicht</b> berücksichtigt.
1	Programmierter Vorschub wird berücksichtigt (Standardeinstellung).

Bei Berücksichtigung des programmierten Vorschubs wird der vorgegebene Glättungsfaktor (siehe MD20460) besser eingehalten, wenn der Override auf 100 % steht.

### Achsspezifische Grenzfrequenzen

Die achsspezifischen Grenzfrequenzen werden festgelegt über das Maschinendatum:

MD32440 \$MA\_LOOKAH\_FREQUENCY (Glättungsfrequenz bei LookAhead)

Beschleunigungs- und Bremsvorgänge, die mit einer höheren Frequenz ablaufen, werden abhängig von der Parametrierung der folgenden Maschinendaten geglättet oder in der Dynamik reduziert:

MD20460 \$MC\_LOOKAH\_SMOOTH\_FACTOR (Glättungsfaktor bei LookAhead)

MD20465 \$MC\_ADAPT\_PATH\_DYNAMIC (Adaption der Bahndynamik)

Weitere Informationen zu MD20465 siehe Kapitel "Anpassung der Bahndynamik (Seite 210)".

---

### Hinweis

Werden Schwingungen in der Mechanik einer Achse angeregt und ist deren Frequenz bekannt, so sollte MD32440 kleiner als diese Frequenz eingestellt werden.

Die benötigten Resonanzfrequenzen können z. B. über die integrierten Messfunktionen ermittelt werden.

---

## Funktionsweise

Aus den an der Bahn beteiligten Achsen wird intern das Minimum von MD32440 ermittelt ( $= f_{\text{Bahn}}$ ). Für die Glättung werden nur Beschleunigungsvorgänge betrachtet, bei denen die Anfangs- oder Endgeschwindigkeit dieser Bewegung innerhalb folgender Zeit wieder erreicht wird:

$$t = t_2 - t_1 = 2 / f_{\text{Bahn}}$$

Auf diese Beschleunigungsvorgänge wird verzichtet, wenn die sich dadurch ergebende Verlängerung der Bearbeitungszeit den über den Glättungsfaktor (MD20460) vorgegebenen Grenzwert nicht überschreitet.



## Beispiel

Folgende Parametrierung ist gegeben:

MD20460 \$MC\_LOOKAH\_SMOOTH\_FACTOR = 10 %

MD32440 \$MA\_LOOKAH\_FREQUENCY[AX1] = 20 Hz

MD32440 \$MA\_LOOKAH\_FREQUENCY[AX2] = 20 Hz

MD32440 \$MA\_LOOKAH\_FREQUENCY[AX3] = 10 Hz

An der Bahn sind die 3 Achsen X = AX1, Y = AX2, Z = AX3 beteiligt.

Das Minimum vom MD32440 dieser 3 Achsen beträgt somit 10 Hz. Daraufhin werden Beschleunigungsvorgänge untersucht, die in einem Zeitraum  $t_2 - t_1 = 2 / 10 \text{ Hz} = 200 \text{ ms}$  ablaufen. Die Zeit  $t_2$  ist diejenige Zeit, zu der nach einem Beschleunigungsvorgang von der Geschwindigkeit  $v_1$  ausgehend diese Geschwindigkeit  $v_1$  wieder erreicht wird. Nur dieser Bereich wird auch für die Verlängerung der Bearbeitungszeit betrachtet.

Wenn der Zeitraum  $t_2 - t_1$  größer als 200 ms ist oder wenn die zusätzliche Programmbearbeitungszeit  $t_3 - t_2$  mehr als 10 % (= MD20460) von  $t_2 - t_1$  beträgt, ergibt sich folgender Zeitverlauf:

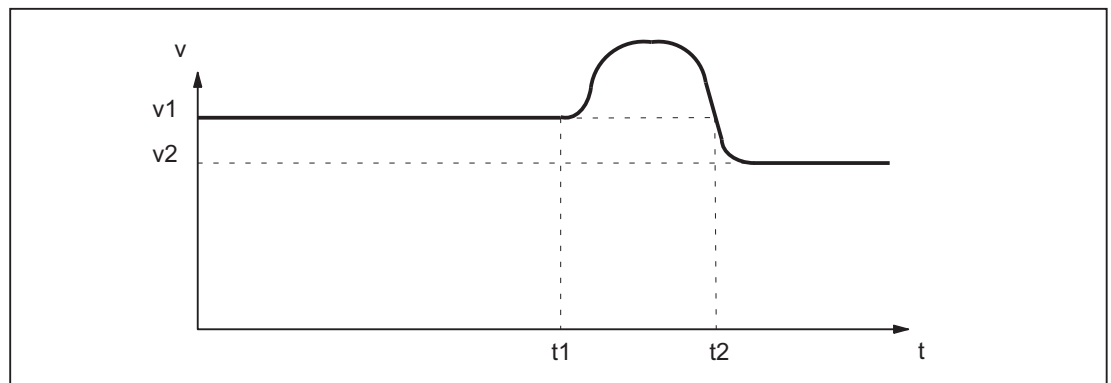


Bild 3-7 Verlauf der zeitoptimalen Bahngeschwindigkeit (ohne Glättung)

Wenn dagegen der Zeitraum  $t_2 - t_1$  kleiner als 200 ms ist und wenn die zusätzliche Programmbearbeitungszeit  $t_3 - t_2$  maximal 10 % von  $t_2 - t_1$  beträgt, dann ergibt sich dieser Zeitverlauf:

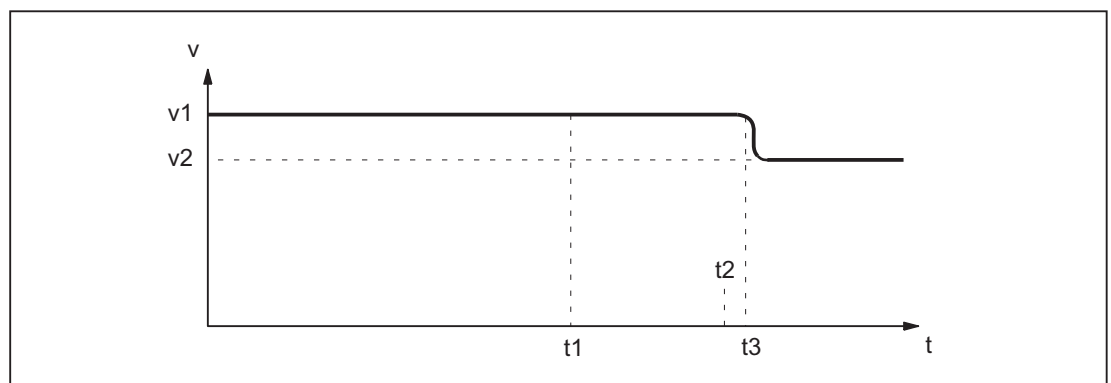


Bild 3-8 Verlauf der geglätteten Bahngeschwindigkeit

### 3.4.2 Anpassung der Bahndynamik

#### Funktion

Hochdynamische Beschleunigungs- und Bremsvorgänge während der Bearbeitung können zur Anregung von mechanischen Schwingungen von Maschinenelementen und in Folge zu einer Verminderung der Oberflächengüte des Werkstücks führen.

Mit der Funktion "Anpassung der Bahndynamik" kann die Dynamik der Beschleunigungs- und Bremsvorgänge an die Maschinengegebenheiten angepasst werden.

---

#### Hinweis

Die Funktion "Anpassung der Bahndynamik" betrachtet nicht die Brems- und Beschleunigungsvorgänge der einzelnen an der Bahn beteiligten Achsen, sondern immer der resultierenden Bahn. Daher kann es auch bei einem konstanten Bahngeschwindigkeitsverlauf durch unetstetige Konturverläufe oder kinematische Transformationen zu bezüglich der Anregung von mechanischen Schwingungen kritischen Brems- und Beschleunigungsvorgängen der Achsen kommen.

---

#### Wirksamkeit

Die Funktion "Anpassung der Bahndynamik" ist nur während Bahnbewegungen wirksam:

- Bahnsteuerbetrieb (G64, G64x)

Im Bahnsteuerbetrieb wird die optimale Wirkung der Dynamikanpassung bei einem wirksamen Override von 100% erreicht. Deutliche Abweichungen von diesem Wert oder Funktionen, durch die die Bahnachsen abgebremst werden müssen (z. B. Hilfsfunktionsausgaben an die PLC), vermindern die gewünschte Wirkung stark.

- Genauhalt (G60)

Unter folgenden Bedingungen ist die Funktion "Anpassung der Bahndynamik" auch bei Bahnbewegungen **nicht** wirksam:

- Programmierem Eilgang (G0)
- Veränderungen des Override-Werts
- Stoppanforderungen während der Bewegung (z. B. NC-Stop, NC-Reset)
- Funktion "Geschwindigkeitsabhängige Bahnbeschleunigung" (DRIVE) ist aktiv

#### Aktivierung / Deaktivierung

Die Funktion wird aktiviert/deaktiviert mit dem Maschinendatum:

MD20465 \$MC\_ADAPT\_PATH\_DYNAMIC (Adaption der Bahndynamik)

Wert	Bedeutung
= 1.0	Dynamikanpassung <b>nicht</b> aktiv (Standardeinstellung)
> 1.0	Dynamikanpassung aktiv

Durch die Aktivierung wird im Bahnsteuerbetrieb intern **immer die Funktion "Glättung der Bahngeschwindigkeit" mitaktiviert** (siehe Kapitel "Glättung der Bahngeschwindigkeit (Seite 206)").

Falls der Glättungsfaktor (MD20460 \$MC\_LOOKAH\_SMOOTH\_FACTOR) auf 0 % eingestellt ist (= Funktion deaktiviert; Voreinstellung!), wird als Ersatz ein Glättungsfaktor von 100 % verwendet. Bei einem von 0 % verschiedenen Glättungsfaktor wird der eingestellte Wert wirksam.

## Parametrierung

### Anpassfaktor der Bahndynamik

Durch den Anpassfaktor der Bahndynamik werden kurzzeitige Veränderungen der Bahngeschwindigkeit mit verminderten Dynamikgrenzwerten durchgeführt.

Der Anpassfaktor ist kanalspezifisch einstellbar:

- Für Verfahrbewegungen mit Beschleunigung ohne Ruckbegrenzung (**BRISK**) über:  
MD20465 \$MC\_ADAPT\_PATH\_DYNAMIC[ 0 ]  
→ Der Anpassfaktor wirkt auf die Beschleunigung.
- Für Verfahrbewegungen mit Beschleunigung mit Ruckbegrenzung (**SOFT**) über:  
MD20465 \$MC\_ADAPT\_PATH\_DYNAMIC[ 1 ]  
→ Der Anpassfaktor wirkt auf den Ruck.

### Achsspezifische Grenzfrequenzen

Die Dynamikbegrenzung soll nur bei den Brems- und Beschleunigungsvorgängen wirken, die mechanische Schwingungen größer einer bestimmten Grenzfrequenz auslösen und dadurch zu einer Anregung von Maschinenresonanzen führen.

Die Grenzfrequenz, ab der die Dynamikbegrenzung einsetzen soll, wird achsspezifisch vorgegeben über das Maschinendatum:

MD32440 \$MA\_LOOKAH\_FREQUENCY (Glättungsfrequenz bei LookAhead)

Weitere Informationen siehe Kapitel "Glättung der Bahngeschwindigkeit (Seite 206)".

### Funktionsweise

Bei der Bearbeitung ermittelt die Steuerung zyklisch über alle an der Bahn beteiligten Achsen das Minimum aller Grenzfrequenzen als die für die Dynamikanpassung relevante Grenzfrequenz (f) und berechnet daraus das relevante Zeitfenster ( $t_{adapt}$ ):

$$t_{adapt} = 1 / f$$

Die Größe des relevanten Zeitfensters  $t_{adapt}$  bestimmt das weitere Verhalten:

1. Die erforderliche Zeit für die Geschwindigkeitsänderung ist kleiner als  $t_{adapt}$ :

Die Beschleunigungen werden reduziert um einen Faktor  $> 1$  und  $\leq$  dem Wert im Maschinendatum:

MD20465 ADAPT\_PATH\_DYNAMIC (Adaption der Bahndynamik)

Durch die geringere Beschleunigung verlängert sich die Zeit für die Geschwindigkeitsänderung.

Folgende Fälle werden unterschieden:

- Die Beschleunigung wird mit einem Wert kleiner als MD20465 so verringert, dass der Vorgang  $t_{adapt}$  [s] dauert. Die erlaubte Reduktion muss nicht voll ausgenutzt werden.
- Die Beschleunigungszeit wird mit dem Wert im MD20465 verringert. Der Vorgang dauert trotz der geringeren Beschleunigung kürzer als  $t_{adapt}$ . Die erlaubte Reduktion wurde voll ausgenutzt.

2. Die erforderliche Zeit für die Geschwindigkeitsänderung ist größer als  $t_{adapt}$ :

Es ist keine Dynamikanpassung erforderlich.

### Beispiel

Das folgende Beispiel soll die Auswirkung der Funktion "Anpassung der Bahndynamik" auf Verfahrbewegungen mit Beschleunigung ohne Ruckbegrenzung (BRISK) veranschaulichen.

Folgende Parametrierung ist gegeben:

```
MD20465 $MC_ADAPT_PATH_DYNAMIC[0] = 1.5
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 1.0
MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 20 Hz      TAX1 = 1/20 Hz = 50 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX2] = 10 Hz      TAX2 = 1/10 Hz = 100 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX3] = 20 Hz      TAX3 = 1/20 Hz = 50 ms
```

---

### Hinweis

Zur Verdeutlichung der Auswirkung der Dynamikanpassung ist der Wert für den Glättungsfaktor (MD20460) auf "1" gesetzt, wodurch die mitaktivierte Funktion "Glättung der Bahngeschwindigkeit" praktisch nicht aktiv ist.

---

An der Bahn sind die 3 Achsen X = AX1, Y = AX2, Z = AX3 beteiligt.

Bei Bahnbewegungen, an denen die Achse AX2 beteiligt ist, werden alle Brems- und Beschleunigungsvorgänge angepasst, die weniger als  $T_{AX2}$  dauern würden.

Sind bei Bahnbewegungen nur die Achsen AX1 und/oder AX3 beteiligt, werden alle Brems- und Beschleunigungsvorgänge angepasst, die weniger als  $T_{AX1} = T_{AX3}$  dauern würden.

Das relevante Zeitfenster wird in den folgenden Bildern mit  $t_{adapt...}$  bezeichnet.

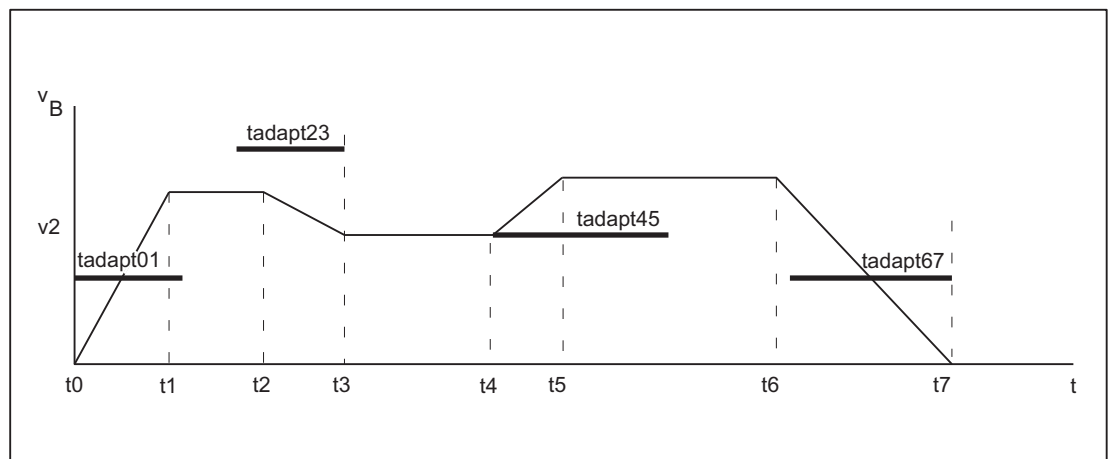


Bild 3-9 Zeitoptimaler Bahngeschwindigkeitsverlauf ohne Glättung und Dynamikanpassung

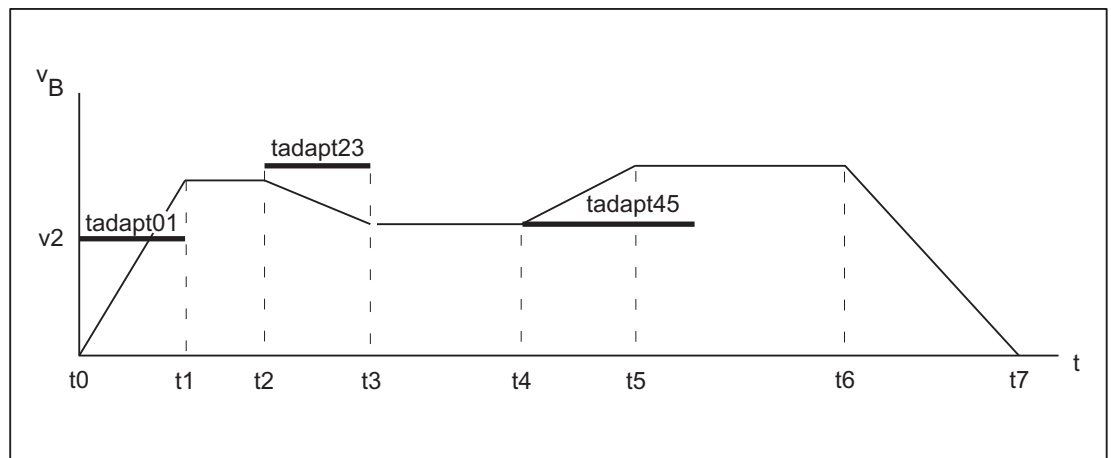


Bild 3-10 Bahngeschwindigkeitsverlauf mit Anpassung der Bahndynamik

- Intervalle  $t_0 - t_1$  und  $t_2 - t_3$ : Der Beschleunigungsvorgang zwischen  $t_0 - t_1$  und der Bremsvorgang zwischen  $t_2 - t_3$  werden durch eine Anpassung der Beschleunigung auf die Zeit  $t_{adapt01}$  bzw.  $t_{adapt23}$  zeitlich verlängert.
- Intervall  $t_4 - t_5$ : Der Beschleunigungsvorgang zwischen  $t_4 - t_5$  wird mit einer um den maximalen Anpassfaktor 1,5 reduzierten Beschleunigung durchgeführt. Dennoch ist der Beschleunigungsvorgang vor der Zeit  $t_{adapt45}$  beendet.
- Intervall  $t_6 - t_7$ : Der Bremsvorgang zwischen  $t_6 - t_7$  bleibt unverändert, da er länger als  $t_{adapt67}$  dauert.

### 3.4.3 Ermittlung der Dynamikgrenzwerte

Die Inbetriebnahme der Funktion "Anpassung der Bahndynamik" erfordert neben der Ermittlung der Eigenfrequenz der Bahnachsen zur Parametrierung der achsspezifischen Grenzfrequenzen (MD32440 \$MA\_LOOKAH\_FREQUENCY) auch die Ermittlung der Dynamikgrenzwerte für Geschwindigkeit, Beschleunigung und Ruck.

#### Vorgehensweise

Im Folgenden ist die Ermittlung der Dynamikgrenzwerte für das Verfahren der Bahnachsen mittels Beschleunigung mit Ruckbegrenzung (SOFT) beschrieben. Das Vorgehen kann sinngemäß auf den Fall der Beschleunigung ohne Ruckbegrenzung (BRISK) übertragen werden.

1. Deaktivieren Sie die Funktion "Anpassung der Bahndynamik":  
MD20465 \$MC\_ADAPT\_PATH\_DYNAMIC [1] = 1
2. Untersuchen Sie das Positionierverhalten der einzelnen Bahnachsen bei unterschiedlichen Verfahrgeschwindigkeiten. Stellen Sie dabei den Ruck so ein, dass die gewünschte Positioniertoleranz eingehalten wird.

---

#### Hinweis

Je höher die Verfahrgeschwindigkeit ist, von der aus der Positioniervorgang begonnen wird, desto höher kann im Allgemeinen der Ruck eingestellt werden.

---

3. Übernehmen Sie den für die unkritischste Verfahrgeschwindigkeit ermittelten maximal zulässigen Ruck:

MD32431 \$MA\_MAX\_AX\_JERK (Maximaler Ruck)

4. Ermitteln Sie für alle Bahnachsen den Faktor  $F_{APD}$  mit:

$F_{APD} = (\text{größter ermittelter Ruck}) / (\text{kleinster ermittelter Ruck})$

---

#### Hinweis

Der kleinste ermittelte Ruck ist der Wert für den Ruck bei der kritischsten Verfahrgeschwindigkeit.

---

5. Tragen Sie den über alle Bahnachsen ermittelten größten Faktor  $F_{APD}$  als Wert für den Anpassfaktor der Bahndynamik ein:

MD20465 \$MC\_ADAPT\_PATH\_DYNAMIC [1] =  $F_{APD}$

### 3.4.4 Zusammenwirken der Funktionen "Glättung der Bahngeschwindigkeit" und "Anpassung der Bahndynamik"

Die folgenden Beispiele sollen das Zusammenwirken der Funktionen "Glättung der Bahngeschwindigkeit" und "Anpassung der Bahndynamik" im Bahnsteuerbetrieb veranschaulichen.

#### Beispiel 1

Beschleunigungsmodus: BRISK

An der Bahn sind die 3 Achsen X = AX1, Y = AX2, Z = AX3 beteiligt.

Folgende Parametrierung ist gegeben:

```
MD20465 $MC_ADAPT_PATH_DYNAMIC[0] = 3
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 80.0
MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 20      TAX1 = 1/20 Hz = 50 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX2] = 20      TAX2 = 1/20 Hz = 50 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX3] = 20      TAX3 = 1/20 Hz = 50 ms
```

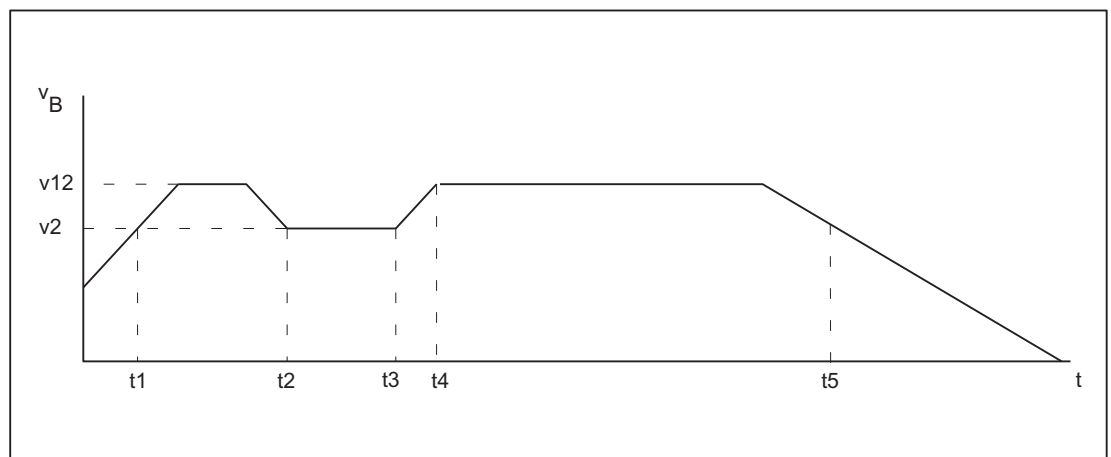


Bild 3-11 Zeitoptimaler Bahngeschwindigkeitsverlauf ohne Glättung und Dynamikanpassung

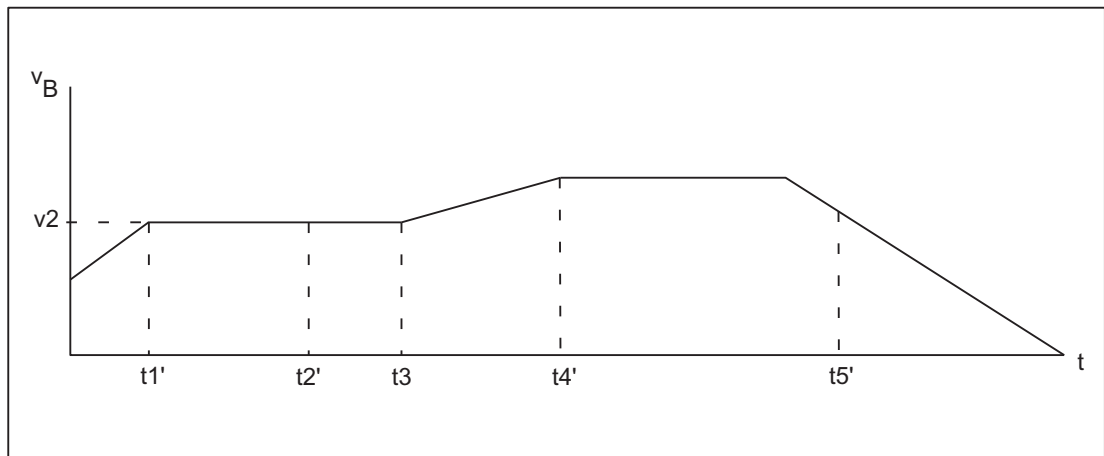


Bild 3-12 Bahngeschwindigkeitsverlauf mit Glättung der Bahngeschwindigkeit und Anpassung der Bahndynamik

Auswirkungen der Glättung der Bahngeschwindigkeit:

Intervall  $t_1 - t_2$ : Der Beschleunigungs- und Bremsvorgang zwischen  $t_1 - t_2$  entfällt, da die Verlängerung der Bearbeitungszeit ohne den Beschleunigungsvorgang auf  $v_{12}$  kleiner als die sich mittels Glättungsfaktor von 80 % ergebende Zeit ist.

Intervall  $t_3 - t_5$ : Der Beschleunigungs- und Bremsverlauf zwischen  $t_3 - t_5$  erfüllt diese Bedingung nicht oder dauert länger als die parametrisierte Glättungszeit  $T_{AXn} = 2/20 \text{ Hz} = 100 \text{ ms}$ .

Auswirkungen der Dynamikanpassung:

Intervall  $t_3 - t_4$ : Der Beschleunigungsvorgang zwischen  $t_3 - t_4$  ist kürzer als  $\text{MIN}(T_{AXn}) = 1/20 \text{ Hz} = 50 \text{ ms}$  und wird daher mit einer um den Anpassfaktor 3 geringeren Beschleunigung durchgeführt.

Intervall bis  $t_1$ : Der nach der Bahnglättung verbleibende Beschleunigungsvorgang bis  $t_1$  wird durch die Dynamikanpassung auf den Zeitraum bis  $t_1'$  gedehnt.

---

#### Hinweis

Das Beispiel zeigt, dass diejenigen Beschleunigungs- oder Bremsvorgänge, die von der Glättung der Bahngeschwindigkeit nicht eliminiert werden, anschließend noch durch die Anpassung der Bahndynamik optimiert werden können. Daher sollten immer möglichst beide Funktionen aktiviert werden.

---



## Beispiel 2

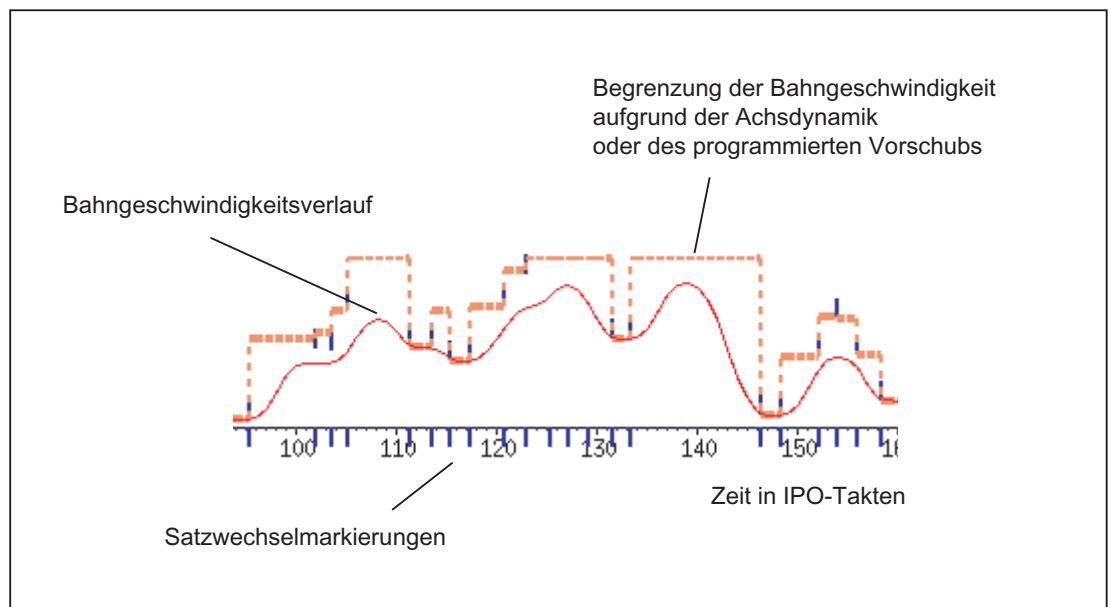
Beschleunigungsmodus: SOFT

An der Bahn sind die 3 Achsen X = AX1, Y = AX2, Z = AX3 beteiligt.

Folgende Parametrierung ist gegeben:

```
MD20465 $MC_ADAPT_PATH_DYNAMIC[1] = 1
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 0.0
MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 10      TAX1 = 1/20 Hz = 100 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX2] = 10      TAX2 = 1/20 Hz = 100 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX3] = 20      TAX3 = 1/20 Hz = 50 ms
```

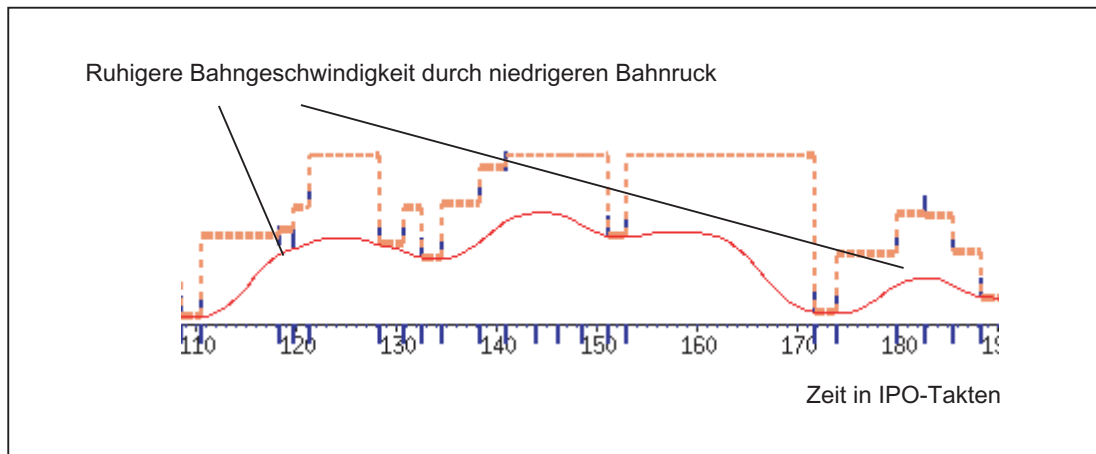
Dies führt zu einem zeitoptimalen Bahngeschwindigkeitsverlauf ohne Glättung der Bahngeschwindigkeit und ohne Anpassung der Bahndynamik:



Die Parametrierung wird wie folgt geändert:

```
MD20465 $MC_ADAPT_PATH_DYNAMIC[1] = 4
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 1.0
```

Daraus ergibt sich ein Bahngeschwindigkeitsverlauf mit Anpassung der Bahndynamik und minimaler und damit fast abgeschalteter Glättung der Bahngeschwindigkeit:

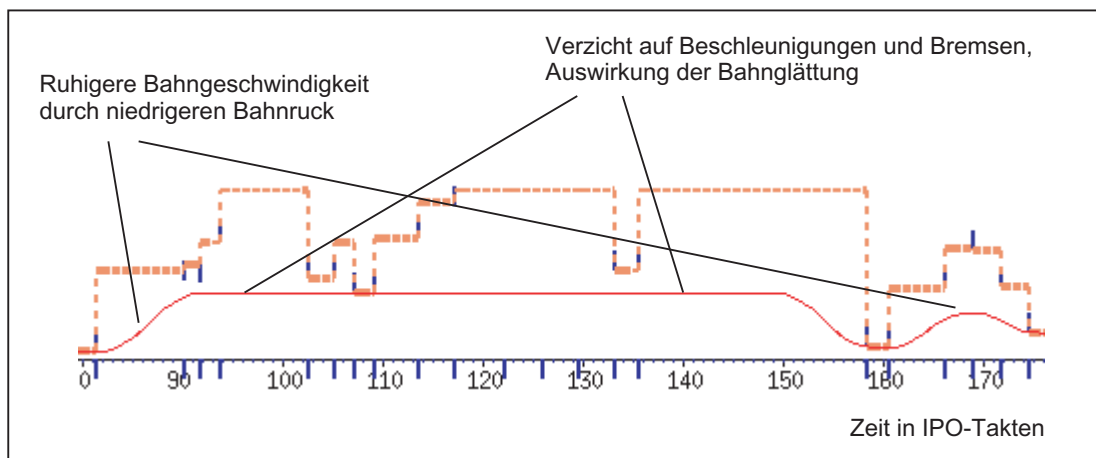


Der Glättungsfaktor wird statt auf 1 % auf 0 % eingestellt (entspricht der Voreinstellung!):

MD20460 \$MC\_LOOKAH\_SMOOTH\_FACTOR = 0.0

Mit dieser Parametrierung wird ein Glättungsfaktor von 100 % wirksam.

Daraus ergibt sich ein Bahngeschwindigkeitsverlauf mit Glättung der Bahngeschwindigkeit und Anpassung der Bahndynamik:



### 3.4.5 Dynamikmodus für Bahninterpolation

#### Funktion

Technologie-spezifische Dynamikeinstellungen können in Maschinendaten hinterlegt und im Teileprogramm über die Befehle der G-Funktionsgruppe 59 (Dynamikmodus für Bahninterpolation) aktiviert werden.

Befehl	Aktiviert die Dynamikeinstellungen für:
DYNNORM	Standard-Dynamikeinstellungen
DYNPOS	Positionierbetrieb, Gewindebohren
DYNROUGH	Schruppen
DYNSEMIFIN	Schlichten
DYNFINISH	Feinschlichten

#### Hinweis

Mit den Befehlen der G-Funktionsgruppe 59 (Dynamikmodus für Bahninterpolation) wird ausschließlich die Dynamik der Bahnachsen bestimmt. Sie haben keinen Einfluss auf:

- Positionierachsen
- PLC-Achsen
- Kommandoachsen
- Bewegungen aufgrund von Achskopplungen
- Überlagerte Bewegungen mit Handrad
- JOG-Bewegungen
- Referenzpunktfahren (G74)
- Festpunktfahren (G75)
- Eilgangbewegung (G0)

Für diese Achsbewegungen ist immer die Standard-Dynamikeinstellung (DYNNORM) wirksam.

#### Anwendung

Durch Umschalten der Dynamikeinstellungen ist es z. B. möglich, ein Werkstück beim Schruppen zeitoptimiert und beim Schlichten oberflächenoptimiert zu bearbeiten.

## Parametrierung

### Axiale Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
Achsspezifische Dynamikeinstellungen		
MD32300	MAX_AX_ACCEL[<n>]	Achsbeschleunigung
MD32431	MAX_AX_JERK[<n>]	Max. axialer Ruck bei Bahnbewegung
MD32432	PATH_TRANS_JERK_LIM[<n>]	Max. axialer Ruck am Satzübergang im Bahnsteuerbetrieb
MD32310	MAX_ACCEL_OVL_FACTOR[<n>]	Überlastfaktor für axiale Geschwindigkeitssprünge
MD32433	SOFT_ACCEL_FACTOR[<n>]	Skalierung der Beschleunigungsbegrenzung bei SOFT
Bahnbezogene Dynamikeinstellungen		
MD20600	MAX_PATH_JERK[<n>]	Bahnbezogener Maximalruck

### Kanalspezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
Bahnbezogene Dynamikeinstellungen		
MD20602	CURV_EFFECT_ON_PATH_ACCEL[<n>]	Einfluss der Bahnkrümmung auf die Bahnbeschleunigung
MD20603	CURV_EFFECT_ON_PATH_JERK[<n>]	Einfluss der Bahnkrümmung auf den Bahnruck

### Wertebereich für den Index n:

Index <n>	Bedeutung
0	Wert für DYNNORM
1	Wert für DYNPOS
2	Wert für DYNROUGH
3	Wert für DYNSEMIFIN
4	Wert für DYNFINISH

### ACHTUNG

Das **Beschreiben** der Maschinendaten **ohne Index** setzt den gleichen Wert in alle Feldelemente des betreffenden Maschinendatums.

Das **Lesen** der Maschinendaten **ohne Index** liefert immer den Wert des Feldes mit dem Index 0.

## G-Befehle unterdrücken

Es wird empfohlen, nicht zur Verwendung vorgesehene G-Befehle der G-Funktionsgruppe 59 (Dynamikmodus für Bahninterpolation) über das folgende Maschinendatum zu unterdrücken:

MD10712 \$MN\_NC\_USER\_CODE\_CONF\_NAME\_TAB[<n>] (Liste umprojektierter NC-Befehle)

Bei Verwendung eines unterdrückten G-Befehls wird, um sicher zu stellen, dass keine nicht parametrisierten Maschinendaten wirksam werden, ein Alarm angezeigt.

### Beispiel

Mit folgenden Einstellungen können die G-Befehle `DYNPOS` und `DYNSEMIFIN` unterdrückt werden:

- MD10712 \$MN\_NC\_USER\_CODE\_CONF\_NAME\_TAB[0]="DYNPOS"
- MD10712 \$MN\_NC\_USER\_CODE\_CONF\_NAME\_TAB[1]=" "
- MD10712 \$MN\_NC\_USER\_CODE\_CONF\_NAME\_TAB[2]="DYNSEMIFIN"
- MD10712 \$MN\_NC\_USER\_CODE\_CONF\_NAME\_TAB[3]=" "

## Literatur

Weitere Informationen zur Programmierung der G-Befehle der G-Funktionsgruppe 59 (Dynamikmodus für Bahninterpolation) finden sich in:

### Literatur:

Programmierhandbuch Grundlagen; Kapitel: Bahnfahrverhalten

### 3.4.6 Freiformflächenmodus: Grundfunktionen

#### Einleitung

Bei Anwendungen im Werkzeug- und Formenbau ist es wichtig, dass möglichst gleichmäßige Oberflächen auf dem Werkstück entstehen. Diese Anforderung ist meistens wichtiger als die Genauigkeit der Werkstückoberfläche.

Ungleichmäßige Werkstückoberflächen können z. B. folgende Ursachen haben:

- Das Teileprogramm zur Fertigung des Werkstücks enthält eine ungleichmäßige Geometrie. Dies betrifft vor allem den Verlauf der Krümmung und der Torsion.

---

#### Hinweis

Die Krümmung  $k$  einer Kontur ist das Inverse des Radius  $r$  des anschmiegenden Kreises in einem Konturpunkt ( $k = 1/r$ ). Die Torsion ist die Änderung der Krümmung (1. Ableitung).

---

Durch die ungleichmäßige Geometrie werden bei der Abarbeitung des Teileprogramms die Dynamikgrenzen der Maschine erreicht und es kommt zu unnötigen Brems- und Beschleunigungsvorgängen. Dies führt je nach Größe des effektiven Nachlaufs der Achsen zu unterschiedlichen Konturabweichungen.

- Unnötige Brems- und Beschleunigungsvorgänge können zur Anregung von Schwingungen der Maschine führen, die als unerwünschte Marken auf dem Werkstück sichtbar werden.

Zur Behebung dieser Ursachen gibt es unterschiedliche Möglichkeiten:

- Die vom CAD-/CAM-System generierten Teileprogramme enthalten einen sehr gleichmäßigen Verlauf der Krümmung und der Torsion, so dass es nicht zu unnötigen Reduzierungen der Bahngeschwindigkeit kommen kann.
- Die Bestimmung der maximalen Bahngeschwindigkeit erfolgt so, dass ungewollte geometrische Schwankungen des Verlaufs der Krümmung und Torsion keinen Einfluss haben.

#### Funktion

Mit der Funktion "Freiformflächenmodus: Grundfunktionen" kann die Festlegung der Begrenzung der Bahngeschwindigkeit unempfindlicher gegenüber kleinen geometrischen Schwankungen der Krümmung bzw. Torsion gemacht werden, ohne dass die dynamischen Begrenzungen der Maschine bezüglich der Beschleunigung und des Rucks der Achsen überschritten werden.

Dadurch ergeben sich folgende Vorteile:

- Gleichmäßigerer Verlauf der Bahngeschwindigkeit
- Gleichmäßigere Oberfläche des Werkstücks
- Verringerung der Bearbeitungszeit (falls die Dynamik der Maschine dies zulässt)

## Anwendungen

Die Funktion wird zur Bearbeitung von Werkstücken verwendet, die vorwiegend aus Freiformflächen bestehen.

## Voraussetzungen

Die Funktion ist nur aktivierbar, wenn bei der Speicherkonfiguration der notwendige Speicher reserviert wird:

MD28610 \$MC\_MM\_PREPDYN\_BLOCKS = 10

Der eingetragene Wert gibt die Anzahl von Sätzen vor, die bei der Festlegung der Bahngeschwindigkeit (Geschwindigkeitspräparation) berücksichtigt werden sollen. Ein sinnvoller Wert ist "10".

Hat MD28610 den Wert "0", dann werden zur Festlegung der maximalen Bahngeschwindigkeit eines Satzes nur die jeweiligen Bewegungen der Achsen in diesem Satz berücksichtigt. Wird bei der Festlegung der Bahngeschwindigkeit auch die Geometrie in benachbarten Sätzen berücksichtigt (Wert > 0), erhält man einen gleichmäßigeren Verlauf der Bahngeschwindigkeit.

## Aktivierung / Deaktivierung

Die Funktion kann für jeden Dynamikmodus (siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 219)") unabhängig ein- oder ausgeschaltet werden:

MD20606 \$MC\_PREPDYN\_SMOOTHING\_ON[<n>] = <Wert>

Index <n>	Dynamikmodus	<Wert>	Freiformflächenmodus: Grundfunktionen
0	Standard-Dynamikeinstellungen (DYNNORM)	0	aus
		1	ein
1	Positionierbetrieb, Gewindebohren (DYNPOS)	0	aus
		1	ein
2	Schruppen (DYNROUGH)	0	aus
		1	ein
3	Schlichten (DYNSEMIFIN)	0	aus
		1	ein
4	Feinschlichten (DYNFINISH)	0	aus
		1	ein

## Hinweis

Die Funktion sollte wegen dem zusätzlichen Speicherbedarf nur in den relevanten Bearbeitungskanälen aktiviert werden.

## Parametrierung

### Veränderung des Konturabtastfaktors

Bei der Interpolation von gekrümmten Konturen ist der dabei entstehende Sekundenfehler von folgenden Faktoren abhängig:

- Krümmung
- Interpolationstakt (Anzeige im MD10071 \$MN\_IPO\_CYCLE\_TIME)
- Geschwindigkeit, mit der die betreffende Kontur abgefahren wird

Der maximal mögliche Sekundenfehler ist für jede Achse festgelegt im Maschinendatum:

MD33100 \$MA\_COMPRESS\_POS\_TOL (Maximale Abweichung bei Kompression)

Wenn der eingestellte Interpolationstakt nicht ausreichend klein ist, kann es vorkommen, dass bei stärker gekrümmten Konturen die maximale Bahngeschwindigkeit reduziert wird. Dies ist notwendig, damit auch in diesem Fall die Oberfläche des Werkstücks noch ausreichend genau hergestellt wird.

Durch die Veränderung des Konturabtastfaktors ist es möglich, das Zeitintervall, mit dem eine gekrümmte Kontur im Interpolator abgetastet wird (Konturabtastzeit), unterschiedlich zum Interpolationstakt einzustellen. Eine Konturabtastzeit niedriger als der Interpolationstakt kann so bei stärker gekrümmten Konturen eine Reduzierung der Bahngeschwindigkeit verhindern.

Der Konturabtastfaktor wird eingestellt mit dem Maschinendatum:

MD10682 \$MN\_CONTOUR\_SAMPLING\_FACTOR

Die wirksame Konturabtastzeit berechnet sich daraus wie folgt:

$$T_s = f * T_1$$

mit:  $T_s$  = wirksame Konturabtastzeit  
 $T_1$  = Interpolationstakt  
 $f$  = Kontur-Abtastfaktor (Wert aus MD10682)

Der Standardwert des Konturabtastfaktors ist "1", d. h. die Konturabtastzeit ist gleich dem Interpolationstakt.

Der Konturabtastfaktor kann sowohl größer als auch kleiner "1" sein.

Wird ein Wert kleiner "1" eingestellt, kann man erreichen, dass die Überprüfung der Genauigkeit der Konturabtastung außer Kraft gesetzt wird.

Die eingestellte Abtastzeit darf die projektierte minimale Konturabtastzeit nicht unterschreiten:

MD10680 \$MN\_MIN\_CONTOUR\_SAMPLING\_TIME

---

### Hinweis

MD10680 ist für jedes Steuerungsmodell spezifisch eingestellt und nicht veränderbar.

---



## Programmierung

Abhängig von der Einstellung im Maschinendatum  
MD20606 \$MC\_PREPDYN\_SMOOTHING\_ON kann die Funktion "Freiformflächenmodus:  
Grundfunktionen" im Teileprogramm durch einen Wechsel des aktiven Dynamikmodus ein-  
oder ausgeschaltet werden.

Beispiel:

Durch die Parametrierung von MD20606 \$MC\_PREPDYN\_SMOOTHING\_ON[2-4] = 1 und  
MD20606 \$MC\_PREPDYN\_SMOOTHING\_ON[0-1] = 0 kann die Funktion über die Befehle  
DYNROUGH, DYNSEMIFIN und DYNFINISH eingeschaltet und über die Befehle DYNNORM und DYNPOS  
ausgeschaltet werden.

## Siehe auch

Überschleifen tangentialer Satzübergänge (G645) (Seite 193)

Geschwindigkeitsabhängige Ruckadaption (achsspezifisch) (Seite 273)

Freiformflächenmodus: Erweiterungsfunktion (Seite 201)

## 3.5 Kompressor-Funktionen

### 3.5.1 NC-Satz-Kompression

#### Funktion

#### COMPON, COMPCURV

Die Kompressor-Funktionen `COMPON` und `COMPCURV` erzeugen aus bis zu 10 aufeinander folgenden Linearsätzen der Form: "G01 X... Y... Z... F..." einen Polynom-Satz. Die Polynom-Sätze der Kompressor-Funktionen haben folgende Eigenschaften:

- `COMPON`: Geschwindigkeits-stetige Satzübergänge
- `COMPCURV`: Geschwindigkeits- und Beschleunigungs-stetige Satzübergänge

#### COMPCAD

Die Kompressor-Funktionen `COMPCAD` kann aus theoretisch beliebig vielen Linear- und Kreissätzen einen Polynom-Satz erzeugen. Die Polynom-Sätze sind an den Satzübergängen Geschwindigkeits- und Beschleunigungs-stetig. Gewollte Ecken werden als solche erkannt und berücksichtigt.

Die maximal tolerierbare Abweichung der berechneten Bahn zu den programmierten Punkten kann für alle Kompressor-Funktionen durch Maschinendaten vorgegeben werden. Anders als bei `COMPON` und `COMPCURV` werden die vorgegebenen Toleranzen bei `COMPCAD` in benachbarten Bahnen nicht in unterschiedlichen Richtungen ausgenutzt. `COMPCAD` versucht vielmehr unter ähnlichen Bedingungen auch ähnliche Abweichungen von den programmierten Punkten zu erreichen.

Gemeinsames Ziel der Kompressor-Funktionen ist eine Optimierung der Oberflächengüte und Bearbeitungsgeschwindigkeit durch stetige Satzübergänge und Vergrößerung der Bahnlänge pro Satz.

`COMPCAD` ist sehr rechenzeit- und speicherplatzintensiv. Es wird empfohlen `COMPCAD` nur dort einzusetzen, wo Oberflächenverbesserung durch Maßnahmen im CAD/CAM-Programm nicht zum Erfolg führen.

#### Allgemein

- Die Positionsangaben in den zu komprimierenden Sätzen können beliebig erfolgen z.B.  $X100, X=AC(100), X=R1*(R2+R3)$
- Der Kompressionsvorgang wird durch jeden anderen Befehl, z. B. Hilfsfunktionsausgabe, in und zwischen den zu komprimierenden Sätzen unterbrochen.

#### Verfügbarkeit

Die NC-Satz-Kompression steht bei SINUMERIK 828D nur für die Fräsvarianten zur Verfügung.

## Orientierungstransformation (TRAORI)

Die Kompressor-Funktionen `COMPON`, `COMPCURV` und `COMPCAD` können bei aktiver Orientierungstransformation (TRAORI) unter bestimmten Voraussetzungen auch Bewegungssätze zur Werkzeugorientierung und Werkzeugdrehung komprimieren. Eine ausführliche Beschreibung findet sich in:

### Literatur:

Funktionshandbuch Sonderfunktionen; 3- bis 5-Achs-Transformation (F2),  
Kapitel: "Komprimierung der Orientierung (`COMPON`, `COMPCURV` und `COMPCAD`)"

## Parametrierung

Zur Parametrierung der NC-Satz-Kompression sind folgende Maschinen- und Settingdaten einzustellen:

### Kanalspezifische Maschinendaten

Nummer	Bezeichner \$MC_	Bedeutung
MD20170	COMPRESS_BLOCK_PATH_LIMIT	Maximale Verfahrlänge eines NC-Satzes bei Kompression
MD20172	COMPRESS_VELO_TOL	Maximal erlaubte Abweichung des Bahnvorschubs bei Kompression
MD20482	COMPRESSOR_MODE	Einstellung der Arbeitsweise des Kompressors. Siehe auch: <b>Literatur:</b> Funktionshandbuch Sonderfunktionen; Mehrachstransformationen (F2), Kapitel: "Orientierung" > "Komprimierung der Orientierung"

### Kanalspezifische Settingdaten

Nummer	Bezeichner \$SC_	Bedeutung
SD42470	CRIT_SPLINE_ANGLE	Ecken-Grenzwinkel für <code>COMPCAD</code>
SD42475	COMPRESS_CONTUR_TOL	Maximal erlaubte Konturabweichung bei Kompression

### Hinweis

#### Ecken-Grenzwinkel und Kompressorfunktion `COMPCAD`

Der über das Settingdatum SD42470 `$SC_CRIT_SPLINE_ANGLE` eingestellte Ecken-Grenzwinkel für `COMPCAD` dient nur als ungefähres Maß zur Eckenerkennung. Der Kompressor erkennt aufgrund von Plausibilitätsbetrachtungen auch flachere Satzübergänge als Ecken und größere Winkel als Ausreißer.

**Axiale Maschinendaten**

Nummer	Bezeichner \$MA_	Bedeutung
MD33100	COMPRESS_POS_TOL	Maximal erlaubte Bahnabweichung bei Kompression

**Einstellempfehlungen für rückwirkende Maschinendaten**

Bei Verwendung der Kompressor-Funktion werden für die nachfolgenden auf die Kompressor-Funktion rückwirkenden Maschinendaten folgende Einstellungen empfohlen:

Nummer	Bezeichner	Empfohlener Wert
MD18360	\$MN_MM_EXT_PROG_BUFFER_SIZE (FIFO-Buffer Größe für Abarbeiten von Extern)	100
MD20490	\$MC_IGNORE_OVL_FACTOR_FOR_ADIS (G641/G642 unabhängig vom Overload-Faktor)	1
MD28520	\$MC_MM_MAX_AXISPOLY_PER_BLOCK (Maximale Anzahl Achspolynome pro Satz)	3
MD28530	\$MC_MM_PATH_VELO_SEGMENTS (Anzahl Speicherelemente z. Begrenzung der Bahngeschwindigkeit)	5
MD28540	\$MC_MM_ARCLENGTH_SEGMENTS (Anzahl Speicherelemente zur Darstellung der Bogenlängenfunktion)	10
MD28060	\$MC_MM_IPO_BUFFER_SIZE (Anzahl der NC-Sätze für die Satzaufbereitung)	100
MD28070	\$MC_MM_NUM_BLOCKS_IN_PREP (Anzahl der Sätze für die Satzaufbereitung)	60
MD32310	\$MA_MAX_ACCEL_OVL_FACTOR (Überlastfaktor für axiale Geschwindigkeitssprünge)	<Wert für G64-Betrieb>

**Programmierung**

**Einschalten**

Die Kompressor-Funktionen werden durch die modalen Befehle `COMPON`, `COMPCURV` bzw. `COMPCAD` eingeschaltet.

Zur weiteren Verbesserung der Oberflächengüte können zusätzlich die Funktionen `G642` (Überschleiffunktion) und `SOFT` (Ruckbegrenzung) verwendet werden. Die Befehle sind gemeinsam am Programmanfang zu schreiben.

Beispiel:

Programmcode	Kommentar
PROC ...	
N10 COMPCAD SOFT G642	; Kompressor COMPCAD einschalten
N20 G01 X... Y... Z... F...	; Verfahrsätze 1 ... n
...	
N1000 COMPOF	; Kompressor ausschalten
N1010 RET	

### Ausschalten

Ausgeschaltet werden alle Kompressor-Funktionen gemeinsam durch den Befehl `COMPOF`.

### Literatur

Die Programmierung von Kompressor-Funktionen ist beschrieben in:  
Programmierhandbuch Arbeitsvorbereitung

Der Einsatz der Kompressor-Funktion bei aktiver Orientierungstransformation ist beschrieben in:  
Funktionshandbuch Sonderfunktionen; Mehrachstransformationen (F2),  
Kapitel: Komprimierung der Orientierung

### Siehe auch

Toleranz und Komprimierung von G0-Sätzen (Seite 236)

## 3.5.2 Zusammenfassung kurzer Spline-Sätze

### Funktion

Bei der Aufbereitung von Spline-Sätzen können zwischen Sätzen mit großen Satzlängen immer wieder Sätze mit kurzen Satzlängen entstehen. Diese kann dann dazu führen, dass die Bahngeschwindigkeit immer wieder signifikant vor diesen kurzen Sätzen reduziert werden muss.

Mit der Funktion "Zusammenfassung kurzer Spline-Sätze" werden die Spline-Sätze so aufbereitet, dass Sätze mit kurzen Satzlängen vermieden werden und dadurch gleichmäßig mit hoher Bahngeschwindigkeit verfahren werden kann.

---

### Hinweis

#### NC-Satz-Kompressor

Der NC-Satz-Kompressor (`COMPON`, `COMPCURV` oder `COMPCAD`) ist bei der Komprimierung von Spline-Sätzen nicht anwendbar, da mit diesem nur Linearsätze komprimiert werden können.

---

### Verfügbarkeit

#### System

SINUMERIK 840D sl  
SINUMERIK 828D

#### Verfügbarkeit

Standard (Grundumfang)  
Option

### Aktivierung

Die Funktion "Zusammenfassung kurzer Spline-Sätze" kann für folgende Spline-Arten aktiviert werden:

- BSPLINE
- BSPLINE / ORICURVE
- CSPLINE

Die Aktivierung erfolgt über das Maschinendatum:

MD20488 \$MC\_SPLINE\_MODE, Bit <n> = <Wert> (Einstellung für Spline-Interpolation)

Bit	<Wert>	Bedeutung: Funktion "Zusammenfassung kurzer Spline-Sätze" ...
0	0	bei BSPLINE nicht aktiv
	1	bei BSPLINE aktiv
1	0	bei BSPLINE/ORICURVE nicht aktiv
	1	bei BSPLINE/ORICURVE aktiv
2	0	bei CSPLINE nicht aktiv
	1	bei CSPLINE aktiv

### Randbedingungen

- Spline-Sätze können nur zusammengefasst werden, wenn außer Verfahrbewegungen und Vorschub keine weiteren Funktionen programmiert sind. Bei z. B. Hilfsfunktionen, die an PLC ausgegeben werden, kann der Spline-Satz nicht zusammengefasst werden..
- Die maximale Anzahl hintereinander liegender Spline-Sätzen, die in einem Programmabschnitt zusammengefasst werden können, hängt von der Größe des Speichers für Sätze in der Satzaufbereitung ab:

MD28070 \$MC\_MM\_NUM\_BLOCKS\_IN\_PREP (Anzahl der Sätze für die Satzaufbereitung)

## Beispiel

Um beim Abarbeiten des folgenden Programms eine höhere Bahngeschwindigkeit zu erreichen, wird die Funktion "Zusammenfassung kurzer Spline-Sätze" für die BSPLINE-Interpolation aktiviert:

MD20488 \$MC\_SPLINE\_MODE, Bit 0 = 1

Programmcode	Kommentar
PROC P1	
N10 G1 G64 X0 Y0 Z0 F1000	
N20 G91 F10000 BSPLINE	
	; Ab hier BSPLINE-Interpolation mit Zusammenfassung kurzer Spline-Sätze
N30 X0.001 Y0.001 Z0.001	
N40 X0.001 Y0.001 Z0.001	
N50 X0.001 Y0.001 Z0.001	
N60 X0.001 Y0.001 Z0.001	
N70 X0.001 Y0.001 Z0.001	
N80 X0.001 Y0.001 Z0.001	
...	
N1000 M30	

## 3.6 Kontur-/Orientierungstoleranz

### Parametrierung der Kontur-/Orientierungstoleranz

Die maximal erlaubte Konturabweichung (Konturtoleranz) und die maximal erlaubte Winkelabweichung der Werkzeugorientierung (Orientierungstoleranz) sind für jede Achse festgelegt im Maschinendatum:

MD33100 \$MA\_COMPRESS\_POS\_TOL (Maximale Abweichung bei Kompression)

Der eingestellte Wert hat Gültigkeit sowohl für die Kompressor-Funktionen als auch für die Überschleif-Funktionen mit Ausnahme von G641 (dort gilt eine mit ADIS/ADISPOS programmierte Wegdistanz zum Satzübergang).

Anstelle von MD33100 können auch die Toleranzwerte aus den folgenden Settingdaten wirksam werden:

SD42465 \$SC\_SMOOTH\_CONTUR\_TOL (Maximale Konturabweichung)

SD42466 \$SC\_SMOOTH\_ORI\_TOL (Maximale Winkelabweichung der Werkzeugorientierung)

Die Art und Weise, wie die Toleranzwerte aus MD33100 und den Settingdaten SD42465 und SD42466 berücksichtigt werden sollen, wird:

- Für die Überschleif-Funktionen eingestellt über die Dezimalstellen im Maschinendatum:  
MD20480 \$MC\_SMOOTHING\_MODE (Verhalten des Überschleifens mit G64x)
- Für die Kompressor-Funktionen eingestellt über die Einerstelle im Maschinendatum:  
MD20482 \$MC\_COMPRESSOR\_MODE (Modus des Kompressors)

Weitere Maschinen- und Settingdaten, die bei der Einstellung der Kontur- und Orientierungstoleranz eine Rolle spielen, sind:

- MD33120 \$MA\_PATH\_TRANS\_POS\_TOL (Max. Abweichung beim Überschleifen mit G645)

Der Wert aus MD33120 ist wirksam beim Überschleifen von tangentialen-, aber nicht krümmungsstetigen Satzübergängen (z. B. Kreis - Gerade) mit G645.

- SD42676 \$SC\_ORI\_SMOOTH\_TOL

Dieses Settingdatum bestimmt die Toleranz beim Überschleifen der Orientierung mit OST.

- SD42678 \$SC\_ORISON\_TOL

Dieses Settingdatum bestimmt die Toleranz beim Glätten der Orientierung mit ORISON.

### Programmierung der Kontur-/Orientierungstoleranz

Bei Programmstart haben die beschriebenen Maschinen- und Settingdaten Gültigkeit und bestimmen die Toleranzen für alle Kompressor-Funktionen, die Überschleif-Funktionen G642, G643, G645, OST und die Orientierungsglättung ORISON.



Das NC-Programm kann diese parametrisierten Toleranzen aber überschreiben. Dazu stehen dem NC-Programmierer folgende Befehle zur Verfügung:

Befehl	Syntax	Bedeutung
CTOL	CTOL=<Wert>	Konturtoleranz
OTOL	OTOL=<Wert>	Orientierungstoleranz
ATOL [<Achse>]	ATOL [<Achse>]=<Wert>	Achsspezifische Toleranz

CTOL und OTOL haben dabei Vorrang vor ATOL.

Die Programmierung löst keinen Vorlaufstopp aus. Wenn möglich, unterbricht sie auch nicht die NC-Satz-Kompression.

Die programmierten Werte gelten, bis sie neu programmiert oder durch Beschreiben mit einem negativen Wert gelöscht werden. Sie werden ferner gelöscht bei Programmende, Kanal-Reset, BAG-Reset, NCK-Reset (Warmstart) und POWER ON (Kaltstart). Nach dem Löschen gelten wieder die Werte aus den Maschinen- und Settingdaten.

Neue Werte können in jedem Satz programmiert und wirksam werden.

---

#### Hinweis

Die programmierte Toleranz wirkt auch auf Funktionen, welche nur implizit von der Toleranz abhängen. Aktuell sind das:

- Die Begrenzung des Sehnenfehlers bei der Sollwertberechnung
  - Die Grundfunktionen des Freiformflächenmodus
- 

#### Hinweis

Folgende Überschleif-Funktionen werden durch die Programmierung von CTOL, OTOL und ATOL nicht beeinflusst:

- Überschleifen der Orientierung mit OSD  
Grund: OSD verwendet keine Toleranz, sondern eine Distanz zum Satzübergang.
  - Überschleifen mit G644  
Grund: G644 dient nicht der Bearbeitung, sondern der Optimierung von Werkzeugwechseln und anderen Bewegungen in Luft
  - Überschleifen von tangentialen-, aber nicht krümmungsstetigen Satzübergängen mit G645  
G645 verhält sich fast immer wie G642 und verwendet somit die programmierten Toleranzen. Nur an tangentialstetigen Satzübergängen mit einem Krümmungssprung, z. B. einem tangentialen Kreis-Gerade-Übergang, wird der Toleranzwert aus dem Maschinendatum MD33120 \$MA\_PATH\_TRANS\_POS\_TOL verwendet. Der Überschleifweg kann sich an diesen Stellen nämlich auch auf der Außenseite der programmierten Kontur befinden, wo viele Anwendungen weniger tolerant sind. Außerdem genügt in der Regel eine kleine, fest eingestellte Toleranz zum Ausgleich der Krümmungsänderungen, über die sich der NC-Programmierer keine Gedanken machen muss.
-

### Toleranzwerte lesen

Für weitergehende Anwendungsfälle oder zur Diagnose sind die aktuell gültigen Toleranzen für die Kompressor-Funktionen (COMPON, COMPCURV, COMPCAD), die Überschleifarten G642, G643, G645, OST und die Orientierungsglättung ORISON unabhängig von der Art des Zustandekommens über Systemvariablen lesbar.

- Für die Anzeige in der Bedienoberfläche, in Synchronaktionen oder mit Vorlaufstopp im Teileprogramm über die Systemvariablen:

\$AC_CTOL	Konturtoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war Falls keine Konturtoleranz wirksam ist, liefert \$AC_CTOL die Wurzel aus der Summe der Quadrate der Toleranzen der Geometrieachsen.
\$AC_OTOL	Orientierungstoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war Falls keine Orientierungstoleranz wirksam ist, liefert \$AC_OTOL während einer aktiven Orientierungstransformation die Wurzel aus der Summe der Quadrate der Toleranzen der Orientierungsachsen, ansonsten den Wert "-1".
\$AA_ATOL[<Achse>]	Achstoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war Falls eine Konturtoleranz aktiv ist, liefert \$AA_ATOL[<Geometrieachse>] die Konturtoleranz geteilt durch die Wurzel aus der Anzahl der Geometrieachsen. Falls eine Orientierungstoleranz und eine Orientierungstransformation aktiv sind, liefert \$AA_ATOL[<Orientierungsachse>] die Orientierungstoleranz geteilt durch die Wurzel aus der Anzahl der Orientierungsachsen.

---

#### Hinweis

Wenn keine Toleranzwerte programmiert wurden, dann sind die \$A-Variablen nicht differenziert genug, um die möglicherweise verschiedenen Toleranzen der einzelnen Funktionen zu unterscheiden, da sie ja nur einen Wert nennen können.

Solche Fälle können auftreten, wenn die Maschinen- und Settingdaten unterschiedliche Toleranzen für Kompressor-Funktionen, Überschleifen und Orientierungsglättung einstellen. Die Variablen liefern dann den größten Wert, der bei den gerade aktiven Funktionen auftritt.

Ist z. B. eine Kompressor-Funktion mit Orientierungstoleranz 0,1° und eine Orientierungsglättung ORISON mit 1° aktiv, liefert die Variable \$AC\_OTOL den Wert "1". Wird die Orientierungsglättung ausgeschaltet, liest man nur noch den Wert "0.1".

---

- Ohne Vorlaufstopp im Teileprogramm über die Systemvariablen:

\$P_CTOL	Programmierte Konturtoleranz
\$P_OTOL	Programmierte Orientierungstoleranz
\$PA_ATOL	Programmierte Achstoleranz

---

**Hinweis**

Wenn keine Toleranzwerte programmiert sind, dann liefern die \$P-Variablen den Wert "-1".

---

## 3.7 Toleranz und Komprimierung von G0-Sätzen

### Funktion

Die Funktion "Toleranz und Komprimierung von G0-Sätzen" ermöglicht ein schnelleres Abfahren von Eilgang-Bewegungen.

Sie besteht aus folgenden Komponenten:

1. **Projektierung/Programmierung eines unabhängigen Toleranzfaktors für G0-Bewegungen**

Mit diesem Faktor können die Toleranzen bei G0-Bewegungen unterschiedlich zu den Werkstückbearbeitungstoleranzen eingestellt werden.

2. **Kompression von G0-Sätzen**

Ist diese Funktionalität angewählt, dann werden bei aktiver NC-Satz-Kompression außer den Verfahransätzen mit G1 (Geradeninterpolation) auch Verfahransätze mit G0 (Eilgang) komprimiert.

### Wirksamkeit

1. Der G0-Toleranzfaktor wird nur wirksam, wenn:

- eine der folgenden Funktionen aktiv ist:
  - Kompressorfunktionen: COMPON, COMPCURV und COMPCAD
  - Überschleiffunktionen: G642 und G645
  - Orientierungsüberschleifen: OST
  - Orientierungsglättung: ORISON
  - Glättung bei bahnrelativer Orientierung: ORIPATH

• im Teileprogramm mehrere ( $\geq 2$ ) G0-Sätze aufeinanderfolgen.

Bei einem einzelnen G0-Satz wird der G0-Toleranzfaktor nicht wirksam, da **beim Übergang** von einer Nicht-G0-Bewegung zu einer G0-Bewegung (und umgekehrt) grundsätzlich die "**kleinere Toleranz**" (Werkstückbearbeitungstoleranz) wirkt!

2. Die Kompression von G0-Sätzen wird wirksam:

- Bei aktiver NC-Satz-Kompression (COMPON, COMPCURV oder COMPCAD)

### Projektierung

#### G0-Toleranzfaktor

Der G0-Toleranzfaktor wird kanalspezifisch eingestellt mit dem Maschinendatum:

MD20560 \$MC\_G0\_TOLERANCE\_FACTOR (Toleranz-Faktor für G0)

Der G0-Toleranzfaktor kann sowohl größer als auch kleiner 1.0 sein. Ist der Faktor gleich 1.0 (Standardwert), sind für G0-Sätze dieselben Toleranzen wirksam wie für Nicht-G0-Sätze. Im Normalfall wird der G0-Toleranzfaktor  $\geq 1.0$  eingestellt sein.

### Kompression von G0-Sätzen

Die Kompression von G0-Sätzen wird kanalspezifisch eingestellt über die **Hunderterstelle** im Maschinendatum:

MD20482 \$MC\_COMPRESSOR\_MODE (Modus des Kompressors)

Wert	Bedeutung
0xx	Kreissätze und G0-Sätze werden <b>nicht</b> komprimiert.
1xx	Kreissätze werden komprimiert. Nur COMPCAD.
2xx	G0-Sätze werden komprimiert. Siehe auch MD20560 \$MC_G0_TOLERANCE_FACTOR bzw. NC-Befehl STOLF
3xx	Kreissätze und G0-Sätze werden komprimiert.

Eine ausführlichere Beschreibung zu MD20482 findet sich in:

**Literatur:**

Funktionshandbuch Sonderfunktionen; Mehrachstransformationen (F2),  
 Kapitel: Komprimierung der Orientierung

### Programmierung

Der mit MD20560 \$MC\_G0\_TOLERANCE\_FACTOR eingestellte Toleranzfaktor kann durch Programmierung von STOLF im Teileprogramm temporär überschrieben werden:

**Syntax:** STOLF=<...>

**Beispiel:**

Programmcode	Kommentar
COMPCAD G645 G1 F10000	; Kompressor-Funktion COMPCAD
X... Y... Z...	; Hier wirken die Maschinen- und Settingdaten.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
G0 X... Y... Z...	; Hier wirkt das Maschinendatum \$MC_G0_TOLERANCE_FACTOR (z.B. =3), also eine Überschleiftoleranz von \$MC_G0_TOLERANCE_FACTOR*\$MA_COMPRESS_POS_TOL.
CTOL=0.02	
STOLF=4	
G1 X... Y... Z...	; Ab hier wirkt eine Konturtoleranz von 0,02mm.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
X... Y... Z...	; Ab hier wirkt ein G0-Toleranzfaktor von 4, also eine Konturtoleranz von 0,08mm.

Durch die Programmierung des Toleranzfaktors wird der Wert im MD20560 nicht verändert. Nach Reset bzw. Teileprogrammende wird der durch MD20560 eingestellte Wert wieder wirksam.

### Toleranzfaktor lesen

Der im Teileprogramm bzw. im aktuellen IPO-Satz wirksame G0-Toleranzfaktor ist über Systemvariablen lesbar.

- Für die Anzeige in der Bedienoberfläche, in Synchronaktionen oder mit Vorlaufstopp im Teileprogramm über die Systemvariable:

\$AC_STOLF	Aktiver G0-Toleranzfaktor
	G0-Toleranzfaktor, der bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.

- Ohne Vorlaufstopp im Teileprogramm über die Systemvariable:

\$P_STOLF	Programmierter G0-Toleranzfaktor
-----------	----------------------------------

Ist im aktiven Teileprogramm kein Wert mit `STOLF` programmiert, dann liefern diese beiden Systemvariablen den durch MD20560 `$MC_G0_TOLERANCE_FACTOR` eingestellten Wert.

Ist in einem Satz kein Eilgang (G0) aktiv, dann liefern diese Systemvariablen immer den Wert 1.

## 3.8 RESET-Verhalten

### MD20150

Durch RESET wird für G-Funktionsgruppen die kanalspezifisch parametrisierte Grundstellung wirksam:

MD20150 \$MC\_GCODE\_RESET\_VALUES (Löschstellung der G-Gruppen)

Bezüglich "Bahnsteuerbetrieb, Genauhalt, LookAhead" sind folgende G-Funktionsgruppen relevant:

- Gruppe 10: Genauhalt - Bahnsteuerbetrieb
- Gruppe 12: Satzwechselkriterium bei Genauhalt
- Gruppe 21: Beschleunigungsprofil
- Gruppe 30: NC-Satz-Kompression
- Gruppe 59: Dynamikmodus für Bahninterpolation

Ausführliche Informationen zur Einstellung der Grundstellungen siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)".

## 3.9 Randbedingungen

### 3.9.1 Satzwechsel und Positionierachsen

Werden in einem Teileprogramm Bahnachsen im Bahnsteuerbetrieb verfahren, können parallel dazu verfahrenende Positionierachsen sowohl das Verhalten der Bahnachsen als auch den Satzwechsel beeinflussen.

Eine ausführliche Beschreibung der Positionierachsen findet sich in:

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

### 3.9.2 Satzwechselverzögerung

Auch wenn für alle im Teileprogrammsatz verfahrenenden Bahn- und Zusatzachsen die jeweils spezifischen Satzwechselkriterien erfüllt sind, kann es aufgrund anderweitiger nicht erfüllter Bedingungen und / oder aktiver Funktionen zu zeitlichen Verzögerungen des Satzwechsels kommen.

**Beispiele:**

- Fehlende Hilfsfunktionsquittierung durch die PLC
- Nicht vorhandene Folgesätze
- Aktive Funktion "Zwischenspeicher leeren"

**Auswirkungen**

Wenn während des Bahnsteuerbetriebs ein Satzwechsel nicht durchgeführt werden kann, dann werden alle in diesem Teileprogrammsatz programmierten Achsen (außer satzübergreifend verfahrenende Zusatzachsen) angehalten. Dabei treten keine Konturfehler auf.

Durch das Anhalten der Bahnachsen **während der Bearbeitung** kann es zum Entstehen von Freischneidmarken auf der Werkstückoberfläche kommen.



### 3.9.3 Überschleifen und Repositionieren (REPOS)

#### Repositionieren innerhalb des Überschleifbereichs

Wird bei Verfahrtsätzen mit programmiertem Überschleifen (Teileprogrammbehl G641, G642, G643, G644 oder G645) die Verfahrbewegung der Bahnachsen innerhalb des Überschleifbereichs unterbrochen, wird bei einem nachfolgenden REPOS-Vorgang abhängig vom aktuellen REPOS-Modus folgendermaßen repositioniert:

REPOS-Modus	REPOS-Zielpunkt
RMBBL	Satzanfang des unterbrochenen Verfahrtsatzes
RMIBL	Satzende des unterbrochenen Verfahrtsatzes
RMEBL	Satzende des unterbrochenen Verfahrtsatzes
RMNBL	Satzende des unterbrochenen Verfahrtsatzes

#### Beispiel

Zwei Verfahrtsätze N10 und N20 mit programmiertem Überschleifen G641. Im Überschleifbereich wird die Verfahrbewegung unterbrochen und anschließend die Achsen z. B. manuell auf den REPOS-Startpunkt verfahren. Abhängig vom aktiven REPOS-Modus wird unterschiedlich an die Kontur repositioniert.

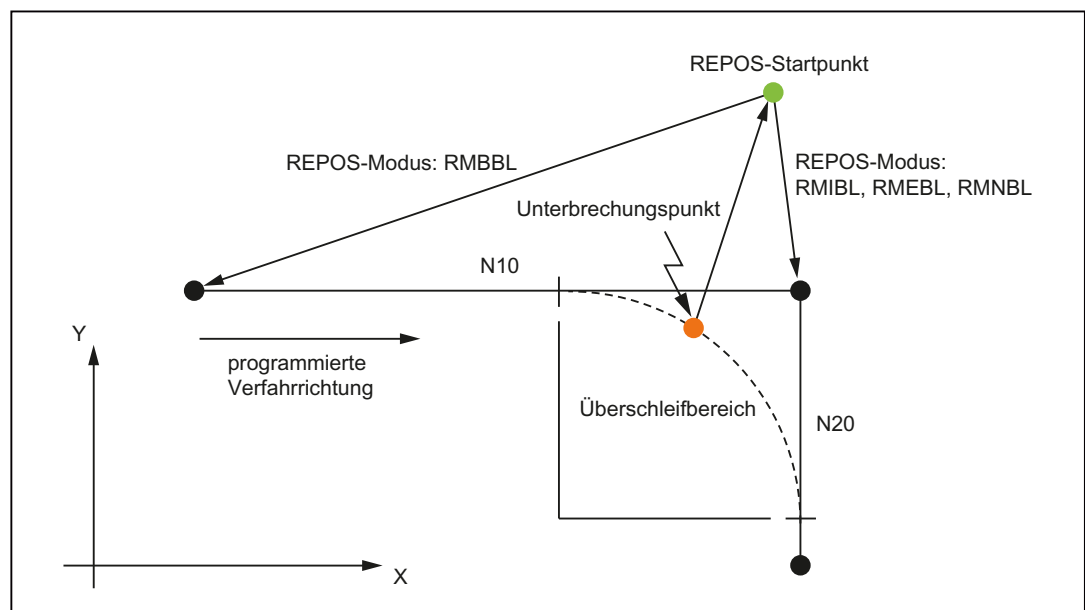


Bild 3-13 Beispiel für Überschleifen und Repositionieren

## 3.10 Datenlisten

### 3.10.1 Maschinendaten

#### 3.10.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10110	PLC_CYCLE_TIME_AVERAGE	Mittlere PLC-Quittierungszeit
10680	MIN_CONTOUR_SAMPLING_TIME	Minimale Kontur-Abtastzeit
10682	CONTOUR_SAMPLING_FACTOR	Konturabtastfaktor
10712	NC_USER_CODE_CONF_NAME_TAB	Liste umprojektierter NC-Befehle
12030	OVR_FACTOR_FEEDRATE	Bewertung des Bahnvorschub-Korrekturschalters
12100	OVR_FACTOR_LIMIT_BIN	Begrenzung bei binärkodiertem Korrektorschalter
18360	MM_EXT_PROG_BUFFER_SIZE	FIFO-Buffer Größe für Abarbeiten von extern (DRAM)

#### 3.10.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20150	GCODE_RESET_VALUES	Löschstellung der Gruppen
20170	COMPRESS_BLOCK_PATH_LIMIT	Maximale Verfahrnlänge eines NC-Satzes bei Kompression
20172	COMPRESS_VELO_TOL	Maximal erlaubte Abweichung des Bahnvorschubs bei Kompression
20400	LOOKAH_USE_VELO_NEXT_BLOCK	LookAhead Folgesatzgeschwindigkeit
20430	LOOKAH_NUM_OVR_POINTS	Anzahl der Korrektorschalter-Eckwerte bei LookAhead
20440	LOOKAH_OVR_POINTS	Korrektorschalter-Eckwerte bei LookAhead
20443	LOOKAH_FFORM	Aktivierung des erweiterten LookAhead
20450	LOOKAH_RELIEVE_BLOCK_CYCLE	Entlastungsfaktor für die Blockzykluszeit
20460	LOOKAH_SMOOTH_FACTOR	Glättungsfaktor bei LookAhead
20462	LOOKAH_SMOOTH_WITH_FEED	Glättung mit programmierten Vorschub
20465	ADAPT_PATH_DYNAMIC	Adaption der Bahndynamik
20480	SMOOTHING_MODE	Verhalten des Überschleifens mit G64x
20482	COMPRESSOR_MODE	Modus des Kompressors
20488	SPLINE_MODE	Einstellung für Spline-Interpolation
20490	IGNORE_OVL_FACTOR_FOR_ADIS	G641/G642 unabhängig vom Überlastfaktor
20550	EXACT_POS_MODE	Genauhaltbedingungen bei G0/G1
20560	G0_TOLERANCE_FACTOR	G0-Toleranzfaktor
20600	MAX_PATH_JERK	Bahnbezogener Maximalruck

Nummer	Bezeichner: \$MC_	Beschreibung
20602	CURV_EFFECT_ON_PATH_ACCEL	Einfluss der Bahnkrümmung auf Bahndynamik
20603	CURV_EFFECT_ON_PATH_JERK	Einfluss der Bahnkrümmung auf Bahnruck
20606	PREPDYN_SMOOTHING_ON	Aktivierung der Glättung der Krümmung
28060	MM_IPO_BUFFER_SIZE	Anzahl der NC-Sätze im IPO-Buffer (DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	Anzahl der NC-Sätze für die Satzaufbereitung (DRAM)
28520	MM_MAX_AXISPOLY_PER_BLOCK	Maximale Anzahl Achspolynome pro Satz
28530	MM_PATH_VELO_SEGMENTS	Anzahl der Speicherelemente zur Begrenzung der Bahngeschwindigkeit im Satz
28533	MM_LOOKAH_FFORM_UNITS	Speicher für den erweiterten LookAhead
28540	MM_ARCLENGTH_SEGMENTS	Anzahl der Speicherelemente zur Darstellung der Bogenlängenfunktion pro Satz
28610	MM_PREPDYN_BLOCKS	Anzahl Sätze zur Geschwindigkeitspräparation

### 3.10.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32310	MAX_ACCEL_OVL_FACTOR	Überlastfaktor für axiale Geschwindigkeitssprünge
32431	MAX_AX_JERK	Maximaler axialer Ruck bei Bahnbewegung
32432	PATH_TRANS_JERK_LIM	Maximaler axialer Ruck am Satzübergang im Bahnsteuerbetrieb
32433	SOFT_ACCEL_FACTOR	Skalierung der Beschleunigungsbegrenzung bei SOFT
32434	G00_ACCEL_FACTOR	Skalierung der Beschleunigungsbegrenzung bei G00
32435	G00_JERK_FACTOR	Skalierung der axialen Ruckbegrenzung bei G00
32440	LOOKAH_FREQUENCY	Glättungsgrenzfrequenz bei LookAhead
33100	COMPRESS_POS_TOL	Maximale Abweichung bei Kompression
33120	PATH_TRANS_POS_TOL	Maximale Abweichung beim Überschleifen mit G645
35240	ACCEL_TYPE_DRIVE	Beschleunigungskennlinie DRIVE für Achsen Ein/Aus
36000	STOP_LIMIT_COARSE	Genauhalt grob
36010	STOP_LIMIT_FINE	Genauhalt fein
36012	STOP_LIMIT_FACTOR	Faktor Genauhalt grob/fein und Stillstandsüberwachung
36020	POSITIONING_TIME	Verzögerungszeit Genauhalt fein

### 3.10.2 Settingdaten

#### 3.10.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42465	SMOOTH_CONTUR_TOL	Maximale Konturabweichung beim Überschleifen
42466	SMOOTH_ORI_TOL	Maximale Abweichung der Werkzeugorientierung beim Überschleifen
42470	CRIT_SPLINE_ANGLE	Ecken-Grenzwinkel Kompressor
42475	COMPRESS_CONTUR_TOL	Maximale Konturabweichung beim Kompressor

### 3.10.3 Signale

#### 3.10.3.1 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Alle Achsen stehen	DB21, ... .DBX36.3	DB3300.DBX4.3

#### 3.10.3.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Position erreicht mit Genauhalt grob	DB31, ... .DBX60.6	DB390x.DBX0.6
Position erreicht mit Genauhalt fein	DB31, ... .DBX60.7	DB390x.DBX0.7

## B2: Beschleunigung

### 4.1 Kurzbeschreibung

#### 4.1.1 Allgemeines

##### Funktionsumfang

Die Funktionsbeschreibung umfasst die Teilfunktionen:

- Beschleunigung
- Ruck
- Geknickte Beschleunigungskennlinie

##### **Beschleunigung und Ruck**

Durch achs- und kanalspezifisch parametrierbare Maximalwerte, sowie in Teileprogrammen und Synchronaktionen programmierbare Beschleunigungsprofile, dynamischen Anpassungen und Begrenzungen, lassen sich die wirksame Beschleunigung und Ruck optimal an die Maschine und die jeweilige Bearbeitungssituation anpassen.

##### **Geknickte Beschleunigungskennlinie**

Durch die geknickte Beschleunigungskennlinie lässt sich für Maschinenachsen mit Motor, insbesondere Schrittmotoren, mit einem stark drehzahlabhängigen Drehmomentenverlauf, die Beschleunigung so einstellen, dass eine optimale Auslastung des Motors bei gleichzeitigem Schutz vor Überlastung erzielt wird.

#### 4.1.2 Merkmale

##### Beschleunigung

##### **Achsspezifische Funktionen:**

- Parametrierbarer Maximalwert der Beschleunigung
- Über Teileprogrammanweisung anwählbares Beschleunigungsprofil:  
Beschleunigung ohne Ruckbegrenzung (BRISKA)
- Vorgabe des Maximalwertes per Teilprogrammanweisung (ACC)
- Eigener Maximalwert zum Verfahren mit programmiertem Eilgang (G00).
- Eigener Maximalwert zum Verfahren mit aktiver Ruckbegrenzung
- Beschleunigungsüberhöhung bei nicht tangentialen Satzübergängen

**Kanalspezifische Funktionen:**

- Über Teileprogrammanweisung anwählbares Beschleunigungsprofil:  
Beschleunigung ohne Ruckbegrenzung (BRISK)
- Parametrierbare Konstantfahrzeit zur Vermeidung von extremen Beschleunigungssprüngen
- Parametrierbare Beschleunigungsreserve für überlagerte Verfahrbewegungen
- Einstellbare Beschleunigungsbegrenzung
- Einstellbare Beschleunigung für spezifische Echtzeitereignisse
- Parametrierbare Beschleunigungsreserve für die Radialbeschleunigung

**Ruck**

**Achsspezifische Funktionen:**

- Über Teileprogrammanweisung anwählbares Beschleunigungsprofil:  
Beschleunigung mit Ruckbegrenzung (SOFTA)
- Parametrierbarer Maximalwert des Rucks für Einzelachsinterpolation
- Parametrierbarer Maximalwert des Rucks für Bahninterpolation

**Kanalspezifische Funktionen:**

- Über Teileprogrammanweisung anwählbares Beschleunigungsprofil:  
Beschleunigung mit Ruckbegrenzung (SOFT)
- Einstellbare Begrenzung des Rucks
- Einstellbarer Bahnruck für spezifische Echtzeitereignisse
- Eigener Maximalwert zum Verfahren mit programmiertem Eilgang (G00)
- Rucküberhöhung bei nicht krümmungsstetigen Satzübergängen

**Geknickte Beschleunigungskennlinie**

Eine geknickte Beschleunigungskennlinie wird über folgende Kenndaten parametrier:

- Maximale Geschwindigkeit  $v_{\max}$
- Maximale Beschleunigung  $a_{\max}$
- Reduziergeschwindigkeit  $v_{\text{red}}$
- Reduzierbeschleunigung  $a_{\text{red}}$
- Verlauf der Beschleunigungsreduktion (konstant, hyperbolisch, linear)

## 4.2 Funktionen

### 4.2.1 Beschleunigung ohne Ruckbegrenzung (BRISK/BRISKA) (kanal-/achsspezifisch)

#### 4.2.1.1 Allgemeine Informationen

##### Allgemeine Informationen

Bei Beschleunigung ohne Ruckbegrenzung (Ruck = unendlich) wird sofort mit dem Maximalwert beschleunigt. Bezüglich einer Beschleunigung mit Ruckbegrenzung ergeben sich folgende Unterschiede:

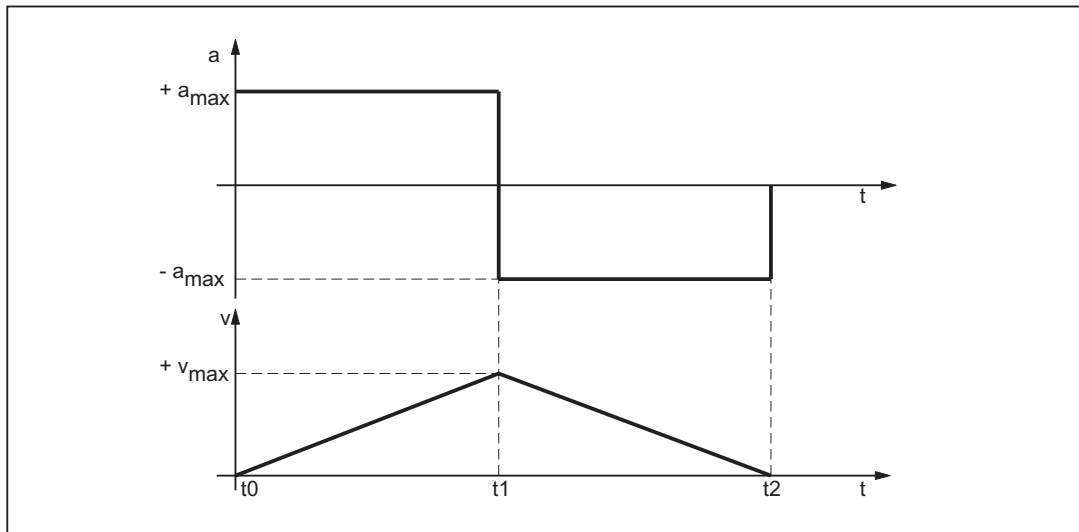
- **Vorteile**

Kürzere Bearbeitungszeiten bei gleichen Maximalwerten für Geschwindigkeit und Beschleunigung.

- **Nachteile**

Höhere Belastung der Maschinenmechanik und Gefahr der Anregung von hochfrequenten, schlecht regelbaren, mechanischen Schwingungen.

### Beschleunigungsprofil



$a_{max}$ : Maximalwert der Beschleunigung  
 $v_{max}$ : Maximalwert der Geschwindigkeit  
 $t$ : Zeit

Bild 4-1 Prinzipieller Verlauf von Geschwindigkeit und Beschleunigung bei sprungförmigem Beschleunigungsprofil

Aus dem obigen Bild lassen sich folgende Eigenschaften des Beschleunigungsprofils erkennen:

- Zeitpunkt:  $t_0$   
Beschleunigungssprung von 0 auf  $+a_{max}$
- Intervall:  $t_0 - t_1$   
Konstante Beschleunigung mit  $+a_{max}$ ; lineare Zunahme der Geschwindigkeit
- Zeitpunkt:  $t_1$   
Beschleunigungssprung von  $2 * a_{max}$  beim unmittelbaren Umschalten von Beschleunigen auf Bremsen

---

#### Hinweis

Der Beschleunigungssprung kann normalerweise durch Vorgabe einer Konstantfahrzeit (siehe Kapitel "Konstantfahrzeit (kanalspezifisch) (Seite 251)") vermieden werden.

---

- Intervall:  $t_1 - t_2$   
Konstante Beschleunigung mit  $-a_{max}$ ; lineare Abnahme der Geschwindigkeit



### 4.2.1.2 Parametrierung

#### Maximale axiale Beschleunigung für Bahnbewegungen

Die maximale axiale Beschleunigung für Bahnbewegungen kann Technologie-spezifisch für jede Maschinenachse über folgendes Maschinendatum eingestellt werden:

MD32300 \$MA\_MAX\_AX\_ACCEL[<Parametersatz-Index>]

mit <Parametersatz-Index> = 0, 1, 2 ... (max. Parametersatz-Nummer - 1)

Zu den Technologie-spezifischen Parametersätzen siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 219)".

Die Bahnparameter werden von der Bahnplanung des Vorlaufs so berechnet, dass die parametrisierten Maximalwerte der an der Bahn beteiligten Maschinenachsen nicht überschritten werden.

---

#### Hinweis

Eine Überschreitung des Maximalwerts für spezifische Bearbeitungssituationen ist möglich (siehe Kapitel "Beschleunigungsanpassung (ACC) (achsspezifisch) (Seite 253)" und "Bahnbeschleunigung für Echtzeitereignisse (kanalspezifisch) (Seite 256)").

---

#### Maximale axiale Beschleunigung für Positionierachsbewegungen

Bei Positionierachsbewegungen wird abhängig vom eingestellten Positionierachsdynamikmodus einer der beiden folgenden Maximalwerte wirksam:

- MD32300 \$MA\_MAX\_AX\_ACCEL [0] (maximale axiale Beschleunigung bei Bahnbewegungen im Dynamikmodus DYNNORM)
- MD32300 \$MA\_MAX\_AX\_ACCEL [1] (maximale axiale Beschleunigung bei Bahnbewegungen im Dynamikmodus DYNPOS)

Der Positionierachsdynamikmodus wird eingestellt im NC-spezifischen Maschinendatum:

MD18960 \$MN\_POS\_DYN\_MODE = <Modus>

<Modus>	Bedeutung
0	wirksame maximale axiale Beschleunigung: MD32300 \$MA_MAX_AX_ACCEL[ 0 ]
1	wirksame maximale axiale Beschleunigung: MD32300 \$MA_MAX_AX_ACCEL[ 1 ]

#### Maximale axiale Beschleunigung für JOG-Bewegungen

Für den JOG-Betrieb kann für jede Maschinenachse ein JOG-spezifischer Beschleunigungsmaximalwert projiziert werden (siehe Kapitel "Beschleunigung und Ruck bei JOG-Bewegungen (Seite 288)").

### 4.2.1.3 Programmierung

#### Bahnbeschleunigung ohne Ruckbegrenzung (BRISK)

##### Syntax

BRISK

##### Funktionalität

Über die Teileprogrammanweisung BRISK wird das Beschleunigungsprofil "ohne Ruckbegrenzung" für die Bahnbeschleunigung angewählt.

G-Gruppe: 21

Wirksamkeit: modal

##### Reset-Verhalten

Durch Reset wird die kanalspezifisch parametrisierte Grundstellung wirksam:

MD20150 \$MC\_GCODE\_RESET\_VALUES[20]

##### Randbedingungen

Wird in einem Teileprogrammen das Beschleunigungsprofil während der Bearbeitung gewechselt (BRISK / SOFT) erfolgt ein Genauhalt am Satzende.

#### Einzelachsbeschleunigung ohne Ruckbegrenzung (BRISKA)

##### Syntax

BRISKA (*Achse*{,*Achse*})

##### Funktion

Über die Teileprogrammanweisung BRISKA wird das Beschleunigungsprofil "ohne Ruckbegrenzung" für Einzelachsbewegungen (JOG, JOG/INC, Positionierachse, Pendelachse, etc.) angewählt.

G-Gruppe: -

Wirksamkeit: modal

*Achse*:

- Wertebereich: Achsbezeichner der Kanalachsen

### Achsspezifische Grundstellung

Beschleunigung ohne Ruckbegrenzung kann als achsspezifische Grundstellung für Einzelachsbewegungen vorgegeben werden:

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE = FALSE

### Reset-Verhalten

Durch Reset wird die achsspezifisch parametrierte Grundstellung wirksam:

MD32420 \$MA\_JOG\_AND\_POS\_ENABLE

## 4.2.2 Konstantfahrzeit (kanalspezifisch)

### 4.2.2.1 Allgemeine Informationen

#### Übersicht

Bei Beschleunigung ohne Ruckbegrenzung tritt beim Wechsel von Beschleunigung und Bremsen ein Beschleunigungssprung von  $2 * a_{max}$  auf. Zur Vermeidung dieses Beschleunigungssprungs kann eine kanalspezifische Konstantfahrzeit parametrierbar werden. Die Konstantfahrzeit legt die Zeit fest, mit der zwischen Beschleunigungs- und Bremsphase mit konstanter Geschwindigkeit verfahren wird:

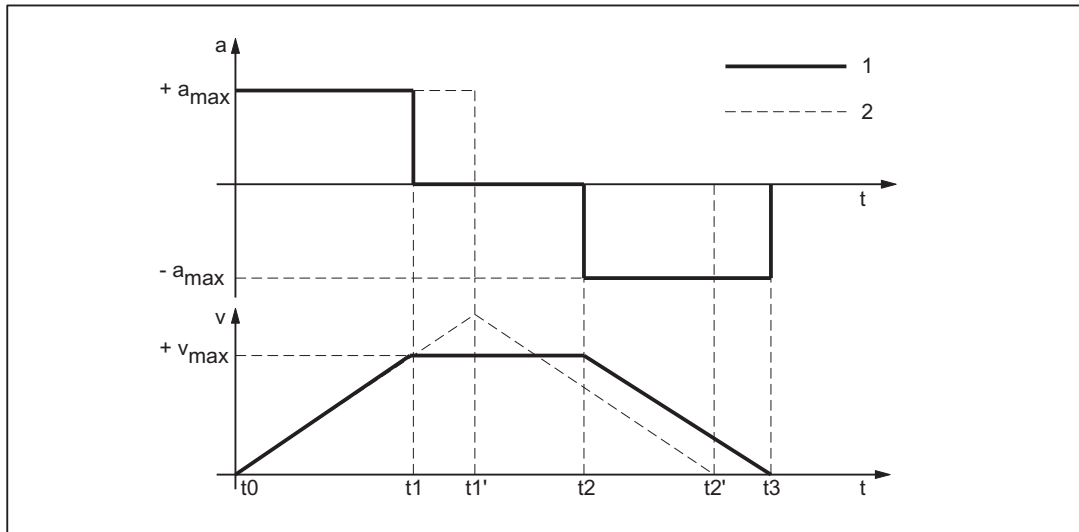
MD20500 \$MC\_CONST\_VELO\_MIN\_TIME (Minimale Zeit mit konstanter Geschwindigkeit)

---

#### Hinweis

Die Konstantfahrzeit wirkt nicht bei:

- Aktiver Funktion: LookAhead
  - In Verfahrssätzen mit einer Verfahrzeit kleiner bzw. gleich dem Interpolations-Takt.
-



- 1: Verlauf mit Konstantfahrzeit
- 2: Verlauf ohne Konstantfahrzeit
- $a_{max}$ : Maximalwert der Beschleunigung
- $v_{max}$ : Maximalwert der Geschwindigkeit
- t: Zeit

Bild 4-2 Prinzipieller Verlauf bei sprunghafter Beschleunigung

Aus dem obigen Bild lässt sich die Wirkung der Konstantfahrzeit erkennen:

- Zeitpunkt:  $t_1$   
Ende der Beschleunigungsphase mit Beschleunigungssprung  $1 * a_{max}$
- Intervall:  $t_1 - t_2$   
Beschleunigung 0; Konstante Geschwindigkeit über die parametrisierte Konstantfahrzeit
- Zeitpunkt:  $t_2$   
Beginn der Bremsphase mit Beschleunigungssprung  $1 * a_{max}$

Die Zeitpunkte  $t_0, t_1'$  und  $t_2'$  kennzeichnen den entsprechenden Verlauf, der sich ohne Konstantfahrzeit ergeben hätte.

#### 4.2.2.2 Parametrierung

Die Parametrierung der Konstantfahrzeit erfolgt kanalspezifisch über Maschinendaten:

MD20500 \$MC\_CONST\_VELO\_MIN\_TIME  
(Minimale Zeit mit konstanter Geschwindigkeit)

## 4.2.3 Beschleunigungsanpassung (ACC) (achsspezifisch)

### 4.2.3.1 Allgemeine Informationen

#### Funktion

Die Beschleunigung kann achsspezifisch per Teileprogrammanweisung ( $_{ACC}$ ) im Bereich von größer 0% bis kleiner gleich 200%, bezogen auf den im Maschinendatum parametrisierten Maximalwert, an die aktuelle Bearbeitungssituation angepasst werden.

#### Wirksamkeit

<b>Wirkt</b>	Die Beschleunigungsanpassung wirkt bei allen Interpolationsarten in den Betriebsarten AUTOMATIK, MDA und bei Probelaufvorschub.
<b>Wirkt nicht</b>	Die Beschleunigungsanpassung wirkt nicht in den Betriebsarten JOG und JOG/REF (Referenzpunktfahren). Die Beschleunigungsanpassung wirkt auch nicht, wenn aufgrund eines erkannten Fehlers die Maschinenachsen mit Schnellstop (Sollwert = 0) stillgesetzt werden.

### 4.2.3.2 Programmierung

#### Syntax

$_{ACC} [Achse] = Anpassfaktor$

#### Funktionalität

Über die Teileprogrammanweisung  $_{ACC}$  wird der Maximalwert der Beschleunigung einer Maschinenachse angepasst.

Achse:

- Wertebereich: Achsbezeichner der Maschinenachsen des Kanals

Anpassfaktor:

- Wertebereich:  $0 < Anpassfaktor \leq 200$
- Einheit: Prozent

Ausschalten:  $_{ACC} [Achse] = 100$

Wirksamkeit: modal

## Reset-Verhalten

Das Verhalten bei Kanal-RESET oder M30 kann über MD32320  
\$MA\_DYN\_LIMIT\_RESET\_MASK gesteuert werden:

Bit 0: 0 Der programmierte ACC-Wert wird mit Kanal-RESET/M30 auf 100 % zurückgesetzt.

Bit 0: 1 Der programmierte ACC-Wert bleibt über Kanal-RESET/M30 hinaus erhalten.

## 4.2.4 Beschleunigungsreserve (kanalspezifisch)

### 4.2.4.1 Allgemeine Informationen

#### Allgemeine Information

Im Normalfall nutzt der Vorlauf für die Bahnbeschleunigung die parametrisierten Maximalwerte der Maschinenachsen zu 100% aus. Um eine Beschleunigungsreserve für überlagerte Bewegungen z.B. im Rahmen der Funktion "Schnellabheben von der Kontur" zu erhalten, kann die Bahnbeschleunigung um einen parametrierbaren Faktor vermindert werden. Bei einem Faktor von z.B. 0,2 wird dann vom Vorlauf die maximal mögliche Bahnbeschleunigung nur zu 80% ausgenutzt. 20% verbleiben als Beschleunigungsreserve für überlagerte Bewegungen.

### 4.2.4.2 Parametrierung

Die Parametrierung der Beschleunigungsreserve erfolgt kanalspezifisch über  
Maschinendatum:  
MD20610 \$MC\_ADD\_MOVE\_ACCEL\_RESERVE  
(Beschleunigungsreserve für überlagerte Bewegungen)

## 4.2.5 Begrenzung der Bahnbeschleunigung (kanalspezifisch)

### 4.2.5.1 Allgemeine Informationen

#### Allgemeine Informationen

Um auf die jeweilige Bearbeitungssituationen flexibel reagieren zu können, kann die vom Vorlauf berechnete Bahnbeschleunigung kanalspezifisch über Settingdaten begrenzt werden:

SD42500 \$SC\_SD\_MAX\_PATH\_ACCEL (Maximale Bahnbeschleunigung)

Der im Settingdatum vorgegebene Wert wird nur dann berücksichtigt, wenn er kleiner ist als die vom Vorlauf berechnete Bahnbeschleunigung.

Die Begrenzung muss kanalspezifisch über Settingdaten freigegeben werden:

SD42502 \$SC\_IS\_SD\_MAX\_PATH\_ACCEL = TRUE

### 4.2.5.2 Parametrierung

Die Parametrierung erfolgt kanalspezifisch über Settingdaten:

SD42500 \$SC\_SD\_MAX\_PATH\_ACCEL (Maximale Bahnbeschleunigung)

SD42502 \$SC\_IS\_SD\_MAX\_PATH\_ACCEL (Aktivierung der Begrenzung der Bahnbeschleunigung)

### 4.2.5.3 Programmierung

#### Begrenzungswert

#### Syntax

`$SC_SD_MAX_PATH_ACCEL = Begrenzungswert`

#### Funktionalität

Die Begrenzung der Bahnbeschleunigung kann durch Programmierung des Settingdatums angepasst werden.

*Begrenzungswert:*

- Wertebereich:  $\geq 0$
- Einheit: m/s<sup>2</sup>

Anwendbarkeit:

- Teileprogramm
- Statische Synchronaktion

## Ein/Ausschalten

### Syntax

`$_SC_IS_SD_MAX_PATH_ACCEL = Wert`

### Funktionalität

Die Begrenzung der Bahnbeschleunigung kann durch Programmierung des Settingdatums ein/ausgeschaltet werden.

Parameter: *Wert*

- Wertebereich: TRUE, FALSE

Anwendbarkeit:

- Teileprogramm
- Statische Synchronaktion

## 4.2.6 Bahnbeschleunigung für Echtzeitereignisse (kanalspezifisch)

### 4.2.6.1 Allgemeine Informationen

#### Allgemeine Informationen

Um keinen Kompromiss zwischen bearbeitungsoptimaler Beschleunigung einerseits und zeitoptimaler Beschleunigung bei folgenden Echtzeitereignissen:

- NC-Stop / NC-Start
- Änderungen des Vorschub-Overrides
- Änderung der Geschwindigkeitsvorgabe für die "sicher reduzierte Geschwindigkeit" im Rahmen der Funktion "Safety Integrated"

andererseits eingehen zu müssen, kann für die genannten Echtzeitereignisse die Bahnbeschleunigung über eine kanalspezifische Systemvariable vorgegeben werden:

`$_AC_PATHACC = Bahnbeschleunigung`

Die Bahnbeschleunigung für Echtzeitereignisse ist nur für die Zeitdauer der Geschwindigkeitsänderung aufgrund eines der genannten Echtzeitereignisse wirksam.



## Begrenzung

Übersteigt die angegebene Bahnbeschleunigung das Leistungsvermögen der an der Bahn beteiligten Maschinenachsen, wird die Bahnbeschleunigung Steuerungs-intern derart begrenzt, dass die resultierende axiale Beschleunigung ( $a_{res}$ ) höchstens das 2-fache des parametrisierten axialen Maximalwertes ( $a_{max}$ ) beträgt.

$$a_{res} = 2 * a_{max}, \text{ mit } a_{max} = \text{MD32300 } \$\text{MA\_MAX\_AX\_ACCEL}$$

### Hinweis

Die Bahnbeschleunigung für Echtzeitereignisse wird ohne Rücksicht auf die Radialbeschleunigung zugelassen.

## Wirksamkeit

<b>Wirkt</b>	Die Bahnbeschleunigung für Echtzeitereignisse wirkt in den Betriebsarten AUTOMATIK und MDA nur im Zusammenhang mit folgenden Echtzeitereignissen: <ul style="list-style-type: none"> <li>• NC-Stop / NC-Start</li> <li>• Override-Änderungen</li> <li>• Änderung der Geschwindigkeitsvorgabe für die "sicher reduzierte Geschwindigkeit" im Rahmen der Funktion "Safety Integrated"</li> </ul>
<b>Wirkt nicht</b>	Die Bahnbeschleunigung für Echtzeitereignisse wirkt nicht bei Änderungen der Bahngeschwindigkeit, die sich aufgrund der Bahnplanung im Vorlauf des Kanals ergeben, wie z. B. Konturkrümmungen, Ecken, kinematische Begrenzungen der Transformation, etc. Die Bahnbeschleunigung für Echtzeitereignisse wirkt nicht, wenn der programmierte Wert kleiner ist als die für den betroffenen Bahnabschnitt vom Vorlauf berechnete Bahnbeschleunigung.

## Programmierung

Zur Programmierung der Systemvariablen im Teileprogramm oder Synchronaktion siehe Kapitel "Programmierung (Seite 258)".

### 4.2.6.2 Programmierung

#### Syntax

`$AC_PATHACC = Bahnbeschleunigung`

#### Funktionalität

Über die kanalspezifische Systemvariable wird die Bahnbeschleunigung für Echtzeitereignisse vorgegeben.

Parameter: *Bahnbeschleunigung*

- Wertebereich: Bahnbeschleunigung  $\geq 0$
- Einheit: m/s<sup>2</sup>

Ausschalten `$AC_PATHACC = 0`

Anwendbarkeit:

- Teileprogramm
- statischen Synchronaktion

#### Reset-Verhalten

Bei Reset wird die Bahnbeschleunigung für Echtzeitereignisse ausgeschaltet.

#### Randbedingungen

Durch Programmierung von `$AC_PATHACC` im Teileprogramm wird implizit ein Vorlaufstopp mit Reorg ausgelöst (`STOPRE`).

## 4.2.7 Beschleunigung bei programmiertem Eilgang (G00) (achsspezifisch)

### 4.2.7.1 Allgemeine Informationen

Oftmals muss die Beschleunigung für die an der Bearbeitung beteiligten Maschinenachsen aufgrund der bearbeitungsspezifischen Randbedingungen niedriger eingestellt werden als es der Leistungsfähigkeit der Maschine entspricht.

Zum zeitoptimalen Verfahren der Maschinenachsen bei programmiertem Eilgang (Teileprogrammanweisung G00), kann ein eigener Maximalwert der achsspezifischen Beschleunigung parametrierbar werden.

#### Einrichtebetriebsart JOG

Die Beschleunigung bei Eilgangüberlagerung in der Einrichtebetriebsart JOG wird durch diese Funktionalität nicht beeinflusst.

### 4.2.7.2 Parametrierung

Die Parametrierung des Maximalwerts der achsspezifischen Beschleunigung bei programmiertem Eilgang (G00) erfolgt über das achsspezifische Maschinendatum:

MD32434 \$MA\_G00\_ACCEL\_FACTOR  
(Skalierung der Beschleunigungsbegrenzung bei G00)

Daraus ergibt sich der von der Bahnplanung im Vorlauf berücksichtigte Maximalwert der achsspezifischen Beschleunigung bei programmiertem Eilgang (G00) zu:

Beschleunigung[Achse] =  
MD32300 \$MA\_MAX\_AX\_ACCEL \* MD32434 \$MA\_G00\_ACCEL\_FACTOR

## 4.2.8 Beschleunigung bei aktiver Ruckbegrenzung (SOFT/SOFTA) (achsspezifisch)

### 4.2.8.1 Allgemeine Informationen

#### Funktion

Beim Beschleunigen mit Ruckbegrenzung entsteht bei gleichem Maximalwert der Beschleunigung ein gewisser Zeitverlust bezogen auf ein Beschleunigen ohne Ruckbegrenzung. Zum Ausgleich dieses Zeitverlustes, kann für das Verfahren der Maschinenachsen bei aktiver Ruckbegrenzung (SOFT/SOFTA), ein eigener Maximalwert der achsspezifische Beschleunigung parametrieren werden.

Die Parametrierung des Maximalwerts der Beschleunigung bei aktiver Ruckbegrenzung erfolgt über einen Faktor bezogen auf den achsspezifischen Maximalwert. Daraus ergibt sich der von der Bahnplanung im Vorlauf berücksichtigte Maximalwert der achsspezifischen Beschleunigung bei aktiver Ruckbegrenzung zu:

$$\text{Beschleunigung[Achse]} = \text{MD32300 \$MA\_MAX\_AX\_ACCEL} * \text{MD32433 \$MA\_SOFT\_ACCEL\_FACTOR}$$

### 4.2.8.2 Parametrierung

Die Parametrierung des Maximalwerts der Beschleunigung bei aktiver Ruckbegrenzung (SOFT/SOFTA) erfolgt über das achsspezifische Maschinendatum:

MD32434 \$MA\_SOFT\_ACCEL\_FACTOR  
(Skalierung der Beschleunigungsbegrenzung bei SOFT)

## 4.2.9 Beschleunigungsüberhöhung bei nicht tangentialen Satzübergängen (achsspezifisch)

### 4.2.9.1 Allgemeine Informationen

#### Funktion

Bei nicht tangentialen Satzübergängen (Ecken) muss von der Steuerung zur Einhaltung der parametrisierten Achsdynamik die Verfahrbewegung der Geometrieachsen unter Umständen stark abgebremst werden. Zur Verminderung bzw. Vermeidung des Abbremsens an nicht tangentialen Satzübergängen kann eine höhere achsspezifische Beschleunigung zugelassen werden.

Die Parametrierung der Beschleunigungsüberhöhung erfolgt über einen Faktor bezogen auf den achsspezifischen Maximalwert. Daraus ergibt sich der von der Bahnplanung im Vorlauf berücksichtigte Maximalwert der achsspezifischen Beschleunigung bei nicht tangentialen Satzübergängen zu:

$$\text{Beschleunigung[Achse]} = \text{MD32300 \$MA\_MAX\_AX\_ACCEL} * \text{MD32310 \$MA\_MAX\_ACCEL\_OVL\_FACTOR}$$

### 4.2.9.2 Parametrierung

Die Parametrierung der Beschleunigungsüberhöhung bei nicht tangentialen Satzübergängen erfolgt über das achsspezifische Maschinendatum:

MD32310 \$MA\_MAX\_ACCEL\_OVL\_FACTOR  
(Überlastfaktor für Geschwindigkeitssprünge)

## 4.2.10 Beschleunigungsreserve für die Radialbeschleunigung (kanalspezifisch)

### 4.2.10.1 Allgemeine Informationen

#### Übersicht

An gekrümmten Konturen wirkt neben der Bahnbeschleunigung (Tangentialbeschleunigung) zusätzlich die Radialbeschleunigung. Wird diese bei der Parametrierung der Bahnparameter nicht berücksichtigt, kann die wirksame axiale Beschleunigung während Beschleunigungs- oder Bremsvorgängen auf der gekrümmten Kontur kurzfristig bis zum 2-fachen des Maximalwertes betragen.

$$\begin{aligned} \text{Wirksame axiale Beschleunigung} &= \\ \text{Anteil der Bahnbeschleunigung} + \text{Anteil der Radialbeschleunigung} &= \\ 2 * (\text{MD32300 \$MA\_MAX\_AX\_ACCEL}) & \end{aligned}$$

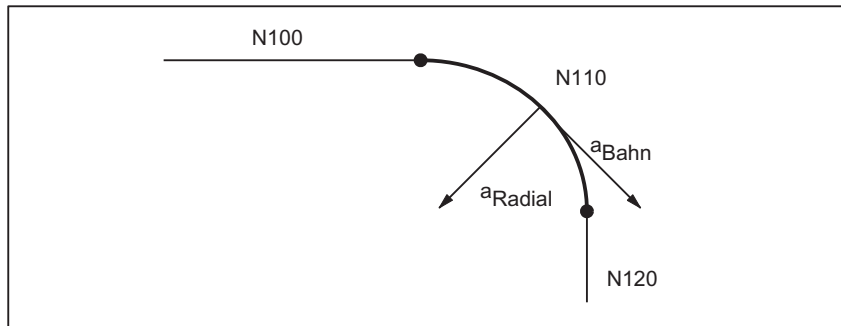


Bild 4-3 Radial- und Bahnbeschleunigung an gekrümmten Konturen

Über das kanalspezifische Maschinendatum:

MD20602 \$MC\_CURV\_EFFECT\_ON\_PATH\_ACCEL

(Einfluss der Bahnkrümmung auf die Bahndynamik)

kann der Anteil der achsspezifischen Beschleunigung eingestellt werden, der als Beschleunigungsreserve für die Radialbeschleunigung berücksichtigt werden soll.

Bei einem Wert von z.B. 0.75 werden 75% der achsspezifischen Beschleunigung für die Radialbeschleunigung und 25% für die Bahnbeschleunigung zur Verfügung gestellt.

Allgemein berechnen sich die entsprechenden Maximalwerte zu:

Radialbeschleunigung =

MD20602 \$MC\_CURV\_EFFECT\_ON\_PATH\_ACCEL \* MD32300 \$MA\_MAX\_AX\_ACCEL

Bahnbeschleunigung =

(1 - MD20602 \$MC\_CURV\_EFFECT\_ON\_PATH\_ACCEL) \* MD32300 \$MA\_MAX\_AX\_ACCEL

### Beispiel

Folgende Maschinenparameter sind gegeben:

- MD32300 \$MA\_MAX\_AX\_ACCEL für alle Geometrieachsen: 3 m/s
- Maximale Bahngeschwindigkeit bei einem Bahnradius von 10 mm aufgrund mechanischer Gegebenheiten an der Maschine: 5 m/min

Die Radialbeschleunigung berechnet sich daraus zu:

$$a_{\text{Radial}} = \frac{v_{\text{Bahn}}^2 \text{ [m/min]}}{r \text{ [mm]} * 3.6 \text{ [m/s}^2\text{]}} = \frac{5^2}{10 * 3.6} = 0.694 \text{ m/s}^2$$

Daraus ergibt sich die Einstellung für die Beschleunigungsreserve zu:

$$\text{MD20602 \$MC\_CURV\_EFFECT\_ON\_PATH\_ACCEL} = \frac{a_{\text{Radial}} [\text{m/s}^2]}{\text{MD32300 \$MA\_MAX\_AX\_ACCEL} [\text{m/s}^2]} = \frac{0.694}{3} \approx 0.23$$

## Linearsätze

Bei Linearsätzen (Geradeninterpolation) ohne aktive kinematische Transformation wirkt die genannte Beschleunigungsreserve nicht.

### 4.2.10.2 Parametrierung

Die Parametrierung des Anteils an der maximalen Achsbeschleunigung, der als Beschleunigungsreserve für die Radialbeschleunigung an gekrümmten Konturen berücksichtigt werden soll, erfolgt über das kanalspezifische Maschinendatum:

MD20602 \$MC\_CURV\_EFFECT\_ON\_PATH\_ACCEL  
(Einfluss der Bahnkrümmung auf die Bahndynamik)

## 4.2.11 Ruckbegrenzung bei Bahninterpolation (SOFT) (kanalspezifisch)

### 4.2.11.1 Allgemeine Informationen

#### Übersicht

Im Rahmen der im weiteren Verlauf beschriebenen Funktionalität wird immer von stetiger Beschleunigung, d. h. von Beschleunigung mit Ruckbegrenzung (Ruck = endlicher Wert) als Beschleunigungsprofil ausgegangen. Bei Beschleunigung mit Ruckbegrenzung wird die Beschleunigung von 0 bis zum Maximalwert linear interpoliert.

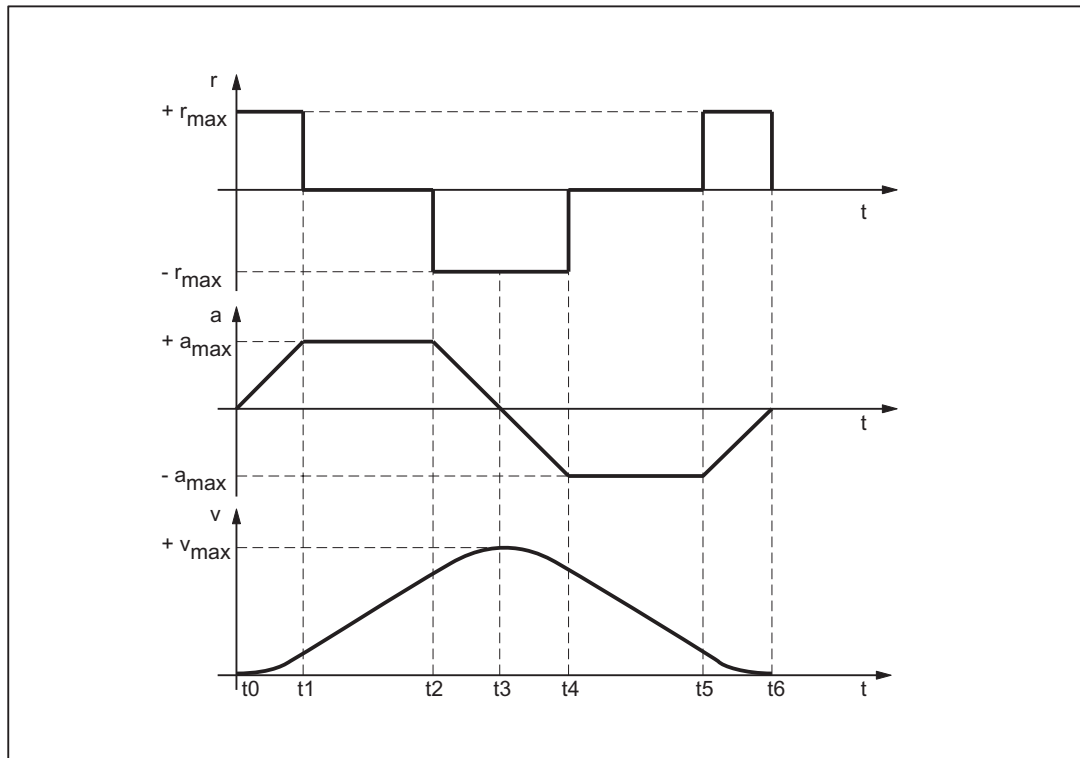
#### Vorteile

Geringe Belastung der Maschinenmechanik und kaum Gefahr der Anregung von hochfrequenten, schlecht regelbaren, mechanischen Schwingungen aufgrund der stetigen Beschleunigungserhöhung.

#### Nachteile

Längere Bearbeitungszeiten bei gleichen Maximalwerten für Geschwindigkeit und Beschleunigung gegenüber dem sprungförmigen Beschleunigungsprofil.

Beschleunigungsprofil



$r_{max}$ : Maximalwert des Rucks  
 $a_{max}$ : Maximalwert der Beschleunigung  
 $v_{max}$ : Maximalwert der Geschwindigkeit  
 t: Zeit

Bild 4-4 Prinzipieller Verlauf von Ruck, Beschleunigung und Geschwindigkeit bei ruckbegrenztem Beschleunigungsprofil

Aus dem obigen Bild lassen sich folgende Eigenschaften des Beschleunigungsprofils erkennen:

- Intervall:  $t_0 - t_1$   
 Konstanter Ruck mit  $+r_{max}$ ; lineare Zunahme der Beschleunigung; quadratische Zunahme der Geschwindigkeit
- Intervall:  $t_1 - t_2$   
 Konstante Beschleunigung mit  $+a_{max}$ ; lineare Zunahme der Geschwindigkeit
- Intervall:  $t_2 - t_3$   
 Konstanter Ruck mit  $-r_{max}$ ; lineare Abnahme der Beschleunigung; quadratische Abnahme der Geschwindigkeitserhöhung bis zum Erreichen des Maximalwertes  $+v_{max}$



- Intervall:  $t_3 - t_4$   
Konstanter Ruck mit  $-r_{\max}$ ; lineare Zunahme der Bremsbeschleunigung; quadratische Abnahme der Geschwindigkeit
- Intervall:  $t_4 - t_5$   
Konstante Bremsbeschleunigung mit  $-a_{\max}$ ; lineare Abnahme der Geschwindigkeit
- Intervall:  $t_5 - t_6$   
Konstanter Ruck mit  $+r_{\max}$ ; lineare Abnahme der Bremsbeschleunigung; quadratische Abnahme der Geschwindigkeitsverminderung bis zum Stillstand  $v = 0$

#### 4.2.11.2 Parametrierung

##### Maximalwert des Rucks für Bahnbewegungen (achsspezifisch)

Der maximale axiale Ruck für Bahnbewegungen kann Technologie-spezifisch für jede Maschinenachse über folgendes Maschinendatum eingestellt werden:

MD32431 \$MA\_MAX\_AX\_JERK[<Parametersatz-Index>]

mit <Parametersatz-Index> = 0, 1, 2 ... (max. Parametersatz-Nummer - 1)

Zu den Technologie-spezifischen Parametersätzen siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 219)".

Die Bahnparameter werden von der Bahnplanung des Vorlaufs so berechnet, dass die parametrisierten Maximalwerte der an der Bahn beteiligten Maschinenachsen nicht überschritten werden.

---

##### Hinweis

Eine Überschreitung des Maximalwerts für spezifische Bearbeitungssituationen ist möglich (siehe Kapitel "Bahnruck für Echtzeitereignisse (kanalspezifisch) (Seite 270)").

---

##### Maximalwert des Rucks für Bahnbewegungen (kanalspezifisch)

Neben der achsspezifischen Einstellung kann der Maximalwert des Rucks auch als kanalspezifischer Bahnparameter über folgendes Maschinendatum vorgegeben werden:

MD20600 \$MC\_MAX\_PATH\_JERK (Bahnbezogener Maximalruck)

Um eine gegenseitige Beeinflussung von achs- und kanalspezifischen Maximalwerten des Rucks auszuschließen, ist der kanalspezifische Maximalwert auf einen Wert größer den axialen Maximalwerten einzustellen.

### 4.2.11.3 Programmierung

#### Syntax

SOFT

#### Funktionalität

Über die Teileprogrammanweisung `SOFT` wird das Beschleunigungsprofil mit Ruckbegrenzung für die Verfahrbewegungen der Geometrieachsen im Kanal angewählt.

G-Gruppe: 21

Wirksamkeit: modal

#### Reset-Verhalten

Durch Reset wird die kanalspezifisch parametrierte Grundstellung wirksam:

MD20150 \$MC\_GCODE\_RESET\_VALUES[20]

#### Randbedingungen

Wird in einem Teileprogramm der Beschleunigungsmodus während der Bearbeitung gewechselt (`BRISK` ↔ `SOFT`) erfolgt auch bei Bahnsteuerbetrieb am Übergang ein Satzwechsel mit Genauhalt am Satzende.

### 4.2.12 Ruckbegrenzung bei Einzelachsinterpolation (SOFTA) (achsspezifisch)

#### 4.2.12.1 Parametrierung

#### Grundeinstellung der axialen Ruckbegrenzung

Die Beschleunigung mit Ruckbegrenzung kann als axiale Grundstellung vorgegeben werden:

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE== TRUE

### Maximaler axialer Ruck für Positionierachsbewegungen

Beim Verfahren von Positionierachsen bei aktiver Ruckbegrenzung wird der Wert aus einem der folgenden Maschinendaten als maximaler axialer Ruck wirksam:

- MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK (maximaler axialer Ruck bei Positionierachsbewegungen)
- MD32431 \$MA\_MAX\_AX\_JERK [0] (maximaler axialer Ruck bei Bahnbewegungen im Dynamikmodus DYNORM)
- MD32431 \$MA\_MAX\_AX\_JERK [1] (maximaler axialer Ruck bei Bahnbewegungen im Dynamikmodus DYNPOS)

Welches Maschinendatum zur Anwendung kommt, wird bestimmt durch den eingestellten Positionierachsdynamikmodus:

MD18960 \$MN\_POS\_DYN\_MODE = <Modus>

<Modus>	Bedeutung
	Als maximaler axialer Ruck bei Positionierachsbewegungen wirkt:
0	MD32430 \$MA_JOG_AND_POS_MAX_JERK Bei aktivem G75/G751 (Festpunkt anfahren): MD32431 \$MA_MAX_AX_JERK[ 0 ]
1	MD32431 \$MA_MAX_AX_JERK[ 1 ]

### Maximaler axialer Ruck für JOG-Bewegungen

Für den JOG-Betrieb kann für jede Maschinenachse ein JOG-spezifischer Ruck-Maximalwert projektiert werden (siehe Kapitel "Beschleunigung und Ruck bei JOG-Bewegungen (Seite 288)").

#### 4.2.12.2 Programmierung

##### Syntax

SOFTA (*Achse*{*Achse*})

##### Funktionalität

Über die Teileprogrammanweisung SOFTA wird Beschleunigung mit Ruckbegrenzung für Einzelachsbewegungen (Positionierachse, Pendelachse, etc.) angewählt.

G-Gruppe: -

Wirksamkeit: modal

*Achse*:

- Wertebereich: Achsbezeichner der Kanalachsen

### Achsspezifische Grundstellung

Beschleunigung mit Ruckbegrenzung kann als achsspezifische Grundstellung für Einzelachsbewegungen vorgegeben werden:

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE = TRUE

### Reset-Verhalten

Durch Reset wird die achsspezifisch parametrisierte Grundstellung wirksam:

MD32420 \$MA\_JOG\_AND\_POS\_ENABLE

## 4.2.13 Begrenzung des Bahnrucks (kanalspezifisch)

### 4.2.13.1 Allgemeine Informationen

#### Übersicht

Um auf die jeweilige Bearbeitungssituationen flexibel reagieren zu können, kann der vom Vorlauf berechnete Bahnruck kanalspezifisch über Settingdaten begrenzt werden:

SD42510 \$SC\_SD\_MAX\_PATH\_JERK (Maximaler Bahnruck)

Der im Settingdatum vorgegebene Wert wird im Kanal nur dann berücksichtigt, wenn er kleiner ist als der vom Vorlauf berechnete Bahnruck.

Die Begrenzung muss kanalspezifisch über Settingdaten freigegeben werden:

SD42512 \$SC\_IS\_SD\_MAX\_PATH\_JERK = TRUE

### 4.2.13.2 Parametrierung

Die Parametrierung erfolgt kanalspezifisch über Settingdaten:

SD42510 \$SC\_SD\_MAX\_PATH\_JERK (Maximaler Bahnruck)

SD42512 \$SC\_IS\_SD\_MAX\_PATH\_JERK  
(Aktivierung der Begrenzung des Bahnrucks)

### 4.2.13.3 Programmierung

#### Maximaler Bahnruck

##### Syntax

$\$SC\_SD\_MAX\_PATH\_JERK = Ruckwert$

##### Funktionalität

Die Begrenzung des Bahnrucks kann durch Programmierung des Settingdatums angepasst werden.

*Ruckwert.*

- Wertebereich:  $\geq 0$
- Einheit:  $m/s^3$

Anwendbarkeit:

- Teileprogramm
- Statische Synchronaktion

#### Ein/Ausschalten

##### Syntax

$\$SC\_IS\_SD\_MAX\_PATH\_JERK = Wert$

##### Funktionalität

Die Begrenzung des Bahnrucks kann durch Programmierung des Settingdatums ein/ausgeschaltet werden.

Parameter: *Wert*

- Wertebereich: TRUE, FALSE

Anwendbarkeit:

- Teileprogramm
- Statische Synchronaktion

## 4.2.14 Bahnruck für Echtzeitereignisse (kanalspezifisch)

### 4.2.14.1 Allgemeine Informationen

#### Übersicht

Um keinen Kompromiss zwischen bearbeitungsoptimalem Ruck einerseits und zeitoptimalem Ruck bei folgenden Echtzeitereignissen:

- NC-Stop / NC-Start
- Änderungen des Vorschub-Overrides
- Änderung der Geschwindigkeitsvorgabe für die "sicher reduzierte Geschwindigkeit" im Rahmen der Funktion "Safety Integrated"

andererseits eingehen zu müssen, kann für die genannten Echtzeitereignisse der Bahnruck über eine kanalspezifische Systemvariable vorgegeben werden:

$\$AC\_PATHJERK = \text{Bahnruck}$

Der Bahnruck für Echtzeitereignisse ist nur für die Zeitdauer der Geschwindigkeitsänderung aufgrund eines der genannten Echtzeitereignisse wirksam.

#### Begrenzung

Da der Ruck keine für den Antrieb relevante physikalische Größe darstellt, erfolgt **keine** Begrenzung des vorgegebenen Rucks.

#### Wirksamkeit

<b>Wirk</b>	Der Bahnruck für Echtzeitereignisse wirkt in den Betriebsarten AUTOMATIK und MDA nur im Zusammenhang mit folgenden Echtzeitereignissen: <ul style="list-style-type: none"><li>• NC-Stop / NC-Start</li><li>• Override-Änderungen</li><li>• Änderung der Geschwindigkeitsvorgabe für die "sicher reduzierte Geschwindigkeit" im Rahmen der Funktion "Safety Integrated"</li></ul>
<b>Wirk nicht</b>	Der Bahnruck für Echtzeitereignisse wirkt nicht bei Änderungen der Bahngeschwindigkeit, die sich aufgrund der Bahnplanung im Vorlauf des Kanals ergeben wie z.B. Konturkrümmungen, Ecken, kinematische Begrenzungen der Transformation, etc. Der Bahnruck für Echtzeitereignisse wirkt nicht, wenn der programmierte Wert kleiner ist als der für den betroffenen Bahnabschnitt vom Vorlauf berechnete Bahnruck.

## Programmierung

Um für Echtzeitereignisse den Ruck entsprechend der Beschleunigung zu setzen, kann die Systemvariable folgendermaßen gesetzt werden:

$\$AC\_PATHJERK = \$AC\_PATHACC / \text{Glättungszeit}$

- $\$AC\_PATHACC$ : Bahnbeschleunigung [m/s<sup>2</sup>]

Glättungszeit: frei wählbar z. B. 0,02 s

Zur Programmierung der Systemvariablen im Teileprogramm oder Synchronaktion siehe Kapitel "Programmierung (Seite 271)".

### 4.2.14.2 Programmierung

#### Syntax

$\$AC\_PATHJERK = \text{Ruckwert}$

#### Funktionalität

Über die kanalspezifische Systemvariable wird der Bahnruck für Echtzeitereignisse vorgegeben.

*Ruckwert.*

- Wertebereich: Bahnruck  $\geq 0$
- Einheit: m/s<sup>3</sup>

Anwendbarkeit:

- Teileprogramm
- statischen Synchronaktion

#### Reset-Verhalten

Bei Reset wird die Funktion ausgeschaltet.

#### Randbedingungen

Durch Programmierung von  $\$AC\_PATHJERK$  im Teileprogramm wird implizit ein Vorlaufstop mit Reorg ausgelöst (*STOPRE*).

## 4.2.15 Ruck bei programmiertem Eilgang (G00) (achsspezifisch)

### 4.2.15.1 Allgemeine Informationen

#### Übersicht

Oftmals muss der maximale Ruck für die an der Bearbeitung beteiligten Maschinenachsen aufgrund der bearbeitungsspezifischen Randbedingungen niedriger eingestellt werden als es der Leistungsfähigkeit der Maschine entspricht.

Zum zeitoptimalen Verfahren der Maschinenachsen bei programmiertem Eilgang (Teileprogrammanweisung G00), kann ein eigener Maximalwert des achsspezifischen Rucks parametrierbar werden.

#### Einrichtebetriebsart JOG

Der Ruck bei Eilgangüberlagerung in der Einrichtebetriebsart JOG wird durch diese Funktionalität nicht beeinflusst.

### 4.2.15.2 Parametrierung

Die Parametrierung des Maximalwerts des achsspezifischen Rucks bei programmiertem Eilgang (G00) erfolgt über das achsspezifische Maschinendatum:

MD32434 \$MA\_G00\_ACCEL\_FACTOR  
(Skalierung der Beschleunigungsbegrenzung bei G00)

Daraus ergibt sich der von der Bahnplanung im Vorlauf berücksichtigte Maximalwert des achsspezifischen Rucks bei programmiertem Eilgang (G00) zu:

Ruck[Achse] =  
MD32431 \$MA\_MAX\_AX\_JERK \* MD32435 \$MA\_G00\_JERK\_FACTOR

## 4.2.16 Rucküberhöhung bei nicht krümmungstetigen Satzübergängen (achsspezifisch)

### 4.2.16.1 Allgemeine Informationen

#### Übersicht

Bei nicht krümmungstetigen Satzübergängen (z. B. Gerade > Kreis) muss von der Steuerung zur Einhaltung der parametrisierten Achsdynamik die Verfahrbewegung der Geometrieachsen unter Umständen stark abgebremst werden. Zur Verminderung bzw. Vermeidung des Abbremsens an nicht krümmungstetigen Satzübergängen kann ein höherer achsspezifischer Ruck zugelassen werden.

Die Parametrierung der Rucküberhöhung erfolgt über einen eigenen achsspezifischen Maximalwert.



#### 4.2.16.2 Parametrierung

Die Parametrierung der Rucküberhöhung bei nicht krümmungsstetigen Satzübergängen erfolgt über das achsspezifische Maschinendatum:

MD32432 \$MA\_PATH\_TRANS\_JERK\_LIM

(Rucküberhöhung bei nicht krümmungsstetigen Satzübergängen)

#### 4.2.17 Geschwindigkeitsabhängige Ruckadaption (achsspezifisch)

##### Funktion

Die Bahndynamik ergibt sich aus den parametrierten, konstanten axialen Maximalwerte für Geschwindigkeit, Beschleunigung und Ruck der an der Bahn beteiligten Achsen:

- MD32000 \$MA\_MAX\_AX\_VELO (max. Achsgeschwindigkeit)
- MD32300 \$MA\_MAX\_AX\_ACCEL (max. Achsbeschleunigung)
- MD32431 \$MA\_MAX\_AX\_JERK (max. axialer Ruck bei Bahnbewegung)

Bei **Konturen mit nicht konstanter Krümmung** (Torsion), wie sie z.B. im Zusammenhang mit Freiformflächen vorkommen, können sich, hauptsächlich aufgrund des axialen Rucks, Schwankungen der Bahngeschwindigkeit, besonders im oberen Geschwindigkeitsbereich, ergeben. Die Schwankungen der Bahngeschwindigkeit führen zu Beeinträchtigungen der Oberflächengüte.

Durch eine geschwindigkeitsabhängige Erhöhung des zulässigen axialen Rucks, vermindert sich bei Konturen mit nicht konstanter Krümmung der Einfluss des axialen Rucks auf die Bahngeschwindigkeit. Bei entsprechender Parametrierung können die Schwankungen der Bahngeschwindigkeit u.U. vollständig vermieden werden.

Die geschwindigkeitsabhängige Erhöhung des zulässigen axialen Rucks hat keinen Einfluss auf die maximal mögliche Bahnbeschleunigung und Bahnruck. Diese ergeben sich auch bei aktiver Ruckadaption weiter aus den die in den Maschinendaten parametrierten, konstanten axialen Maximalwerte.

Da bei Linearbewegungen sind sowohl Krümmung als auch Torsion gleich Null sind, hat die geschwindigkeitsabhängige Ruckadaption bei Linearbewegungen keine Auswirkungen.

##### Verfügbarkeit

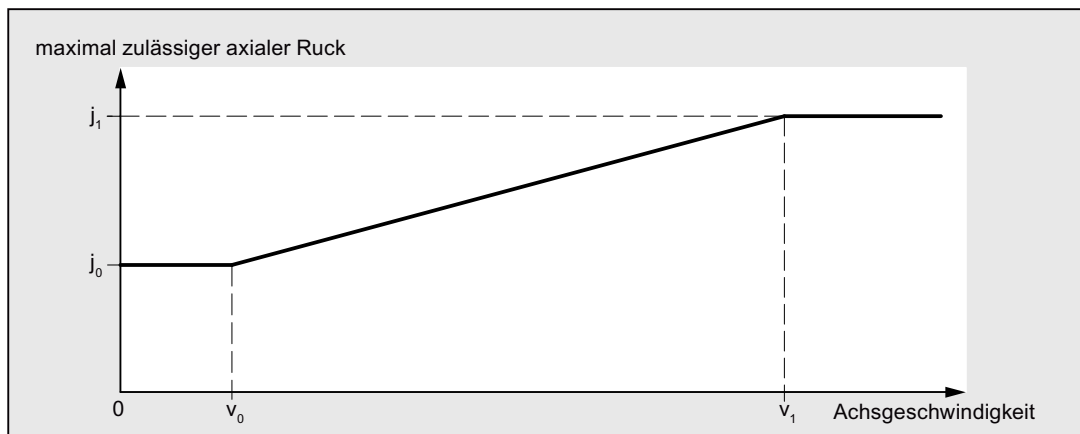
Die Funktion "Geschwindigkeitsabhängige Ruckadaption" ist unabhängig von der Funktion "Freiformflächenmodus: Grundfunktionen (Seite 222)" verfügbar.

**Parametrierung**

Die Parametrierung der Funktion "Geschwindigkeitsabhängige Ruckadaption" erfolgt achsspezifisch mit folgenden Maschinendaten:

- MD32437 \$MA\_AX\_JERK\_VEL0[<n>] = <Schwellwert<sub>unten</sub>>  
 Untere Geschwindigkeitsschwelle der Ruckadaption. Ab dieser Geschwindigkeit wird geschwindigkeitsabhängige Ruckadaption wirksam.  
 Die untere Geschwindigkeitsschwelle ist über den Index n, für jeden Dynamikmodus (siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 219)") separat einstellbar.
- MD32438 \$MA\_AX\_JERK\_VEL1[<n>] = <Schwellwert<sub>oben</sub>>  
 Obere Geschwindigkeitsschwelle, der Ruckadaption. Bei dieser Geschwindigkeit erreicht der geschwindigkeitsabhängige Ruck seinen mit MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR parametrierten Maximalwert  $j_{max}$ .  
 Die obere Geschwindigkeitsschwelle ist über den Index n, für jeden Dynamikmodus (siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 219)") separat einstellbar.
- MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR = <Faktor>  
 Faktor zur Parametrierung des maximalen geschwindigkeitsabhängigen Rucks  $j_{max}$  bei Erreichen der obere Geschwindigkeitsschwelle MD32438 \$MA\_AX\_JERK\_VEL1[<n>]:  

$$j_{max} = (MD32431 \$MA\_MAX\_AX\_JERK) * (MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR)$$
 Bei einem Wert > 1.0 ist die geschwindigkeitsabhängige Ruckadaption aktiv.  
 Bei einem Wert = 1.0, ist die geschwindigkeitsabhängige Ruckadaption inaktiv.



- v<sub>0</sub>: MD32437 \$MA\_AX\_JERK\_VEL0
- v<sub>1</sub>: MD32438 \$MA\_AX\_JERK\_VEL1
- j<sub>0</sub>: MD32431 \$MA\_MAX\_AX\_JERK
- j<sub>1</sub>: MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR \* MD32431 \$MA\_MAX\_AX\_JERK

Bild 4-5 Axialer Ruck als Funktion der Achsgeschwindigkeit

---

### Hinweis

Die geschwindigkeitsabhängige Ruckadaption wird nur aktiv, wenn:

MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR > 1.0

---

### Beispiel

Beispiel für Parametrierung:

- MD32437 \$MA\_AX\_JERK\_VELO = 3000 mm/min
- MD32438 \$MA\_AX\_JERK\_VELO1 = 6000 mm/min
- MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR[AX1] = 2.0
- MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR[AX2] = 3.0
- MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR[AX3] = 1.0

### Auswirkung

- Die geschwindigkeitsabhängige Ruckadaption wird für die 1. und 2. Achse aktiv, während die Funktion für die 3. Achse nicht aktiv wird.
- Für Achsgeschwindigkeiten im Bereich 0 bis 3000 mm/min wird der parametrierte Ruck wirksam.
- Für Achsgeschwindigkeiten im Bereich 3000 mm/min bis 6000 mm/min wird der maximale Ruck linear erhöht.
- Der maximal erlaubte Ruck der 1. Achse wird dabei für Achsgeschwindigkeiten größer als 6000 mm/min um den Faktor 2 erhöht, bei der 2. Achse um den Faktor 3.
- Die parametrierten Werte gelten in jedem Dynamikmodus.

## 4.2.18 Ruckfilter (achsspezifisch)

### 4.2.18.1 Allgemeine Informationen

#### Übersicht

In einigen Anwendungsfällen, z. B. beim Fräsen von Freiformflächen, kann es vorteilhaft sein, die Lagesollwertverläufe der Maschinenachsen zu glätten. Dadurch lassen sich höhere Oberflächengüten durch Verminderung der Anregungen von mechanischen Schwingungen an der Maschine erreichen.

Zum Glätten des Lagesollwertverlaufs einer Maschinenachse kann unabhängig von den kanal- und achsspezifischen Ruckbegrenzungen, die auf Interpolatorebene berücksichtigt werden, ein Ruckfilter auf Lagereglerebene aktiviert werden.

Die Wirkung des Ruckfilters muss möglichst ausgeprägt sein, ohne dabei die Konturgenauigkeit unzulässig zu beeinträchtigen, und zusätzlich ein möglichst "symmetrisches" Glättungsverhalten aufweisen. D. h. fährt man dieselbe Kontur vorwärts und rückwärts ab, sollte der vom Filter verrundete Konturverlauf in beiden Richtungen möglichst ähnlich sein.

Zur optimalen Anpassung des Ruckfilters an die Gegebenheiten an der Maschine stehen verschiedenen Filter-Modi zur Verfügung:

- Filter 2. Ordnung (PT2)
- Gleitende Mittelwertbildung
- Bandsperre

#### Modus: Filter 2. Ordnung

Der Filter-Modus "Filter 2. Ordnung" erfüllt als einfacher Tiefpass-Filter die oben genannten Anforderungen nur bei relativ kleinen Filterzeitkonstanten (ca. 10 ms). Bei größeren Zeitkonstanten ergeben sich schnell unzulässige Konturabweichungen. Die Filterwirkung ist relativ gering.

Dieser Filter-Modus bietet Vorteile wenn sehr große Filterzeitkonstanten benötigt werden und die Konturgenauigkeit eine untergeordnete Rolle spielt (z. B. Positionierachsen).

Aus historischen Gründen ist dieser Filter-Modus voreingestellt.

### Modus: Gleitende Mittelwertbildung

Mit der gleitenden Mittelwertbildung als Filter-Modus können bei geringen Konturabweichungen Filterzeitkonstanten im Bereich von 20 - 40 ms eingestellt werden. Die Glättungswirkung ist weitgehend symmetrisch.

Die Anzeige des berechneten Faktors der Regelkreisverstärkung (Kv-Faktor) in der Bedienoberfläche zeigt kleinere Werte an als anhand der Filterwirkung angemessen wäre. Die Konturgenauigkeit ist höher als der angezeigte Kv-Faktor erwarten lässt.

Bei einer Umstellung von Filter-Modus "Filter 2. Ordnung" nach "Gleitende Mittelwertbildung" kann sich daher der angezeigte Kv-Faktor verkleinern (bei gleicher Filterzeitkonstanten), obwohl die Konturgenauigkeit besser wird.

### Modus: Bandsperre

Bei der Bandsperre handelt es sich um einen Filter 2. Ordnung in Zähler und Nenner:

$$H(s) = \frac{\frac{s^2}{(2 \cdot \pi \cdot f_Z)^2} + \frac{2 \cdot s \cdot D_Z}{(2 \cdot \pi \cdot f_Z)}}{\frac{s^2}{(2 \cdot \pi \cdot f_N)^2} + \frac{2 \cdot s \cdot D_N}{2 \cdot \pi \cdot f_N}}$$

mit:

- f<sub>Z</sub>: Zähler-Eigenfrequenz
- f<sub>N</sub>: Nenner-Eigenfrequenz
- D<sub>Z</sub>: Zähler-Dämpfung
- D<sub>N</sub>: Nenner-Dämpfung

Da erwartet wird, dass eine schwingfähige Filtereinstellung ohnehin nicht zu brauchbaren Ergebnissen führt, steht wie beim Tiefpassfilter (PT2) Filter-Modus "Filter 2. Ordnung" (PT2) des Ruckfilters keine Einstellmöglichkeit für die Nenner-Dämpfung D<sub>N</sub> zur Verfügung. Die Nenner-Dämpfung D<sub>N</sub> ist fest auf den Wert 1 eingestellt.

Die Bandsperre kann auf 2 verschiedene Arten parametrisiert werden:

- Echte Bandsperre
- Bandsperre mit zusätzlicher Betragsanhebung/-absenkung bei hohen Frequenzen

#### Echte Bandsperre

Die echte Bandsperre ergibt sich, wenn Zähler- und Nenner-Eigenfrequenz identisch gewählt werden:

$$f_Z = f_N = f_{\text{Sperr}} \text{ (Sperrfrequenz)}$$

Wählt man die Zähler-Dämpfung = 0, ergibt sich für die Sperrfrequenz vollständige Auslöschung. Die 3dB-Bandbreite ergibt sich dann zu:

$$f_{\text{3dB-Bandbreite}} = 2 \cdot f_{\text{Sperr}}$$

Soll keine komplette Auslöschung erfolgen, sondern nur eine Absenkung um einen Faktor k, so ist die Zählerdämpfung entsprechend k zu wählen. Die obige Formel für die 3dB-Bandbreite gilt dann nicht.

**Bandsperre mit zusätzlicher Betragsanhebung/-absenkung bei hohen Frequenzen**

In diesem Fall werden Zähler- und Nenner-Eigenfrequenz unterschiedlich eingestellt. Die Zähler-Eigenfrequenz legt dabei die Sperrfrequenz fest.

Durch Wahl einer kleineren/größeren Nenner-Eigenfrequenz als der Zähler-Eigenfrequenz wird bei hohen Frequenzen der Amplitudengang entsprechend angehoben/abgesenkt. Eine Amplitudengang-Anhebung bei hohen Frequenzen ist in den meisten Fällen vertretbar, da die Regelstrecke i. A. selbst Tiefpasscharakter besitzt, d. h. bei hohen Frequenzen der Amplitudengang ohnehin abfällt.

**Randbedingungen**

Wird die Zähler-Eigenfrequenz zu groß gewählt, wird der Filter abgeschaltet. Die Grenzfrequenz  $f_{Zmax}$  ist dabei abhängig vom Lagereglertakt:

$$f_{Zmax} = \frac{1}{2 * \pi * T_{Zmin}} = \frac{1}{2 * \pi * T_{Lagereglertakt}} \quad (\text{Shannon-Theorem})$$

**4.2.18.2 Parametrierung**

**Aktivierung**

Die Aktivierung des Ruckfilters erfolgt über das Maschinendatum:  
 MD32400 \$MA\_AX\_JERK\_ENABLE (Axiale Ruckbegrenzung)  
 Der Ruckfilter ist in jeder Betriebs- und Interpolationsart aktiv.

**Filter-Modus**

Die Auswahl des Filter-Modus erfolgt über das Maschinendatum:  
 MD32402 \$MA\_AX\_JERK\_MODE

Wert	Filter-Modus
1	Filter 2. Ordnung
2	Gleitende Mittelwertbildung
3	Bandsperre

**Zeitkonstante**

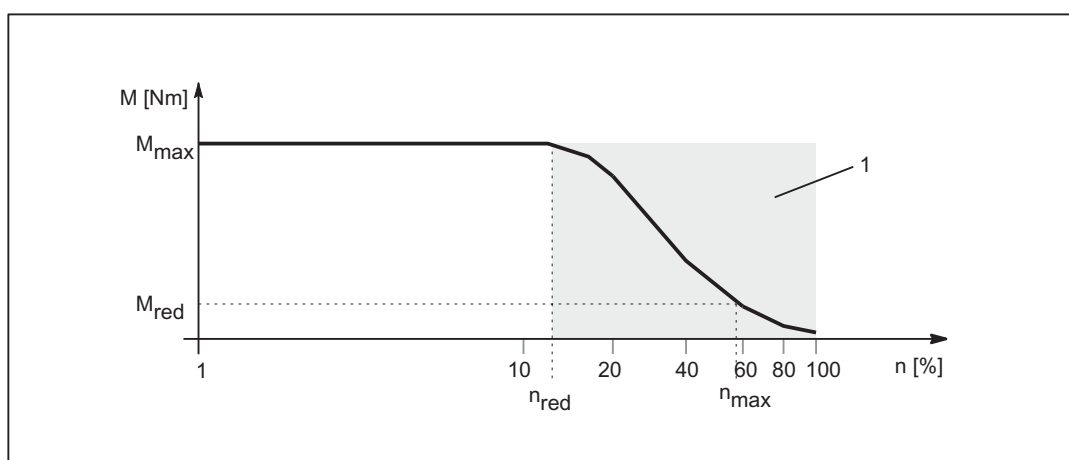
Die Zeitkonstante für den axialen Ruckfilter wird eingestellt mit dem Maschinendatum:  
 MD32410 \$MA\_AX\_JERK\_TIME  
 Der Ruckfilter ist nur dann wirksam, wenn die Zeitkonstante größer ist als ein Lagereglertakt.

## 4.2.19 Geknickte Beschleunigungskennlinie

### 4.2.19.1 Anpassung an die Motorkennlinie

#### Funktion

Verschiedene Motortypen, insbesondere Schrittmotoren, weisen einen stark drehzahlabhängigen Drehmomentenverlauf mit einem steilen Abfall des Drehmoments im oberen Drehzahlbereich auf. Zur optimalen Ausnutzung der Motorkennlinie ist es erforderlich, die Beschleunigung ab einer bestimmten Drehzahl zu reduzieren.



- 1: Bereich des Drehmomentabfalls
- $n_{red}$ : Drehzahl ab der mit reduziertem Drehmoment gerechnet wird
- $n_{max}$ : Maximale Drehzahl
- $M_{max}$ : Maximales Drehmoment
- $M_{red}$ : Drehmoment bei  $n_{max}$  (entspricht Reduzierbeschleunigung)

Bild 4-6 Drehmomentkennlinie eines Motors mit stark drehzahlabhängigem Drehmomentverlauf

#### Nachbildung des Drehmomentverlaufs

Zur Nachbildung des Drehmomentverlaufs der Motorkennlinie können über folgendes Maschinendatum:

MD35242 \$MA\_ACCEL\_REDUCTION\_TYPE = *Verlauf*

verschiedene Kennlinientypen ausgewählt werden:

0	= Konstanter Verlauf
1	= Hyperbolischer Verlauf
2	= Linearer Verlauf

Die folgenden Bilder zeigen die typischen Geschwindigkeits- und Beschleunigungskennlinien des jeweiligen Kennlinientyps:

**Konstanter Verlauf**

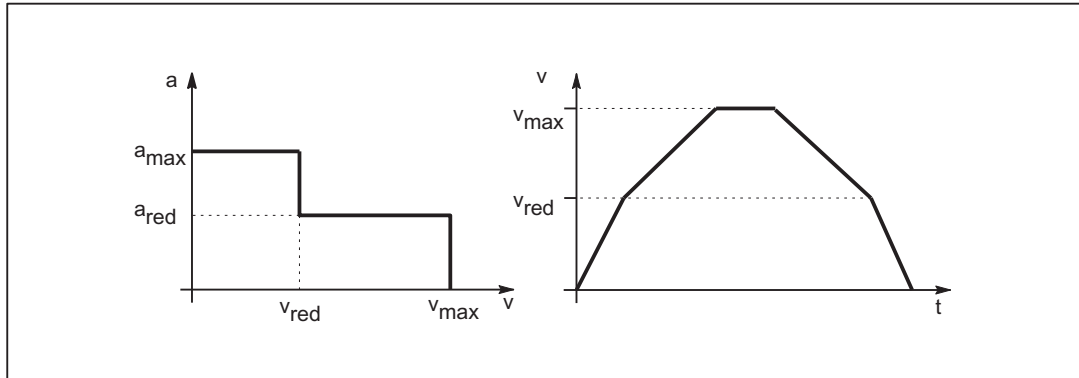


Bild 4-7 Beschleunigungs- und Geschwindigkeitsverlauf bei Beschleunigungsreduktion:0 = konstant

**Hyperbolischer Verlauf**

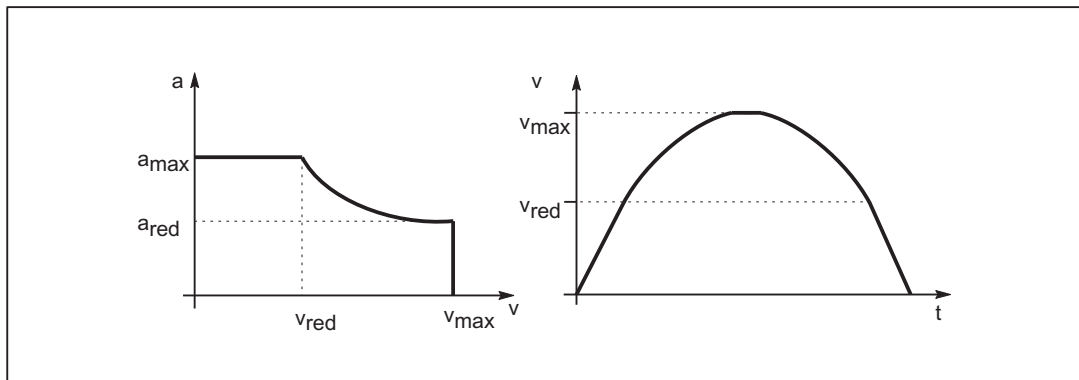


Bild 4-8 Beschleunigungs- und Geschwindigkeitsverlauf bei Beschleunigungsreduktion:1 = hyperbolisch



### Linearer Verlauf

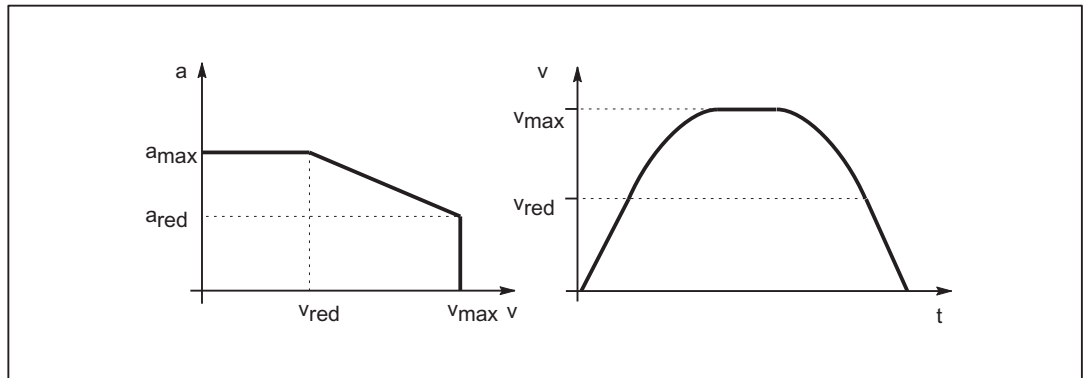


Bild 4-9 Beschleunigungs- und Geschwindigkeitsverlauf bei Beschleunigungsreduktion:2=linear

Die Eckdaten der Kennlinien ergeben sich zu:

$$\begin{aligned}v_{\max} &= \$MA\_MAX\_AX\_VELO \\v_{\text{red}} &= \$MA\_ACCEL\_REDUCTION\_SPEED\_POINT * \$MA\_MAX\_AX\_VELO \\a_{\max} &= \$MA\_MAX\_AX\_ACCEL \\a_{\text{red}} &= (1 - \$MA\_ACCEL\_REDUCTION\_FACTOR) * \$MA\_MAX\_AX\_ACCEL\end{aligned}$$

#### 4.2.19.2 Auswirkungen auf die Bahnbeschleunigung

##### Funktion

Die Kennlinie der Bahnbeschleunigung ergibt sich aus den Kennlinientypen der an der Bahn beteiligten Achsen. Werden Achsen mit unterschiedlichen Kennlinientypen gemeinsam interpoliert, bestimmt der restriktivste Reduktionstyp das Beschleunigungsprofil der Bahnbeschleunigung.

Es gilt folgende Reihenfolge der Prioritäten, mit 1. = Höchste Priorität:

1. Beschleunigungsreduktion: 0 = konstanter Verlauf
2. Beschleunigungsreduktion: 1 = hyperbolischer Verlauf
3. Beschleunigungsreduktion: 2 = linearer Verlauf
4. Keine Beschleunigungsreduktion wirksam

Keine wirksame Beschleunigungsreduktion ergibt sich z. B. durch:

MD35220 \$MA\_ACCEL\_REDUCTION\_SPEED\_POINT = 1

und / oder

MD35230 \$MA\_ACCEL\_REDUCTION\_FACTOR = 0

---

##### Hinweis

Maschinenachsen mit Schrittmotor- und Gleichstromantrieb können gemeinsam interpoliert werden.

---

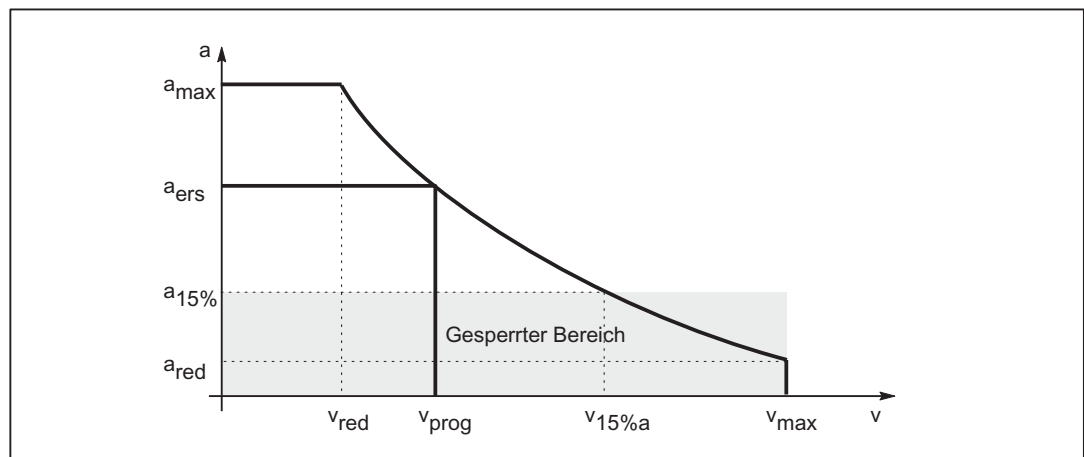
### 4.2.19.3 Ersatzkennlinie

#### Funktion

Kann die programmierte Bahn nicht mit der parametrisierten Beschleunigungskennlinie gefahren werden (z. B. aktiver kinematische Transformation), wird eine Ersatzkennlinie durch Reduzierung der dynamischen Grenzwerte erzeugt. Die dynamischen Grenzwerte werden dabei so berechnet, dass sich als Ersatzkennlinie ein optimierter Kompromiss zwischen maximaler Geschwindigkeit und konstanter Beschleunigung ergibt.

#### Ersatzkennlinie bei linearen Bahnabschnitten

Wird eine Bahngeschwindigkeit programmiert, die größer der Geschwindigkeit ist bei der das Beschleunigungsvermögen noch 15 % des Maximalwertes beträgt ( $v_{15\%a}$ ), wird auf diesen Wert begrenzt. Somit stehen in jeder Bearbeitungssituation mindestens 15 % des maximalen Beschleunigungsvermögens bzw. Drehmoments des Motors zur Verfügung.



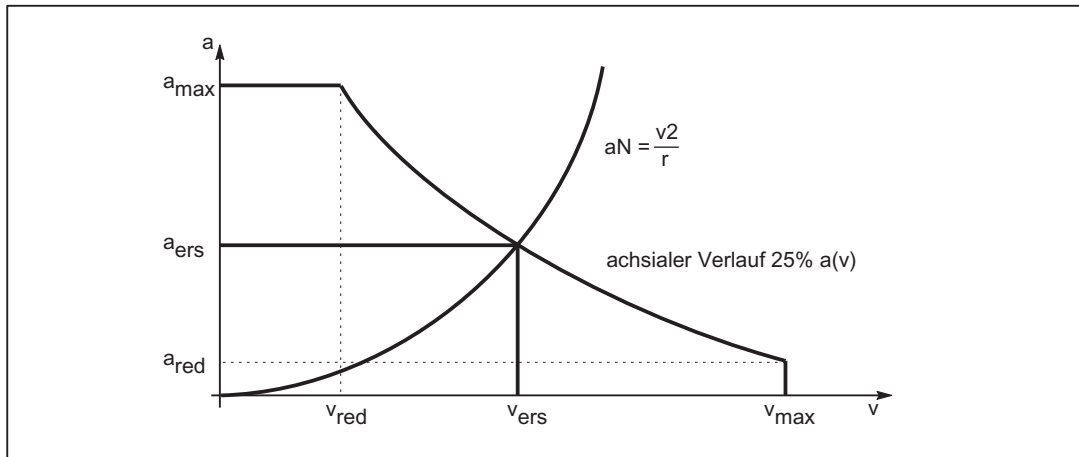
- $a_{ers}$ : Konstante Beschleunigung der Ersatzkennlinie
- $a_{15\%}$ : Minimale konstante Beschleunigung  

$$a_{15\%} = 0,15 * (a_{max} - a_{red}) + a_{red}$$
- $v_{ers}$ : Geschwindigkeit der Ersatzkennlinie
- $v_{prog}$ : Programmierte Geschwindigkeit
- $v_{15\%a}$ : Geschwindigkeit bei  $a_{15\%}$

Bild 4-10 Ersatzkennlinie der Bahn: Lineare Bahn

**Ersatzkennlinie bei gekrümmten Bahnabschnitten**

Bei gekrümmten Bahnabschnitten werden Normalen- und Tangentialbeschleunigung gemeinsam betrachtet. Die Bahngeschwindigkeit wird dabei soweit reduziert, dass nur maximal 25 % des geschwindigkeitsabhängigen Beschleunigungsvermögens der Achsen für die Normalenbeschleunigung benötigt wird. Die verbleibenden 75 % des Beschleunigungsvermögens werden für die Tangentialbeschleunigung, also dem Bremsen bzw. Beschleunigen auf der Bahn, reserviert.



- $a_N$ : Normalbeschleunigung
- $a_{ers}$ : Konstante Beschleunigung der Ersatzkennlinie
- $v_{ers}$ : Geschwindigkeit der Ersatzkennlinie
- $r$ : Bahnradius

Bild 4-11 Ersatzkennlinie der Bahn: Gekrümmte Bahn

**Satzübergänge bei Bahnsteuerbetrieb**

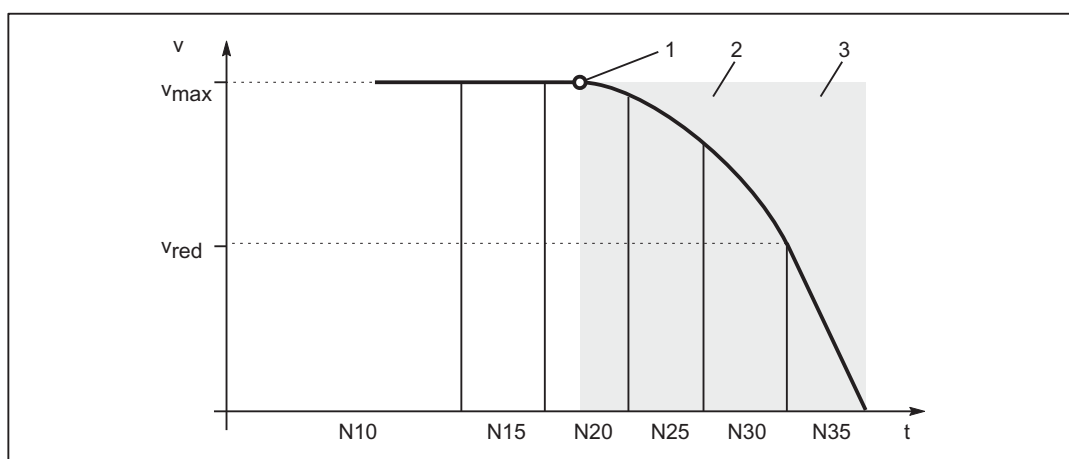
Bei aktivem Bahnsteuerbetrieb kommt es an nichttangentialen Satzübergängen beim Verfahren mit der programmierten Bahngeschwindigkeit zu axialen Geschwindigkeitssprüngen.

Die Bahngeschwindigkeit wird dann derart geführt, dass beim Satzübergang kein axialer Geschwindigkeitsanteil größer der Reduziergeschwindigkeit  $v_{red}$  ist.

### Bremsrampe bei Bahnsteuerbetrieb und LookAhead

Bei aufeinander folgenden Teileprogrammssätzen mit kurzen Bahnwegen kann sich ein Beschleunigungs- oder Bremsvorgang über mehrere Teileprogrammssätzen erstrecken.

LookAhead berücksichtigt dabei auch den parametrisierten geschwindigkeitsabhängigen Verlauf der Beschleunigung.



- 1: Bremseseinsatzpunkt
- 2: Bereich des Drehmomentabfalls
- 3: Bereich des maximalen Drehmoments
- $v_{red}$ : Reduziergeschwindigkeit
- $v_{max}$ : Maximale Geschwindigkeit
- $N_{xy}$ : Teileprogrammssatz mit Satznummer  $N_{xy}$

Bild 4-12 Bremsvorgang mit LookAhead

#### 4.2.19.4 Parametrierung

Die Aktivierung der geknickten Beschleunigungskennlinie erfolgt Maschinenachs-spezifisch über das Maschinendatum:

MD35240 \$MA\_ACCEL\_TYPE\_DRIVE = TRUE

Die Parametrierung der geknickten Beschleunigungskennlinie erfolgt achsspezifisch über die Maschinendaten:

MD32000 \$MA\_MAX\_AX\_VELO (Maximale Achsgeschwindigkeit)

MD35220 \$MA\_ACCEL\_REDUCTION\_SPEED\_POINT  
(Drehzahl für reduzierte Beschleunigung)

MD35230 \$MA\_ACCEL\_REDUCTION\_FACTOR (Reduzierte Beschleunigung)

MD32300 \$MA\_MAX\_AX\_ACCEL (Maximale Achsbeschleunigung)

MD35242 \$MA\_ACCEL\_REDUCTION\_TYPE  
(Art der Beschleunigungsreduktion: 0=konstant, 1=hyperbolisch, 2=linear)

#### 4.2.19.5 Programmierung

### Kanalspezifisches Einschalten (DRIVE)

#### Syntax

DRIVE

#### Funktionalität

Über die Teileprogrammanweisung `DRIVE` wird die geknickte Beschleunigungskennlinie für die Bahnbeschleunigung aktiviert

G-Gruppe: 21

Wirksamkeit: modal

#### Reset-Verhalten

Durch Reset wird die kanalspezifisch parametrisierte Grundstellung wirksam:

MD20150 \$MC\_GCODE\_RESET\_VALUES[20]

#### Abhängigkeiten

Ist für eine Maschinenachsen die geknickten Beschleunigungskennlinie parametrisiert, wird sie standardmäßig nach diesem Beschleunigungsprofil verfahren.

Wird für einen bestimmten Bahnabschnitt durch Teileprogrammanweisung `SOFT` oder `BRISK` das wirksame Beschleunigungsprofil umgeschaltet, wird statt der geknickten Beschleunigungskennlinie eine entsprechende Ersatzkennlinie mit reduzierten dynamischen Grenzwerten verwendet.

Mit erneuter Programmierung von `DRIVE` kann die geknickte Beschleunigungskennlinie wieder aktiviert werden.

### Achsspezifisches Einschalten (DRIVEA)

#### Syntax

DRIVEA (*Achse* {*Achse*})

## Funktionalität

Über die Teileprogrammanweisung wird die geknickte Beschleunigungskennlinie achsspezifisch für alle Einzelachs-Interpolationen (Positionierachse, Pendelachse, etc.) eingeschaltet.

G-Gruppe: -

Wirksamkeit: modal

*Achse:*

- Wertebereich: Achsbezeichner der Kanalachsen

## Reset-Verhalten

Durch Reset wird die kanalspezifisch parametrisierte Grundstellung wirksam:

MD20150 \$MC\_GCODE\_RESET\_VALUES[20]

## Abhängigkeiten

Ist für eine Maschinenachse die geknickte Beschleunigungskennlinie parametrisiert, wird sie standardmäßig nach diesem Beschleunigungsprofil verfahren.

Wird durch die Teileprogrammanweisung `SOFTA` oder `BRISKA` achsspezifisch das wirksame Beschleunigungsprofil umgeschaltet, wird statt der geknickten Beschleunigungskennlinie eine entsprechende Ersatzkennlinie verwendet.

Mit Programmierung von `DRIVEA` kann achsspezifisch wieder auf die geknickte Beschleunigungskennlinie umgeschaltet werden.

### 4.2.19.6 Randbedingungen

#### Einzelachs-Interpolation

Ab Aktivierung der geknickten Beschleunigungskennlinie wird bei Einzelachs-Interpolationen (Positionierachse, Pendelachse, Handfahren, etc.) die Maschinenachse ausschließlich im Modus `DRIVEA` verfahren.

Ein Umschalten des Beschleunigungsprofils über folgende Teileprogrammanweisungen ist nicht möglich:

- Sprungförmige Beschleunigung (`BRISKA`)
- Ruckbegrenzte Beschleunigung (`SOFTA`)

#### Bahn-Interpolation

Ist für eine an einer programmierten Bahn beteiligten Maschinenachse die geknickte Beschleunigungskennlinie parametrisiert ohne dass die Teileprogrammanweisung `DRIVE` aktiv ist, wird für die Bahn eine Ersatzkennlinie mit reduzierten dynamischen Grenzwerten bestimmt.

## Kinematische Transformation

Im Zusammenhang mit einer aktiven kinematischen Transformation wird die geknickte Beschleunigungskennlinie nicht berücksichtigt. Steuerungsintern wird auf Beschleunigung ohne Ruckbegrenzung (BRISK) umgeschaltet und für die Bahnbeschleunigung eine Ersatzkennlinie wirksam.

### 4.2.20 Beschleunigung und Ruck bei JOG-Bewegungen

Um im JOG-Betrieb störende Maschinenruckbewegungen zu vermeiden, können für JOG-Bewegungen eigene axiale Beschleunigungs- und Ruckbegrenzungswerte vorgegeben werden.

Darüber hinaus ist es möglich, Beschleunigung und Ruck für das Handfahren von Geometrie- und Orientierungsachsen kanalspezifisch zu begrenzen. Dies ermöglicht eine bessere Handhabbarkeit für Kinematiken, die kartesische Bewegungen vollständig über Rundachsen erzeugen (Roboter).

#### 4.2.20.1 Parametrierung

### Axiale Begrenzung von Beschleunigung und Ruck

#### Maximale axiale Beschleunigung bei JOG-Bewegungen

Die maximale axiale Beschleunigung bei JOG-Bewegungen kann für jede Maschinenachse vorgegeben werden über das Maschinendatum:

MD32301 \$MA\_JOG\_MAX\_ACCEL

Bei MD32301 = 0 ist statt des JOG-spezifischen Maximalwerts der Wert aus MD32300 \$MA\_MAX\_AX\_ACCEL wirksam.

#### Maximaler axialer Ruck bei JOG-Bewegungen

Der maximale axiale Ruck bei JOG-Bewegungen kann für jede Maschinenachse vorgegeben werden über das Maschinendatum:

MD32436 \$MA\_JOG\_MAX\_JERK

Bei MD32436 = 0 ist statt des JOG-spezifischen Maximalwerts der Wert aus MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK wirksam.

---

#### Hinweis

MD32436 \$MA\_JOG\_MAX\_JERK ist nur wirksam, wenn für die zu verfahrenen Maschinenachsen die axiale Ruckbegrenzung im JOG-Betrieb freigegeben wurde:

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE [<Achse>] == TRUE

Dies ist auch über die Programmierung von SOFTA (<Achse1>, <Achse2>, ...) im Teileprogramm möglich.

---



## Kanalspezifische Begrenzung von Beschleunigung und Ruck

### Maximale Beschleunigung beim Handfahren von Geometrieachsen

Die maximale Beschleunigung beim Handfahren von Geometrieachsen kann kanalspezifisch vorgegeben werden über das Maschinendatum:

MD21166 \$MC\_JOG\_ACCEL\_GEO [<Geometrieachse>]

mit <Geometrieachse> = 0, 1, 2

Bei MD21166 = 0 ist statt der kanalspezifischen Beschleunigungsbegrenzung der achsspezifische Grenzwert aus MD32301 \$MA\_JOG\_MAX\_ACCEL wirksam.

---

#### Hinweis

Für MD21166 \$MC\_JOG\_ACCEL\_GEO [<Geometrieachse>] gibt es keine direkte Begrenzung auf MD32300 \$MA\_MAX\_AX\_ACCEL.

---

#### Hinweis

Bei einer aktiven Transformation bestimmt MD32300 \$MA\_MAX\_AX\_ACCEL die maximal mögliche axiale Beschleunigung.

---

### Maximaler Ruck beim Handfahren von Geometrieachsen

Der maximale Ruck beim Handfahren von Geometrieachsen im Beschleunigungsmodus SOFT (Beschleunigung mit Ruckbegrenzung) kann kanalspezifisch vorgegeben werden über das Maschinendatum:

MD21168 \$MC\_JOG\_JERK\_GEO [<Geometrieachse>]

mit <Geometrieachse> = 0, 1, 2

Bei MD21168 = 0 ist statt der kanalspezifischen Ruckbegrenzung der achsspezifische Grenzwert aus MD32436 \$MA\_JOG\_MAX\_JERK wirksam.

---

#### Hinweis

MD21168 \$MC\_JOG\_JERK\_GEO ist nur wirksam, wenn für die zugrunde liegenden Maschinenachsen die axiale Ruckbegrenzung im JOG-Betrieb freigegeben wurde:

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE [<Achse>] == TRUE

---

### Maximaler Ruck beim Handfahren von Orientierungsachsen

Der maximale Ruck beim Handfahren von Orientierungsachsen kann kanalspezifisch vorgegeben werden über das Maschinendatum:

MD21158 \$MC\_JOG\_JERK\_ORI [<Orientierungsachse>]

Damit MD21158 wirksam werden kann, muss die kanalspezifische Ruckbegrenzung für das Handfahren von Orientierungsachsen über das folgende Maschinendatum freigegeben werden:

MD21159 \$MC\_JOG\_JERK\_ORI\_ENABLE == TRUE

---

### Hinweis

Orientierungsachsen sind von den Maschinendaten MD32301 \$MA\_JOG\_MAX\_ACCEL und MD32436 \$MA\_JOG\_MAX\_JERK nicht betroffen.

---

## 4.2.20.2 Randbedingungen

### Wegüberlagerung / Überlagerte Bewegungen

Bei Wegüberlagerung / überlagerten Bewegungen (z. B. DRF) sind die JOG-spezifischen Maximalwerte für Beschleunigung und Ruck (MD32301 \$MA\_JOG\_MAX\_ACCEL und MD32436 \$MA\_JOG\_MAX\_JERK) **nicht** wirksam. Es wirken stattdessen die Werte für Positionierachsbewegungen:

- Beschleunigung:
  - MD32300 \$MA\_MAX\_AX\_ACCEL [0] (maximale axiale Beschleunigung bei Bahnbewegungen im Dynamikmodus DYNNORM)  
  
Siehe auch Kapitel "Beschleunigung ohne Ruckbegrenzung (BRISK/BRISKA) (kanal-/achsspezifisch) (Seite 247)".
- Ruck
  - MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK (maximaler axialer Ruck bei Positionierachsbewegungen)  
  
bzw. (bei G75/751):
  - MD32431 \$MA\_MAX\_AX\_JERK [0] (maximaler axialer Ruck bei Bahnbewegungen im Dynamikmodus DYNNORM)  
  
Siehe auch Kapitel "Ruckbegrenzung bei Einzelachsinterpolation (SOFTA) (achsspezifisch) (Seite 266)".

---

### Hinweis

Für den JOG-Betrieb ist immer nur der Dynamikmodus DYNNORM wirksam.

---

### Verhalten beim Handfahren von Geometrieachsen bei aktiver Rotation

Beim Handfahren von Geometrieachsen im Beschleunigungsmodus SOFT (Beschleunigung mit Ruckbegrenzung) wird der Wert aus MD32436 \$MA\_JOG\_MAX\_JERK bzw. MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK auch bei aktiver Rotation oder aktivem orientierbaren Werkzeugträger verwendet.

### Teileprogrammanweisung SOFTA / BRISKA / DRIVEA

Die Teileprogrammanweisung `SOFTA (<Achse1>, <Achse2>, ...)` wirkt auch im JOG-Betrieb, d. h. für die angegebenen Achsen ist beim Verfahren im JOG-Betrieb der maximale axiale Ruck aus MD32436 \$MA\_JOG\_MAX\_JERK wirksam (genau wie beim Setzen von MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE [<Achse>] == TRUE).

---

#### Hinweis

Die Teileprogrammanweisung `SOFT` hat dagegen keine Auswirkung auf den JOG-Betrieb.

---

Wie `SOFTA` sind auch die Teileprogrammanweisungen `BRISKA` und `DRIVEA` im JOG-Betrieb wirksam, d. h. die Beschleunigung erfolgt ohne Ruckbegrenzung, auch wenn für die betroffenen Maschinenachsen MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE auf "TRUE" gesetzt ist.

---

#### Hinweis

Das Handfahren von Orientierungsachsen wird von `BRISKA/SOFTA/DRIVEA` nicht beeinflusst.

---

## 4.3 Beispiele

### 4.3.1 Beschleunigung

#### 4.3.1.1 Bahngeschwindigkeitsverlauf

##### Kernaussage

Im Folgenden wird beispielhaft ein Teileprogrammausschnitt mit dem dazugehörigen Verlauf der Bahngeschwindigkeit aufgezeigt, um daran zu erläutern, wie die Bahngeschwindigkeit aufgrund der verschiedenen Ereignisse und der sich daraus ergebenden Änderung der Beschleunigung angepasst wird.

##### Teileprogramm (Ausschnitt, schematisch)

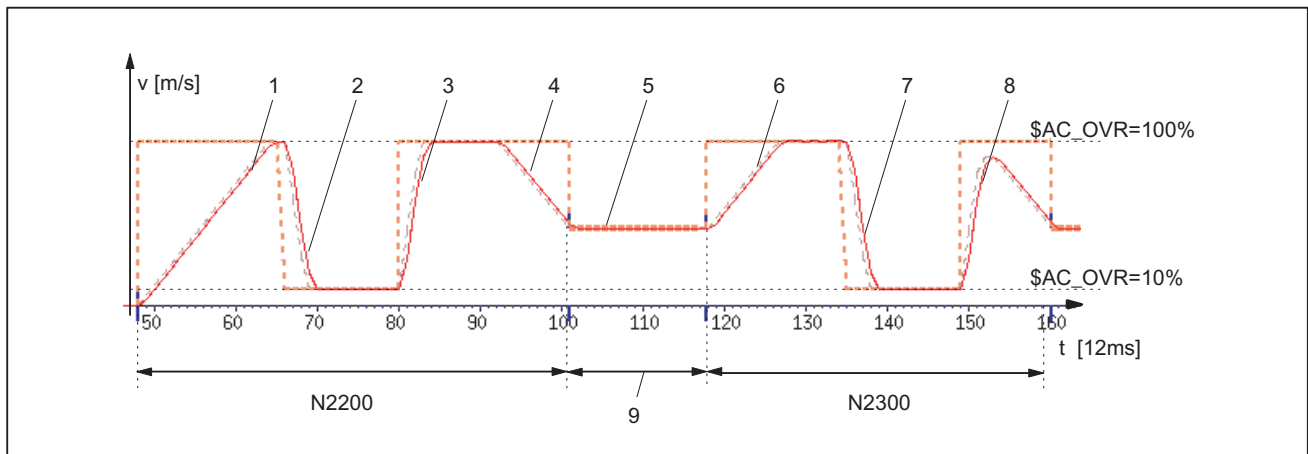
**Programmcode**

```
; Beschleunigungsumschaltung abhängig von schnellem Eingang 1 ($A_IN[1]):
N53 ID=1 WHEN $A_IN[1] == 1 DO $AC_PATHACC = 2.*$MA_MAX_AX_ACCEL[X]

; Test-Override-Profil (simuliert externen Eingriff):
N54 ID=2 WHENEVER ($AC_TIMEC > 16) DO $AC_OVR=10
N55 ID=3 WHENEVER ($AC_TIMEC > 30) DO $AC_OVR=100

; Anfahren
N1000 GO X0 Y0 BRISK
N1100 TRANS Y=-50
N1200 AROT Z=30 G642

; Kontur
N2100 X0 Y0
N2200 X = 70 G1 F10000 RNDM=10 ACC[X]=30 ACC[Y]=30
N2300 Y = 70
N2400 X0
N2500 Y0
M30
```



### Beschleunigungsprofil: BRISK

- 1: Beschleunigen auf 100% Bahngeschwindigkeit (F10000) gemäß Beschleunigungsvorgabe: ACC (N2200...)
- 2: Bremsen auf 10% Bahngeschwindigkeit aufgrund der Override-Änderung ( $\$AC\_OVR$ ) gemäß Echtzeitbeschleunigung  $\$AC\_PATHACC$  (N53/N54...)
- 3: Beschleunigen auf 100% Bahngeschwindigkeit aufgrund der Override-Änderung ( $\$AC\_OVR$ ) gemäß Echtzeitbeschleunigung  $\$AC\_PATHACC$  (N53/N55...)
- 4: Bremsen auf Satzendgeschwindigkeit für den Verrundungs-Zwischensatz gemäß Beschleunigungsvorgabe: ACC (N2200...)
- 5: Geschwindigkeitsbegrenzung aufgrund der Verrundungen (siehe 9)
- 6: Beschleunigen auf 100% Bahngeschwindigkeit ( $\$AC\_OVR$ ) gemäß Beschleunigungsvorgabe: ACC (N2300...)
- 7: Abbremsen aufgrund der Override-Änderung mit Beschleunigung gemäß Echtzeitbeschleunigung  $\$AC\_PATHACC$  (N53/N54...)
- 8: Beschleunigen auf 100% Bahngeschwindigkeit aufgrund der Override-Änderung ( $\$AC\_OVR$ ) gemäß Echtzeitbeschleunigung  $\$AC\_PATHACC$  (N53/N55...)
- 9: Steuerungsintern eingefügter Zwischensatz aufgrund der programmierten Verrundung (RNDM) (N2200...)

Bild 4-13 Umschalten zwischen im Vorlauf bestimmter Bahnbeschleunigung und Echtzeitbeschleunigung

## 4.3.2 Ruck

### 4.3.2.1 Bahngeschwindigkeitsverlauf

#### Kernaussage

Im Folgenden wird beispielhaft ein Teileprogrammausschnitt mit dem dazugehörigen Verlauf der Bahngeschwindigkeit aufgezeigt, um daran zu erläutern, wie die Bahngeschwindigkeit aufgrund der verschiedenen Ereignisse und der sich daraus ergebenden Änderung des Rucks angepasst wird.

#### Teileprogramm (Ausschnitt, schematisch)

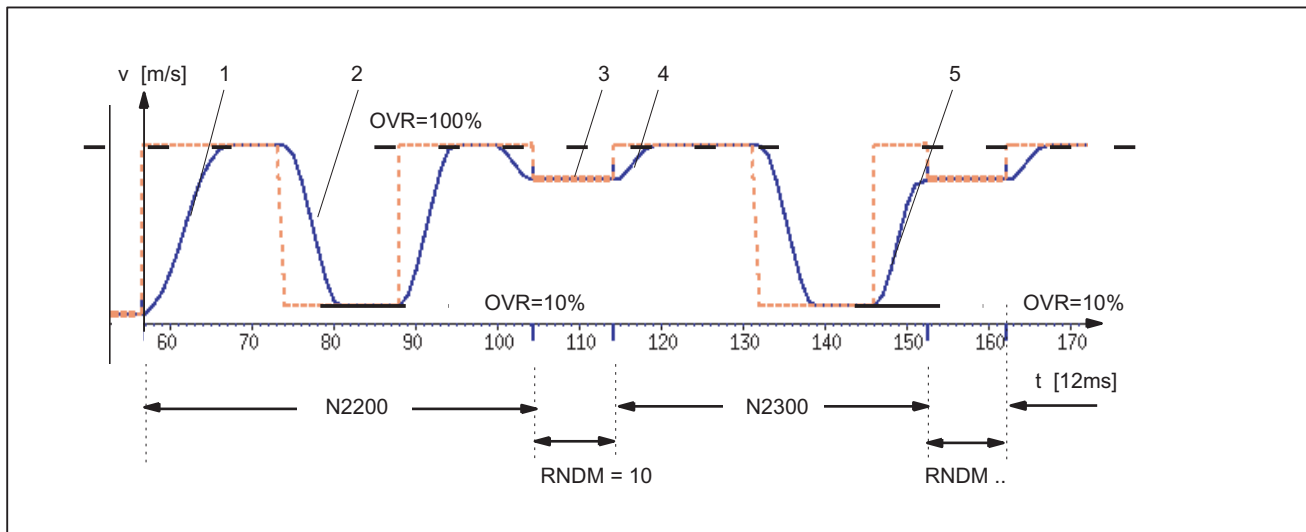
**Programmcode**

```
; Einstellen von Bahnbeschleunigung und Bahnruck bei externem Eingriff:
N0100 $AC_PATHACC = 0.
N0200 $AC_PATHJERK = 4. * ($MA_MAX_AX_JERK[X] + $MA_MAX_AX_JERK[Y]) / 2.

; Synchronaktionen zum Variieren des Overrides (simuliert externen Eingriff):
N53 ID=1 WHENEVER ($AC_TIMEC > 16) DO $AC_OVR=10
N54 ID=2 WHENEVER ($AC_TIMEC > 30) DO $AC_OVR=100

; Anfahren
N1000 GO X0 Y0 SOFT
N1100 TRANS Y=-50
N1200 AROT Z=30 G642

; Kontur
N2100 X0 Y0
N2200 X = 70 G1 F10000 RNDM=10
N2300 Y = 70
N2400 X0
N2500 Y0
M30
```



**Beschleunigungsprofil: SOFT**

- 1: Ruck gemäß \$MA\_MAX\_AX\_JERK[.]
- 2: Ruck gemäß \$AC\_PATHJERK
- 3: Ruck gemäß \$MA\_MAX\_AX\_JERK[.] (Anfahren der Satzendgeschwindigkeit)
- 4: Geschwindigkeitsbegrenzung wegen Kreisbogen
- 5: Ruck gemäß \$AC\_PATHJERK

Bild 4-14 Umschalten zwischen im Vorlauf bestimmtem Bahnrick und \$AC\_PATHJERK

### 4.3.3 Beschleunigung und Ruck

**Kernaussage**

Das folgende Beispiel zeigt anhand eines kurzen Teileprogramms den Verlauf von Geschwindigkeit und Beschleunigung der X-Achse und welche Geschwindigkeits- und Beschleunigungs-relevanten Maschinendaten für welchen Abschnitt der Kontur maßgeblich sind.

Teileprogramm

Programmcode	Kommentar
N90 F5000 SOFT G64	; Bahnsteuerbetrieb, rückbegrenzte Beschleunigung
N100 G0 X0 Y0 Z0	; Eilgang
N110 G1 X10	; Gerade
N120 G3 CR=5 X15 Y5	; Kreisbogen, Radius 5 mm, Satzübergang: tangential
N130 G3 CR=10 X5 Y15	; Kreisbogen, Radius 10 mm, Satzübergang: tangential
N140 G1 X-5 Y17.679	; Gerade, 15° Knick
N200 M30	

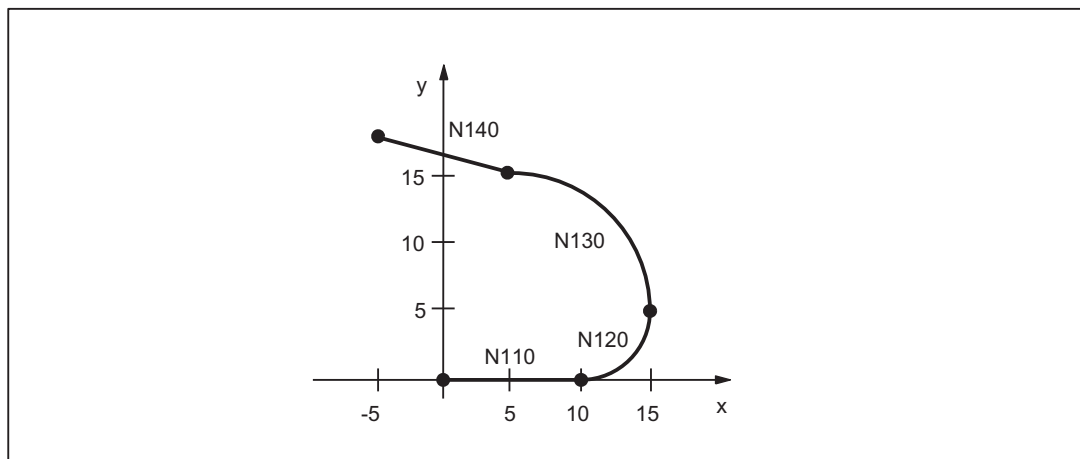


Bild 4-15 Kontur des Teileprogramms



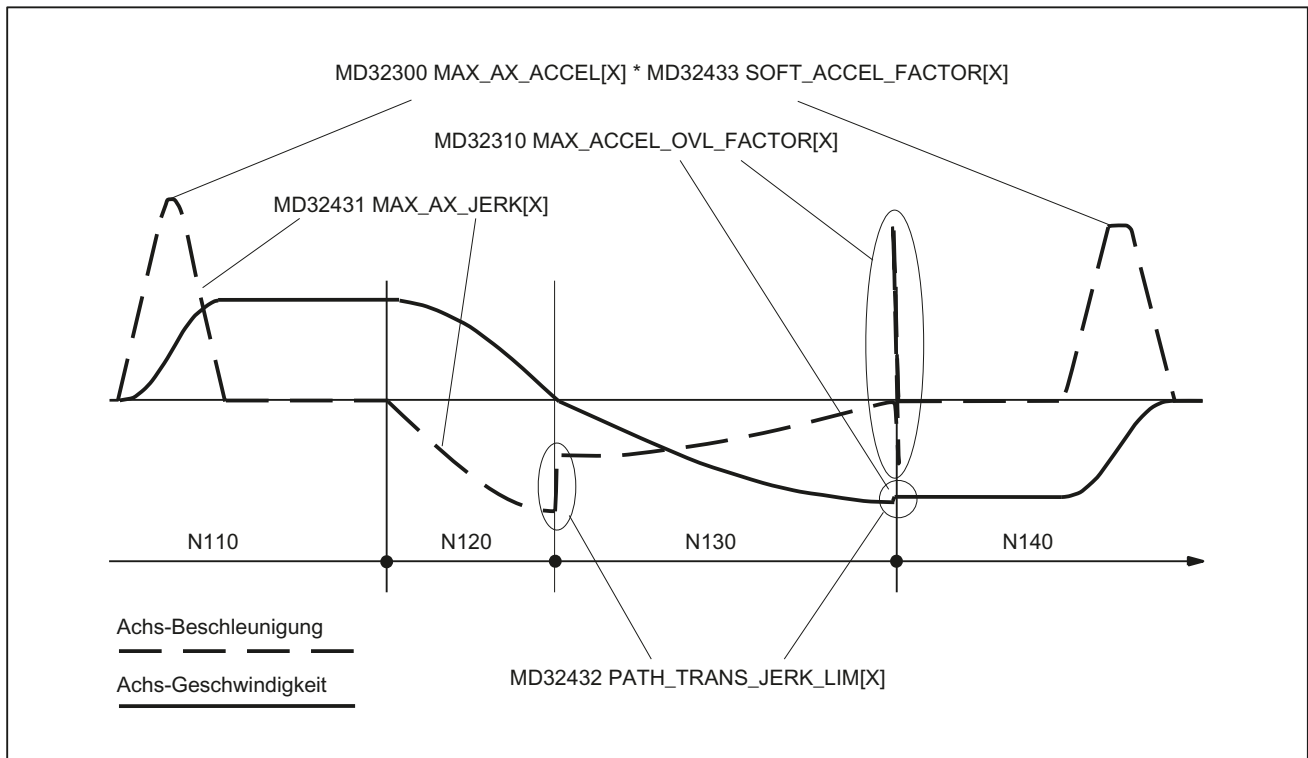


Bild 4-16 X-Achse: Verlauf von Geschwindigkeit und Beschleunigung

## 4.3.4 Geknickte Beschleunigungskennlinie

### 4.3.4.1 Aktivierung

#### Kernaussage

Das Beispiel zeigt die Aktivierung der geknickten Beschleunigungskennlinie anhand von:

- Maschinendaten
- Teileprogrammanweisung

**Maschinendaten**

- Parametrieren der Kennlinie (beispielhaft)

```

Programmcode
-----
X-Achse
MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT[X] = 0.4
MD35230 $MA_ACCEL_REDUCTION_FACTOR[X] = 0.85
MD35242 $MA_ACCEL_REDUCTION_TYPE[X] = 2
MD35240 $MA_ACCEL_TYPE_DRIVE[X] = TRUE

Y-Achse
MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT[Y] = 0.0
MD35230 $MA_ACCEL_REDUCTION_FACTOR[Y] = 0.6
MD35242 $MA_ACCEL_REDUCTION_TYPE[Y] = 1
MD35240 $MA_ACCEL_TYPE_DRIVE[Y] = TRUE

Z-Achse
MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT[Z] = 0.6
MD35230 $MA_ACCEL_REDUCTION_FACTOR[Z] = 0.4
MD35242 $MA_ACCEL_REDUCTION_TYPE[Z] = 0
MD35240 $MA_ACCEL_TYPE_DRIVE[Z] = FALSE
    
```

- Aktivierung durch Vorgabe als kanalspezifische Grundstellung  
 MC\_GCODE\_RESET\_VALUE[20] = 3 (DRIVE)

**Teileprogramm (Ausschnitt)**

Programmcode	Kommentar
N10 G1 X100 Y50 Z50 F700	; Bahnbewegung (X,Y, Z) mit DRIVE
N15 Z20	; Bahnbewegung (Z) mit DRIVE
N20 BRISK	; Umschaltung auf BRISK
N25 G1 X120 Y70	; Bahnbewegung (Y, Z) mit Ersatzkennlinie
N30 Z100	; Bahnbewegung (Z) mit BRISK
N35 POS[X] = 200 FA[X] = 500	; Positionierbewegung (X) mit DRIVEA
N40 BRISKA(Z)	; Aktivieren BRISKA für Z
N40 POS[Z] = 50 FA[Z] = 200	; Positionierbewegung (Z) mit BRISKA
N45 DRIVEA(Z)	; Aktivieren DRIVEA für Z
N50 POS[Z] = 100	; Positionierbewegung (Z) mit DRIVE
N55 BRISKA(X)	; führt zu Fehlermeldung
...	

## 4.4 Datenlisten

### 4.4.1 Maschinendaten

#### 4.4.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
18960	POS_DYN_MODE	Art der Positionierachsdynamik

#### 4.4.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen
20500	CONST_VELO_MIN_TIME	Minimale Zeit mit konstanter Geschwindigkeit
20600	MAX_PATH_JERK	Bahnbezogener Maximalruck
20602	CURV_EFFECT_ON_PATH_ACCEL	Einfluss der Bahnkrümmung auf Bahndynamik
20610	ADD_MOVE_ACCEL_RESERVE	Beschleunigungsreserve für überlagerte Bewegungen
21158	JOG_JERK_ORI	Maximaler Ruck der Orientierungsachsen beim Verfahren in JOG
21159	JOG_JERK_ORI_ENABLE	Grundstellung der kanalspezifischen Ruckbegrenzung von Orientierungsachsen beim Verfahren in JOG
21166	JOG_ACCEL_GEO	Maximale Beschleunigung der Geometrieachsen beim Verfahren in JOG
21168	JOG_JERK_GEO	Maximaler Ruck der Geometrieachsen beim Verfahren in JOG

## 4.4.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32000	MAX_AX_VELO	Maximale Achsgeschwindigkeit
32300	MAX_AX_ACCEL	Maximale Achsbeschleunigung
32301	JOG_MAX_ACCEL	Maximale Beschleunigung im JOG-Betrieb
32310	MAX_ACCEL_OVL_FACTOR	Überlastfaktor für Geschwindigkeitssprung
32320	DYN_LIMIT_RESET_MASK	Resetverhalten von Dynamikbegrenzungen
32400	AX_JERK_ENABLE	Axiale Ruckbegrenzung
32402	AX_JERK_MODE	Filtertyp für axiale Ruckbegrenzung
32410	AX_JERK_TIME	Zeitkonstante für den axialen Ruckfilter
32420	JOG_AND_POS_JERK_ENABLE	Grundstellung der axialen Ruckbegrenzung
32430	JOG_AND_POS_MAX_JERK	Axialer Ruck bei Einzelachsbewegungen
32431	MAX_AX_JERK	Maximaler axialer Ruck am Satzübergang im Bahnsteuerbetrieb
32432	PATH_TRANS_JERK_LIM	Max. axialer Ruck einer Geo-Achse an Satzgrenze
32433	SOFT_ACCEL_FACTOR	Skalierung der Beschleunigungsbegrenzung bei <i>SOFT</i>
32434	G00_ACCEL_FACTOR	Skalierung der Beschleunigungsbegrenzung bei <i>G00</i>
32435	G00_JERK_FACTOR	Skalierung der axialen Ruckbegrenzung bei <i>G00</i>
32436	JOG_MAX_JERK	Maximaler axialer Ruck bei JOG-Bewegung
32437	AX_JERK_VELO	Erste Geschwindigkeitsschwelle für die geschwindigkeitsabhängige Ruckadaption
32438	AX_JERK_VEL1	Zweite Geschwindigkeitsschwelle für die geschwindigkeitsabhängige Ruckadaption
32439	MAX_AX_JERK_FACTOR	Faktor zur Einstellung des maximalen Rucks bei höheren Geschwindigkeiten (geschwindigkeitsabhängige Ruckadaption)
35220	ACCEL_REDUCTION_SPEED_POINT	Drehzahl für reduzierte Beschleunigung
35230	ACCEL_REDUCTION_FACTOR	Reduzierte Beschleunigung
35240	ACCEL_TYPE_DRIVE	Beschleunigungskennlinie <i>DRIVE</i> für Achsen Ein/Aus
35242	ACCEL_REDUCTION_TYPE	Art der Beschleunigungsreduktion

## 4.4.2 Settingdaten

### 4.4.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42500	SD_MAX_PATH_ACCEL	Max. Bahnbeschleunigung
42502	IS_SD_MAX_PATH_ACCEL	Auswerten von SD 42500: Ein/Aus
42510	SD_MAX_PATH_JERK	Max. bahnbezogener Ruck
42512	IS_SD_MAX_PATH_JERK	Auswerten von SD 42510: Ein/Aus

## 4.4.3 Systemvariablen

Bezeichner	Beschreibung
\$AC_PATHACC	Bahnbeschleunigung für Echtzeitereignisse
\$AC_PATHJERK	Bahnruck für Echtzeitereignisse



## F1: Fahren auf Festanschlag

### 5.1 Kurzbeschreibung

#### Funktion

Mit der Funktion "Fahren auf Festanschlag" können bewegliche Maschinenteile, z. B. Reitstock oder Pinole, mit einem definierten Moment gegen andere Maschinenteile gefahren werden.

#### Merkmale

- Das Klemmmoment und ein Festanschlags-Überwachungsfenster sind im Teileprogramm programmierbar und, nachdem der Festanschlag erreicht wurde, auch über Settingdaten änderbar.
- Fahren auf Festanschlag ist für Achsen und Spindeln möglich.
- Fahren auf Festanschlag ist für mehrere Achsen gleichzeitig und parallel zum Verfahren anderer Achsen möglich.
- In Synchronaktionen kann die Reduzierung des Klemmmoments oder Klemmkraft ein- und ausgeschaltet werden.
- Die Funktionen des Fahrens auf Festanschlag können auch über Synchronaktionen aktiviert werden.
- Es ist ein mehrkanaliger Satzsuchlauf mit Berechnung aller erforderlichen Zusatzdaten möglich (SERUPRO).
- Die Achsen können simuliert mit FXS und FOC gefahren werden.
- "Hängende" Achsen können bei FXS-Alarmen auf Festanschlag gefahren werden.

## 5.2 Allgemeine Funktionalität

### 5.2.1 Programmierung

#### Funktion

##### Fahren auf Festanschlag

Über die Befehle `FXS`, `FXST` und `FXSW` wird die Funktion "Fahren auf Festanschlag" gesteuert.

Die Aktivierung kann auch ohne Verfahrbewegung der betreffenden Achse erfolgen. Das Moment wird sofort begrenzt. Auf Anschlag wird überwacht, sobald die Achse verfahren wird.

---

##### Hinweis

##### Synchronaktionen

Die Funktion "Fahren auf Festanschlag" kann auch über Synchronaktionen gesteuert werden.

##### Literatur:

Funktionshandbuch Synchronaktionen

---

##### Fahren mit begrenztem Moment / Kraft

Über die Befehle `FOCON`, `FOCOF` und `FOC` wird das Verfahren mit begrenztem Moment / Kraft gesteuert (siehe Kapitel "Fahren mit begrenztem Moment/Kraft FOC (Seite 323)").

#### Syntax

```
FXS [<Achse>] = <Anforderung>  
FXST [<Achse>] = <Klemmmoment>  
FXSW [<Achse>] = <Fensterbreite>  
FOCON [<Achse>]  
FOCOF [<Achse>]  
FOC [<Achse>]
```



## Bedeutung

Parameter	Bedeutung
FXS:	Fahren auf Festanschlag Wirksamkeit: modal
<Anforderung>:	Anforderung an die Funktion "Fahren auf Festanschlag": 0 = Ausschalten 1 = Einschalten
FXST:	Klemmmoment einstellen
<Klemmmoment>:	Klemmmoment in % des maximalen Moments des Antriebs. SINAMICS S120: p2003
FXSW:	Überwachungsfensters einstellen
<Fensterbreite>:	Breite des Toleranzfensters um den Festanschlag Einheit: mm, inch oder Grad
FOCON:	Moment / Kraft-Begrenzung: einschalten
FOCOF:	Moment / Kraft-Begrenzung: ausschalten
FOC:	Satzweise wirksame Moment/Kraft-Begrenzung
<Achse>:	Bezeichner der Kanalachse Typ: AXIS

### Achsbezeichner

Statt eines Maschinenachsbezeichners können die Bezeichner von Geometrie- oder Zusatzachsen verwendet werden, wenn dieser genau eine Maschinenachse zugeordnet ist und für die Maschinenachse nicht eine der folgenden Funktionen aktiv ist:

- Transformation
- Kopplung
- Frame

### Klemmmoment ändern

Mit dem Befehl `FXST` kann das Klemmmoment verändert werden. Die Änderung wird vor den Verfahrbewegungen des Satzes wirksam. Die Momentenbegrenzung wirkt zusätzlich zur Beschleunigungsbegrenzung (`ACC`).

### Rampenförmige Änderung

Über das folgende Maschinendatum kann eine Zeit eingestellt werden, in der die Änderung des Klemmmoments erfolgen soll. Dadurch erfolgt die Änderung des Klemmmoments nicht mehr sprunghaft sondern rampenförmig.

MD37012 \$MA\_FIXED\_STOP\_TORQUE\_RAMP\_TIME (Zeitdauer bis zum Erreichen der neuen Momentengrenze)

### Überwachungsfenster ändern

Mit dem Befehl `FXSW` kann das Überwachungsfenster verändert werden. Die Änderungen werden vor den Verfahrbewegungen des Satzes wirksam.

Wird ein neues Überwachungsfenster programmiert, ändert sich nicht nur die Fensterbreite. Hat sich die Achse vor der Änderung bewegt, ändert sich auch der Bezugspunkt für die Fenstermitte auf die aktuelle Istposition der Achse.

### Funktionsanwendung bei aktivem Bahnsteuerbetrieb G64

Über das folgende Maschinendatum kann eingestellt werden, dass durch die Anwahl der Funktion (`FXS`) während aktivem Bahnsteuerbetrieb (`G64`) kein Genauhalt zum Satzwechsel (`G60`) ausgelöst wird:

MD37060 \$MA\_FIXED\_STOP\_ACKN\_MASK (Beachtung von PLC-Quittierungen für Fahren auf Festanschlag)

Bit	Wert	Bedeutung
0	0	Start der Verfahrbewegung ohne Quittierung durch die PLC
	1	Start der Verfahrbewegung nach Quittierung durch die PLC

### Randbedingungen

- Die Verfahrbewegung zum Festanschlag kann als Bahn- oder satzbezogene oder satzübergreifende Positionierachsbewegung programmiert werden.
- Fahren auf Festanschlag kann gleichzeitig für mehrere Maschinenachsen erfolgen.
- Im Teileprogramm muss der Verfahrweg und das Aktivieren der Funktion in einem Satz programmiert werden.
- Wird das Fahren auf Festanschlag über die Synchronaktionen aktiviert, können die Programmierung des Verfahrwegs und das Aktivieren der Funktion in getrennten Sätzen erfolgen.

## Beispiel

Voraussetzung: Abbildung der Kanalachse X auf Maschinenachse X1

Programmierung mit Kanalachsbezeichner

Programmcode	Kommentar
FXS[X] = 1	; Anwahl für X -> X1
FXST[X] = 12.3	; neues Moment X -> X1
FXSW[X] = 2000	; neues Fenster für X -> X1

Programmierung mit Maschinenachsbezeichner

Programmcode	Kommentar
FXS[X1] = 1	; Anwahl für X1
FXST[X1] = 12.3	; neues Moment für X1
FXSW[X1] = 2000	; neues Fenster für X1

### Literatur:

Programmierhandbuch Grundlagen

## 5.2.2 Funktionsablauf

### 5.2.2.1 Anwahl

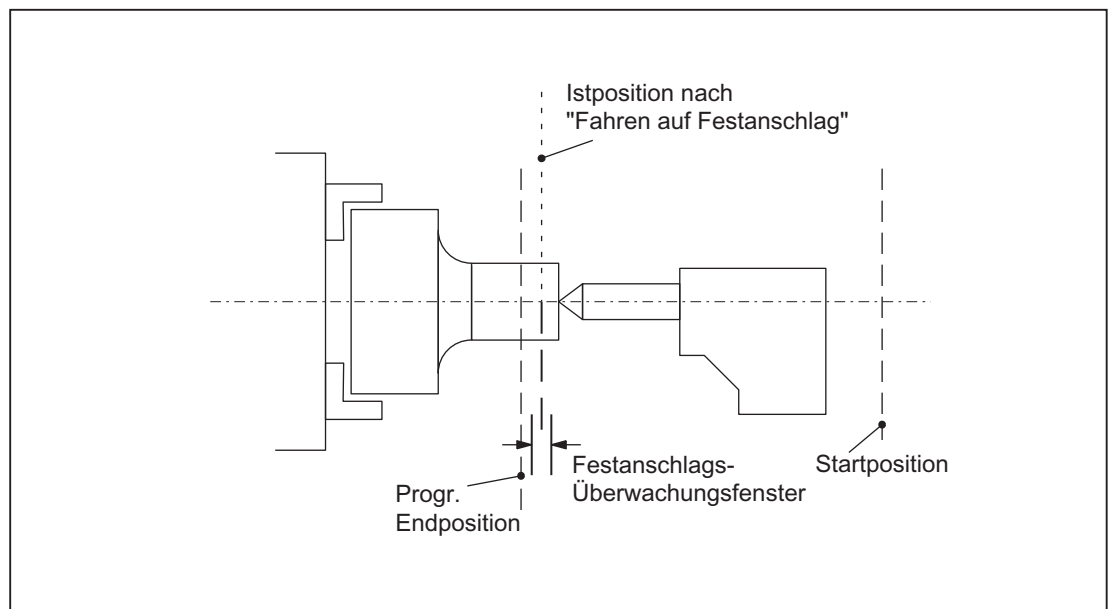


Bild 5-1 Beispiel für Fahren auf Festanschlag

**Vorgang**

Die NC erkennt die Funktionsanwahl "Fahren auf Festanschlag" über den Befehl `FXS [x] =1` und meldet der PLC durch das NST DB31, ... DBX62.4 ("Fahren auf Festanschlag aktivieren"), dass die Funktion angewählt wurde.

Wenn das Maschinendatum:

MD37060 \$MA\_FIXED\_STOP\_ACKN\_MASK (Beachtung von PLC-Quittierungen für Fahren auf Festanschlag)

entsprechend gesetzt ist, wird die Quittierung der PLC durch das NST DB31, ... DBX3.1 ("Fahren auf Festanschlag freigeben") abgewartet.

Anschließend wird von der Startposition aus mit der programmierten Geschwindigkeit auf die programmierte Zielposition gefahren. Der Festanschlag muss sich zwischen der Start- und Zielposition der Achse/Spindel befinden. Eine programmierte Momentenbegrenzung (Klemmmoment vorgegeben über `FXST [<Achse>]`) wirkt ab Satzbeginn, d.h. auch das Anfahren des Anschlags erfolgt mit reduziertem Moment. Dies wird in der NC durch automatische Reduzierung der Beschleunigung berücksichtigt.

Wurde in dem Satz oder seit Programmbeginn kein Moment programmiert, so gilt der Wert, der in das achsspezifische Maschinendatum:

MD37010 \$MA\_FIXED\_STOP\_TORQUE\_DEF (Voreinstellung Festanschlag-Klemmmoment)

eingetragen ist.

**5.2.2.2 Festanschlag wird erreicht**

**Vorgang**

Hat die Achse den Festanschlag erreicht, wird das Moment im Antrieb bis zum programmierten Klemmmoment erhöht. Wie die Steuerung das Erreichen des Festanschlags erkennt, kann über folgendes Maschinendatum eingestellt werden:

MD37040 \$MA\_FIXED\_STOP\_BY\_SENSOR = <Wert> (Festanschlagserkennung über Sensor)

<Wert>	Bedeutung
0	Der Festanschlag ist erreicht, wenn die axiale Konturabweichung (Differenz zwischen tatsächlichem und erwartetem Schleppabstand) den eingestellten Wert im Maschinendatum überschritten hat: MD37030 \$MA_FIXED_STOP_THRESHOLD (Schwelle für Festanschlagserkennung)
1	Das Erreichen des Festanschlags wird von einem externen Sensor erkannt und über folgendes axiale NC/PLC-Nahtstellensignal der Steuerung mitgeteilt: DB31, ... DBX1.2 == 1 (Sensor Festanschlag)
2	Der Festanschlag ist erreicht, wenn die Bedingung entsprechend Wert == 0 oder Wert == 1 zutreffen.

### Unwirksame NC/PLC-Nahtstellensignale

Ist die Achse im Zustand "Festanschlag erreicht", sind folgende NC/PLC-Nahtstellensignale unwirksam:

- DB31, ... DBX1.3 (Achsen-/Spindelsperre)
- DB31, ... DBX2.1 (Reglerfreigabe)

### Interne Abläufe

Nachdem der Zustand "Festanschlag erreicht" von der NC erkannt wurde, wird der Restweg gelöscht und der Lagesollwert wird nachgeführt. Die Reglerfreigabe bleibt aktiv.

Anschließend wird die PLC durch das NST DB31, ... DBX62.5 ("Festanschlag erreicht") informiert.

Wenn das Maschinendatum:

MD37060 \$MA\_FIXED\_STOP\_ACKN\_MASK

entsprechend gesetzt ist, wird die Quittierung der PLC durch das NST DB31, ... DBX1.1 ("Festanschlag erreicht quittieren") abgewartet.

Danach führt die NC einen Satzwechsel durch bzw. betrachtet die Positionierbewegung als beendet, lässt aber weiterhin einen Sollwert anstehen, damit das Klemmmoment wirken kann.

Nachdem der Festanschlag erreicht wurde, wird die Festanschlags-Überwachung aktiviert.

### Überwachungsfenster

Wurde in dem Satz oder seit Programmbeginn kein Festanschlags-Überwachungsfenster programmiert, so gilt der Wert der in das Maschinendatum:

MD37020 \$MA\_FIXED\_STOP\_WINDOW\_DEF

(Voreinstellung für Festanschlags-Überwachungsfenster)

eingetragen ist.

Verlässt die Achse die Position, die sie beim Erkennen des Anschlags hatte, um mehr als das gewählte Fenster, so wird der Alarm 20093 "Festanschlags-Überwachung hat angesprochen" ausgelöst und die Funktion "Fahren auf Festanschlag" abgewählt.

Das Fenster muss vom Anwender so gewählt werden, dass nur ein Wegbrechen des Anschlags zur Auslösung führt.

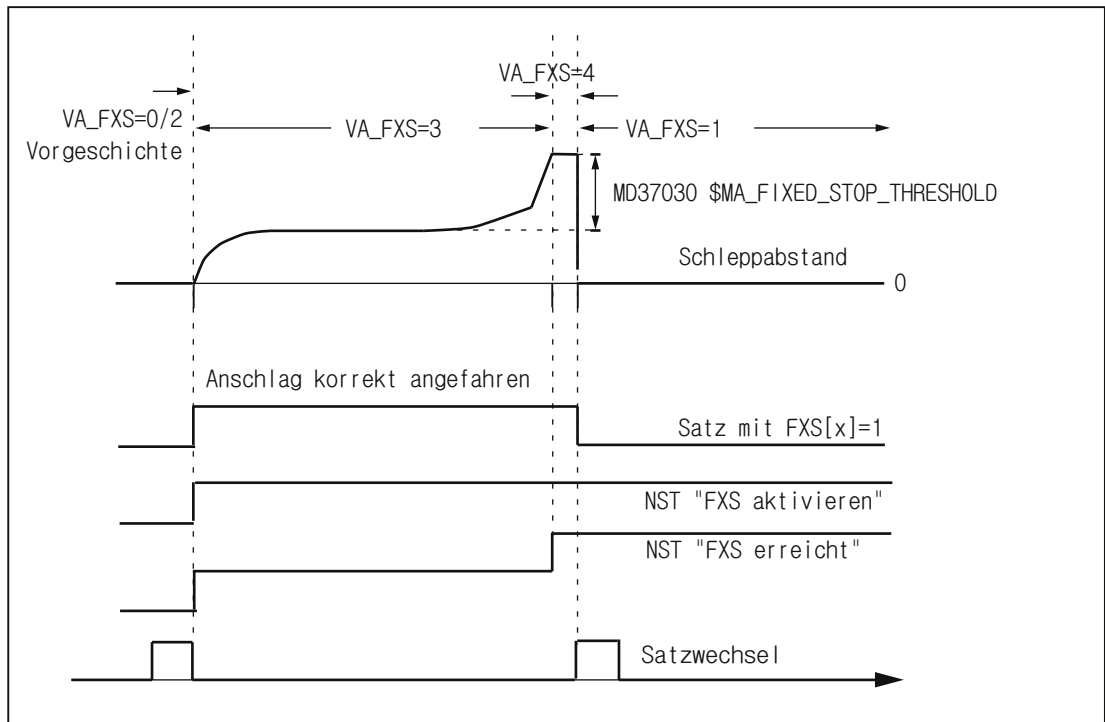


Bild 5-2 Festanschlag wird erreicht

### 5.2.2.3 Festanschlag wird nicht erreicht

#### Vorgang

Wird die programmierte Endposition erreicht, ohne dass der Zustand "Festanschlag erreicht" erkannt wurde, so wird abhängig vom Zustand des Maschinendatums:

MD37050 \$MA\_FIXED\_STOP\_ALARM\_MASK = 0 (Freigabe Festanschlagsalarme)

der Alarm 20091 "Festanschlag nicht erreicht" ausgegeben.

#### Abbruch ohne Alarm

Das Fahren auf Festanschlag kann von der PLC aus im Anfahrtsatz ohne einen Alarm auszulösen (zum Beispiel beim Eintreffen eines Tastendruckes des Bedieners) abgebrochen werden, wenn im Maschinendatum:

MD37050 \$MA\_FIXED\_STOP\_ALARM\_MASK = 2

der Alarm 20094 unterdrückt wird.

Sowohl bei "Festanschlag nicht erreicht" als auch bei "Festanschlag abgebrochen" wird die Funktion "Fahren auf Festanschlag" abgewählt.

### Funktion wird abgebrochen

Wird die Funktion "Fahren auf Festanschlag" wegen des Auftretens einer Impulssperre, Wegnahme der PLC-Quittungen oder Reset im Anfahrtsatz abgebrochen, kann die Anzeige bzw. Unterdrückung des Alarms 20094 über das Maschinendatum:

MD37050 \$MA\_FIXED\_STOP\_ALARM\_MASK = 3

gesteuert werden.

Alle anderen Werte  $\leq 7$  unterdrücken keine Alarmer.

Mit dem Teileprogramm-Befehl `NEWCONF` kann eine neue Einstellung aktiviert werden.

### Ablauf bei Störung oder Abbruch

Rücksetzen des axialen NC/PLC-Nahtstellensignals:

DB31, ... DBX62.4 = 0 (Fahren auf Festanschlag aktivieren)

Abhängig von der Einstellung in Maschinendatum:

MD37060 \$MA\_FIXED\_STOP\_ACKN\_MASK

wird die Quittierung abgewartet:

DB31, ... DBX3.1 == 0 (Fahren auf Festanschlag freigeben).

Anschließend wird die Momentenbegrenzung aufgehoben und ein Satzwechsel durchgeführt.

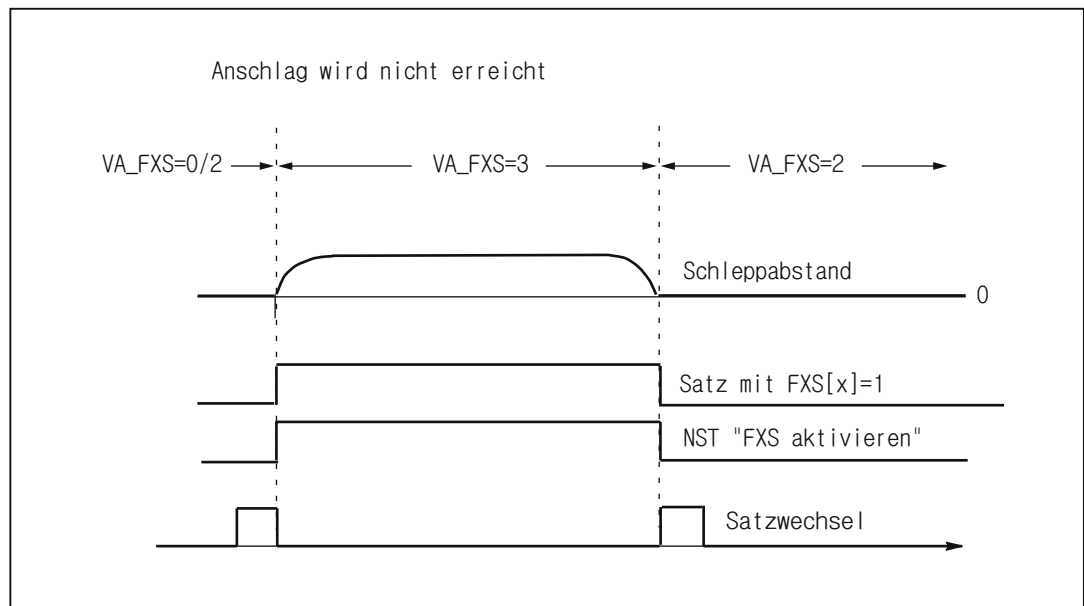


Bild 5-3 Festanschlag wird nicht erreicht

5.2.2.4 Abwahl

Vorgang

Mit der Abwahl der Funktionsabwahl  $FXS[<Achse>]=0$  wird ein Vorlaufstop (STOPRE) ausgelöst. Die Momentenbegrenzung und die Überwachung des Festanschlags-Überwachungsfensters werden aufgehoben.

Die NC/PLC-Nahtstellensignale werden zurückgesetzt:

DB31, ... DBX62.4 = 0 (Fahren auf Festanschlag aktivieren)

DB31, ... DBX62.5 = 0 (Festanschlag erreicht)

Abhängig vom Maschinendatum:

MD37060 \$MA\_FIXED\_STOP\_ACKN\_MASK

wird die Quittierung abgewartet:

DB31, ... DBX3.1 == 0 (Fahren auf Festanschlag freigeben)

DB31, ... DBX1.1 == 0 (Festanschlag erreicht quittieren)

Anschließend geht die Achse in Lageregelung. Das Nachführen des Lagesollwertes wird beendet und auf die neue Istposition synchronisiert. Die Achse kann wieder frei verfahren werden.

Verhalten bei Wegnahme der Impulsfreigabe

Eine Wegnahme der Impulsfreigabe bzw. Impulssperre kann erfolgen über:

- Antrieb: über Klemme EP (Enable Pulses)
- NC/PLC-Nahtstellensignal: DB31, ... DBX21.7 ("Impulsfreigabe")

Über das folgende Maschinendatum kann das Verhalten am Festanschlag eingestellt werden:

MD37002 \$MA\_FIXED\_STOP\_CONTROL (Ablaufkontrolle für Fahren auf Festanschlag)

Bit	Wert	Bedeutung
0	Verhalten bei Impulssperre am Anschlag	
	0	Fahren auf Festanschlag wird abgebrochen
1	1	Fahren auf Festanschlag wird unterbrochen, d.h. der Antrieb wird kraftlos. Sobald die Impulssperre wieder aufgehoben wird, drückt der Antrieb wieder mit dem Anschlagsmoment. Steuerung der Momentenaufschaltung siehe Bit 1.
	Verhalten bei Impulsfreigabe am Anschlag	
	0	Das Moment wird sprunghaft aufgeschaltet
1	1	Das Moment wird rampenförmig aufgeschaltet Siehe auch: MD37012 \$MA_FIXED_STOP_TORQUE_RAMP_TIME

Wird während einer aktiven Abwahl der Funktion (Zustand: "warten auf PLC-Quittungen") die Impulsfreigabe gelöscht, wird die Momentengrenze auf Null abgesenkt. In dieser Phase wird beim erneuten Setzen der Impulsfreigabe kein Moment mehr aufgebaut. Sobald die Abwahl vollzogen ist, kann wieder normal verfahren werden.



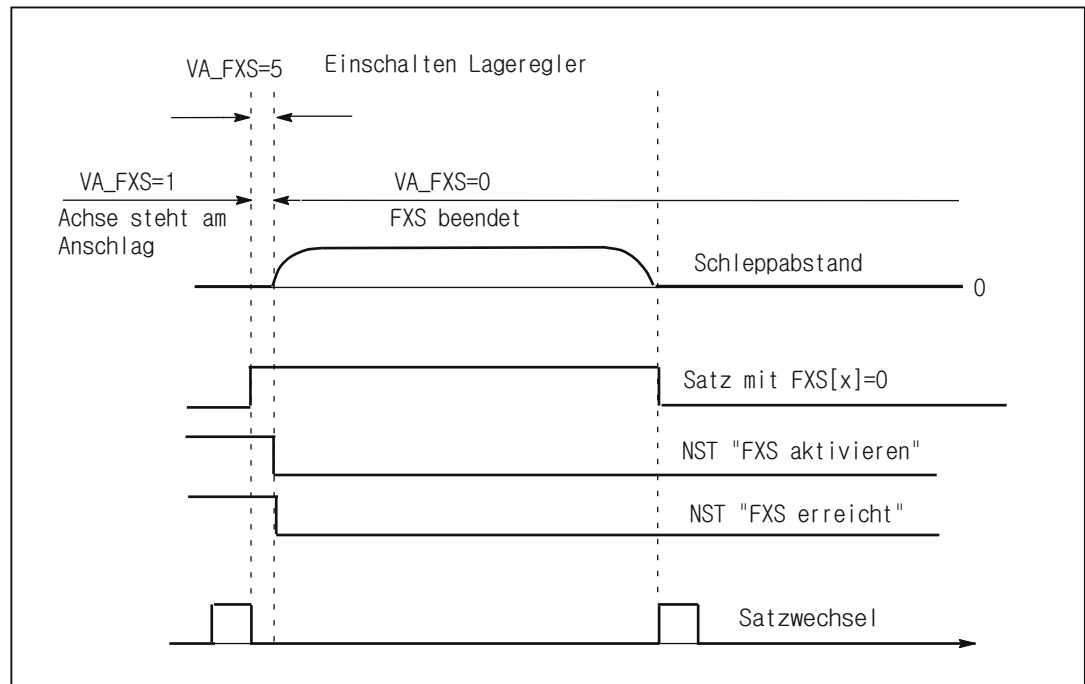


Bild 5-4 Festanschlag Abwahl

### 5.2.3 Verhalten bei Satzsuchlauf

#### Satzsuchlauf mit Berechnung

- Wenn der Zielsatz in einem Programmabschnitt liegt, in dem die Achse am Festanschlag stehen soll, wird der Festanschlag angefahren, wenn dieser nicht bereits erreicht ist.
- Wenn der Zielsatz in einem Programmabschnitt liegt, in dem die Achse nicht am Festanschlag stehen soll, wird der Festanschlag verlassen, wenn die Achse noch dort steht.
- Ist die Achse im Zustand "Festanschlag erreicht", wird die Meldung 10208 "Zur Programmfortsetzung NC-Start geben" angezeigt. Mit NC-Start kann das Programm fortgesetzt werden.
- Klemmmoment  $FXST$  und Überwachungsfenster  $FXSW$  haben beim Start des Zielsatzes jeweils den Wert, den sie auch bei normaler Programmbearbeitung haben.

#### Satzsuchlauf ohne Berechnung

Die Befehle  $FXS$ ,  $FXST$  und  $FXSW$  werden während des Suchlaufs ignoriert.

#### Wirksamkeit von FOCON/FOCOF

Der Zustand der modal wirkenden Momenten/Kraft-Reduzierung  $FOCON/FOCOF$  wird während des Suchlaufs mitgeführt und ist im Anfahrtsatz wirksam.

**Satzsuchlauf mit FXS oder FOC**

Der Anwender markiert in einem Programmbereich des Zielsatzes die Funktion `FXS` oder `FOC`, um alle zuletzt gültigen Zustände und Funktionen dieser Bearbeitung zu erfassen. Die NC startet selbsttätig das angewählte Programm im Modus Programmtest. Nach gefundenem Zielsatz stoppt die NC am Beginn des Zielsatzes, wählt intern Programmtest wieder ab, und zeigt anschließend die Stopp-Bedingung "Suchziel gefunden" in ihrer Satzanzeige an.

Befindet sich zwischen Programmanfang und Suchziel die Anweisung `FXS`, wird diese von der NC nicht real ausgeführt. Die Bewegung wird bis zum programmierten Endpunkt nur simuliert.

 <b>VORSICHT</b>
SERUPRO-Anfahren berücksichtigt die Anweisung <code>FXS</code> nicht real. Die programmierte Endposition des <code>FXS</code> -Satzes wird <b>ohne Momentenbegrenzung</b> nur simuliert angefahren.

Der Anwender kann im Teileprogramm das Ein- und Ausschalten von `FXS` mitprotokollieren. Bei Bedarf kann der Benutzer ein ASUP starten, um `FXS` in diesem SERUPRO-ASUP ein- oder auszuschalten.

**Systemvariable**

Der Zustand der Funktion kann über folgende Systemvariable gelesen werden:

- `$AA_FXS` (Sollzustand)
- `$VA_FXS` (Istzustand)

Wert	Beschreibung
0	Achse nicht im Anschlag
1	Anschlag wurde erfolgreich angefahren
2	Anfahren des Festanschlags fehlgeschlagen
3	Anwahl Fahren auf Festanschlags aktiv
4	Anschlag wurde erkannt
5	Abwahl Fahren auf Festanschlag aktiv

**SERUPRO: \$AA\_FXS**

Während SERUPRO liefert `$AA_FXS`, abhängig vom Aktivierungsstatus der Funktion "Fahren auf Festanschlag", folgende Werte:

FXS[<Achse>]=<Status>	
<Status>	\$AA_FXS
0 (Ausschalten)	0 (Achse nicht im Anschlag)
1 (Einschalten)	3 (Anwahl Fahren auf Festanschlags aktiv)

Außerhalb von SERUPRO liefern beide Systemvariable immer den gleichen Wert.

#### Hinweis

Während SERUPRO liefert die Systemvariable \$AA\_FXS nur die Werte 0 und 3. Dadurch kann bei Programmverzweigungen, basierend auf \$AA\_FXS, der Programmablauf bei SERUPRO gegenüber der normalen Programmabarbeitung verändert werden.

#### SERUPRO: \$VA\_FXS

Während SERUPRO liefert die Variable \$VA\_FXS immer den realen Zustand der Achse an der Maschine.

### SERUPRO-ASUP: Vergleich Soll- / Ist-Zustand

Im SERUPRO-ASUP kann über die Systemvariablen \$AA\_FXS und \$VA\_FXS der aktuelle Zustand der Funktion "Fahren auf Festanschlag" ermittelt und entsprechend reagiert werden:

FXS\_SERUPRO\_ASUP.MPF

Programmcode
N100 WHEN (\$AA_FXS[X]==3) AND (\$VA_FXS[X]==0) DO FXS[X]=1
N200 WHEN (\$AA_FXS[X]==0) AND (\$VA_FXS[X]==1) DO FXS[X]=0
N1020 REPOSA

### REPOS-Verschiebung anzeigen

Nachdem das Suchziel gefunden ist, wird für jede Achse der an der Maschine herrschende FXS-Zustand über folgende axialen NC/PLC-Nahtstellensignale angezeigt:

- DB31, ... DBX62.4 (Fahren auf Festanschlag aktivieren)
- DB31, ... DBX62.5 (Festanschlag erreicht)

#### Beispiel:

Wenn die Achse am Festanschlag steht, und der Zielsatz nach der Abwahl von FXS liegt, wird die neue Zielposition über DB31, ... DBX62.5 (Festanschlag erreicht) als REPOS-Verschiebung angezeigt.

### REPOS und FXS

Mit REPOS wird die Funktionalität von FXS selbsttätig wiederholt und im Folgenden mit FXS-REPOS bezeichnet. Diese Sequenz ist vergleichbar mit dem Programm FXS\_SERUPRO\_ASUP.MPF. Dabei wird jede Achse berücksichtigt und es wird das zuletzt vor dem Suchziel programmierte Moment verwendet.

Der Anwender kann FXS in einem SERUPRO-ASUP gesondert behandeln.

Dafür gilt:

Jede im SERUPRO-ASUP durchgeführte **FXS**-Aktion sorgt automatisch für

$\$AA\_FXS[ <Achse> ] = \$VA\_FXS[ <Achse> ]$ .

Damit wird **FXS-REPOS** für die Achse X inaktiv.

### FXS-REPOS deaktivieren

**FXS-REPOS** wird deaktiviert durch:

- Eine **FXS**-Synchronaktion, die sich auf **REPOSA** bezieht
- $\$AA\_FXS[X] = \$VA\_FXS[X]$  im **SERUPRO\_ASUP**

---

#### Hinweis

Ein **SERUPRO-ASUP** **ohne** **FXS**-Behandlung oder ein nicht vorhandenes **SERUPRO-ASUP** führt automatisch zu **FXS-REPOS**.



#### VORSICHT

**FXS-REPOS** fährt alle Bahnachsen gemeinsam auf den Zielpunkt. Damit fahren Achsen mit und ohne **FXS**-Behandlung gemeinsam mit dem im Zielsatz gültigen G-Code und Vorschub. Dies kann dazu führen, dass der Festanschlag mit Eilgang (**G0**) oder hoher Geschwindigkeit angefahren wird.

### FOC im REPOS

**FOC-REPOS** verhält sich analog zu **FXS-REPOS**.

Ein während des Programmvorlaufs auftretender wechselnder Momentenverlauf kann mit **FOC-REPOS** nicht realisiert werden.

#### Beispiel

Achse X wird von Position 0 nach 100 verfahren. Dabei wird alle 20 Millimeter für jeweils 10 Millimeter **FOC** eingeschaltet. Der sich dabei ergebende Momentenverlauf wird mit satzweisem **FOC** erzeugt und kann durch **FOC-REPOS** nicht nachvollzogen werden. Durch **FOC-REPOS** wird die Achse X mit oder ohne **FOC** fahren, entsprechend der letzten Programmierung vor dem Zielsatz.

Programmierbeispiele zu **FXS** "Fahren auf Festanschlag" siehe Kapitel "Programmtest (Seite 511)".

## 5.2.4 Verhalten bei Reset und Funktionsabbruch

### NC-Reset

Solange die Funktion noch nicht im Zustand "Anschlag wurde erfolgreich angefahren" ist, kann das Fahren auf den Festanschlag mit NC-Reset abgebrochen werden.

Auch wenn der Festanschlag bereits angefahren wurde, das vorgegebene Anschlagsmoment aber noch nicht vollständig erreicht, dann die Funktion mit NC-Reset noch abgebrochen werden. Dabei wird der Sollposition der Achse auf die aktuelle Istposition synchronisiert.

Sobald die Funktion im Zustand "Anschlag wurde erfolgreich angefahren" ist, bleibt die Funktion auch über NC-Reset hinaus wirksam.

### Funktionsabbruch

Ein Funktionsabbruch wird durch folgende Ereignisse ausgelöst:

- Not-Halt

---

#### Hinweis

NC und der Antrieb sind bei "Not-Halt" stromlos, d. h. die PLC muss reagieren.

 <b>VORSICHT</b>
---

Es ist sicher zu stellen, dass durch das Auslösen und Rücksetzen von "Not-Halt" während Fahren auf Festanschlag aktiv ist, keine gefährlichen Maschinensituationen entstehen.

Z.B. Verhalten beim Setzen und Wegnehmen der Impulssperre:  
MD37002 \$MA\_FIXED\_STOP\_CONTROL, Bit 0 (Verhalten bei Impulssperre am Anschlag)

- Bit 0 = 0: Fahren auf Festanschlag wird abgebrochen
- Bit 0 = 1: Fahren auf Festanschlag wird unterbrochen, d.h. der Antrieb wird kraftlos. Sobald die Impulssperre wieder aufgehoben wird, drückt der Antrieb wieder mit dem Anschlagsmoment.

- Funktionszustand: "Festanschlag nicht erreicht"
- Funktionszustand: "Festanschlag abgebrochen"
- Abbruch durch das PLC-Anwenderprogramm:  
DB31, ... DBX62.4 = 0 ("Fahren auf Festanschlag aktivieren")
- Wegnahme der Impulsfreigabe und Maschinendatenparametrierung:  
MD37002 \$MA\_FIXED\_STOP\_CONTROL, Bit 0 = 0 (siehe oben)

## 5.2.5 Verhalten bezüglich anderer Funktionen

### Messen mit Restweglöschen

"Fahren auf Festanschlag" ( $_{FXS}$ ) kann nicht gemeinsam in einem Satz mit "Messen mit Restweglöschen" ( $_{MEAS}$ ) programmiert werden. Außer eine Funktion wirkt auf eine Bahnachse und die andere auf eine Positionierachse oder beide wirken auf Positionierachsen.

### Konturüberwachung

Während "Fahren auf Festanschlag" aktiv ist, erfolgt keine axiale Konturüberwachung.

### Positionierachsen

Bei "Fahren auf Festanschlag" mit Positionierachsen  $_{POSA}$  wird der Satzwechsel ausgeführt, auch wenn die Positionierachse bis zu diesem Zeitpunkt den Festanschlag noch nicht erreicht hat.

### Hängende Achsen

Das "Fahren auf Festanschlag" mit hängenden Achsen ist auch bei Alarmmeldungen möglich.

Tritt bei einer hängenden Achse während des Fahrens auf Festanschlag ein funktionspezifischer Alarm auf, wird das NC/PLC-Nahtstellensignal DB11, DBX6.3 (BAG betriebsbereit) nicht zurückgesetzt. Dadurch wird der entsprechende Antrieb nicht stromlos geschaltet.

Dies entspricht für die hängende Achse einem elektronischen Gewichtsausgleich und ist über das folgende Maschinendatum projektierbar:

MD37052 \$MA\_FIXED\_STOP\_ALARM\_REACTION

### Literatur

Weitere Informationen zu hängenden Achsen finden sich in:

- Funktionshandbuch SINAMICS S120
- Funktionshandbuch Erweiterungsfunktionen; Kompensation (K3), Kapitel: Elektronischer Gewichtsausgleich

## 5.2.6 Settingdaten

Die über die funktionsspezifischen Befehle `FXS`, `FXST` und `FXSW` programmierten Werte, werden satzsynchron in folgende sofort wirksamen, achsspezifischen Settingdaten geschrieben:

### Ein/Ausschalten der Funktion

SD43500 `$SA_FIXED_STOP_SWITCH` (An/Abwahl Fahren auf Festanschlag)

### Klemmmoment

SD43510 `$SA_FIXED_STOP_TORQUE` (Klemmmoment)

---

### Hinweis

#### Klemmmoment größer als 100%

Ein Wert für das Klemmmoment in SD43510 größer 100% des maximalen Motormoments ist nur kurzzeitig sinnvoll. Außerdem wird das maximale Motormoment durch den Antrieb begrenzt. Begrenzend wirken z.B. folgende Antriebsparameter:

- p1520/p1521 Drehmomentgrenze/Kraftgrenze oben / Drehmomentgrenze/Kraftgrenze unten
- p1522/p1523 Drehmomentgrenze/Kraftgrenze oben / Drehmomentgrenze/Kraftgrenze unten
- p1530/p1531 Leistungsgrenze motorisch / Leistungsgrenze generatorisch
- p0640 Stromgrenze
- p0326 Motor-Kippkraftkorrekturfaktor

Ausführliche Informationen zu den Antriebsparametern und den Funktionen finden sich in:

### Literatur

- SINAMICS S120/S150 Listenhandbuch
  - SINAMICS S120 Funktionshandbuch
- 

### Überwachungsfenster

SD43520 `$SA_FIXED_STOP_WINDOW` (Überwachungsfenster)

## Vorbelegung

Über folgende Maschinendaten wird die Vorgebelegung der Settingdaten eingestellt:

- Klemmmoment:  
MD37010 `$MA_FIXED_STOP_TORQUE_DEF` (Vorbelegung Klemmmoment)
- Überwachungsfenster:  
MD37020 `$MA_FIXED_STOP_WINDOW_DEF` (Vorbelegung Überwachungsfenster)

### Wirksamkeit

Die Settingdaten für Klemmmoment und Überwachungsfenster sind sofort wirksam. Dadurch kann vom Bediener oder über das PLC-Anwenderprogramm der Klemmzustand jederzeit an die Bearbeitungssituation angepasst werden.

### Literatur

Weiter ausführliche Informationen zu Maschinen- und Settingdaten finden sich in:  
Listenhandbuch Ausführliche Maschinendatenbeschreibung

### Siehe auch

Z1: NC/PLC-Nahtstellensignale (Seite 1791)

## 5.2.7 Systemvariable

Über folgende Systemvariable können in Teileprogrammen und Synchronaktionen Informationen zum Status der Funktion gelesen werden:

- \$AA\_FXS (Status Sollzustand "Fahren auf Festanschlag")
- \$VA\_FXS (Status Istzustand "Fahren auf Festanschlag")
- \$VA\_FXS\_INFO (Zusatzinformation bei "Fahren auf Festanschlag")  
Die Systemvariable enthält Zusatzinformationen für den Fall, dass das Anfahren des Festanschlags fehlgeschlagen ist: \$VA\_FXS == 2

---

### Hinweis

#### Istposition am Festanschlag

Nach Erreichen des Festanschlages kann z.B. für Auswertungs- oder Messzweck über die Systemvariablen \$AA\_IM die Istposition der Maschinenachse gelesen werden.

---



## Anwendungsbeispiel für \$AA\_FXS

Damit ein Satzwechsel erfolgt, soll im Fehlerfall kein Alarm ausgelöst werden. Die Ursache kann anschließend über die Systemvariable ermittelt und spezifisch darauf reagiert werden.

Voraussetzung: MD37050 \$MA\_FIXED\_STOP\_ALARM\_MASK = 0 (keinen Alarm auslösen)

### Programmcode

```
X300 Y500 F200 FXS[X1]=1 FXST[X1]=25 FXSW[X1]=5
; IF Festanschlag == erreicht => Normalfall
IF $AA_FXS[X1]==2 GOTOF FXS_ERROR
; Normalfall
G01 X400 Y200
...
GOTOF PROG_END
; ELSE Fehlerbehandlung
FXS_ERROR: ...
PROG_END: M30
```

## 5.2.8 Festanschlagsalarme

### Alarm 20091 "Festanschlag nicht erreicht"

Wird beim Fahren auf Festanschlag die Zielposition nicht erreicht, wird Alarm 20091 "Festanschlag nicht erreicht" angezeigt und ein Satzwechsel durchgeführt.

### Alarm 20092 "Fahren auf Festanschlag noch aktiv"

Erfolgt für eine Achse nach Erreichen des Festanschlags eine Fahranforderung oder erneute Funktionsanwahl, wird der Alarm 20092 "Fahren auf Festanschlag noch aktiv" angezeigt.

### Alarm 20093 "Stillstandsüberwachung am Anschlag hat ausgelöst"

Wird eine Achse nach Erreichen des Festanschlags um mehr als den im Settingdatum SD43520 FIXED\_STOP\_WINDOW (Festanschlags-Überwachungsfenster)

angegebenen Wert aus der Position bewegt, wird Alarm 20093 "Stillstandsüberwachung am Anschlag hat ausgelöst" angezeigt, Fahren auf Festanschlag für diese Achse abgewählt und folgende Systemvariable gesetzt:

\$AA\_FXS[x] = 2

### Alarm 20094 "Funktion wurde abgebrochen"

Das Fahren auf Festanschlag wird abgebrochen, wenn durch die Wegnahme der Impulsfreigabe das Klemmmoment nicht länger aufgebracht werden kann, oder das angeforderte Quittungssignal an der NC/PLC-Nahtstelle zurückgesetzt wurde:

- Quittungssignal erforderlich: MD37060 \$MA\_FIXED\_STOP\_ACKN\_MASK, Bit 0 = 1
- Quittungssignal: DB31, ... DBX3.1 == 0 (Fahren auf Festanschlag freigeben)

### Freigabe der Festanschlagsalarme

Über das folgende Maschinendatum kann eingestellt werden, ob die Festanschlagsalarme

- Alarm 20091 "Festanschlag nicht erreicht",
- Alarm 20094 "Festanschlag abgebrochen"

angezeigt werden:

MD37050 \$MA\_FIXED\_STOP\_ALARM\_MASK (Freigabe der Festanschlagsalarme)

### Einstellbares Funktionsverhalten bei Festanschlagsalarmen

Über das folgende Maschinendatum kann eingestellt werden, dass die Funktion auch beim Auftreten funktionspezifischer Alarme nicht abgebrochen wird:

- Alarm 20090 Fahren auf Festanschlag nicht möglich
- Alarm 20091 Festanschlag nicht erreicht
- Alarm 20092 Fahren auf Festanschlag noch aktiv
- Alarm 20093 Stillstandsüberwachung am Anschlag hat ausgelöst
- Alarm 20094 Fahren auf Festanschlag abgebrochen

MD37052 \$MA\_FIXED\_STOP\_ALARM\_REACTION (Reaktion bei Festanschlagsalarmen)

### Alarmunterdrückung nach neuer Programmierung

Fahren auf Festanschlag kann für einfache Messvorgänge eingesetzt werden.

Beispielsweise kann auf Werkzeugbruch kontrolliert werden, indem die Länge des Werkzeugs durch Auffahren auf ein definiertes Hindernis gemessen wird. Dies erfordert die Unterdrückung des Anschlagalarms. Bei anschließender "normaler" Nutzung der Funktion zum Klemmen von Werkstücken ist eine Aktivierung des Alarms durch Teileprogrammbeefehle möglich.

## 5.2.9 Fahren mit begrenztem Moment/Kraft FOC

### Funktion

Für Anwendungen, bei denen Moment oder Kraft dynamisch weg- oder zeitabhängig oder von anderen Größen abhängig geändert werden sollen (z. B. Pressen), wird die Funktionalität FOC (Force Control) bereitgestellt.

Hiermit sind Kraft/Weg- oder Kraft/Zeitprofile in der Auflösung für den Interpolationstakt möglich.

Die Funktion gestattet es, über Synchronaktionen zu jeder Zeit Moment/Kraft zu ändern.

Die Funktion kann modal oder satzbezogen aktiviert werden.

### Verfügbarkeit

System	Verfügbarkeit
SINUMERIK 840D sl	Standard (Grundumfang)
SINUMERIK 828D	Option

### Modale Aktivierung (FOCON/FOCOF)

Die Aktivierung der Funktion nach POWER ON und RESET wird bestimmt durch das Maschinendatum:

MD37080 \$MA\_FOC\_ACTIVATION\_MODE (Steuerung der Grundstellung der modalen Begrenzung von Moment/Kraft )

Bit	Bedeutung
0	nach POWER ON wirksam:
	= 0 FOCO F
	= 1 FOCO N
1	nach RESET wirksam:
	= 0 FOCO F
	= 1 FOCO N

FOCON: Aktivierung der modal wirksamen Moment/Kraft-Begrenzung

FOCOF: Abschalten der Moment/Kraft-Begrenzung

Die modale Aktivierung wirkt über das Programmende hinaus.

Es wirkt, wenn schon programmiert, das mit FXST eingestellte Moment/Kraft.

Eine Programmierung von FXST kann unabhängig von FOCON erfolgen, die Wirkung tritt jedoch erst nach Aktivierung der Funktion ein.

## Programmierung

Die Programmierung der Achse erfolgt in eckigen Klammern.

Zulässig sind:

- Geometrieachs-Bezeichner
- Kanalachs-Bezeichner
- Maschinenachs-Bezeichner

Beispiel:

Programmcode	Kommentar
N10 FOCON[X]	; modale Aktivierung der Momentenbegrenzung
N20 X100 Y200 FXST[X]=15	; X fährt mit reduziertem Moment (15%)
N30 FXST[X]=75 X20	; Änderung des Moments auf 75%, ; X fährt mit diesem begrenzten Moment
N40 FOCOF[X]	; Abschalten der Momentenbegrenzung

## Satzbezogene Begrenzung (FOC)

Der Teileprogrammbehehl `FOC` schaltet die Momentenbegrenzung für einen Satz ein.

Eine Aktivierung aus einer Synchronaktion wirkt bis zum Satzende des aktuellen Teileprogrammsatzes.

## Vorrang FXS/FOC

Eine Aktivierung von FXS bei aktivem FOC hat Vorrang, d. h. FXS wird ausgeführt.

Eine Abwahl von FXS hebt die Klemmung auf.

Eine modale Momenten/Kraftbegrenzung bleibt aktiv.

Nach POWER ON wirkt bei einer Aktivierung das Maschinendatum:

MD37010 \$MA\_FIXED\_STOP\_TORQUE\_DEF

Dieses Moment kann jederzeit durch Programmierung von FXST verändert werden.

## Synchronaktionen

Die Sprachbefehle `FOC`, `FOCON`, `FOCOF` können wie die Befehle für "Fahren auf Festanschlag" auch in Synchronaktionen programmiert werden.

### Status FOC feststellen

Durch die Statusvariable \$AA\_FOC kann jederzeit der Aktivierungszustand gelesen werden.  
Wird FXS zusätzlich aktiviert, wird der Status nicht verändert.

Wert	Bedeutung
0	FOC nicht aktiv
1	FOC modal aktiv
2	FOC satzbezogen aktiv

### Status Momentgrenze feststellen

Mit der Systemvariablen \$VA\_TORQUE\_AT\_LIMIT kann in Systemen jederzeit gelesen werden, ob das aktuell wirksame Moment der vorgegebenen Momentengrenze entspricht.

Wert	Bedeutung
0	wirksames Moment < Momentengrenzwert
1	wirksames Moment hat den Momentengrenzwert erreicht

### Einschränkungen

Die Funktion FOC hat folgende Einschränkungen:

- Die Veränderung der Momenten-/Kraftbegrenzung, die sich als Beschleunigungsbegrenzung darstellt, wird **nur an Satzgrenzen** in die Verfahrbewegung eingerechnet (siehe Befehl ACC).
- Nur FOC:  
Es ist **keine Überwachung** auf das Erreichen der aktiven Momentengrenze aus der NC/PLC-Nahtstelle möglich.
- Durch Nichtanpassung der Beschleunigungsbegrenzung kommt es während der Verfahrbewegung zu einer Erhöhung des Schleppabstandes.
- Durch Nichtanpassung der Beschleunigungsbegrenzung wird der Satzendpunkt eventuell später erreicht, als im Maschinendatum:  
MD36040 \$MA\_STANDSTILL\_DELAY\_TIME  
angegeben.  
Dafür wird das Maschinendatum:  
MD36042 \$MA\_FOC\_STANDSTILL\_DELAY\_TIME  
eingeführt, auf das in diesem Zustand überwacht wird.

### **Einsatzmöglichkeit bei Link-und Containerachsen**

Alle Achsen, die in einem Kanal verfahren werden können, d. h. auch Linkachsen und Containerachsen, können auf Festanschlag gefahren werden.

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3)

Der Zustand der Maschinenachse bleibt bei einer Container-Drehung erhalten, d. h. eine geklemmte Maschinenachse bleibt am Anschlag.

Wurde mit FOCON eine modale Momentenbegrenzung aktiviert, bleibt diese auch nach einer Container-Drehung für die Maschinenachse erhalten.

## 5.3 Beispiele

### Statische Synchronaktionen

Fahren auf Festanschlag (FXS), ausgelöst durch eine Synchronaktion.

Programmcode	Kommentar
N10 IDS=1 WHENEVER ((R1==1) AND (\$AA_FXS[Y]==0)) DO R1=0 FXS[Y]=1 FXST[Y]=10 FA[Y]=200 POS[Y]=150	; statische Synchronaktion aktivieren: Durch das Setzen von R1=1 wird für die Achse Y FXS aktiviert das wirksame Moment auf 10% reduziert und eine Fahrbewegung in Richtung des Anschlags gestartet
N11 IDS=2 WHENEVER (\$AA_FXS[Y]==4) DO FXST[Y]=30	; Sobald der Anschlag erkannt wurde (\$AA_FXS[Y]==4), wird das Moment auf 30% heraufgesetzt
N12 IDS=3 WHENEVER (\$AA_FXS[Y]==1) DO FXST[Y]=\$R0	; Nach dem Erreichen des Anschlags wird das Moment abhängig von R0 gesteuert
N13 IDS=4 WHENEVER ((R3==1) AND (\$AA_FXS[Y]==1)) DO FXS[Y]=0 FA[Y]=1000 POS[Y]=0	; Abwahl in Abhängigkeit von R3 und zurückfahren
N20 FXS[Y]= 0 G0 G90 X0 Y0	; normaler Programmablauf
N30 RELEASE(Y)	; Achse Y für die Bewegung in Synchronaktion freigeben
N40 G1 F1000 X100 N50 .....	; Bewegung einer anderen Achse
N60 GET(Y)	; Achse Y wieder in den Bahnverbund aufnehmen

### Mehrfache Anwahl

Eine Anwahl darf nur einmal erfolgen. Wird durch eine fehlerhafte Programmierung die Funktion nach der Aktivierung ( $FXS[Achse]=1$ ) nochmals aufgerufen wird der Alarm 20092 "Fahren auf Festanschlag noch aktiv" ausgelöst.

Eine Programmierung, die in der Bedingung entweder  $\$AA\_FXS[]$  oder einen eigenen Merker (hier R1) abfragt, vermeidet eine mehrfache Aktivierung der Funktion.

#### Programmierbeispiel (Teileprogrammfragment):

Programmcode
N10 R1=0
N20 IDS=1 WHENEVER ( $\$R1 == 0$ AND $\$AA\_IW[AX3]>7$ ) DO R1=1 FXS[AX1]=1
FXST[AX1]=12

### Satzbezogene Synchronaktionen

Durch die Programmierung einer satzbezogenen Synchronaktion kann Fahren auf Festanschlag während einer Anfahrbewegung zugeschaltet werden.

#### Programmierbeispiel:

Programmcode	Kommentar
N10 G0 G90 X0 Y0	
N20 WHEN $\$AA\_IW[X]>17$ DO FXS[X]=1	; Erreicht X eine Position größer
N30 G1 F200 X100 Y110	; 17mm wird FXS aktiviert



## 5.4 Datenlisten

### 5.4.1 Maschinendaten

#### 5.4.1.1 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
36042	FOC_STANDSTILL_DELAY_TIME	Verzögerungszeit Stillstandsüberwachung bei aktiver Momenten-/Kraftbegrenzung
37000	FIXED_STOP_MODE	Modus Fahren auf Festanschlag
37002	FIXED_STOP_CONTROL	Ablaufkontrolle für Fahren auf Festanschlag
37010	FIXED_STOP_TORQUE_DEF	Voreinstellung Festanschlag-Klemmmoment
37012	FIXED_STOP_TORQUE_RAMP_TIME	Zeitdauer bis zum Erreichen der geänderten Momentengrenze
37020	FIXED_STOP_WINDOW_DEF	Voreinstellung für Festanschlags-Überwachungsfenster
37030	FIXED_STOP_THRESHOLD	Schwelle für Festanschlagserkennung
37040	FIXED_STOP_BY_SENSOR	Festanschlagserkennung über Sensor
37050	FIXED_STOP_ALARM_MASK	Freigabe der Festanschlagsalarme
37052	FIXED_STOP_ALARM_REACTION	Reaktion bei Festanschlagsalarmen
37060	FIXED_STOP_ACKN_MASK	Beachtung von PLC-Quittierungen für Fahren auf Festanschlag
37070	FIXED_STOP_ANA_TORQUE	Momentengrenze beim Anfahren des Festanschlags für analoge Antriebe
37080	FOC_ACTIVATION_MODE	Grundstellung der modalen Momenten-/Kraftbegrenzung

### 5.4.2 Settingdaten

#### 5.4.2.1 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: \$SA_	Beschreibung
43500	FIXED_STOP_SWITCH	Anwahl Fahren auf Festanschlag
43510	FIXED_STOP_WINDOW	Festanschlagsklemmmoment
43520	FIXED_STOP_TORQUE	Festanschlags-Überwachungsfenster

### 5.4.3 Signale

#### 5.4.3.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Festanschlag erreicht quittieren	DB31, ... .DBX1.1	DB380x.DBX1.1
Sensor Festanschlag	DB31, ... .DBX1.2	DB380x.DBX1.2
Achsen-/Spindelsperre	DB31, ... .DBX1.3	DB380x.DBX1.3
Reglerfreigabe	DB31, ... .DBX2.1	DB380x.DBX2.1
Fahren auf Festanschlag freigeben	DB31, ... .DBX3.1	DB380x.DBX3.1

#### 5.4.3.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Fahren auf Festanschlag aktivieren	DB31, ... .DBX62.4	DB390x.DBX2.4
Festanschlag erreicht	DB31, ... .DBX62.5	DB390x.DBX2.5

# G2: Geschwindigkeiten, Soll-/Istwertsysteme, Regelung

# 6

## 6.1 Kurzbeschreibung

Diese Funktionsbeschreibung beschreibt die Parametrierung einer Maschinenachse bezüglich:

- Istwert- bzw. Messsysteme
- Sollwertsystem
- Bediengenauigkeit
- Fahrbereiche
- Achsgeschwindigkeiten
- Regelparameter

## 6.2 Geschwindigkeiten, Verfahrbereiche, Genauigkeiten

### 6.2.1 Geschwindigkeiten

#### Maximale Bahn-, Achsgeschwindigkeit und Spindeldrehzahl

Die maximale Bahn-, Achsgeschwindigkeit und Spindeldrehzahl wird beeinflusst durch die Maschinenkonstruktion, Antriebsdynamikauslegung und die Grenzfrequenz der Istwerterfassung (Gebergrenzfrequenz).

Die maximale Achsgeschwindigkeit wird in dem Maschinendatum:

MD32000 \$MA\_MAX\_AX\_VELO (Maximale Achsgeschwindigkeit)

definiert.

Die maximal zulässige Spindeldrehzahl wird über das Maschinendatum:

MD35100 \$MA\_SPIND\_VELO\_LIMIT (Maximale Spindeldrehzahl)

vorgegeben.

Erläuterungen siehe Kapitel "S1: Spindeln (Seite 1383)".

Bei einem höheren (resultierend aus programmierten und über Vorschubkorrektur beeinflussten) Vorschub wird auf  $V_{\max}$  begrenzt.

Diese automatische Vorschubbegrenzung kann bei von CAD-Systemen generierten Programmen, die extrem kurze Sätze enthalten, zu einer Absenkung der Geschwindigkeit über mehrere Sätze führen.

#### Beispiel:

IPO-Takt = 12 ms

N10 G0 X0 Y0; [mm]

N20 G0 X100 Y100; [mm]

⇒ programmierte Weglänge im Satz = 141,42 mm

⇒  $V_{\max} = (141,42 \text{ mm} / 12 \text{ ms}) \cdot 0,9 = 10606,6 \text{ mm/s} = 636,39 \text{ m/min}$

#### Minimale Bahn-, Achsgeschwindigkeit

Für die minimale Bahn- oder Achsgeschwindigkeit gilt folgende Einschränkung:

$$V_{\min} \geq \frac{10^{-3}}{\text{Rechenfeinheit} \left[ \frac{\text{Inkr.}}{\text{mm oder Grad}} \right] \cdot \text{IPO-Takt [s]}}$$

Die Rechenfeinheit wird mit dem Maschinendatum:

MD10200 \$MN\_INT\_INCR\_PER\_MM (Rechenfeinheit für Linearpositionen)

bzw.  
 MD10210 \$MN\_INT\_INCR\_PER\_DEG (Rechenfeinheit für Winkelpositionen)  
 definiert.:  
 Bei Unterschreitung von  $V_{min}$  erfolgt keine Verfahrbewegung.

**Beispiel:**

MD10200 \$MN\_INT\_INCR\_PER\_MM = 1000 [Inkr. / mm] ;

IPO-Takt = 12 ms;

$\Rightarrow V_{min} = 10^{-3} / (1000 \text{ Inkr./mm} \times 12 \text{ ms}) = 0,005 \text{ mm / min};$

Der Wertebereich der Vorschübe ist abhängig von der gewählten Rechenfeinheit.

Bei der Standardvorbelegung des Maschinendatums:

MD10200 \$MN\_INT\_INCR\_PER\_MM  
 (Rechenfeinheit für Linearpositionen) (1000 Inkr./mm)

bzw.

MD10210 \$MN\_INT\_INCR\_PER\_DEG  
 (Rechenfeinheit für Winkelpositionen) (1000 Inkr./Grad)

kann folgender Wertebereich mit der angegebenen Auflösung programmiert werden:

<b>Wertebereich für Bahnvorschub F bei Geometrieachsen:</b>	
metrisches System:	inch-System:
$0,001 \leq F \leq 999.999,999$ [mm/min, mm/U, Grad/min, Grad/U]	$0,001 \leq F \leq 399.999,999$ [inch/min, inch/U]

<b>Wertebereich für Vorschub bei Positionierachsen:</b>	
metrisches System:	inch-System:
$0,001 \leq FA \leq 999.999,999$ [mm/min, mm/U, Grad/min, Grad/U]	$0,001 \leq FA \leq 399.999,999$ [inch/min, inch/U]

<b>Wertebereich für Spindeldrehzahl S:</b>
$0,001 \leq S \leq 999.999,999$ [U/min]

Wird die Rechenfeinheit um einen Faktor erhöht/erniedrigt, so ändern sich die Wertebereiche entsprechend.

## 6.2.2 Verfahrbereiche

### Wertebereich der Verfahrbereiche

Der Wertebereich der Verfahrbereiche ist abhängig von der gewählten Rechenfeinheit.

Bei der Standardbelegung des Maschinendatums:

MD10200 \$MN\_INT\_INCR\_PER\_MM

(Rechenfeinheit für Linearpositionen) (1000 Inkr./mm)

bzw.

MD10210 \$MN\_INT\_INCR\_PER\_DEG

(Rechenfeinheit für Winkelpositionen) (1000 Inkr./Grad)

kann folgender Wertebereich mit der angegebenen Auflösung programmiert werden:

Tabelle 6- 1 Verfahrbereiche der Achsen

	G71 [mm, Grad]	G70 [inch, Grad]
	Bereich	Bereich
Linearachsen X, Y, Z, ...	± 999.999,999	± 399.999,999
Rundachsen A, B, C, ...	± 999.999,999	± 999.999,999
Interpolationsparameter I, J, K	± 999.999,999	± 399.999,999

Rundachsen tragen immer die Maßeinheit Grad.

Wird die Rechenfeinheit um den Faktor 10 erhöht/erniedrigt, so ändern sich die Wertebereiche entsprechend.

Der Verfahrbereich kann durch SW-Endschalter und Arbeitsbereiche begrenzt werden (siehe Kapitel "A3: Achsüberwachungen, Schutzbereiche (Seite 85)").

Besonderheiten bei großem Verfahrbereich für Linear- und Rundachsen siehe Kapitel "R1: Referenzieren (Seite 1319)".

Der Verfahrbereich für Rundachsen kann über Maschinendaten beschränkt werden.

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Rundachsen (R2)

## 6.2.3 Positioniergenauigkeit der Steuerung

### Istwertauflösung und Rechenfeinheit

Die Positioniergenauigkeit der Steuerung ist abhängig von der Istwertauflösung (= Geberinkremente / (mm oder Grad)) und der Rechenfeinheit (= interne Inkremente / (mm oder Grad)).

Die gröbere Auflösung der beiden bestimmt die Positioniergenauigkeit der Steuerung.

Die Wahl der Eingabefinheit, des Interpolator- und Lageregelaktkes haben keinen Einfluss auf diese Genauigkeit.

Neben der Begrenzung durch MD32000 begrenzt die Steuerung situationsbedingt die maximale Bahngeschwindigkeit nach folgender Formel:

$$\frac{\text{Interne Inkremente / mm}}{\text{Geberinkremente / mm}} = \frac{1}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}}$$
$$* \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}}$$
$$* \frac{\text{DRIVE\_AX\_RATIO\_DENOM [n]}}{\text{DRIVE\_AX\_RATIO\_NUMERA [n]}}$$
$$* \text{LEADSCREW\_PITCH}$$
$$* \text{INT\_INCR\_PER\_MM}$$

## 6.2.4 Eingabe-/Anzeigefinheit, Rechenfeinheit

### Feinheiten: Unterschiede

Bei den Feinheiten, d. h. der Auflösung von Linear- und Winkelpositionen, Geschwindigkeiten, Beschleunigungen und Ruck, ist zu unterscheiden zwischen:

- Eingabefinheit  
Eingabe von Daten über die Bedientafelfront oder über Teileprogramme.
- Anzeigefinheit  
Anzeige von Daten über die Bedientafelfront.
- Rechenfeinheit  
Interne Darstellung der über Bedientafelfront oder Teileprogramm eingegebenen Daten.

Die Eingabe- und Anzeigefinheit wird durch die verwendete Bedientafelfront vorgegeben, wobei die Anzeigefinheit für Positionswerte mit dem Maschinendatum:

MD9004 \$MM\_DISPLAY\_RESOLUTION (Anzeigefinheit)

veränderbar ist.

Über das Maschinendatum:

MD9011 \$MM\_DISPLAY\_RESOLUTION\_INCH (Anzeigefinheit INCH-Maßsystem)

kann die Anzeigefinheit für Positionswerte bei Inch-Einstellung projiziert werden.

Damit ist es möglich, bei Inch-Einstellung bis zu sechs Nachkommastellen anzeigen zu können.

Für die Programmierung in Teileprogrammen gelten die in der Programmieranleitung aufgeführten Eingabefinheiten.

Die gewünschte Rechenfeinheit wird mit den Maschinendaten:

MD10200 \$MN\_INT\_INCR\_PER\_MM (Rechenfeinheit für Linearpositionen)

und

MD10210 \$MN\_INT\_INCR\_PER\_DEG (Rechenfeinheit für Winkelpositionen)

festgelegt.

Sie ist unabhängig von der Eingabe-/Anzeigefinheit, sollte aber mindestens die gleiche Auflösung haben.

Mit der Rechenfeinheit wird die Anzahl der max. wirksamen Nachkommastellen für Positionswerte, Geschwindigkeiten usw. im Teileprogramm, sowie die Anzahl der Nachkommastellen für Werkzeugkorrekturen, Nullpunktverschiebungen etc. (und somit auch für die max. erreichbare Genauigkeit) festgelegt.

Die Genauigkeit der Eingabe von Winkel- und Linear-Positionen wird auf die Rechenfeinheit begrenzt, indem das Produkt des programmierten Wertes mit der Rechenfeinheit auf eine ganze Zahl gerundet wird.

Um die ausgeführte Rundung leicht nachvollziehbar zu halten, ist es sinnvoll, für die Rechenfeinheit 10er-Potenzen zu verwenden.

### Beispiel für Rundung:

Rechenfeinheit : 1000 Inkremente / mm

Programmierter Weg : 97,3786 mm

Wirksamer Wert = 97,379 mm

### Beispiel für Programmierung im 1/10-µm-Bereich:

Alle Linearachsen einer Maschine sollen im Wertebereich 0,1 ... 1000 µm programmiert und verfahren werden.

⇒ Um auf 0,1 µm genau zu positionieren, muss die Rechenfeinheit auf  $\geq 10^4$  Inkr./mm gesetzt werden.

⇒ MD10200 \$MN\_INT\_INCR\_PER\_MM = 10000 [Inkr./mm]:

⇒ Beispiel für zugehöriges Teileprogramm:

Programmcode	Kommentar
N20 G0 X 1.0000 Y 1.0000	; Achsen fahren auf die Position X=1.0000 mm, Y=1.0000 mm
N25 G0 X 5.0002 Y 2.0003	; Achsen fahren auf die Position =5.0002 mm, Y=2.0003 mm



## 6.2.5 Normierung physikalischer Größen der Maschinen- und Settingdaten

### Ein-/Ausgabeeinheiten

Maschinen- und Settingdaten, die eine physikalische Größe besitzen, werden je nach Grundsystem (metrisch/inch) standardmäßig in folgenden Ein-/Ausgabeeinheiten interpretiert:

Physikalische Größe:	Ein- /Ausgabeeinheiten für Standardgrundsystem:	
	Metrisch	Inch
Linear-Position	1 mm	1 inch
Winkel-Position	1 Grad	1 Grad
Linear-Geschwindigkeit	1 mm/min	1 inch/min
Winkel-Geschwindigkeit	1 Umdr./min	1 Umdr./min
Linear-Beschleunigung	1 m/s <sup>2</sup>	1 inch/s <sup>2</sup>
Winkel-Beschleunigung	1 Umdr./s <sup>2</sup>	1 Umdr./s <sup>2</sup>
Linear-Ruck	1 m/s <sup>3</sup>	1 inch/s <sup>3</sup>
Winkel-Ruck	1 Umdr./s <sup>3</sup>	1 Umdr./s <sup>3</sup>
Zeit	1 s	1 s
Lageregler-Kreisverstärkung	1/s	1/s
Umdrehungsvorschub	1 mm/Umdr.	inch/Umdr.
Kompensationswert Linear-Position	1 mm	1 inch
Kompensationswert Winkel-Position	1 Grad	1 Grad

Für die interne Ablage werden folgende unten aufgeführte Einheiten benutzt. Unabhängig von dem gewählten Grundsystem arbeitet die Steuerung intern immer mit diesen Einheiten.

Physikalische Größe:	Interne Einheit:
Linear-Position	1 mm
Winkel-Position	1 Grad
Linear-Geschwindigkeit	1 mm/s
Winkel-Geschwindigkeit	1 Grad/s
Linear-Beschleunigung	1 mm/s <sup>2</sup>
Winkel-Beschleunigung	1 Grad/s <sup>2</sup>
Linear-Ruck	1 mm/s <sup>3</sup>
Winkel-Ruck	1 Grad/s <sup>3</sup>
Zeit	1 s
Lageregler-Kreisverstärkung	1/s
Umdrehungsvorschub	1 mm/Grad
Kompensationswert Linear-Position	1 mm
Kompensationswert Winkel-Position	1 Grad

Der Anwender hat die Möglichkeit, andere Ein-/Ausgabeeinheiten für Maschinen- und Settingdaten zu definieren.

Dazu muss über die Maschinendaten:

MD10220 \$MN\_SCALING\_USER\_DEF\_MASK

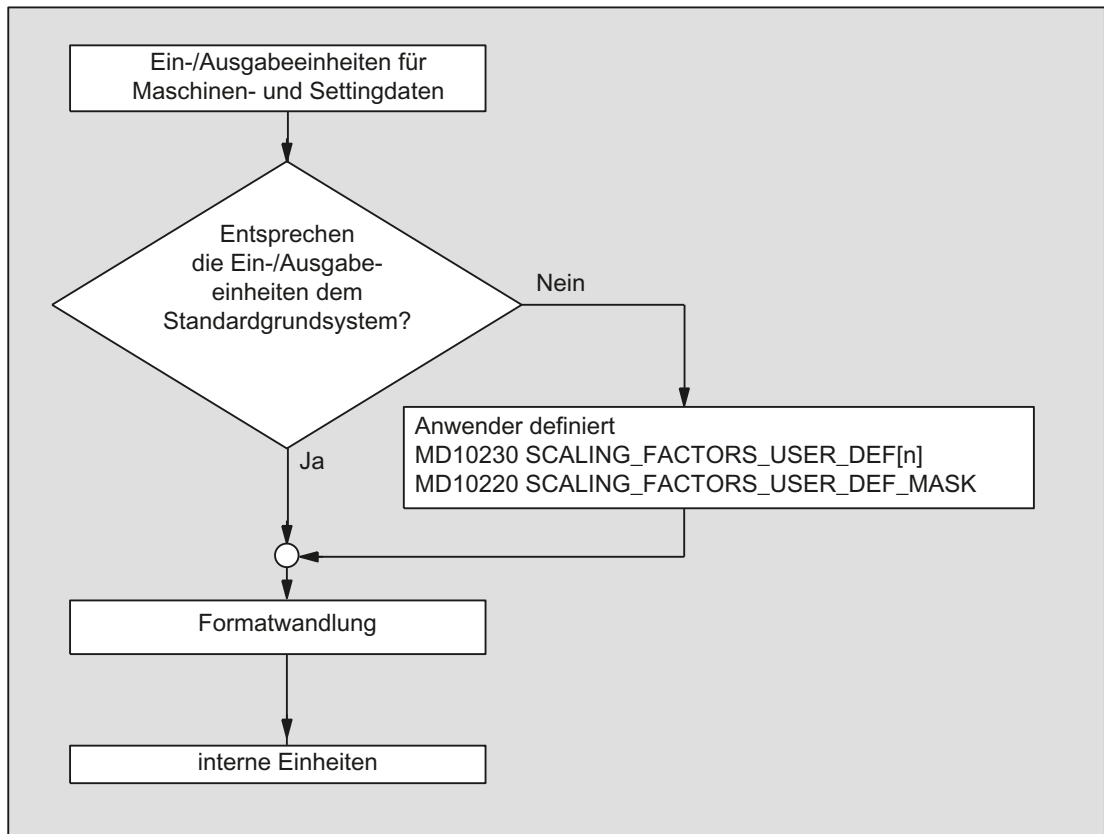
(Aktivierung der Normierungsfaktoren)

und

MD10230 \$MN\_SCALING\_FACTORS\_USER\_DEF[n]

(Normierungsfaktoren der physikalischen Größen)

eine Anpassung zwischen den neu gewählten Ein-/Ausgabeeinheiten und den internen Einheiten erfolgen.



Dabei gilt:

Gewählte Ein-/Ausgabeeinheit = MD10230 \* interne Einheit

In das Maschinendatum:

MD10230 \$MN\_SCALING\_FACTORS\_USER\_DEF[n]

ist also jeweils die gewählte Ein-/Ausgabeeinheit ausgedrückt in den internen Einheiten 1 mm, 1 Grad und 1 s einzugeben.

### Beispiel 1:

Die Maschinendaten-Ein-/Ausgabe von Lineargeschwindigkeiten soll statt in mm/min (Grundstellung) in m/min erfolgen.

(Die interne Einheit ist mm/s)

⇒ Der Normierungsfaktor für Lineargeschwindigkeiten soll von der Standardeinstellung abweichen. Dazu muss im Maschinendatum:

MD10220 \$MN\_SCALING\_USER\_DEF\_MASK

das Bit Nummer 2 gesetzt werden.

⇒ MD10220 \$MN\_SCALING\_USER\_DEF\_MASK = 'H4'; (Bit-Nr. 2 als Hex-Wert)

⇒ Der Normierungsfaktor für Lineargeschwindigkeiten soll von der Standardeinstellung abweichen. Dazu muss im Maschinendatum:

MD10220 \$MN\_SCALING\_USER\_DEF\_MASK

das Bit Nummer 2 gesetzt werden.

⇒ MD10220 \$MN\_SCALING\_USER\_DEF\_MASK = 'H4'; (Bit-Nr. 2 als Hex-Wert)

⇒ Der Normierungsfaktor errechnet sich nach folgender Formel:

$$\begin{aligned} \text{MD10230 SCALING_FACTORS\_USER\_DEF}[n] &= \frac{\text{Gewählte Ein-/Ausgabeeinheit}}{\text{interne Einheit}} \\ \text{MD10230 SCALING_FACTORS\_USER\_DEF}[n] &= \frac{1 \frac{\text{m}}{\text{min}}}{1 \frac{\text{mm}}{\text{s}}} = \frac{1000 \frac{\text{mm}}{60 \text{ s}}}{1 \frac{\text{mm}}{\text{s}}} = \frac{1000}{60} = 16,667; \\ \Rightarrow \text{MD10230 SCALING_FACTORS\_USER\_DEF}[n] &= 16,667 \end{aligned}$$

Der Index n spezifiziert in der Liste der "Normierungsfaktoren der physikalischen Größen" die "Linear-Geschwindigkeit".

**Beispiel 2:**

Zusätzlich zu der Änderung von Beispiel 1 soll die Maschinendaten-Ein-/Ausgabe von Linear-Beschleunigungen statt in  $m/s^2$  (Grundstellung) in  $ft/s^2$  erfolgen.  
(Die interne Einheit ist  $mm/s^2$ )

```
⇒ MD10220 SCALING_USER_DEF_MASK = H14 ; (Bit-Nr. 4 und Bit-Nr. 2 von Beispiel 1  
als Hex-Wert)  
  
⇒ MD10220 SCALING_FACTORS_USER_DEF[4] =  $\frac{1 \frac{ft}{s^2}}{1 \frac{mm}{s^2}} = \frac{12*25,4 \frac{mm}{s^2}}{1 \frac{mm}{s^2}} = 304,8 ;$   
  
⇒ MD10230 SCALING_FACTORS_USER_DEF[4] = 304,8
```

Der Index 4 spezifiziert in der Liste der "Normierungsfaktoren der physikalischen Größen" die "Linear-Beschleunigung".

## 6.3 Metrisches-/Inch-Maßsystem

### 6.3.1 Umrechnung des Grundsystems mittels Teileprogramm

#### Programmierbare Maßsystemumschaltung

Das Grundsystem kann innerhalb eines Teileprogramms über die G-Funktionen  $G70/G71/G700/G710$  (G-Gruppe 13) umgeschaltet werden. Das programmierte Maßsystem ( $G70/G71/G700/G710$ ) und das Grundsystem können zu jeder Zeit gleich oder ungleich sein. Mit der Umschaltung des Maßsystems im einem Teileprogrammabschnitt kann z.B. bei metrischem Grundsystem an einem Werkstück ein Zoll-Gewinde bearbeitet werden.

Der folgende Teileprogrammabschnitt wird im Grundsystem "metrisch" ausgeführt:  
MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC = 1

Programmcode	Kommentar
N100 G71	; Umschalten auf Metrische Bearbeitung
	; Der Umrechnungsfaktor wirkt nicht, da das prog.
	; Maßsystem nicht vom Grundsystem abweicht
....	; Metrische Bearbeitung
N200 G70	; Umschalten auf Inch Bearbeitung
	; Der Umrechnungsfaktor wirkt
...	; Inch Bearbeitung
N300	; Umschalten auf Metrische Bearbeitung
...	; Metrische Bearbeitung

#### Grundstellung der G-Funktionen

Die Grundstellung der G-Funktionen kann kanalspezifisch über folgendes Maschinendatum eingestellt werden werden:

MD20150 \$MC\_GCODE\_RESET\_VALUES[12] (Löschstellung der G-Gruppe 13)

Bei der Umschaltung des Maßsystems über die HMI-Bedienoberfläche wird die Löschstellung passend zum neuen Maßsystem automatisch mit  $G700$  oder  $G710$  vorbesetzt.

### Anzeigen längenbehafteter Daten am HMI

Längenbehaftete Daten werden am HMI im parametrierten Grundsystem angezeigt  
MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC (Grundsystem metrisch)

Die Anzeige der folgenden längenbehaftete Daten erfolgt im parametrierten Grundsystem:

- Maschinendaten
- Daten im Maschinenkoordinatensystem
- Werkzeugdaten
- Nullpunktverschiebungen

Die Anzeige der folgenden längenbehaftete Daten im programmierten Maßsystem:

- Daten im Werkstückkoordinatensystem

### Einlesen von Teileprogrammen von Extern

Werden Teileprogramme inklusive Datensätze (Nullpunktverschiebungen, Werkzeugkorrekturen, etc.) die in einem vom Grundsystem abweichenden Maßsystem programmiert wurden von Extern eingelesen, muss vorher die Grundstellung über das Maschinendatum MD10240 geändert werden.

### NC-PLC-Nahtstellensignale

Bei NC-PLC-Nahtstellensignalen, die maßabhängige Informationen enthalten, z.B. Vorschub für Bahn- und Positionierachsen, erfolgt der Datenaustausch mit der PLC im parametrierten Grundsystem.

### G-Funktionen G700/G710

Die G-Funktionen G700/G710 stellen eine Erweiterung von G70/G71 um folgende Funktionalität dar:

1. Vorschub wird im programmierten Maßsystem interpretiert:
  - G700: Längenangaben [Inch]; Vorschübe [Inch/min]
  - G710: Längenangaben [mm]; Vorschübe [mm/min]

Der programmierte Vorschub ist modal wirksam, bleibt also über nachfolgende G70/G71/G700/G710 hinaus wirksam. Soll der Vorschub im neuen G70/G71/G700/G710 Kontext wirksam werden, so muss dieser neu programmiert werden.

2. Das Lesen und Schreiben von längenbehafteten Systemvariablen und Maschinendaten im Teileprogramm erfolgt im programmierten Maßsystem.

## Unterschiede beim Lesen und Schreiben von Maschinendaten und Systemvariable

Es bestehen folgende Unterschiede zwischen G70/G71 und G700/G710 bezüglich dem Lesen und Schreiben von Maschinendaten und Systemvariable im Teileprogramm:

- G70/G71: Lesen und Schreiben erfolgt im parametrisierten Grundsystem
- G700/G710: Lesen und Schreiben erfolgt im programmierten Maßsystem

### Beispiel

Das folgende Teileprogramm wird mit metrischer Grundstellung ausgeführt:  
MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC = 1

Programmcode	Kommentar
N100 R1=0 R2=0	;
N120 G01 G70 X1 F1000	; prog. Maßsystem: inch
N130 \$MA_LUBRICATION_DIST[X]=10	; MD = 10 [mm] (Grundsystem)
N150 IF (\$AA_IW[X]>\$MA_LUBRICATION_DIST[X])	; SYS [mm] > MD [mm] (beide Grundsystem)
N160 R1=1	;
N170 ENDIF	;
N180 IF (\$AA_IW[X]>10)	; SYS [mm] (Grundsystem) > 10 [inch]
	; (prog. Maßsystem)
N190 R2=1	;
N200 ENDIF	;
N210 IF ( (R1+R2) = 1 )	; Alarm, wenn nur eine der beiden
	; Bedingungen (N150, N180) TRUE ist
N220 SETAL(61000)	;
N230 ENDIF	;
N240 M30	;

N120: wird G70 durch G700 ersetzt, tritt der Alarm 61000 (N220) nicht auf.

## Synchronaktionen

Damit bei Synchronaktion nicht der aktuelle Teileprogrammkontext das verwendete Maßsystem im Bedingungs- und/oder Aktionsteil bestimmt, muss die Festlegung des Maßsystems innerhalb der Synchronaktion (Bedingungs- und/oder Aktionsteil) erfolgen. Erst damit erreicht man ein definiertes und reproduzierbares Verhalten bei der Verwendung längenbehafteter Daten innerhalb einer Synchronaktion.

**Beispiel 1**

Innerhalb der Synchronaktion erfolgt keine Festlegung des Maßsystems. Die Verfahrbewegung der X-Achse erfolgt daher im Maßsystem der parametrisierten Grundstellung:

Programmcode	Kommentar
N100 R1=0	;
N110 G0 X0 Z0	;
N120 WAITP(X)	;
N130 ID=1 WHENEVER \$R1==1 DO POS[X]=10	; X = 10 inch oder mm, abhängig vom weiteren Teileprogramm
N140 R1=1	;
N150 G71 Z10 F10	; Z = 10 mm X = 10 mm
N160 G70 Z10 F10	; Z = 10 inch X = 10 inch
N170 G71 Z10 F10	; Z = 10 mm X = 10 mm
N180 M30	;

**Beispiel 2**

Innerhalb der Synchronaktion wird mit G71 explizit das Maßsystem "metrisch" programmiert. Die Verfahrbewegung der X-Achse erfolgt daher metrischen Maßsystem:

Programmcode	Kommentar
N100 R1=0	;
N110 G0 X0 Z0	;
N120 WAITP(X)	;
N130 ID=1 WHENEVER \$R1==1 DO G71 POS[X]=10	; X = 10 mm, unabhängig vom weiteren Teileprogramm
N140 R1=1	;
N150 G71 Z10 F10	; Z = 10 mm X = 10 mm
N160 G70 Z10 F10	; Z = 10 inch X = 10 mm
N170 G71 Z10 F10	; Z = 10 mm X = 10 mm
N180 M30	;



## Lesen und Schreiben von Daten bei G70/G71 und G700/G710 im Teileprogramm

Datenbereich	G70/G71		G700/G710	
	Lesen	Schreiben	Lesen	Schreiben
Anzeigen, Nachkommastellen (WKS)	P	P	P	P
Anzeigen, Nachkommastellen (MKS)	G	G	G	G
Vorschübe	G	G	P	P
Positionsangaben X, Y, Z	P	P	P	P
Interpolationsparameter I, J, K	P	P	P	P
Kreisradius (CR)	P	P	P	P
Polarradius (RP)	P	P	P	P
Gewindesteigung	P	P	P	P
Programmierbarer FRAME	P	P	P	P
Einstellbare FRAMES	G	G	P	P
Basisframes	G	G	P	P
Externe Nullpunktverschiebungen	G	G	P	P
axiale Preset-Verschiebung	G	G	P	P
Arbeitsfeldbegrenzungen (G25/G26)	G	G	P	P
Schutzbereiche	P	P	P	P
Werkzeugkorrekturen	G	G	P	P
Längenbehaftete Maschinendaten	G	G	P	P
Längenbehaftete Settingdaten	G	G	P	P
Längenbehaftete Systemvariablen	G	G	P	P
GUD's	G	G	G	G
LUD's	G	G	G	G
PUD's	G	G	G	G
R-Parameter	G	G	G	G
Siemenszyklen	P	P	P	P
Inkrementbewertung Jog/Handrad	G	G	G	G
P: Schreiben/Lesen erfolgt im programmierten Maßsystem G: Schreiben/Lesen erfolgt im parametrisierten Grundsystem				

**ACHTUNG****Lesen von Positionsdaten in Synchronaktionen**

Ohne explizite Programmierung des Maßsystems in der Synchronaktion (Bedingungsteil und/oder Aktionsteil bzw. Technologiefunktion) werden **längenbehaftete Positionsdaten** in der Synchronaktion immer im **parametrisierten Grundsystem** gelesen.

**Literatur:**

Programmierhandbuch Grundlagen; Liste der Adressen

### NC-spezifischer Umrechnungsfaktor

Standardmäßig ist der Umrechnungsfaktor im Maschinendatum:

MD10250 \$MN\_SCALING\_VALUE\_INCH (Umrechnungsfaktor für Umschaltung auf Inch-System)

auf 25,4 für die Umrechnung vom metrischen in das Inch-Maßsystem eingestellt. Durch Änderung des Umrechnungsfaktors kann die Steuerung auch an kundenspezifische Maßsysteme angepasst werden.

### Achsspezifischer Umrechnungsfaktor

Standardmäßig ist der Umrechnungsfaktor im achsspezifischen Maschinendatum:

MD31200 \$MA\_SCALING\_FACTOR\_G70\_G71 (Faktor für die Umrechnung bei aktivem G70/G71)

auf 25,4 für die Umrechnung vom metrischen in das Inch-Maßsystem eingestellt. Durch Änderung des Umrechnungsfaktors kann die Steuerung achsspezifisch auch an kundenspezifische Maßsysteme angepasst werden.

## 6.3.2 Manuelle Umschaltung des Grundsystems

### Allgemeines

Die Steuerung kann mit metrischen oder Inch-Maßsystemen arbeiten. Die Grundstellung des Maßsystems (Grundsystem) wird über folgende Maschinendaten festgelegt:

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC (Grundsystem metrisch).

Entsprechend des Grundsystems werden alle längenbehafteten Daten als metrische oder Inch-Maßangaben interpretiert.

Die Maßsystemumschaltung der Steuerung erfolgt über den entsprechenden Softkey der HMI-Bedienoberfläche im Bedienbereich "Maschine".

Die Maßsystemumschaltung wird nur unter folgenden Randbedingungen durchgeführt:

- MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM=1
- Bit 0 des MD20110 \$MC\_RESET\_MODE\_MASK ist in jedem Kanal gesetzt.
- Alle Kanäle sind im Reset-Zustand.
- Achsen werden nicht über JOG, DRF oder die PLC verfahren.
- Konstante Scheibenumfangsgeschwindigkeit (SUG) ist nicht aktiv.

Für die Dauer der Maßsystemumschaltung werden Aktionen wie Teileprogrammstart oder BA-Wechsel gesperrt.

Kann die Maßsystemumschaltung nicht vorgenommen werden, so wird das mit einer entsprechenden Meldung an der Bedienoberfläche angezeigt. Diese Festlegung stellt sicher, dass eine laufende Programmabarbeitung im Bezug auf das Maßsystem immer einem konsistenten Datensatz vorfindet.

Die eigentliche Umschaltung des Maßsystems wird über das Schreiben aller notwendigen Maschinendaten und anschließendem Wirksamsetzen mit `RESET` vorgenommen.

Das Maschinendatum:

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC

und die entsprechenden `G70/G71/G700/G710`-Einstellungen in Maschinendatum:

MD20150 \$MN\_GCODE\_RESET\_VALUES

werden für alle projektierten Kanäle automatisch und konsistent umgeschaltet.

**Dabei wechselt der Wert von Maschinendatum:**

**MD20150 \$MC\_GCODE\_RESET\_VALUES[12]**

**zwischen G700 und G710.**

Dieser Vorgang wird unabhängig von der aktuell eingestellten Schutzstufe durchgeführt.

---

#### **Hinweis**

Die Verfügbarkeit des Softkeys und damit der Funktionalität ist über das Kompatibilitätsmaschinendatum:

MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM

projektierbar.

Sind mehrere NCUs durch NCU-Link verbunden, so wirkt sich die Umschaltung einheitlich auf alle verbundenen NCUs aus. Sind die Voraussetzungen für eine Umschaltung auf einer der verbundenen NCUs nicht erfüllt, so findet die Umschaltung auf keiner der NCUs statt. Es wird davon ausgegangen, dass bei vorhandenem NCU-Link NCU-übergreifende Interpolationen stattfinden sollen, die nur bei einheitlichem Maßsystem korrekte Ergebnisse liefern können.

#### **Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3)

---

## Systemdaten

Bei der Umschaltung des Maßsystems werden aus Sicht des Bediener alle längenbehafteten Angaben in das neue Maßsystem automatisch umgerechnet.

Dazu zählen:

- Positionen
- Vorschübe
- Beschleunigungen
- Ruck
- Werkzeugkorrekturen
- Programmierbare, einstellbare und externe Nullpunktverschiebungen, DRF-Verschiebungen
- Kompensationswerte
- Schutzbereiche
- Maschinendaten
- Jog u. Handradbewertungen

Nach der Umschaltung sind alle o.g. Daten in physikalischen Größen verfügbar.

Daten, für die keine eindeutigen physikalischen Einheiten definiert sind, werden **keiner** automatischen Umrechnung unterzogen.

Dazu zählen:

- R-Parameter
- GUDs (Global User Data)
- LUDs (Local User Data)
- PUDs (Program global User Data)
- Analoge Ein-/Ausgänge
- Datenaustausch über den FC21

Hier ist der Anwender gefordert, das aktuell gültige Maßsystem:

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC

zu berücksichtigen.

An der PLC-Nahtstelle kann die aktuelle Maßsystemeinstellung über das Signal:

DB10 DBX107.7 (Inch-Maßsystem)

gelesen werden.

Über das Signal:

DB10 DBB71 (Änderungszähler Maßsystem inch/metrisch)

kann der "Maßsystem-Änderungszähler" ausgelesen werden

- Die Projektierung des Maßsystems für Durchhangkompensation erfolgt über:

MD32711 \$MA\_CEC\_SCALING\_SYSTEM\_METRIC

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

- Die Projektierung des Maßsystems für Positionsangaben der Teilungsachstabellen und der Schaltpunkte für Softwareschaltungen erfolgt über:

MD10270 \$MN\_POS\_TAB\_SCALING\_SYSTEM

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Softwareschaltungen, Wegschaltensignale (N3) / Teilungsachsen (T1)

## Anwenderwerkzeugdaten

Für die anwenderdefinierten Werkzeugdaten:

MD18094 \$MN\_MM\_NUM\_CC\_TDA\_PARAM

und Werkzeugschneidendaten:

MD18096 \$MN\_MM\_NUM\_CC\_TOA\_PARAM

werden jeweils zusätzliche Maschinendatensätze eingeführt:

MD10290 \$MN\_CC\_TDA\_PARAM\_UNIT [MM\_NUM\_CC\_TDA\_PARAM]

MD10292 \$MN\_CC\_TOA\_PARAM\_UNIT [MM\_NUM\_CC\_TOA\_PARAM]

Über diese Maschinendaten kann eine physikalische Einheit projiziert werden.

Entsprechend der Eingabe werden alle längenbehafteten anwenderdefinierten

Werkzeugdaten beim Umschalten automatisch in das neue Maßsystem umgerechnet.

## Referenzpunkt

Der Referenzpunkt bleibt erhalten. Ein erneutes Referenzieren ist nicht notwendig.

### Eingabe- und Rechenfeinheit

Die Eingabe-/Rechenfeinheit wird in der Steuerung über das Maschinendatum:

MD10200 \$MN\_INT\_INCR\_PER\_MM

eingestellt.

Standardeinstellungen:

Metrisches System	Inch System
1000 (0.001 mm)	0.0001

#### Beispiel:

1 Inch = 25.4 mm  $\Rightarrow$  0.0001 Inch = 0.00254 mm = 2.54  $\mu$ m

Um die letzten 40 nm noch programmieren und darstellen zu können, muss ein Wert von 100000 in das MD10200 eingegeben werden.

Erst mit dieser für beide Maßsysteme gleichen Einstellung können Maßsystemumschaltungen ohne einen nennenswerten Genauigkeitsverlust realisiert werden. Einmal so eingestellt, muss das MD10200 nicht mehr bei jeder Maßsystemumschaltung verändert werden.

### JOG und Handradbewertung

Das Maschinendatum:

MD31090 \$MA\_JOG\_INCR\_WEIGHT

besteht aus zwei Werten, die achsiale Inkrementbewertungen für jedes der beiden Maßsysteme beinhaltet.

In Abhängigkeit von der aktuellen Einstellung in Maschinendatum:

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC

wählt die Steuerung automatisch den passenden Wert aus.

Bereits zum Zeitpunkt der Inbetriebnahme legt der Anwender beide Inkrementbewertungen z.B. für die 1. Achse fest:

- Metrisch:

MD31090 \$MA\_JOG\_INCR\_WEIGHT[0;AX1]=0.001 mm

- Inch:

MD31090 \$MA\_JOG\_INCR\_WEIGHT[1;AX1]=0.00254 mm  $\triangleq$  0.0001 Inch

Damit muss nicht mehr bei jeder Inch/Metrisch-Umschaltung MD31090 geschrieben werden.

Ein inkrementelles Verfahren über JOG akkumuliert bei wechselnden Maßsystemen keine Wegreste, da alle internen Positionen sich immer auf mm beziehen.

## Datensicherung

Datensätze, die aus der Steuerung separat gelesen werden können und die über maßsystemrelevante Daten verfügen, erhalten beim Lesevorgang in Abhängigkeit von Maschinendatum:

MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM

eine mit Maschinendatum:

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC

korrespondierende INCH- bzw. METRIC-Kennung.

Damit soll festgehalten werden, in welchem Maßsystem die Daten ursprünglich ausgelesen worden sind.

Mit dieser Information soll verhindert werden, dass Datensätze mit einer Maßsystemeinstellung, die unterschiedlich zur aktuell eingestellten ist, in die Steuerung eingelesen werden können. In so einem Fall wird ein passender Alarm (15030) ausgelöst und der Schreibvorgang unterbrochen.

Da die Sprachanweisung auch in Teileprogrammen ausgewertet wird, können auch diese auf die o.g. Art und Weise gegen Fehlbedienung "abgesichert" werden. Man kann damit verhindern, dass Teileprogramme die z.B. nur metrische Angaben enthalten, in einem Inch-Maßsystem ablaufen können.

Archive und Maschinendatensätze sind bei einer Einstellung von:

MD11220 \$MN\_INI\_FILE\_MODE = 2

abwärtskompatibel.

---

### Hinweis

Die INCH/METRIC-Anweisung wird nur beim gesetzten Kompatibilitätsmaschinendatum:

MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM

generiert.

---

## Runden von Maschinendaten

Um Rundungsproblemen vorzubeugen, werden alle längenbehafteten Maschinendaten beim Schreiben im Inch-Maßsystem (MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC=0 und MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM=1) auf 1 pm gerundet.

Durch dieses Vorgehen werden die heute störenden Genauigkeitsverluste, die durch die Wandlung nach ASCII beim Auslesen einer Datensicherung im Inch-System entstehen, wieder beim Einlesen korrigiert.

### 6.3.3 FGROUP und FGREF

#### Programmierung

Für Bearbeitungsvorgänge, bei denen das Werkzeug oder das Werkstück oder beide von einer Rundachse bewegt werden (z.B. Laser-Bearbeitung von drehenden Rohren), soll der wirksame Bearbeitungsvorschub in gewohnter Weise als Bahnvorschub über den F-Wert programmiert werden können.

Dazu muss für jede der beteiligten Rundachsen ein effektiver Radius (Bezugsradius) angegeben werden. Dies wird ermöglicht durch die Programmierung der modal wirksamen NC-Adresse:

`FGREF [<Rundachse>] = <Bezugsradius>`

Die Einheit des Bezugsradius ist abhängig von der `G70/G71/G700/G710`-Einstellung.

Um überhaupt zur Berechnung des Bahnvorschubs beizutragen, müssen alle mitwirkenden Achsen wie bisher in den `FGROUP`-Befehl aufgenommen werden.

Um kompatibel zum Verhalten ohne `FGREF`-Programmierung zu bleiben, wird nach dem System-Hochlauf und bei `RESET` die Bewertung 1 Grad entspricht 1 mm wirksam.

Dies entspricht einem Bezugsradius von:

$$FGREF = 360\text{mm} / (2\pi) = 57.296 \text{ mm}$$

Diese Voreinstellung ist unabhängig vom aktiven Grundsystem (MD10240 `$MN_SCALING_SYSTEM_IS_METRIC`) und von der aktuell wirksamen `G70/G71/G700/G710`-Einstellung.

Besonderheiten der Vorschubbewertung von Rundachsen in `FGROUP`:

Programmcode
N100 FGROUP (X, Y, Z, A)
N110 G1 G91 A10 F100
N120 G1 G91 A10 X0.0001 F100

Bei angegebener Programmierung wird der programmierte F-Wert in Satz `N110` als Rundachsvorschub in Grad/min bewertet, während die Vorschubbewertung in Satz `N120` abhängig von der aktuell wirksamen inch/metrisch-Einstellung entweder 100 inch/min oder 100 mm/min ist.

VORSICHT
Die <code>FGREF</code> -Bewertung wirkt auch, wenn nur Rundachsen im Satz programmiert sind. Die gewohnte F-Wert-Interpretation als Grad/min gilt in diesem Fall nur, wenn der Radiusbezug entsprechend der <code>FGREF</code> -Voreinstellung ist:
<ul style="list-style-type: none"><li>• Bei <code>G71/G710</code>: <code>FGREF [A] = 57.296</code></li><li>• Bei <code>G70/G700</code>: <code>FGREF [A] = 57.296 / 25.4</code></li></ul>



**Beispiel**

Das folgende Beispiel soll die Wirkung von `FGROUP` auf den Bahnweg und Bahnvorschub verdeutlichen. Die Variable `$AC_TIME` enthält die Zeit vom Satzanfang in Sekunden. Sie ist nur in Synchronaktionen verwendbar.

Programmcode	Kommentar
N100 G0 X0 A0	
N110 FGROUP(X,A)	
N120 G91 G1 G710 F100	; Vorschub= 100mm/min bzw. 100Grad/min
N130 DO \$R1=\$AC_TIME	
N140 X10	; Vorschub= 100mm/min, Bahnweg= 10mm, R1= ca.6s
N150 DO \$R2=\$AC_TIME	
N160 X10 A10	; Vorschub= 100mm/min, Bahnweg= 14.14mm, R2= ca.8s
N170 DO \$R3=\$AC_TIME	
N180 A10	; Vorschub= 100Grad/min, Bahnweg= 10Grad, R3= ca.6s
N190 DO \$R4=\$AC_TIME	
N200 X0.001 A10	; Vorschub= 100mm/min, Bahnweg= 10mm, R4= ca.6s
N210 G700 F100	; Vorschub= 2540mm/min bzw. 100Grad/min
N220 DO \$R5=\$AC_TIME	
N230 X10	; Vorschub= 2540mm/min, Bahnweg= 254mm, R5= ca.6s
N240 DO \$R6=\$AC_TIME	
N250 X10 A10	; Vorschub= 2540mm/min, Bahnweg= 254,2mm, R6= ca.6s
N260 DO \$R7=\$AC_TIME	
N270 A10	; Vorschub= 100Grad/min, Bahnweg= 10Grad, R7= ca.6s
N280 DO \$R8=\$AC_TIME	
N290 X0.001 A10	; Vorschub= 2540mm/min, Bahnweg= 10mm, R8= ca.0.288s
N300 FGREF[A]=360/(2*\$PI)	; 1 Grad = 1 inch über den effektiven Radius einstellen.
N310 DO \$R9=\$AC_TIME	
N320 X0.001 A10	; Vorschub= 2540mm/min, Bahnweg= 254mm, R9= ca.6s
N330 M30	

## Diagnose

### Bezugsradius lesen

Der Wert des Bezugsradius einer Rundachse kann über Systemvariablen gelesen werden:

- Für die Anzeige in der Bedienoberfläche, in Synchronaktionen oder mit Vorlaufstopp im Teileprogramm über die Systemvariable:

`$AA_FGREF[<Achse>]` Aktueller Hauptlauf-Wert

- Ohne Vorlaufstopp im Teileprogramm über die Systemvariable:

`$PA_FGREF[<Achse>]` Programmierter Wert

Sind keine Werte programmiert, liest man in beiden Variablen für Rundachsen die Voreinstellung  $360 \text{ mm} / (2\pi) = 57.296 \text{ mm}$  (entspricht 1 mm pro Grad).

Für Linearachsen liest man in beiden Variablen immer den Wert 1 mm.

### Geschwindigkeitsbestimmende Bahnachsen lesen

Die an der Bahninterpolation beteiligten Achsen können über Systemvariablen gelesen werden:

- Für die Anzeige in der Bedienoberfläche, in Synchronaktionen oder mit Vorlaufstopp im Teileprogramm über die Systemvariablen:

`$AA_FGROUP[<Achse>]` Liefert den Wert "1", wenn die angegebene Achse per Grundeinstellung oder durch `FGROUP`-Programmierung einen Einfluss auf die Bahngeschwindigkeit im aktuellen Hauptlaufsatz hat. Wenn nicht, dann liefert die Variable den Wert "0".

`$AC_FGROUP_MASK` Liefert einen Bitschlüssel der mit `FGROUP` programmierten Kanalachsen, die zur Bahngeschwindigkeit beitragen sollen.

- Ohne Vorlaufstopp im Teileprogramm über die Systemvariablen:

`$PA_FGROUP[<Achse>]` Liefert den Wert "1", wenn die angegebene Achse per Grundeinstellung oder durch `FGROUP`-Programmierung einen Einfluss auf die Bahngeschwindigkeit hat. Wenn nicht, dann liefert die Variable den Wert "0".

`$P_FGROUP_MASK` Liefert einen Bitschlüssel der mit `FGROUP` programmierten Kanalachsen, die zur Bahngeschwindigkeit beitragen sollen.

## 6.4 Soll-/Istwertsystem

### 6.4.1 Allgemeines

#### Regelkreis

Für jede geregelte Achse/Spindel ist ein Regelkreis mit folgendem Aufbau konfigurierbar:

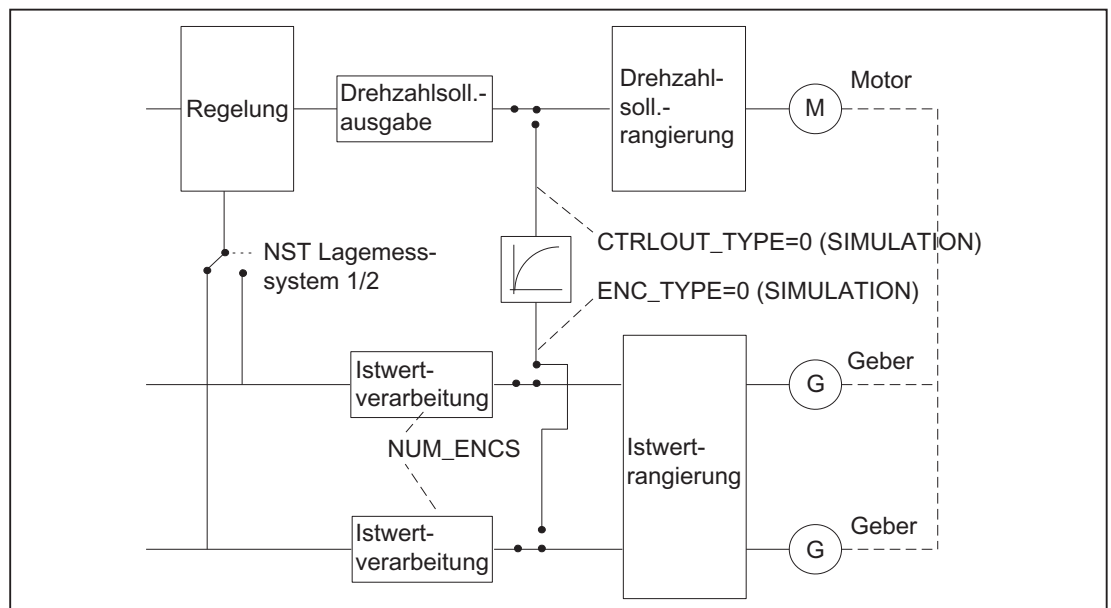


Bild 6-1 Prinzipschaltung eines Regelkreises

#### Sollwertausgabe

Je Achse/Spindel kann ein Sollwerttelegramm ausgegeben werden. Die Sollwertausgabe an den Steller erfolgt bei SINUMERIK 840D sl.

#### Istwerterfassung

Je Achse/Spindel können max. zwei Messsysteme angeschlossen werden, z. B. ein direktes Messsystem für den Bearbeitungsprozess mit hoher Anforderung an die Genauigkeit und ein indirektes Messsystem für schnelle Positionieraufgaben.

Die Anzahl der verwendeten Geber wird eingetragen in das Maschinendatum:

MD30200 \$MA\_NUM\_ENCS (Anzahl der Geber)

Bei zwei vorhandenen Istwertzweigen erfolgt die Istwerterfassung über beide Zweige.

Für die Lageregelung, Absolutwertberechnung und Anzeige wird immer das aktive Messsystem verwendet. Werden von der PLC-Nahtstelle beide gleichzeitig aktiviert, so wird steuerungsintern das Lagemesssystem 1 ausgewählt.

Das Referenzpunktfahren wird mit dem angewählten Messsystem durchgeführt.

Jedes Lagemesssystem muss getrennt referiert werden.

Erläuterungen zu Geberüberwachungen siehe Kapitel "A3: Achsüberwachungen, Schutzbereiche (Seite 85)".

Erläuterungen zu Kompensationsfunktionen für die Istwerterfassung siehe:

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

### Umschalten der Messsysteme

Über die folgenden NC/PLC-Nahtstellensignale kann zwischen den beiden Messsystemen umgeschaltet werden:

DB31, ... DBX1.5 (Lagemesssystem 1)

DB31, ... DBX1.6 (Lagemesssystem 2)

Weitere Informationen siehe Kapitel "A2: Diverse NC/PLC-Nahtstellensignale und Funktionen (Seite 33)".

Die Umschaltung zwischen den Messsystemen kann jederzeit geschehen, sie ist nicht vom Stillstand der Achsen abhängig. Die Umschaltung erfolgt nur, wenn eine zulässige Abweichung zwischen den Istwerten der beiden Messsysteme nicht überschritten wird.

Die zugehörige Toleranz wird eingetragen in das Maschinendatum:

MD36500 \$MA\_ENC\_CHANGE\_TOL (Maximale Toleranz bei Lageistwertumschaltung)

Bei der Umschaltung wird die aktuelle Abweichung zwischen Lagemesssystem 1 und 2 sofort verfahren.

### Überwachung

Die zulässige Abweichung zwischen den Istwerten der beiden Messsysteme ist einzutragen in das Maschinendatum:

MD36510 \$MA\_ENC\_DIFF\_TOL

Diese Differenz darf beim zyklischen Vergleich der beiden verwendeten Messsysteme nicht überschritten werden, ansonsten wird der Alarm 25105 "Messsysteme laufen auseinander" generiert.

Nicht aktiv ist die zugehörige Überwachung bei MD36510 = 0, wenn keine 2 Messsysteme in der Achse aktiv/vorhanden sind bzw. wenn die Achse nicht referenziert ist (zumindest akt. Regelungs-Messsystem).

### Arten der Istwerterfassung

Der verwendete Gebertyp muss festgelegt werden über das Maschinendatum:

MD30240 \$MA\_ENC\_TYPE (Art der Istwerterfassung (Lageistwert))

## Simulationsachsen

Zu Testzwecken kann der Drehzahlregelkreis einer Achse simuliert werden.

Die Achse "fährt" dadurch mit Schleppfehler, ähnlich wie eine echte Achse.

Eine Simulationsachse wird definiert, indem die beiden folgenden Maschinendaten auf "0" gesetzt werden:

MD30130 \$MA\_CTRL\_OUT\_TYPE[n] (Ausgabeart des Sollwertes)

MD30240 \$MA\_ENC\_TYPE[n] (Art der Istwerterfassung)

Nach Laden der Standardmaschinendaten sind die Achsen auf Simulation gesetzt.

Mit Referenzpunktfahren können Soll- und Istwert auf den Referenzpunktwert gesetzt werden.

Über das Maschinendatum:

MD30350 \$MA\_SIMU\_AX\_VDI\_OUTPUT (Ausgabe der Achssignale bei Simulationsachsen)

kann festgelegt werden, ob während der Simulation die achsspezifischen Nahtstellensignale an die PLC ausgegeben werden.

## Istwertkorrektur

Sollen Istwertkorrekturen, die von der NC auf dem für die Lageregelung ausgewählten Geber vorgenommen werden, nicht den Istwert eines weiteren in der gleichen Achse definierten Gebers beeinflussen, so ist dieser mit dem folgenden Maschinendatum "independent" (unabhängig) zu erklären:

MD30242 \$MA\_ENC\_IS\_INDEPENDENT

Zu den Istwertkorrekturen zählt man:

- Modulobehandlung
- Referenzpunktfahren
- Messsystemabgleich
- PRESET

## 6.4.2 Sollwert- und Geberzuordnung

### Sollwert-Rangierung

Für die Sollwert-Zuordnung einer Maschinenachse sind folgende Maschinendaten relevant.

MD30100		\$MA_CTRLOUT_SEGMENT_NR[ n ]
Sollwertzuordnung Bussegment		
System	Wert	Bedeutung
840D sl	5	PROFIBUS-DP / PROFINET (Voreinstellung)

MD30110		\$MA_CTRLOUT_MODULE_NR[ n ]
Sollwertzuordnung: Antriebsnummer / Baugruppennummer		
System	Wert	Bedeutung
840D sl	x	Es ist der Index x des MD13050 \$MN_DRIVE_LOGIC_ADDRESS[ x ] einzutragen, das auf den angeschlossenen Antrieb verweist. MD30110 \$MA_CTRLOUT_MODULE_NR[ n ] = x verweist auf: MD13050 \$MN_DRIVE_LOGIC_ADDRESS[ x ] <b>Hinweis</b> Wird der Antrieb simuliert (MD30130 \$MA_CTRLOUT_TYPE[ n ] = 0), hat das Maschinendatum keine Bedeutung.

MD30120		\$MA_CTRLOUT_NR[ n ]
Sollwertzuordnung: Sollwertausgang auf Antriebsmodul / Baugruppe		
System	Wert	Bedeutung
840D sl	1	Modularer Antrieb am PROFIBUS / PROFINET mit Profil PROFIdrive (Voreinstellung)

MD30130		\$MA_CTRLOUT_TYPE[ n ]
Ausgabeart des Sollwerts		
System	Wert	Bedeutung
840D sl	0	Simulation (Betrieb ohne Antrieb)
	1	Sollwertausgang aktiv

## Geberzuordnung

Für die Zuordnung der im PROFIdrive-Telegramm übertragenen Geber-Informationen des Antriebs zu den Gebereingängen der Maschinenachse sind folgende Maschinendaten relevant:

MD30210		\$MA_ENC_SEGMENT_NR[ n ]
Istwertzuordnung Bussegment		
System	Wert	Bedeutung
840D sl	5	PROFIBUS-DP / PROFINET

MD30220		\$MA_ENC_MODULE_NR[ n ]
Istwertzuordnung: Antriebsmodulnummer / Messkreisnummer		
System	Wert	Bedeutung
840D sl	x	Es ist die Nummer des über MD13050 \$MN_DRIVE_LOGIC_ADDRESS[ x ] zugeordneten Antriebs einzutragen. MD30220 \$MA_ENC_MODULE_NR[ n ] = x verweist auf: MD13050 \$MN_DRIVE_LOGIC_ADDRESS[ x ]

MD30230		\$MA_ENC_INPUT_NR[ n ]
Istwertzuordnung: Eingang auf Antriebsmodul / Messkreiskarte		
System	Wert	Bedeutung
840D sl	x	Nummer der Geber-Schnittstelle innerhalb des PROFIdrive-Telegramms Beispiele PROFIdrive-Telegramm 103 x = 1 → 1. Geber-Schnittstelle (G1_ZSW, G1_XIST1, G1_XIST2) x = 2 → 2. Geber-Schnittstelle (G2_ZSW, G2_XIST1, G2_XIST2) PROFIdrive-Telegramm 118 x = 1 → 1. Geber-Schnittstelle (G2_ZSW, G2_XIST1, G2_XIST2) x = 2 → 2. Geber-Schnittstelle (G3_ZSW, G3_XIST1, G3_XIST2) <b>Hinweis</b> Bei SINAMICS S120: - Geber 1 (G1...): Motorgeber - Geber 2 (G2...): Direktes Messsystem - Geber 3 (G3...): Zusätzliches Messsystem

6.4 Soll-/Istwertsystem

MD30240		\$MA_ENC_TYPE[ n ]
<b>Geber-Typ der Istwerterfassung (Lageistwert)</b>		
System	Wert	Bedeutung
840D sl	0	Simulation (Betrieb ohne Geber)
	1	Inkrementalgeber
	4	Absolutgeber
Hinweis Korrespondiert mit PROFIdrive-Parameter p979		

MD30242		\$MA_ENC_IS_INDEPENDENT[ n, Achse ]
<b>Geber ist unabhängig</b>		
System	Wert	Bedeutung
840D sl	0	Der Geber ist nicht unabhängig.
	1	Der Geber ist unabhängig. Sollen Istwertkorrekturen, die auf dem für die Lageregelung ausgewählten Geber vorgenommen werden, nicht den Istwert des zweiten in der gleichen Achse definierten Gebers beeinflussen, so ist dieser als unabhängig (independent) zu erklären. Istwertkorrekturen sind: <ul style="list-style-type: none"> <li>• - Modulobehandlung,</li> <li>• - Referenzpunktfahren,</li> <li>• - Messsystemabgleich,</li> <li>• - PRESET</li> </ul> Beispiel: eine Achse, 2 Geber, 2. Geber ist unabhängig MD30200 \$MA_NUM_ENCS[ AX1 ] = 2 MD30242 \$MA_ENC_IS_INDEPENDENT[ 0, AX1 ] = 0 MD30242 \$MA_ENC_IS_INDEPENDENT[ 1, AX1 ] = 1 Auswahl Lagemesssystem 1 / 2: DB31.DBX1.5 / 1.6 Ist Geber 1 zur Lageregelung ausgewählt, werden die Istwertkorrekturen nur auf diesem Geber durchgeführt, da Geber 2 unabhängig ist. Ist Geber 2 zur Lageregelung ausgewählt, werden die Istwertkorrekturen auf beiden Gebern durchgeführt, da Geber 1 nicht unabhängig ist. Das Maschinendatum hat also nur Auswirkung auf die passiven Geber einer Maschinenachse.
	2	Der passive Geber ist abhängig. Der Geberistwert wird durch den aktiven Geber verändert. In der Kombination mit MD35102 \$MA_REFP_SYNC_ENCS = 1 wird der passive Geber beim Referenzpunktfahren auf den aktiven Geber abgeglichen NICHT aber referenziert. Im Referenziermodus MD34200 \$MA_ENC_REFP_MODE = 3 (abstandscodierte Referenzmarken) wird der passive Geber mit der nächsten Verfahrbewegung nach dem Überfahren der Nullmarkendistanz automatisch referenziert. Dieses geschieht unabhängig von der aktuellen Betriebsarteneinstellung.



MD30242		\$MA_ENC_IS_INDEPENDENT[ n, Achse ]
Geber ist unabhängig		
System	Wert	Bedeutung
	3	Der Geber ist unabhängig. Bei Modulo-Rundachsen werden Modulo-Istwertkorrekturen auch im passiven Geber durchgeführt.

**Hinweis****Maschinendaten-Index [ n ]**

Der Maschinendaten-Index [ n ] für die Geberzuordnung hat folgende Bedeutung:

- n = 0: Erster der Maschinenachse zugeordneter Geber
- n = 1: Zweiter der Maschinenachse zugeordneter Geber

Die Zuordnung erfolgt über die Maschinendaten:

- MD30220\$MA\_ENC\_MODULE\_NR[ n ]
- MD30230\$MA\_ENC\_INPUT\_NR[ n ]

### 6.4.3 Anpassungen der Motor/Last-Verhältnisse

#### Getriebearten

Zur Anpassung der mechanischen Verhältnisse existieren folgende Getriebearten:

Getriebeart	Aktivierung	Anpassung	Anbauort
Motor-/Last-Getriebe	Parametersatz	fest projektiert	Getriebekasten
Geber-Mess-Getriebe	Power-On	geberabhängig	geberseitig
Last-Vorsatz-Getriebe	NewConfig	lastabhängig	werkzeugseitig

**Örtliche Lage der Getriebe / Geber**

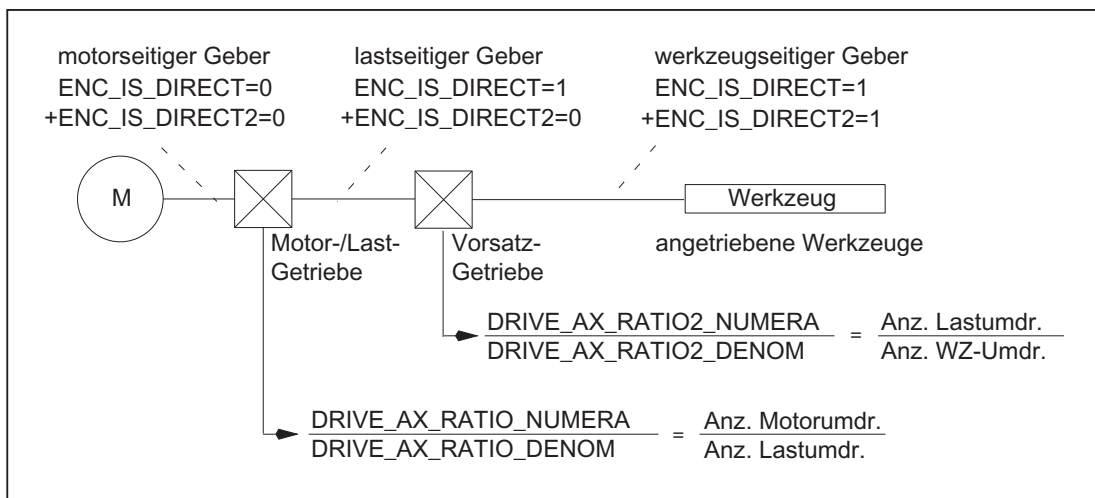


Bild 6-2 Getriebearten und Geberanbauorte

**Motor-/Last-Getriebe**

Das von SINUMERIK unterstützte Motor-/Last-Getriebe wird über folgende Maschinendaten projektiert:

MD31060 \$MA\_DRIVE\_AX\_RATIO\_NUMERA (Zähler Lastgetriebe)

MD31050 \$MA\_DRIVE\_AX\_RATIO\_DENOM (Nenner Lastgetriebe)

Die Getriebeübersetzung ergibt sich aus dem Verhältnis Zähler zu Nenner der beiden Maschinendaten. Über die dazugehörigen Parametersätze wird der Lageregler standardmäßig automatisch von der Steuerung auf die jeweiligen Übersetzungsverhältnisse synchronisiert.

Da ein Getriebestufenwechsel nicht immer automatisch erfolgen muss und es außerdem mehrere Möglichkeiten gibt, eine neue Getriebestufe anzufordern, wird nicht immer die Lageregelung über die Parametersätze einbezogen.

**Hinweis**

Weitere Informationen zu den Parametersätzen beim Getriebestufenwechsel siehe Kapitel "S1: Spindeln (Seite 1383)".


## Vorsatz-Getriebe

Von der Steuerung werden zusätzlich projektierbare Last-Vorsatz-Getriebe unterstützt:

MD31066 \$MA\_DRIVE\_AX\_RATIO2\_NUMERA (Zähler Vorsatzgetriebe)

MD31064 \$MA\_DRIVE\_AX\_RATIO2\_DENOM (Nenner Vorsatzgetriebe)

Angetriebene Werkzeuge bringen meistens ihr "eigenes" Vorsatz-Getriebe mit. Solche variablen Mechaniken können durch das multiplikativ zum Motor-/Last-Getriebe wirkende Vorsatz-Getriebe parametrierbar werden.

 <b>VORSICHT</b>
<p>Im Unterschied zum Motor-/Last-Getriebe gibt es beim Vorsatz-Getriebe keinen Parametersatz und damit auch keine Möglichkeit, die zeitsynchrone Umschaltung zum Teileprogramm oder zur PLC (NC/PLC-Nahtstelle) steuern zu können. Eine Teile-Bearbeitung während des Getriebewechsels ist deshalb auszuschließen. Es bleibt Aufgabe des Anwenders, die Synchronisation der jeweils geänderten Maschinendaten mit der entsprechenden mechanischen Umschaltung abzustimmen und zu aktivieren. Bei einer Umschaltung während einer Bewegung sind aufgrund der sich sprungartig ändernden Normierungsfaktoren Ausgleichsvorgänge <b>nicht</b> auszuschließen. Diese werden nicht auf Einhaltung der maximalen Beschleunigung überwacht.</p>

### Geber direkt am Werkzeug

Für das Vorsatz-Getriebe wird eine weitere Anschlussvariante für einen "werkzeugseitigen Geber" durch Projektierung von Maschinendatum:

MD31044 \$MA\_ENC\_IS\_DIRECT2

möglich.

### Geber nicht direkt am Werkzeug

Bei einer Getriebeumschaltung des Vorsatz-Getriebes im lagegeregelten Betrieb gelten folgende Randbedingungen:

- Die umzuschaltende Getriebe-Übersetzung geht in diesem Fall auch in eine Umnormierung der Geber-Informationen ein.

In diesem Fall gilt für Achsen/Spindeln im Positionierbetrieb:

- Eine Getriebeumschaltung ist **nur bei Stillstand sprungfrei** möglich.

Hierfür wird die werkzeugseitige Position softwareseitig vor und nach einer Getriebeumschaltung bei einer Übersetzungs-Änderung gleichgesetzt, da sich die mechanische Position während eines Getriebestufenwechsels nur wenig oder gar nicht ändert.

#### Empfehlung:

Zur Vermeidung von Alarm 21612 "Reglerfreigabe während der Bewegung zurückgesetzt" ist die Umschaltung "nur bei Stillstand" vorzuziehen. Es bleibt weiterhin zulässig und auch sinnvoll, die Achse bzw. Spindel vor oder während einer Getriebe-Umschaltung zusätzlich in den Drehzahl-Steuerbetrieb oder in den Nachführbetrieb zu schalten.

## Randbedingungen

Ist der für die Lageregelung genutzte **Geber direkt am Werkzeug** angeschlossen, so wirkt ein Getriebestufenwechsel des Motor-/Last-Getriebes oder des Vorsatz-Getriebes ausschließlich auf die physikalischen Größen an der Drehzahlschnittstelle zwischen NC und Antrieb. Die steuerungsinternen Parametersätze werden dabei nicht umgeschaltet.

## Referenzpunkt und Positionsbezug

Bei Getriebeumschaltungen kann mit Auswirkung auf die Geber-Normierung keine Aussage über Referenzpunkt- oder Maschinenpositions-Bezug getroffen werden. Teilweise nimmt die Steuerung in solchen Fällen den Status "Achse referenziert/synchronisiert" weg.

Ist der Positionsbezug zu Maschine, Werkzeug o. ä. verlorengegangen, dann muss durch einen entsprechenden Justage- oder Referenzvorgang der verloren gegangene Bezugspunkt erst wieder hergestellt werden. Dies ist insbesondere für die Funktionen "Fahren auf Festanschlag", "Referenzieren auf Bero-, Nocken- und/oder Nullmarke" wichtig.

### VORSICHT

Die Steuerung kann nicht alle möglichen Situationen erkennen, die zum Verlust des Maschinenpositions-Bezugs führen. Es bleibt deshalb der Verantwortung des Inbetriebnehmers bzw. Anwenders überlassen, in solchen Fällen ein explizites Referenzieren bzw. Nullmarken-Synchronisieren anzustoßen.

### Hinweis

Um ein Neu-Referenzieren ohne unterbrechenden RESET zu ermöglichen, sind die Maschinendaten:

MD34080 \$MA\_REFP\_MOVE\_DIST

und

MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR

auf NewConfig-Wirksamkeit umgestellt.

Weitere Erläuterungen siehe Kapitel "R1: Referenzieren (Seite 1319)".

## 6.4.4 Drehzahlsollwertausgabe

### Regelsinn und Verfahrrichtung der Vorschubachsen

Vor Beginn der Arbeit ist die Verfahrrichtung der Vorschubachse zu klären.

#### Regelsinn

Vor der Inbetriebnahme der Lageregelung müssen der Drehzahlregler und der Stromregler des Antriebs in Betrieb genommen und optimiert werden.

#### Verfahrrichtung

Mit dem Maschinendatum:

MD32100 \$MA\_AX\_MOTION\_DIR (Verfahrrichtung)

kann die Bewegungsrichtung der Achse umgekehrt werden,

ohne Auswirkung auf den Regelsinn der Lageregelung.

### Drehzahlsollwertanpassung

#### SINUMERIK 840D sl

Beim Drehzahlsollwertabgleich wird der NC zur Parametrierung der axialen Regelung und Überwachung mitgeteilt, welchem Drehzahlsollwert welche Motordrehzahl im Antrieb entspricht. Dieser Abgleich erfolgt automatisch.

Für PROFIBUS-DP Antriebe ist alternativ auch der manuelle Drehzahlsollwertabgleich möglich.

- **Manueller Abgleich**

In das Maschinendatum:

MD32250 \$MA\_RATED\_OUTVAL

wird ein Wert ungleich Null eingetragen.

---

#### Hinweis

##### **Geschwindigkeitsabgleich und Maximaler Drehzahlsollwert**

Ein Geschwindigkeitsabgleich ist bei einer SINUMERIK 840D sl aufgrund des automatischen Drehzahlsollwertabgleichs nicht erforderlich!

---

### Maximaler Drehzahlsollwert

Bei SINUMERIK 840D sl ist der maximale Drehzahlsollwert in Prozent festgelegt. 100% bedeutet maximaler Drehzahlsollwert bzw. Maximaldrehzahl bei PROFIdrive-Antrieben (herstellerspezifischer Einstellparameter im Antrieb, z.B. p1082 bei SINAMICS).

Die Ausgabe der Spindeldrehzahl ist bei der SINUMERIK 840D sl in der NC realisiert.

In der Steuerung sind Daten für 5 Getriebestufen realisiert.

Die Getriebestufen sind durch eine Minimal- und Maximaldrehzahl für die Getriebestufe und eine Minimaldrehzahl und eine Maximaldrehzahl für den automatischen Getriebestufenwechsel definiert. Die Ausgabe einer neuen Soll-Getriebestufe erfolgt nur, wenn der neu programmierte Drehzahlsollwert nicht in der aktuellen Getriebestufe gefahren werden kann.

Mit dem Maschinendatum:

MD36210 \$MA\_CTRLLOUT\_LIMIT[n] (Maximaler Drehzahlsollwert)

wird der Drehzahlsollwert prozentual begrenzt.

Werte bis 200% sind möglich.

Bei Überschreiten der Grenze wird ein Alarm ausgegeben.

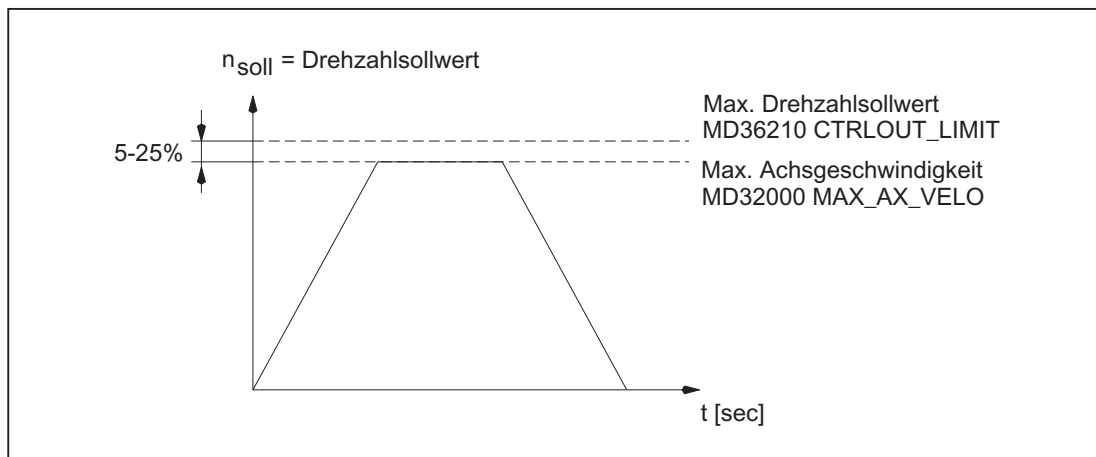


Bild 6-3 Maximaler Drehzahlsollwert

Die Achsen sollten wegen Regelungsvorgängen jedoch nicht erst bei 100% Drehzahlsollwert ihre Maximalgeschwindigkeit (MD32000 \$MA\_MAX\_AX\_VELO) erreichen, sondern bereits bei 80% bis 95%.

Bei Achsen, deren maximale Geschwindigkeit bei ca. 80% des Drehzahlsollwertbereiches erreicht wird, kann der Standardwert (default 80%) des Maschinendatums:

MD32000 \$MA\_MAX\_AX\_VELO (Maximale Achsgeschwindigkeit)

übernommen werden.

## 6.4.5 Istwertverarbeitung

### Istwertauflösung

Um einen korrekt geschlossenen Lageregelkreis zu erzeugen, ist es nötig, der Steuerung die Istwertauflösung mitzuteilen. Dazu dienen die folgenden achsspezifischen Maschinendaten, die teilweise vom Antrieb vorgegeben werden (MD31000, 31010, 31020, 31025).

Anhand der Maschinendaten wird die Istwertauflösung automatisch von der Steuerung errechnet. Die Regelungsparametersätze des Lagereglers werden als Servo-Parametersätze bezeichnet.

Basis der Lageistwerterfassung ist der Bearbeitungsprozess der Maschine.

Direktes Messsystem (DM) ist direkt an der Maschine:	Lastseitiger Geber
Indirektes Messsystem (IM) ist indirekt am Motor:	Motorseitiger Geber

Abhängig von der Achsart (Linearachse, Rundachse) und der Art der Istwerterfassung (direkt an Maschine/Werkzeug, indirekt am Motor) sind für die Berechnung der Istwertauflösung folgende Maschinendaten zu parametrieren:

Maschinendatum	Linearachse	Linearachse		Rundachse	
	Linearmaßstab/ oder als direktes Messsystem	Geber am Motor	Geber an Maschine und/oder Werkzeug	Geber am Motor	Geber an Maschine und/oder Werkzeug
MD30300 \$MA_IS_ROT_AX	0	0	0	1	1
MD31000 \$MA_ENC_IS_LINEAR[n]	1	0	0	0	0
MD31010 \$MA_ENC_GRID_POINT_DIST[n]	Teilung	-	-	-	-
MD34320 \$MA_ENC_INVERS[n]	◆	-	-	-	-
MD31040 \$MA_ENC_IS_DIRECT[n]	- / 1	0	1	0	1
MD31044 \$MA_ENC_IS_DIRECT2[n]	- / 1	0	1	0	1
MD31020 \$MA_ENC_RESOL[n]	-	Striche/ Umdr.	Striche/ Umdr.	Striche/ Umdr.	Striche/ Umdr.
MD31025 \$MA_ENC_PULSE_MULT[n]	Gebervervielfachung				
MD31030 \$MA_LEADSCREW_PITCH	-	mm/Umdr.	mm/Umdr.	-	-
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[n]	-	Last- umdr.	-/1	Last- umdr.	•
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[n]	-	Motor- umdr. falls Zustell- getr. vorh.	-/1	Motor- umdr.	•

6.4 Soll-/Istwertsystem

Maschinendatum	Linearachse	Linearachse		Rundachse	
	Linearmaßstab/ oder als direktes Messsystem	Geber am Motor	Geber an Maschine und/oder Werkzeug	Geber am Motor	Geber an Maschine und/oder Werkzeug
MD31070 \$MA_DRIVE_ENC_RATIO_DENOM[n]	-	Geber- umdr.	Geber- umdr.	Geber- umdr.	Geber- umdr.
MD31080 \$MA_DRIVE_ENC_RATIO_NUMERA[n]	-	Motor- seitiger Geber*	Motor- umdr.	Motor- umdr.	Last- umdr.

- = Für diese Kombination irrelevant
- \* Der motorseitige Geber ist ein Einbaugeber und hat damit **kein** Messgetriebe. Das Übersetzungsverhältnis ist immer 1:1.
- ◆ Bei abstandscodierten Messsystemen
- Diese Maschinendaten werden nicht für die Geberanpassung (Wegbewertung) benötigt. Sie müssen jedoch für die Sollwertberechnung richtig eingegeben werden! Es stellt sich sonst nicht der gewünschte  $K_v$ -Faktor ein. In Maschinendatum MD31050 \$MA\_DRIVE\_AX\_RATIO\_DENOM werden die Lastumdrehungen, in Maschinendatum MD31060 \$MA\_DRIVE\_AX\_RATIO\_NUMERA die Motorumdrehungen eingetragen.

Maschinendaten

Geber-abhängige Maschinendaten	Bedeutung
MD31000 \$MA_ENC_IS_LINEAR[n]	Direktes Messsystem Linearmaßstab
MD31044 \$MA_ENC_IS_DIRECT2[n]	Geber am Vorsatz-Getriebe
MD31070 \$MA_DRIVE_ENC_RATIO_DENOM[n]	Nenner Messgetriebe
MD31080 \$MA_DRIVE_ENC_RATIO_NUMERA[n]	Zähler Messgetriebe
MD31010 \$MA_ENC_GRID_POINT_DIST[n]	Teilungsperiode bei Linearmaßstäben
MD31020 \$MA_ENC_RESOL[n]	Geberstriche pro Umdrehung bei rotierenden Geber
MD31040 \$MA_ENC_IS_DIRECT[n]	Geber ist direkt an der Maschine
MD34080 \$MA_REFP_MOVE_DIST [n]	Referenzpunktabstand
MD34090 \$MA_REFP_MOVE_DIST_CORR[n]	Referenzpunktverschiebung
MD34320 \$MA_ENC_INVERS[n]	Längenmesssystem ist gegensinnig
n: Geber-Index, mit n = 0, 1, ... (1. Geber, 2. Geber, ...)	

Parametersatz-abhängige Maschinendaten	Bedeutung
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[m]	Nenner Lastgetriebe
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[m]	Zähler Lastgetriebe
m: Parametersatz-Index, mit m = 0, 1, ... (1. Parametersatz, 2. Parametersatz, ...)	



Geber- und Parametersatz-unabhängige Maschinendaten	Bedeutung
MD30200 \$MA_NUM_ENCS	Anzahl der Geber
MD30300 \$MA_IS_ROT_AX	Rundachse
MD31030 \$MA_LEADSCREW_PITCH	Steigung der Kugelrollspindel
MD31064 \$MA_DRIVE_AX_RATIO2_DENOM	Nenner Vorsatz-Gertriebe
MD31066 \$MA_DRIVE_AX_RATIO2_NUMERA	Zähler Vorsatz-Gertriebe
MD32000 \$MA_MAX_AX_VELO	Maximale Achsgeschwindigkeit

#### Hinweis

Maschinendaten mit Wirksamkeitskriterium NEWCONFIG können entweder im Teileprogramm mit den Befehl `NEWCONF` oder über die Bedienoberfläche per Softkey aktiviert werden.

## 6.4.6 Istwertauflösung

### 6.4.6.1 Funktionsbeschreibung

Das Verhältnis von steuerungsinterner Rechenfeinheit zur Istwertauflösung ist ein Maß dafür, wie exakt die von der Steuerung berechneten Werte an der Maschine umgesetzt werden können.

#### Linearachsen

$$\frac{\text{Rechenfeinheit}}{\text{Istwertauflösung}} = \frac{\text{Interne Inkremente / (mm)}}{\text{Geberinkremente / (mm)}}$$

#### Rundachsen

$$\frac{\text{Rechenfeinheit}}{\text{Istwertauflösung}} = \frac{\text{Interne Inkremente / (Grad)}}{\text{Geberinkremente / (Grad)}}$$

### Relevante Maschinendaten für die Istwertauflösung

Die Istwertauflösung ergibt sich aus dem Aufbau der Maschine, ob Getriebe vorhanden sind und deren Übersetzung, der Steigung der Kugelrollspindel bei Linearachsen und der Auflösung des verwendeten Gebers. An der Steuerung sind dazu folgende Maschinendaten einzustellen:

Nummer	Bezeichner \$MA_	Bedeutung
30300	IS_ROT_AX	Achse ist Rundachse / Spindel
31000	ENC_IS_LINEAR[ n ]	Messsystem ist ein Linearmaßstab
31010	ENC_GRID_POINT_DIST	Teilungsperiode des Linearmaßstabs
31020	ENC_RESOL[ n ]	Geberstriche pro Umdrehung
31025	ENC_PULSE_MULT[ n ]	Geber-Vervielfachung (Hochauflösung)
31030	LEADSCREW_PITCH	Steigung der Kugelrollspindel
31040	ENC_IS_DIRECT[ n ]	Direktes oder indirektes Messsystem
31044	ENC_IS_DIRECT2[ n ]	Geber am Vorsatzgetriebe
31050	DRIVE_AX_RATIO_DENOM[ m ]	Nenner Lastgetriebe
31060	DRIVE_AX_RATIO_NUMERA[ m ]	Zähler Lastgetriebe
31064	DRIVE_AX_RATIO2_DENOM	Nenner Vorsatzgetriebe
31066	DRIVE_AX_RATIO2_NUMERA	Zähler Vorsatzgetriebe
31070	DRIVE_ENC_RATIO_DENOM[ n ]	Nenner Messgetriebe
31080	DRIVE_ENC_RATIO_NUMERA[ n ]	Zähler Messgetriebe
n: Geber-Index, mit n = 0, 1, ... (1. Geber, 2. Geber, ...)		
m: Parametersatz-Index, mit m = 0, 1, ... (1. Parametersatz, 2. Parametersatz, ...)		

### Relevante Maschinendaten für die Rechenfeinheit

Die Rechenfeinheit, d.h. die Auflösung mit der steuerungsintern alle wegbezogenen Daten gerechnet werden, ist für Linear- und Rundachsen getrennt, über folgende Maschinendaten einzustellen:

Nummer	Bezeichner \$MA_	Bedeutung
10200	INT_INCR_PER_MM	Rechenfeinheit für Linearpositionen
10210	INT_INCR_PER_DEG	Rechenfeinheit für Winkelpositionen

### Einstellempfehlung

Die oben genannten Komponenten und Einstellungen, die für die Istwertauflösung verantwortlich sind, sollte so gewählt werden, dass die Istwertauflösung größer ist als die parametrisierte Rechenfeinheit.

$$\frac{\text{Rechenfeinheit}}{\text{Istwertauflösung}} \leq 1$$

6.4.6.2 Beispiel: Linearachse mit Linearmaßstab

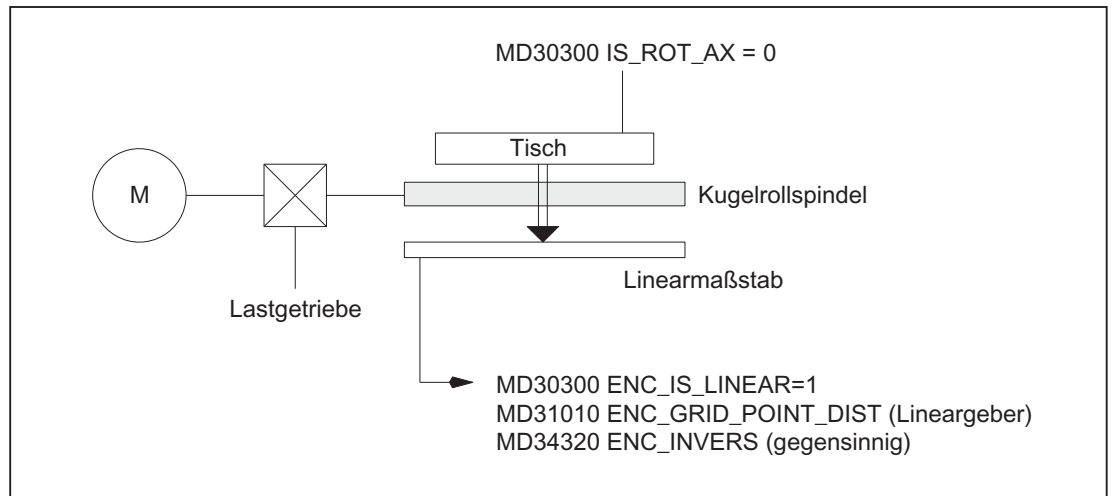


Bild 6-4 Linearachse mit Linearmaßstab

Das Verhältnis von internen Inkrementen zu Geberinkrementen pro mm berechnet sich zu:

$$\frac{\text{Interne Inkremente / mm}}{\text{Geberinkremente / mm}} = \frac{\text{ENC\_GRID\_POINT\_DIST [n]} * \text{INT\_INCR\_PER\_MM}}{\text{ENC\_PULSE\_MULT[n]}}$$

6.4.6.3 Beispiel: Linearachse mit rotatorischem Geber am Motor

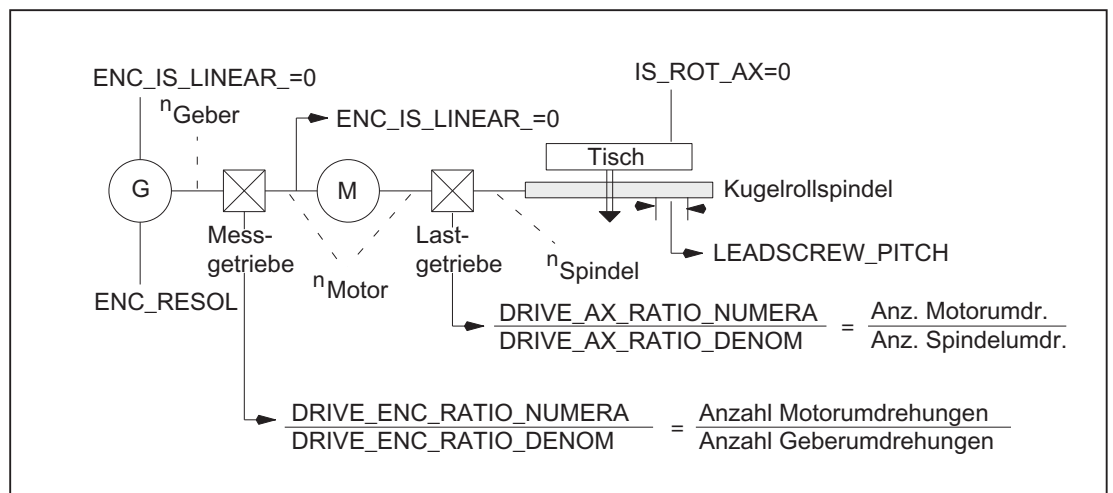


Bild 6-5 Linearachse mit rotatorischem Geber am Motor

Das Verhältnis von internen Inkrementen zu Geberinkrementen pro mm berechnet sich zu:

$$\frac{\text{Interne Inkremente / mm}}{\text{Geberinkremente / mm}} = \frac{1}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}}$$

$$* \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}}$$

$$* \frac{\text{DRIVE\_AX\_RATIO\_DENOM [n]}}{\text{DRIVE\_AX\_RATIO\_NUMERA [n]}}$$

$$* \text{LEADSCREW\_PITCH}$$

$$* \text{INT\_INCR\_PER\_MM}$$

**Beispiel**

Annahmen:

- Rotatorischer Geber am Motor: 2048 Impulse / Umdrehung
- Interne Impulsvervielfachung: 2048
- Getriebe, Motor / Kugelrollspindel: 5:1
- Steigung der Kugelrollspindel: 10 mm / Umdrehung
- Rechenfeinheit: 10000 Inkremente pro mm

Maschinendatum	Wert
MD30300 \$MA_IS_ROT_AX	0
MD31000 \$MA_ENC_IS_LINEAR[0]	0
MD31040 \$MA_ENC_IS_DIRECT[0]	0
MD31020 \$MA_ENC_RESOL[0]	2048
MD31025 \$MA_ENC_PULSE_MULT	2048
MD31030 \$MA_LEADSCREW_PITCH	10
MD31080 \$MA_DRIVE_ENC_RATIO_NUMERA[0]	1
MD31070 \$MA_DRIVE_ENC_RATIO_DENOM[0]	1
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[0]	5
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[0]	1
MD10210 \$MN_INT_INCR_PER_DEG	10000

$$\Rightarrow \frac{\text{Interne Inkremente / mm}}{\text{Geberinkremente / mm}} = \frac{1}{2048 * 2048} * \frac{1}{1} * \frac{1}{5} * 10 \text{ mm} * 10000 \text{ Inkr./mm} = 0,004768$$

Ein Geberinkrement entspricht 0,004768 internen Inkrementen bzw. 209,731543 Geberinkremente entsprechen einem internen Inkrement.

#### 6.4.6.4 Beispiel: Linearachse mit rotatorischem Geber an der Maschine

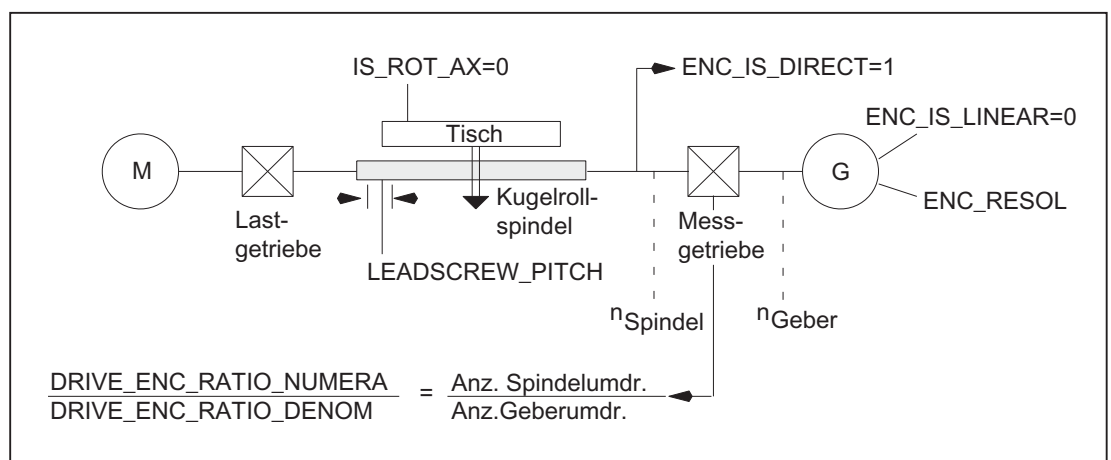


Bild 6-6 Linearachse mit rotatorischem Geber an der Maschine

Das Verhältnis von internen Inkrementen zu Geberinkrementen pro mm berechnet sich zu:

$$\frac{\text{Interne Inkremente / mm}}{\text{Geberinkremente / mm}} = \frac{1}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}} * \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}} * \text{LEADSCREW\_PITCH} * \text{INT\_INCR\_PER\_MM}$$

6.4.6.5 Beispiel: Rundachse mit rotatorischem Geber am Motor

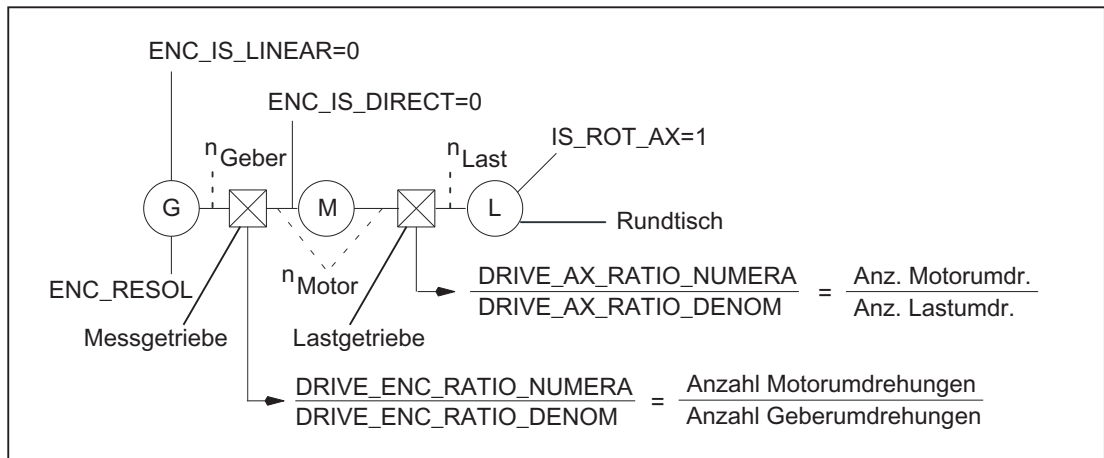


Bild 6-7 Rundachse mit rotatorischem Geber am Motor

Das Verhältnis von internen Inkrementen zu Geberinkrementen pro Grad berechnet sich zu:

$$\frac{\text{Interne Inkremente / Grad}}{\text{Geberinkremente / Grad}} = \frac{360 \text{ Grad}}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}}$$

$$* \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}}$$

$$* \frac{\text{DRIVE\_AX\_RATIO\_DENOM [n]}}{\text{DRIVE\_AX\_RATIO\_NUMERA [n]}}$$

$$* \text{INT\_INCR\_PER\_DEG}$$

Beispiel

Annahmen:

- Rotatorischer Geber am Motor: 2048 Impulse / Umdrehung
- Interne Impulsvervielfachung: 2048
- Getriebe, Motor / Rundachse: 5:1
- Rechenfeinheit: 1000 Inkremente pro Grad

Maschinendatum	Wert
MD30300 \$MA_IS_ROT_AX	1
MD31000 \$MA_ENC_IS_LINEAR[0]	0
MD31040 \$MA_ENC_IS_DIRECT[0]	0
MD31020 \$MA_ENC_RESOL[0]	2048
MD31025 \$MA_ENC_PULSE_MULT	2048
MD31080 \$MA_DRIVE_ENC_RATIO_NUMERA[0]	1
MD31070 \$MA_DRIVE_ENC_RATIO_DENOM[0]	1
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[0]	5
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[0]	1
MD10210 \$MN_INT_INCR_PER_DEG	1000

$$\Rightarrow \frac{\text{Interne Inkremente / Grad}}{\text{Geberinkremente / Grad}} = \frac{360 \text{ Grad}}{2048 * 2048} * \frac{1}{1} * \frac{1}{5}$$

$$* 1000 \text{ Inkr./Grad} = 0,017166$$

Ein Geberinkrement entspricht 0,017166 internen Inkrementen bzw. 58,254689 Geberinkremente entsprechen einem internen Inkrement.

#### 6.4.6.6 Beispiel: Rundachse mit rotatorischem Geber an der Maschine

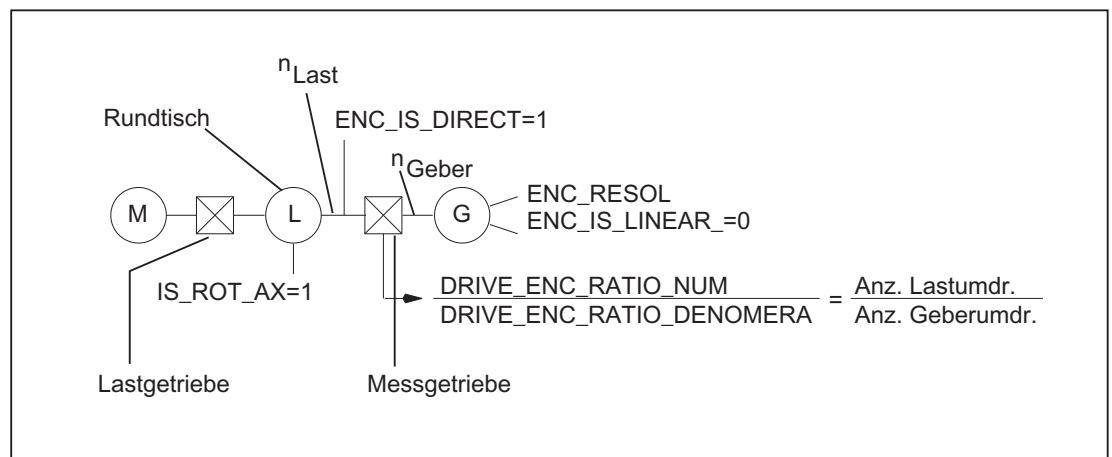


Bild 6-8 Rundachse mit rotatorischem Geber an der Maschine

Das Verhältnis von internen Inkrementen zu Geberinkrementen pro Grad berechnet sich zu:

$$\frac{\text{Interne Inkremente / Grad}}{\text{Geberinkremente / Grad}} = \frac{360 \text{ Grad}}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}}$$

$$* \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}}$$

$$* \text{INT\_INCR\_PER\_DEG}$$

6.4.6.7 Beispiel: Vorsatz-Getriebe mit Geber am Werkzeug

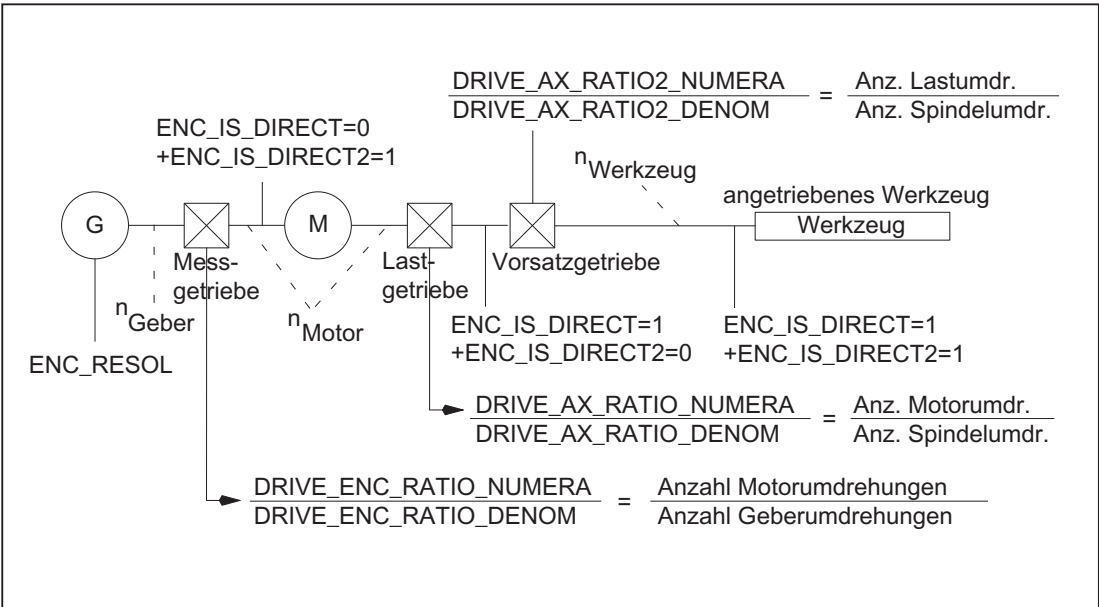


Bild 6-9 Vorsatz-Getriebe mit Geber direkt am angetriebenen Werkzeug

Das Verhältnis von internen Inkrementen zu Geberinkrementen pro Grad berechnet sich zu:

$$\frac{\text{Interne Inkremente / Grad}}{\text{Geberinkremente / Grad}} = \frac{360 \text{ Grad}}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}}$$

$$* \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}}$$

$$* \text{INT\_INCR\_PER\_DEG}$$



## 6.5 Regelung

### 6.5.1 Allgemeines

#### Lageregelung einer Achse/Spindel

Die Regelung einer Achse besteht aus dem Strom- und Drehzahlregelkreis des Antriebes und einem übergeordneten Lageregelkreis in der NC.

Die Lageregelung einer Achse/Spindel ist im Prinzip wie folgt aufgebaut:

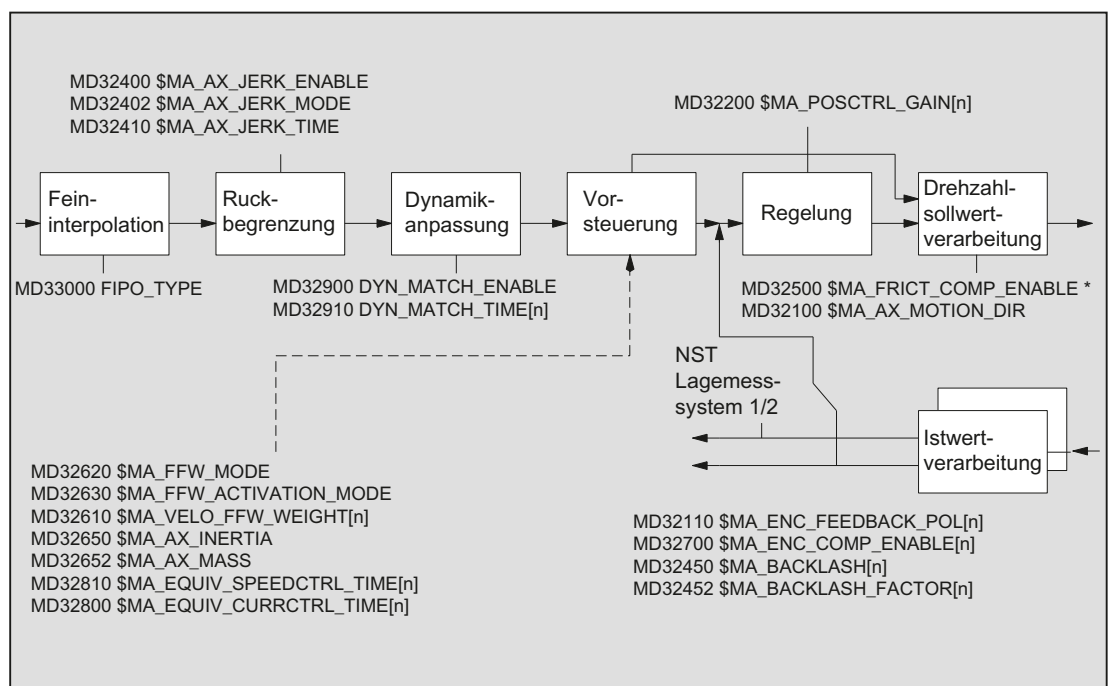


Bild 6-10 Prinzipielle Darstellung der Sollwert-Verarbeitung und Regelung

Informationen zur Ruckbegrenzung siehe Kapitel "B2: Beschleunigung (Seite 245)".

Beschreibung der Vorsteuerung, Umkehrlose-, Reibkompensation mit weiteren Maschinendaten und Spindelsteigungsfehlerkompensation siehe:

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

### Feininterpolation

Mit dem Feininterpolator (FIPO) kann die Konturgüte durch die Verringerung des Treppeneffektes beim Drehzahlsollwert weiter erhöht werden. Es können 3 Arten der Feininterpolation eingestellt werden:

MD33000 \$MA\_FIPO\_TYPE = <FIPO-Mode>

<FIPO-Mode>	Bedeutung
1	Differenzielle Feininterpolation mit Mittelwertbildung (Glättung) über einen IPO-Takt
2	Kubische Feininterpolation
3	Für die Verwendung mit der Vorsteuerung optimierte kubische Feininterpolation für höchste Konturgenauigkeit

### K<sub>v</sub>-Faktor

Damit im Bahnsteuerbetrieb nur geringe Konturabweichungen auftreten, ist ein hoher K<sub>v</sub>-Faktor erforderlich:

MD32200 \$MA\_POSCTRL\_GAIN[n]

Ein zu hoher K<sub>v</sub>-Faktor führt jedoch zu Instabilität, Überschwingungen und evtl. zu unzulässig hohen Maschinenbelastungen.

Der maximal zulässige K<sub>v</sub>-Faktor ist abhängig von:

- Auslegung und Dynamik des Antriebs  
(Anregelzeit, Beschleunigungs- und Bremsvermögen)
- Güte der Maschine  
(Elastizität, Schwingungsdämpfung)
- Lageregeltakt bzw. Drehzahlregeltakt bei aktivem DSC

Der K<sub>v</sub>-Faktor ist definiert als:

$$K_v = \frac{\text{Geschwindigkeit}}{\text{Schleppabstand}} ; \frac{[\text{m/min}]}{[\text{mm}]}$$

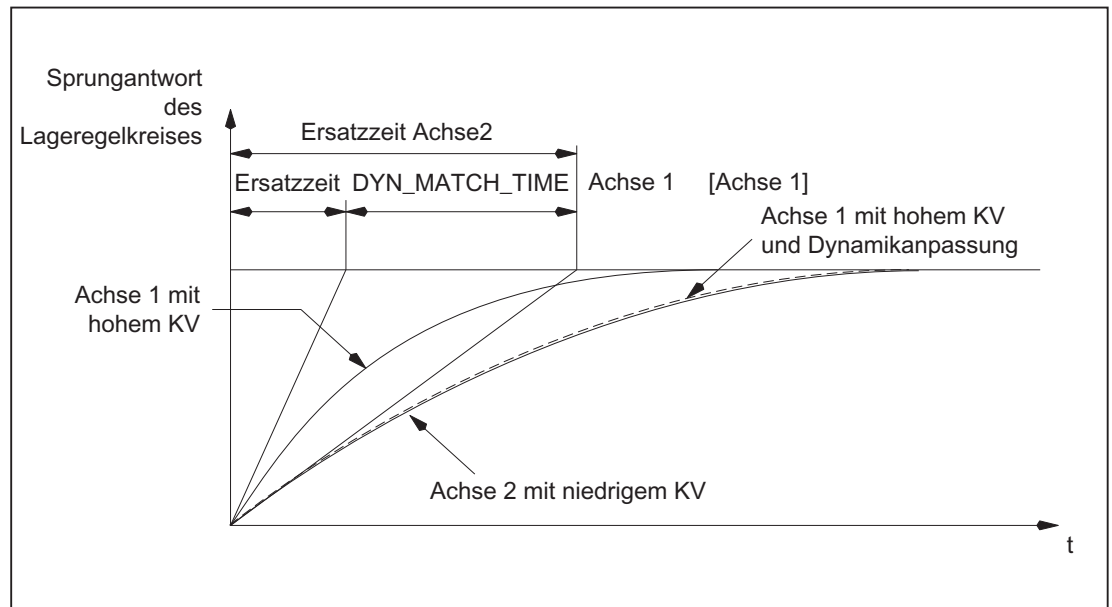
K<sub>v</sub>-Faktor-Einstellung bei SINUMERIK 840D sl

Bild 6-11 Dynamikanpassung

## Dynamikanpassung

Achsen die miteinander interpolieren aber unterschiedliche K<sub>v</sub>-Faktoren aufweisen, können durch die Dynamikanpassung auf gleichen Schleppabstand eingestellt werden. Dadurch lässt sich eine optimale Konturgenauigkeit ohne Verlust von Regelgüte durch Verringerung der K<sub>v</sub>-Faktoren auf die dynamisch schwächste Achse erreichen.

Die Funktion wird aktiviert über:

MD32900 \$MA\_DYN\_MATCH\_ENABLE = 1 (Dynamikanpassung)

Die Dynamikanpassung erfolgt durch Vorgabe einer neuen Ersatzzeitkonstante. Diese berechnet sich aus der Differenz der Ersatzzeitkonstanten der dynamisch schwächsten Achse und der anzupassenden Achse:

MD32910 \$MA\_DYN\_MATCH\_TIME[n] = <Differenz der Ersatzzeitkonstanten>

**Beispiel für eine Dynamikanpassung von 3 Achsen ohne Drehzahlvorsteuerung**

Die Ersatzzeitkonstante des Lageregelkreises betrage:	
Achse 1:	30 ms
Achse 2:	20 ms
Achse 3:	24 ms

Mit einer Ersatzzeitkonstante von 30 ms ist die Achse 1 die dynamisch schwächste Achse.

Daraus ergeben sich für die Achsen folgende neuen Ersatzzeitkonstanten:

- Achse 1: MD32910 \$MA\_DYN\_MATCH\_TIME = 0 ms
- Achse 2: MD32910 \$MA\_DYN\_MATCH\_TIME = 30 ms - 20 ms = 10 ms
- Achse 3: MD32910 \$MA\_DYN\_MATCH\_TIME = 30 ms - 24 ms = 6 ms

**Näherungsformeln für die Ersatzzeitkonstante des Lageregelkreises einer Achse**

Die Ersatzzeitkonstante  $T_{Ersatz}$  des Lageregelkreises einer Achse berechnet sich abhängig von der Art der Vorsteuerung näherungsweise zu:

- Ohne Vorsteuerung:

$$T_{Ersatz} \approx \frac{1}{MD32200 POSCTRL\_GAIN [1/s]}$$

- Mit Drehzahlvorsteuerung:

$$T_{Ersatz} \approx MD32810 EQUIV\_SPEEDCTRL\_TIME$$

- Bei kombinierter Momenten-/Drehzahl-Vorsteuerung

$$T_{Ersatz} \approx MD32800 EQUIV\_CURRCTRL\_TIME$$

**Hinweis**

Wird eine Dynamikanpassung bei einer Geometrieachse vorgenommen, müssen alle anderen Geometrieachsen auf die gleiche Dynamik eingestellt werden.

**Literatur:**

Inbetriebnahmehandbuch CNC: NCK, PLC, Antriebe

## 6.5.2 Parametersätze des Lagereglers

Die Lageregelung kann

Zur schnellen Anpassung der Lageregelung an veränderte Eigenschaften der Maschine während des Betriebes, z. B. bei Getriebeumschaltung der Spindel, oder zur Dynamikanpassung an eine andere Achse, z.B. beim Gewindebohren, stehen pro Maschinenachse sechs Parametersätze zur Verfügung.

Ein Parametersatz umfasst folgende Maschinendaten:

Nummer	Bezeichner \$MA_	Bedeutung
31050	DRIVE_AX_RATIO_DENOM	Nenner Lastgetriebe
31060	DRIVE_AX_RATIO_NUMERA	Zähler Lastgetriebe
32200	POSCTRL_GAIN	K <sub>v</sub> -Faktor
32452	BACKLASH_FACTOR	Losekompensation
32610	VELO_FFW_WEIGHT	Vorsteuerfaktor
36012	STOP_LIMIT_FACTOR	Faktor Genauhalt grob/fein und Stillstand
32800	EQUIV_CURRCTRL_TIME	Ersatzzeitkonst. Stromregelkreis für Vorsteuerung
32810	EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonst. Drehzahlregelkreis für Vorsteuerung
32910	DYN_MATCH_TIME	Zeitkonstante der Dynamikanpassung
36200	AX_VELO_LIMIT	Schwellwert für Geschwindigkeitsüberwachung

### Gewindebohren / -schneiden

Für das Gewindebohren / -schneiden gilt bezüglich der Parametersätze von Achsen:

- Bei Maschinenachsen, die **nicht** am Gewindebohren oder -schneiden beteiligt sind, ist immer Parametersatz 1 (Index = 0) aktiv. Die anderen Parametersätze müssen nicht berücksichtigt werden.
- Bei Maschinenachsen, die am Gewindebohren oder -schneiden beteiligt sind, wird die gleiche Parametersatznummer wie die der aktuellen Getriebestufe der Spindel aktiv.  
Alle Parametersätze entsprechen den Getriebestufen und müssen deshalb parametrieren werden.

Der aktuelle Parametersatz wird auf der Bedienoberfläche angezeigt unter:

#### SINUMERIK Operate

Bedienbereichumschaltung" > "Diagnose" > "Service Achse"

**Parametersätze beim Getriebestufenwechsel**

Jeder Getriebestufe einer Spindel ist ein eigener Parametersatz zugeordnet. Die Anwahl der Getriebestufe erfolgt über folgendes NC/PLC-Nahtstellensignal:

DB31, ... DBX16.0 - 16.2 = <Istgetriebestufe>

Istgetriebestufe	DB31, ... DBX16.0 - 16.2	Parametersatz	
1. Getriebestufe	000	2	(Index=1)
1. Getriebestufe	001	2	(Index=1)
2. Getriebestufe	010	3	(Index=2)
3. Getriebestufe	011	4	(Index=3)
4. Getriebestufe	100	5	(Index=4)
5. Getriebestufe	101 110 111	6	(Index=5)

Weitere Informationen zu Getriebestufen für Spindeln siehe Kapitel "S1: Spindeln (Seite 1383)".

## 6.6 Optimierung der Regelung

### 6.6.1 Lageregler Lagesollwertfilter: Symmetrierfilter

#### Anwendung

Für Drehzahl- und Drehmoment-Vorsteuerung.

Bei aktiver Vorsteuerung wird der Lagesollwert vor Erreichen des eigentlichen Reglers über ein sogenanntes Symmetrierfilter geschickt. Dadurch ist es möglich, den Drehzahlsollwert zu 100 % vorzusteuern, ohne dass beim Positionieren Überschwinger entstehen.

#### Filteraktivierung

Die Aktivierung des Filters und die Auswahl der Vorsteuerungsvariante erfolgen durch Änderung des axialen Maschinendatums:

MD32620 \$MA\_FFW\_MODE

Wert	Bedeutung
3	Drehzahl-Vorsteuerung
4	Kombinierte Momenten-/Drehzahl-Vorsteuerung

#### Aktivierung der Vorsteuerung

Die Aktivierung und Deaktivierung der Vorsteuerung für alle Achsen kann per Teileprogramm durch die Anweisungen `FFWON` und `FFWOF` vorgenommen werden, indem die Wirkungen des folgenden Maschinendatums unverändert bleiben:

MD32630 \$MA\_FFW\_ACTIVATION\_MODE

#### Steuerungsverhalten bei POWER ON, RESET, REPOS etc.

Bei POWER ON und RESET sowie bei "Maschinendaten wirksam setzen" werden die Einstelldaten der Vorsteuerung neu eingelesen (siehe hierzu die entsprechenden Angaben der jeweiligen Maschinendaten).

Betriebsartenwechsel, Satzsuchlauf, Repositionieren, haben **keine** Auswirkung auf die Vorsteuerung.

### Empfohlene Einstellung bei neuer Inbetriebnahme

Bei einer Neuinbetriebnahme oder wenn zuvor Standardwerte geladen sind (Schalterstellung 1 am Inbetriebnahmeschalter und POWER ON), gelten folgende Maschinendaten-Vorbelegungen:

MD32620 \$MA\_FFW\_MODE = 3

MD32610 \$MA\_VELO\_FFW\_WEIGHT = 1

Für die Drehzahlvorsteuerung ist dann nur noch die Symmetrierzeit im folgenden Maschinendatum anzupassen:

MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME

### Einstellung der Ersatzzeitkonstante des Drehzahlregelkreises

#### MD32810 Drehzahlvorsteuerung

Es empfiehlt sich, die Achse in der Betriebsart "AUTOMATIK" mit einem Teileprogramm hin- und herfahren zu lassen und das Einfahren in die Zielposition, d. h. den Lageistwert des aktiven Messsystems, mit dem Servo-Trace zu beobachten.

Man kann auch den Lageistwert auf die D/A-Wandler des Antriebsmoduls ausgeben und ein Oszilloskop zur Beobachtung verwenden.

Startwert für die Einstellung ist die Zeitkonstante des Drehzahlregelkreises. Diese kann aus dem Führungsfrequenzgang des Drehzahlregelkreises abgelesen werden. In dem häufig vorkommenden Fall PI-Regler mit Drehzahlsollwertglättung kann die Ersatzzeit näherungsweise aus den Antriebsmaschinendaten p1414, p1415, p1416 und p1421 abgelesen werden.

Eine weitere Möglichkeit besteht darin, Drehzahlsollwert und Drehzahlwert bei konstanter Beschleunigung mit einem Oszilloskop aufzuzeichnen und den zeitlichen Nachlauf des Drehzahlwerts auszumessen.

Man trägt nun diesen Startwert (z. B. 1,5 ms) ein:

MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME = 0.0015

Dann fährt man die Achse hin und her und beobachtet den Verlauf des Lage-Istwerts an der Zielposition in starker Vergrößerung.

Für den manuellen Feinabgleich gelten folgende Regeln:

Überschwinger beobachtet:	→ MD32810 vergrößern.
Zu langsame Annäherung beobachtet:	→ MD32810 verkleinern.



### **Vergrößern von MD32810**

Vergrößern des Werts von MD32810 bewirkt, dass die Achse etwas langsamer und der geometrische Konturfehler an Krümmungen etwas größer wird.

Es wirkt ähnlich wie ein Verkleinern der Lagereglerverstärkung:

MD32200 \$MA\_POSCTRL\_GAIN

Man kann dies auch im Bereich "Diagnose" im Bild "Service Achse" anhand des errechneten KV-Werts beobachten.

### **Verkleinern von MD32810**

Verkleinern des Werts von MD32810 lässt die Achse schneller werden.

Man versucht also, MD32810 so klein wie möglich auszulegen, wobei die Überschwinger beim Positionieren die Grenze setzen.

### **Feinabstimmung MD32810**

Erfahrungsgemäß verstellt man beim Feinabgleich den Anfangswert nur sehr wenig, typisch um 0,25 ms nach oben oder unten.

Bei einem Anfangswert von z. B. 1,5 ms liegt das manuell gefundene Optimum normalerweise im Bereich 1,25 ms bis 1,75 ms.

An Achsen mit direkten Messsystemen (Lastgeber) und ausgeprägter Elastizität kann es vorkommen, dass man kleine Überschwinger von einigen Mikrometern akzeptieren muss.

Diese lassen sich mit Hilfe der Lagesollwertfilter für Dynamik-Anpassung (MD32910 \$MA\_DYN\_MATCH\_TIME) und für Ruck (MD32410 \$MA\_AX\_JERK\_TIME) reduzieren, wodurch die Achse wieder etwas langsamer wird.

### **Gleiche Achsdaten eines Interpolationsverbunds**

Alle Achsen eines Interpolationsverbunds sollten in den folgenden Daten **gleich eingestellt sein**:

MD32200 \$MA\_POSCTRL\_GAIN (angepasst über MD32910)

MD32620 \$MA\_FFW\_MODE

MD32610 \$MA\_VELO\_FFW\_WEIGHT

MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME (oder MD32800 \$MA\_EQUIV\_CURRCTRL\_TIME) (abhängig von Mechanik und Antrieb)

MD32400 \$MA\_AX\_JERK\_ENABLE

MD32402 \$MA\_AX\_JERK\_MODE

MD32410 \$MA\_AX\_JERK\_TIME

Zur Kontrolle dient die K<sub>v</sub>-Anzeige im Service-Bild Achse.

### Ungleiche Achsdaten eines Interpolationsverbunds

Sind gleiche Werte in den o. g. Daten nicht möglich, so kann mit Hilfe des folgenden Maschinendatums eine Anpassung bewirkt werden:

MD32910 \$MA\_DYN\_MATCH\_TIME

Dadurch ist es möglich, den gleichen Kv-Wert anzuzeigen.

Normalerweise sind unterschiedliche Kv-Anzeige-Werte ein Indiz für folgende Fälle:

- In einer oder in mehreren Achsen passen die Getriebefaktoren nicht.
- Die Einstelldaten der Vorsteuerung stimmen nicht.

### Einstellung der Ersatzzeitkonstante des Stromregelkreises

#### MD32800 Ersatzzeitkonstante Stromregelkreis für Vorsteuerung

Für die Einstellung der Zeitkonstante des Stromregelkreises gelten dieselben Regeln und Empfehlungen wie bei der Drehzahlvorsteuerung.

Die Aktivierung des Filters für Drehmoment-Vorsteuerung mit:

MD32620 \$MA\_FFW\_MODE = 4

muss zur Einstellung der Zeitkonstante mit:

MD32800 \$MA\_EQUIV\_CURRCTRL\_TIME

aber wie bisher im Antrieb freigeschaltet werden.

#### Einschränkung auf steife Maschinen

Dieser Aufwand lohnt sich erfahrungsgemäß nur an sehr steifen Maschinen und setzt entsprechende Erfahrung voraus. Oft werden durch die Aufschaltung des Drehmoments die Elastizitäten der Maschine so stark angeregt, dass die entstehenden Schwingungen den Gewinn an Konturtreue neutralisieren.

Es lohnt sich in diesem Fall, alternativ den Einsatz von Dynamic Stiffness Control (DSC) zu erproben:

MD32640 \$MA\_STIFFNESS\_CONTROL\_ENABLE = 1

---

#### Hinweis

Soll DSC aktiviert werden (MD32640 = 1), darf in der NC keine Istwertinvertierung parametrisiert sein (MD32110 \$MA\_ENC\_FEEDBACK\_POL = -1). Es kommt sonst zum Fehler 26017.

Bei DSC-Betrieb darf eine Istwertinvertierung nur im Antrieb (SINAMICS-Parameter p0410) vorgenommen werden.

---

## 6.6.2 Lageregler Lagesollwertfilter: Ruckfilter

### Anwendung

In einigen Anwendungen, z. B. beim Fräsen von Freiformflächen, kann es vorteilhaft sein, die Lagesollwertverläufe zu glätten, um bessere Oberflächen aufgrund weniger Anregungen von Maschinenschwingungen zu erzeugen.

### Funktionalität

Die Filterwirkung der Lagesollwerte muss möglichst ausgeprägt sein, ohne aber die Konturgenauigkeit unzulässig zu beeinträchtigen.

Außerdem muss das Filter ein möglichst "symmetrisches" Glättungsverhalten haben, d. h. fährt man dieselbe Kontur vorwärts und rückwärts ab, sollte der vom Filter verrundete Verlauf in beiden Richtungen möglichst ähnlich sein.

Anhand der Anzeige des effektiven  $K_V$ -Faktors im Servicebild Achse kann die Wirkung des Filters beobachtet werden. Durch die Filterwirkung werden die Lagesollwerte geringfügig verschliffen und deshalb die Bahngenauigkeit etwas reduziert, weshalb mit wachsender Filterzeit ein kleinerer effektiver  $K_V$ -Faktor angezeigt wird.

### Filterfreigabe mit MD32402

Mit dem Maschinendatum:  
MD32400 \$MA\_AX\_JERK\_ENABLE  
wird das Lagesollwertfilter freigegeben  
und durch folgende Festlegungen bestimmt:

MD32402 \$MA_AX_JERK_MODE	= 2	Filtermodus gleitender Mittelwert
MD32410 \$MA_AX_JERK_TIME	= 0.02	Filterzeit in Sekunden einstellen (z. B. 20 ms)
MD32400 \$MA_AX_JERK_ENABLE	= 1	Filterberechnung freigegeben

Wenn vorher kein Filtermodus:  
MD32402 \$MA\_AX\_JERK\_MODE = 2  
aktiviert war, muss einmalig "Power On" ausgelöst werden.

Andernfalls reicht "Maschinendatum wirksam setzen" oder "Reset" an der Maschinensteuertafel.

---

### Hinweis

Generell wird das Filter mittels:  
MD32402 \$MA\_AX\_JERK\_MODE = 2  
eingestellt.

---

## Feineinstellung

Die Feineinstellung des Ruckfilters wird wie folgt vorgenommen:

1. Fahrverhalten der Achse beurteilen  
(z.B. anhand von Positioniervorgängen bei Servo-Trace).
2. Filterzeit in MD32410 \$MA\_AX\_JERK\_TIME ändern.
3. Geänderte Zeit mittels "Maschinendatum wirksam setzen" oder "Reset" an der Maschinensteuertafel aktivieren.

## Ausschalten

Ausschalten des Ruckfilters:

1. Filterberechnung sperren:  
MD32410 \$MA\_AX\_JERK\_ENABLE = 0
2. Sperre mittels "Maschinendatum wirksam setzen" oder "Reset" an der Maschinensteuertafel aktivieren.

## Randbedingungen

Das Ruckfilter ist in allen Steuerungsvarianten wie folgt verfügbar:

- Wirksame Filterzeiten sind von minimal 1 Lageregeltakt bis maximal 32 Lageregeltakte begrenzt (31 Lageregeltakte verfügbar).

Weitere Randbedingungen zur Filterwirkung:

- Die Anzeige des berechneten  $K_v$ -Faktors im Servicebild Achse zeigt kleinere Werte an als anhand der Filterwirkung angemessen wäre.
- Die Bahngenaugkeit ist besser, als der angezeigte  $K_v$  erwarten lässt.

Daher kann sich bei Umstellung von  
MD32400 \$MA\_AX\_JERK\_MODE = 1  
nach  
MD32400 \$MA\_AX\_JERK\_MODE = 2  
der angezeigte  $K_v$  bei gleicher Filterzeit verkleinern,  
obwohl die Bahngenaugkeit besser wird.

Miteinander interpolierende Achsen müssen gleich eingestellt sein.

Wenn man für solche Achsen jeweils ein Optimum gefunden hat, muss unter diesen Einstellungen diejenige mit der größten Filterzeit für alle Achsen des Interpolationsverbundes eingetragen werden.

Weiteren Informationen zur Ruckbegrenzung auf Interpolatorebene siehe Kapitel "Ruckbegrenzung bei Einzelachsinterpolation (SOFTA) (achsspezifisch) (Seite 266)" und "Achs-/Spindel-spezifische Maschinendaten (Seite 300)".

### 6.6.3 Lageregelung mit PI-Regler

#### Funktion

Im Standardfall ist der Kern des Lagereglers ein P-Regler.

Für besondere Einsätze (wie Elektronisches Getriebe) ist die Zuschaltung eines Integralteils möglich. Der dann vorliegende PI-Regler regelt den Fehler zwischen Soll- und Istposition bei entsprechender Einstellung der zugehörigen Maschinendaten zu Null in endlicher, einstellbarer Zeit aus.

#### VORSICHT

Bei aktiviertem PI-Regler tritt ein Überschwingen der Istposition auf. Der Anwender muss in diesem Fall prüfen, ob dies für die jeweilige Nutzung zulässig bzw. tolerierbar ist. Für die Nutzung der Funktion sind regeltechnische Fachkenntnisse und Messungen mit Servotrace unumgänglich. Bei falschen Einstellungen der entsprechenden Maschinendaten besteht durch das mögliche Auftreten von Instabilitäten Maschinenschadengefahr.

#### Vorgehen

1. Optimieren Sie den Lageregelkreis zunächst als P-Regler mit den in den vorangehenden Unterkapiteln beschriebenen Möglichkeiten.
2. Vergrößern Sie die Toleranzen folgender Maschinendaten für die Dauer der Messungen zur Feststellung der Qualität der Lageregelung mit PI-Lageregler:
  - MD36020 \$MA\_POSITIONING\_TIME
  - MD36030 \$MA\_STANDSTILL\_POS\_TOL
  - MD36040 \$MA\_STANDSTILL\_DELAY\_TIME
  - MD36400 \$MA\_CONTOUR\_TOL
3. Aktivieren Sie den Lageregelkreis als PI-Regler durch Einstellung folgender Maschinendaten:
  - MD32220 \$MA\_POSCTRL\_INTEGR\_ENABLE ; Wert 1 setzen
  - MD32210 \$MA\_POSCTRL\_INTEGR\_TIME ; Nachstellzeit [s]
 Wirkungsweise der Nachstellzeit:
  - $T_n \rightarrow 0$ :  
Der Regelfehler wird schnell ausgeregelt, der Regelkreis kann jedoch instabil werden.
  - $T_n \rightarrow \infty$ :  
Der Regelfehler wird langsamer ausgeregelt.
4. Suchen Sie zwischen diesen beiden Extremfällen den für die Anwendung richtigen Kompromiss für  $T_n$ .  
 $T_n$  darf nicht zu nahe der Instabilitätsgrenze gewählt werden, da bei Auftreten einer Instabilität Maschinenschadengefahr besteht.

5. Zeichnen Sie mit dem Servo-Trace das Einfahren eines Automatikprogramms zum Hin- und Herfahren in eine Zielposition auf.
6. Lassen Sie im Servo-Trace darstellen:
  - Schleppfehler
  - Istgeschwindigkeit
  - Istposition
  - Sollposition
7. Setzen Sie die Toleranzwerte in den folgenden Maschinendaten wieder auf die notwendigen Werte zurück, wenn für  $T_n$  der optimale Wert gefunden ist:
  - MD36020 \$MA\_POSITIONING\_TIME
  - MD36030 \$MA\_STANDSTILL\_POS\_TOL
  - MD36040 \$MA\_STANDSTILL\_DELAY\_TIME
  - MD36400 \$MA\_CONTOUR\_TOL

### Beispiel

Einstellerggebnis nach mehreren Iterationsschritten für  $K_R$  und  $T_n$ .

Es wurden Schleppabstand, Istgeschwindigkeit, Lageistwert und Lagesollwert jeweils per Servotrace aufgezeichnet. Beim Fahren im JOG-Betrieb wurde schließlich der im folgenden Bild dargestellte Verlauf der einzelnen Daten aufgezeichnet.

Eingestellte Maschinendaten:

MD32220 \$MA\_POSCTRL\_INTEGR\_ENABLE = 1

MD32210 \$MA\_POSCTRL\_INTEGR\_TIME = 0.003

MD32200 \$MA\_POSCTRL\_GAIN[1] = 5.0

Parametersatzanwahl 0

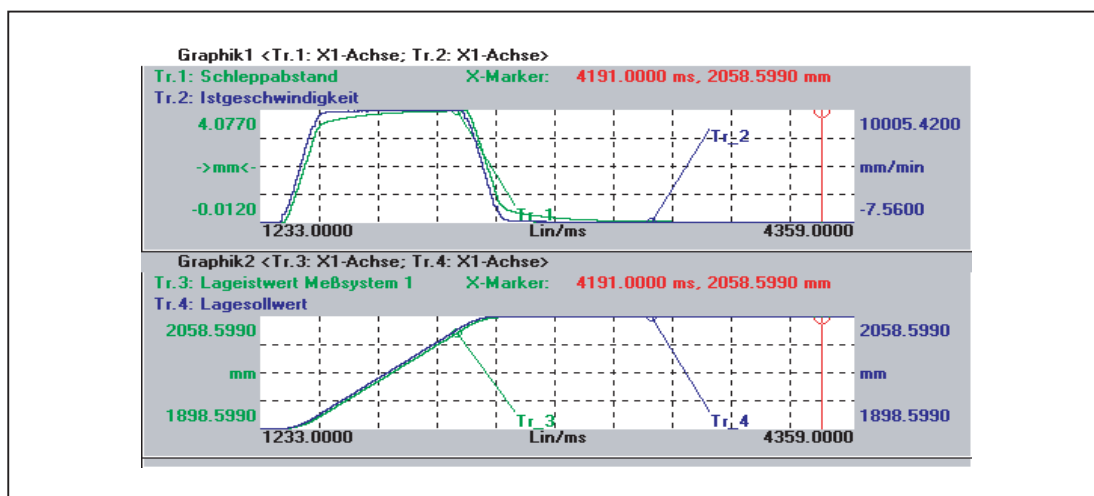


Bild 6-12 Schleppabstand (1), Istgeschwindigkeit (2), Lageistwert (3), Lagesollwert (4)

## 6.7 Datenlisten

### 6.7.1 Maschinendaten

#### 6.7.1.1 Anzeige-Maschinendaten

Nummer	Bezeichner: \$MM_	Beschreibung
9004	DISPLAY_RESOLUTION	Anzeigefeinheit
9010	SPIND_DISPLAY_RESOLUTION	Anzeigefeinheit für Spindeln
9011	DISPLAY_RESOLUTION_INCH	Anzeigefeinheit INCH-Maßsystem

#### 6.7.1.2 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10000	AXCONF_MACHAX_NAME_TAB	Maschinenachsname
10050	SYSCLOCK_CYCLE_TIME	Systemgrundtakt
10070	IPO_SYSCLOCK_TIME_RATIO	Faktor für Interpolortakt
10060	POSCTRL_SYSCLOCK_TIME_RATIO	Faktor für Lageregeltakt
10200	INT_INCR_PER_MM	Rechenfeinheit für Linearpositionen
10210	INT_INCR_PER_DEG	Rechenfeinheit für Winkelpositionen
10220	SCALING_USER_DEF_MASK	Aktivierung der Normierungsfaktoren
10230	SCALING_FACTORS_USER_DEF	Normierungsfaktoren der physikalischen Größen
10240	SCALING_SYSTEM_IS_METRIC	Grundsystem metrisch
10250	SCALING_VALUE_INCH	Umrechnungsfaktor für Umschaltung auf Inch-System
10260	CONVERT_SCALING_SYSTEM	Grundsystem Umschaltung aktiv
10270	POS_TAB_SCALING_SYSTEM	Maßsystem der Positionstabellen
10290	CC_TDA_PARAM_UNIT	Physikalische Einheiten der Werkzeugdaten für CC
10292	CC_TOA_PARAM_UNIT	Physikalische Einheiten der Schneidendaten für CC
13050	DRIVE_LOGIC_ADDRESS	Logische Antriebsadressen
13060	DRIVE_TELEGRAM_TYPE	Standard-Telegramm-Typ für PROFIBUS-DP
13070	DRIVE_FUNCTION_MASK	Benutzte DP-Funktion
13080	DRIVE_TYPE_DP	Antriebsart PROFIBUS-DP

#### 6.7.1.3 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen

## 6.7.1.4 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30110	CTRL_OUT_MODULE_NR	Sollwertzuordnung: Antriebsnummer
30120	CTRL_OUT_NR	Sollwertzuordnung: Sollwertausgang auf Antriebsmodul
30130	CTRL_OUT_TYPE	Ausgabeart des Sollwertes
30200	NUM_ENCS	Anzahl der Geber
30220	ENC_MODULE_NR	Istwertzuordnung: Antriebsmodulnr.
30230	ENC_INPUT_NR	Istwertzuordnung: Eingang auf Antriebsmodul
30240	ENC_TYPE	Art der Istwerterfassung (Lageistwert)
30242	ENC_IS_INDEPENDENT	Geber ist unabhängig
30300	IS_ROT_AX	Rundachse
31000	ENC_IS_LINEAR	Direktes Messsystem (Linearmaßstab)
31010	ENC_GRID_POINT_DIST	Teilungsperiode bei Linearmaßstäben
31020	ENC_RESOL	Geberstriche pro Umdrehung
31030	LEADSCREW_PITCH	Steigung der Kugelrollspindel
31040	ENC_IS_DIRECT	Geber ist direkt an der Maschine angebracht
31044	ENC_IS_DIRECT2	Geber am Vorsatz-Getriebe
31050	DRIVE_AX_RATIO_DENOM	Nenner Lastgetriebe
31060	DRIVE_AX_RATIO_NUMERA	Zähler Lastgetriebe
31064	DRIVE_AX_RATIO2_DENOM	Nenner Vorsatz-Getriebe
31066	DRIVE_AX_RATIO2_NUMERA	Zähler Vorsatz-Getriebe
31070	DRIVE_ENC_RATIO_DENOM	Nenner Messgetriebe
31080	DRIVE_ENC_RATIO_NUMERA	Zähler Messgetriebe
31090	JOG_INCR_WEIGHT	Bewertung des Inkrements bei INC/Handrad
31200	SCALING_FACTOR_G70_G71	Faktor für die Umrechnung der Werte bei aktivem G70/G71
32000	MAX_AX_VELO	Maximale Achsgeschwindigkeit
32100	AX_MOTION_DIR	Verfahrriichtung
32110	ENC_FEEDBACK_POL	Vorzeichen Istwert (Regelsinn)
32200	POSCTRL_GAIN	KV-Faktor
32210	POSCTRL_INTEGR_TIME	Nachstellzeit Lageregler
32220	POSCTRL_INTEGR_ENABLE	Aktivierung Integralanteil Lageregler
32250	RATED_OUTVAL	Nenn-Ausgangsspannung
32260	RATED_VELO	Nenn-Motordrehzahl
32450	BACKLASH	Umkehrlose
32500	FRICT_COMP_ENABLE	Reibkompensation aktiv
32610	VELO_FFWEIGHT	Vorsteuerfaktor für Drehzahlvorsteuerung
32620	FFW_MODE	Vorsteuerungsart
32630	FFW_ACTIVATION_MODE	Vorsteuerung aktivieren von Programm
32650	AX_INERTIA	Trägheitsmoment für Drehmomentvorsteuerung



Nummer	Bezeichner: \$MA_	Beschreibung
32652	AX_MASS	Achsmasse für Drehmomentvorsteuerung
32711	CEC_SCALING_SYSTEM_METRIC	Maßsystem der Durchhangkompensation
32800	EQUIV_CURRCTRL_TIME	Ersatzzeitkonstante Stromregelkreis für Vorsteuerung
32810	EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung
32900	DYN_MATCH_ENABLE	Dynamik-Anpassung
32910	DYN_MATCH_TIME[n]	Zeitkonstante der Dynamikanpassung
32930	POSCTRL_OUT_FILTER_ENABLE	Aktivierung des Tiefpassfilters am Legereglerausgang
33000	FIPO_TYPE	Feininterpolatortyp
34320	ENC_INVERS[n]	Längen-Messsystem ist gegensinnig
35100	SPIND_VELO_LIMIT	Maximale Spindeldrehzahl
36200	AX_VELO_LIMIT[n]	Schwellwert für Geschwindigkeitsüberwachung
36210	CTRL_OUT_LIMIT[n]	Maximaler Drehzahlsollwert
36400	AX_JERK_ENABLE	Axiale Ruckbegrenzung
36410	AX_JERK_TIME	Zeitkonstante für den axialen Ruckfilter
36500	ENC_CHANGE_TOL	Max. Toleranz bei Lageistwertumschaltung
36510	ENC_DIFF_TOL	Toleranz Messsystem-Gleichlauf
36700	ENC_COMP_ENABLE[n]	Interpolatorische Kompensation



## H2: Hilfsfunktionsausgaben an PLC

### 7.1 Kurzbeschreibung

#### 7.1.1 Funktion

Hilfsfunktionen bieten die Möglichkeit, Systemfunktionen des NCK und PLC-Anwenderfunktionen zu aktivieren. Hilfsfunktionen können programmiert werden in:

- Teileprogrammen
- Synchronaktionen
- Anwenderzyklen

Ausführliche Informationen zur Verwendung von Hilfsfunktionsausgaben in Synchronaktionen siehe:

**Literatur:**

Funktionshandbuch Synchronaktionen

#### Vordefinierte Hilfsfunktionen

Vordefinierte Hilfsfunktionen aktivieren Systemfunktionen. Zusätzlich wird die Hilfsfunktion an die NC/PLC-Nahtstelle ausgegeben.

Vordefiniert sind folgende Hilfsfunktionen:

Typ	Funktion	Beispiel	Bedeutung
M	Zusatzfunktion	M30	Programmende
S	Spindelfunktion	S100	Spindeldrehzahl 100 (z. B. 1/min)
T	Werkzeugnummer	T3	Werkzeugnummer 3
D, DL	Werkzeugkorrektur	D1	Werkzeugschneidenummer 1
F	Vorschub	F1000	Vorschub 1000 (z. B. mm/min)

#### Anwenderdefinierte Hilfsfunktionen

Anwenderdefinierte Hilfsfunktionen sind entweder erweiterte vordefinierte Hilfsfunktionen oder anwenderspezifische Hilfsfunktionen.

**Erweiterung von vordefinierten Hilfsfunktionen**

Die Erweiterung von vordefinierten Hilfsfunktionen bezieht sich auf den Parameter "Adresserweiterung". Über die Adresserweiterung wird die Nummer der Spindel definiert, auf die sich die Hilfsfunktion bezieht. Vordefiniert ist z. B. die Spindelfunktion M3 (Spindel rechts) für die Masterspindel eines Kanals. Ist einem Kanal eine 2. Spindel zugeordnet, ist eine entsprechende anwenderdefinierte Hilfsfunktionen zu definieren, welche die vordefinierte Hilfsfunktion erweitert.

Typ	Funktion	Beispiel	Bedeutung
M	Zusatzfunktion	M2=3	2. Spindel: Spindel rechts
S	Spindelfunktion	S2=100	2. Spindel: Spindeldrehzahl = 100 (z. B. 1/min)
T	Werkzeugnummer	T2=3	

**Anwenderspezifische Hilfsfunktionen**

Über anwenderspezifische Hilfsfunktionen werden keine Systemfunktionen aktiviert. Anwenderspezifische Hilfsfunktionen werden lediglich an die NC/PLC-Nahtstelle ausgegeben. Die Funktionalität der Hilfsfunktion ist vom Maschinenhersteller/Anwender im PLC-Anwenderprogramm zu realisieren.

Typ	Funktion	Beispiel	Bedeutung
H <sup>1)</sup>	Hilfsfunktion	H2=5	Anwenderspezifische Funktion

1) Empfehlung

**7.1.2 Definition einer Hilfsfunktion**

Eine Hilfsfunktion ist über folgende Parameter definiert:

- **Typ, Adresserweiterung und Wert**

Die 3 Parameter werden an die NC/PLC-Nahtstelle ausgegeben.

- **Ausgabeverhalten**

Über das Hilfsfunktionsspezifische Ausgabeverhalten wird festgelegt, wie lange eine Hilfsfunktion an die NC/PLC-Nahtstelle ausgegeben wird und wann die Ausgabe, bezogen auf die im gleichen Teileprogrammsatz programmierte Verfahrbewegung, erfolgt.

- **Gruppenzuordnung**

Eine Hilfsfunktion kann einer bestimmten Hilfsfunktionsgruppe zugeordnet werden. Für jede Hilfsfunktionsgruppe kann ein eigenes Ausgabeverhalten festgelegt werden. Dieses wird wirksam, wenn kein Hilfsfunktionsspezifisches Ausgabeverhalten festgelegt wurde. Darüber hinaus beeinflusst die Gruppenzugehörigkeit die Ausgabe einer Hilfsfunktion nach Satzsuchlauf.

Ausführliche Informationen zur Hilfsfunktionsausgabe an die NC/PLC-Nahtstelle siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)".

### 7.1.3 Übersicht der Hilfsfunktionen

#### M-Funktionen

M (Zusatzfunktion)					
Adresserweiterung			Wert		
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>8)</sup></b>
0 (implizit)	- - -	0 ... 99	INT	Funktion	5
<b>Anmerkungen:</b> Für den Wertebereich zwischen 0 und 99 ist die Adresserweiterung 0. Zwingend ohne Adresserweiterung: M0, M1, M2, M17, M30					
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>8)</sup></b>
1 ... 20	Spindelnummer	1 ... 99	INT	Funktion	5
<b>Anmerkungen:</b> M3, M4, M5, M19, M70 mit Adresserweiterung als Spindel-Nummer. (z. B. M2=5; Spindel-Halt für Spindel 2). Ohne Adresserweiterung wirkt die Funktion auf die Masterspindel.					
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>8)</sup></b>
0 ... 99	beliebig	100 ... 2147483647	INT	Funktion	5
<b>Anmerkungen:</b> Anwenderspezifische M-Funktionen.					

<sup>8)</sup> Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

#### Verwendung

Maschinenfunktionen synchron zum Teileprogramm steuern.

#### Weitere Informationen

- Folgende M-Funktionen haben vordefinierte Bedeutung: M0, M1, M2, M17, M30  
M3, M4, M5, M6, M19, M70, M40, M41, M42, M43, M44, M45.
- Zu jeder M-Funktion (M0 - M99) gehört an der NC/PLC-Nahtstelle ein dynamisches Signal, das die Gültigkeit (neue Ausgabe) der M-Funktion anzeigt. Zusätzlich können 64 weitere Signale für eigene M-Funktionen zugeordnet werden (siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)").
- Für Unterprogramme ist über Maschinendatum einstellbar, ob eine Ausgabe der M-Funktion für das Teileprogrammende M17, M2 und M30 an die PLC erfolgen soll:  
MD20800 \$MC\_SPF\_END\_TO\_VDI (Unterprogrammende an PLC)
- Für die vordefinierten M-Funktionen M40 – M45 kann die Ausgabespezifikation nur eingeschränkt umprojektiert werden.

7.1 Kurzbeschreibung

- Die vordefinierten Hilfsfunktionen M0, M1, M17, M30, M6, M4, M5 lassen sich nicht umprojektieren.
- M-Funktionsspezifische Maschinendaten:
  - MD10800 \$MN\_EXTERN\_CHAN\_SYNC\_M\_NO\_MIN
  - MD10802 \$MN\_EXTERN\_CHAN\_SYNC\_M\_NO\_MAX
  - MD10804 \$MN\_EXTERN\_M\_NO\_SET\_INT
  - MD10806 \$MN\_EXTERN\_M\_NO\_DISABLE\_INT
  - MD10814 \$MN\_EXTERN\_M\_NO\_MAC\_CYCLE
  - MD10815 \$MN\_EXTERN\_M\_NO\_MAC\_CYCLE\_NAME
  - MD20094 \$MC\_SPIND\_RIGID\_TAPPING\_M\_NR
  - MD20095 \$MC\_EXTERN\_RIGID\_TAPPING\_M\_NR
  - MD20096 \$MC\_T\_M\_ADDRESS\_EXT\_IS\_SPINO
  - MD22200 \$MC\_AUXFU\_M\_SYNC\_TYPE
  - MD22530 \$MC\_TOCARR\_CHANGE\_M\_CODE
  - MD22532 \$MC\_GEOAX\_CHANGE\_M\_CODE
  - MD22534 \$MC\_TRAFO\_CHANGE\_M\_CODE
  - MD22560 \$MC\_TOOL\_CHANGE\_M\_CODE

S-Funktionen

S (Spindelfunktion)					
Adresserweiterung			Wert		
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>3)</sup></b>
0 ... 20	Spindelnummer <sup>5)</sup>	0 ... ± 3.4028 exp38 <sup>3)</sup>	REAL	Spindeldrehzahl	3
<b>Anmerkungen:</b>					
Ohne Angabe einer Adresserweiterung wird die Master-Spindel des Kanals adressiert.					

<sup>3)</sup> , <sup>5)</sup>, <sup>8)</sup>Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

**Verwendung**

Spindeldrehzahl.

**Weitere Informationen**

- Die S-Funktionen sind standardmäßig der 3. Hilfsfunktionsgruppe zugeordnet.
- Ohne Adresserweiterung beziehen sich die S-Funktionen auf die Masterspindel des Kanals.
- S-Funktionsspezifische Maschinendaten:
  - MD22210 \$MC\_AUXFU\_S\_SYNC\_TYPE (Ausgabezeitpunkt der S-Funktionen)

## H-Funktionen

H (Hilfsfunktion)					
Adresserweiterung			Wert		
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>8)</sup></b>
0 ... 99	beliebig	- 2147483648 ... + 2147483647	INT	beliebig	3
		0 ... ± 3.4028 exp38 <sup>2) 3) 4)</sup>	REAL		
<b>Anmerkungen:</b> Die Funktionalität ist durch den Anwender im PLC-Anwenderprogramm zu realisieren.					

<sup>2) 3) 4) 8)</sup> Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

**Verwendung**

Anwenderspezifische Hilfsfunktionen.

**Weitere Informationen**

- H-Funktionspezifische Maschinendaten:
  - MD22110 \$MC\_AUXFU\_H\_TYPE\_INT (Typ von H-Hilfsfunktionen ist Integer)
  - MD22230 \$MC\_AUXFU\_H\_SYNC\_TYPE (Ausgabezeitpunkt der H-Funktionen)

## T-Funktionen

T (Werkzeugnummer) <sup>5) 6)</sup>					
Adresserweiterung			Wert		
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>8)</sup></b>
1 ... 12	Spindelnummer (bei aktiver Werkzeug- verwaltung)	0 ... 32000 (auch symbolische Werkzeugnamen bei aktiver Werkzeugverwaltung)	INT	Anwahl des Werkzeugs	1
<b>Anmerkungen:</b> Werkzeugnamen werden nicht an die PLC ausgegeben. <sup>1)</sup>					

<sup>1) 5) 6) 8)</sup> Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

**Verwendung**

Werkzeuganwahl.

**Weitere Informationen**

- Identifikation der Werkzeuge, wahlweise über Werkzeugnummer oder Platznummer (siehe Kapitel "W1: Werkzeugkorrektur (Seite 1567)").

**Literatur:**

Funktionshandbuch Werkzeugverwaltung

- Mit T0 wird das aktuelle Werkzeug aus der Werkzeughalterung entfernt und kein neues eingewechselt (Grundeinstellung).
- T-Funktionsspezifische Maschinendaten:  
MD22220 \$MC\_AUXFU\_T\_SYNC\_TYPE (Ausgabezeitpunkt der T-Funktionen)

**D-Funktionen**

D (Werkzeugkorrektur)					
Adresserweiterung			Wert		
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>8)</sup></b>
---	---	0 ... 9	INT	Anwahl der Werkzeugkorrektur	1
<b>Anmerkungen:</b> Abwahl der Werkzeugkorrektur mit D0. Vorbesetzung ist D1.					

<sup>8)</sup> Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

**Verwendung**

Anwahl der Werkzeugkorrektur.

**Weitere Informationen**

- Grundstellung: D1
- Die Default-Werkzeugschneide nach Werkzeugwechsel ist parametrierbar über:  
MD20270 \$MC\_CUTTING\_EDGE\_DEFAULT (Grundstellung der Werkzeugschneide ohne Programmierung)
- Abwahl der Werkzeugkorrektur: D0
- D-Funktionsspezifische Maschinendaten:  
MD22250 \$MC\_AUXFU\_D\_SYNC\_TYPE (Ausgabezeitpunkt der D-Funktionen)



## DL-Funktionen

DL (Werkzeugsummenkorrektur)					
Adresserweiterung			Wert		
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl <sup>8)</sup>
---	---	0 ... 6	INT	Anwahl der Werkzeugsummenkorrektur	1
<b>Anmerkungen:</b> Die mit DL angewählte Werkzeugsummenkorrektur bezieht sich auf aktive D-Nummer.					

<sup>8)</sup> Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

**Verwendung**

Anwahl der Werkzeugsummenkorrektur bezüglich einer aktiven Werkzeugkorrektur.

**Weitere Informationen**

- Grundstellung: DL = 0
- DL-Werte können nicht über Synchronaktionen an die PLC ausgegeben werden.
- Voreinstellung der Werkzeugsummenkorrektur ohne aktive DL-Funktion:  
MD20272 \$MC\_SUMCORR\_DEFAULT (Grundstellung Summenkorrektur ohne Programm)
- Abwahl der Werkzeugsummenkorrektur: DL = 0
- DL-Funktionsspezifische Maschinendaten:  
MD22252 \$MC\_AUXFU\_DL\_SYNC\_TYPE (Ausgabezeitpunkt DL-Funktionen)

## F-Funktionen

F (Bahnvorschub)					
Adresserweiterung			Wert		
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl <sup>8)</sup>
---	---	0.001 ... 999 999.999	REAL	Bahnvorschub	6
<b>Anmerkungen:</b> ---					

<sup>8)</sup> Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

**Verwendung**

Bahngeschwindigkeit.

**Weitere Informationen**

- F-Funktionsspezifische Maschinendaten:  
MD22240 \$MC\_AUXFU\_F\_SYNC\_TYPE (Ausgabezeitpunkt der F-Funktionen)

**FA-Funktionen**

FA (axialer Vorschub)					
Adresserweiterung			Wert		
<b>Wertebereich</b>	<b>Bedeutung</b>	<b>Wertebereich</b>	<b>Typ</b>	<b>Bedeutung</b>	<b>Anzahl <sup>8)</sup></b>
1 - 31	Achsnummer	0.001 ... 999 999.999	REAL	Achsvorschub	6
<b>Anmerkungen:</b>					
- - -					

<sup>8)</sup> Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

**Verwendung**

Axiale Geschwindigkeit.

**Weitere Informationen**

- F-Funktionsspezifische Maschinendaten:  
MD22240 \$MC\_AUXFU\_F\_SYNC\_TYPE (Ausgabezeitpunkt der F-Funktionen)

## Bedeutung der Fußnoten

- 1) Bei aktiver Werkzeugverwaltung wird weder ein T-Änderungssignal noch ein T-Wort auf die Nahtstelle (Kanal) ausgegeben.
- 2) Der Typ für die Werte kann durch MD22110 \$MC\_AUXFU\_H\_TYPE\_INT vom Anwender gewählt werden.
- 3) Aufgrund der begrenzten Anzeigemöglichkeiten auf den Bildschirmen der Bediengeräte sind die angezeigten Werte des Typs REAL begrenzt auf:  
–999 999 999.9999 bis 999 999 999.9999  
Die NC rechnet intern aber mit der vollen Genauigkeit.
- 4) Die REAL-Werte werden gerundet an die PLC ausgegeben, bei der Einstellung des Maschinendatums:  
MD22110 \$MC\_AUXFU\_H\_TYPE\_INT = 1 (Typ von H-Hilfsfunktionen ist Integer)  
Das PLC-Anwenderprogramm muss den übergebenen Wert entsprechend der Maschinendaten-Einstellung interpretieren.
- 5) Bei aktiver Werkzeugverwaltung kann die Bedeutung der Adresserweiterung parametrisiert werden. Adresserweiterung = 0 bedeutet, der Wert ist durch den der Masterspindelnummer zu ersetzen; ist somit gleichbedeutend mit Nichtprogrammierung der Adresserweiterung.  
Bei Satzsuchlauf aufgesammelte Hilfsfunktionen M19 "Spindel positionieren" werden nicht an die PLC ausgegeben.
- 6) M6: Wertebereich der Adresserweiterung:  
- ohne Werkzeugverwaltung: 0 ... 99  
- mit Werkzeugverwaltung: 0 ... maximale Spindelnummer  
0: ersetzen durch Wert der Masterspindelnummer bzw. des Masterwerkzeughalters
- 7) Mit aktiver Werkzeugverwaltung kann die Hilfsfunktion M6 "Werkzeugwechsel" nur einmal in einem Teileprogrammabsatz programmiert werden, unabhängig davon, welche Adresserweiterungen programmiert werden.
- 8) Maximale Anzahl von Hilfsfunktionen pro Teileprogrammabsatz.

## 7.2 Vordefinierte Hilfsfunktionen

### Funktion

Jeder vordefinierten Hilfsfunktion ist nicht änderbar eine Systemfunktion zugeordnet. Ist in einem Teileprogramm / Zyklus eine vordefinierte Hilfsfunktion programmiert, wird diese über die NC/PLC-Nahtstelle an die PLC ausgegeben und im NCK die entsprechende Systemfunktion ausgeführt.

### Definition einer vordefinierten Hilfsfunktion

Die Parameter der vordefinierten Hilfsfunktionen sind in Maschinendaten hinterlegt und können teilweise geändert werden. Alle Maschinendaten die zu einer Hilfsfunktion gehören, haben den gleichen Index <n>.

- MD22040 \$MC\_AUXFU\_PREDEF\_GROUP[<n>] (Gruppenzuordnung von vordefinierten Hilfsfunktionen)
- MD22050 \$MC\_AUXFU\_PREDEF\_TYPE[<n>] (Typ von vordefinierten Hilfsfunktionen)
- MD22060 \$MC\_AUXFU\_PREDEF\_EXTENSION[<n>] (Adresserweiterung für vordefinierte Hilfsfunktionen)
- MD22070 \$MC\_AUXFU\_PREDEF\_VALUE[<n>] (Wert von vordefinierten Hilfsfunktionen)
- MD22080 \$MC\_AUXFU\_PREDEF\_SPEC[<n>] (Ausgabeverhalten von vordefinierten Hilfsfunktionen)

### 7.2.1 Übersicht: vordefinierten Hilfsfunktionen

Bedeutung der in den nachfolgenden Tabellen aufgeführten Parameter:

Parameter	Bedeutung
Index <n>	Maschinendatenindex der Parameter einer Hilfsfunktion
Typ	MD22050 \$MC_AUXFU_PREDEF_TYPE[<n>]
Adresserweiterung	MD22060 \$MC_AUXFU_PREDEF_EXTENSION[<n>]
Wert	MD22070 \$MC_AUXFU_PREDEF_VALUE[<n>]
Gruppe	MD22040 \$MC_AUXFU_PREDEF_GROUP[<n>]

## Vordefinierte Hilfsfunktionen

Allgemeine Hilfsfunktionen, Teil 1					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Halt	0	M	0	0	1
bedingter Halt	1	M	0	1	1
Unterprogramm Ende	2	M	0	2	1
	3	M	0	17	1
	4	M	0	30	1
Werkzeugwechsel	5	M	(0)	6 <sup>1)</sup>	(1)

Spindel-spezifische Hilfsfunktionen, Spindel 1					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	6	M	1	3	(2)
Spindel links	7	M	1	4	(2)
Spindel halt	8	M	1	5	(2)
Spindel positionieren	9	M	1	19	(2)
Achsbetrieb	10	M	1	70 <sup>2)</sup>	(2)
automatische Getriebestufe	11	M	1	40	(4)
Getriebestufe 1	12	M	1	41	(4)
Getriebestufe 2	13	M	1	42	(4)
Getriebestufe 3	14	M	1	43	(4)
Getriebestufe 4	15	M	1	44	(4)
Getriebestufe 5	16	M	1	45	(4)
Spindel-Drehzahl	17	S	1	-1	(3)

Allgemeine Hilfsfunktionen, Teil 2					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Vorschub	18	F	0	-1	(1)
Schneidenwahl	19	D	0	-1	(1)
DL	20	L	0	-1	(1)
Werkzeugwahl	21	T	(0)	-1	(1)
Halt (assoziiert)	22	M	0	-1 <sup>3)</sup>	1
bedingter Halt (assoziiert)	23	M	0	-1 <sup>4)</sup>	1
Unterprogramm Ende	24	M	0	-1 <sup>5)</sup>	1
Nibbeln	25	M	0	20 <sup>6)</sup>	(10)
Nibbeln	26	M	0	23 <sup>6)</sup>	(10)
Nibbeln	27	M	0	22 <sup>6)</sup>	(11)
Nibbeln	28	M	0	25 <sup>6)</sup>	(11)
Nibbeln	29	M	0	26 <sup>6)</sup>	(12)
Nibbeln	30	M	0	122 <sup>6)</sup>	(11)
Nibbeln	31	M	0	125 <sup>6)</sup>	(11)
Nibbeln	32	M	0	27 <sup>6)</sup>	(12)

Spindel-spezifische Hilfsfunktionen, Spindel 2					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	33	M	2	3	(72)
Spindel links	34	M	2	4	(72)
Spindel halt	35	M	2	5	(72)
Spindel positionieren	36	M	2	19	(72)
Achsbetrieb	37	M	2	70 <sup>2)</sup>	(72)
automatische Getriebestufe	38	M	2	40	(74)
Getriebestufe 1	39	M	2	41	(74)
Getriebestufe 2	40	M	2	42	(74)
Getriebestufe 3	41	M	2	43	(74)
Getriebestufe 4	42	M	2	44	(74)
Getriebestufe 5	43	M	2	45	(74)
Spindel-Drehzahl	44	S	2	-1	(73)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 3</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	45	M	3	3	(75)
Spindel links	46	M	3	4	(75)
Spindel halt	47	M	3	5	(75)
Spindel positionieren	48	M	3	19	(75)
Achsbetrieb	49	M	3	70 <sup>2)</sup>	(75)
automatische Getriebestufe	50	M	3	40	(77)
Getriebestufe 1	51	M	3	41	(77)
Getriebestufe 2	52	M	3	42	(77)
Getriebestufe 3	53	M	3	43	(77)
Getriebestufe 4	54	M	3	44	(77)
Getriebestufe 5	55	M	3	45	(77)
Spindel-Drehzahl	56	S	3	-1	(76)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 4</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	57	M	4	3	(78)
Spindel links	58	M	4	4	(78)
Spindel halt	59	M	4	5	(78)
Spindel positionieren	60	M	4	19	(78)
Achsbetrieb	61	M	4	70 <sup>2)</sup>	(78)
automatische Getriebestufe	62	M	4	40	(80)
Getriebestufe 1	63	M	4	41	(80)
Getriebestufe 2	64	M	4	42	(80)
Getriebestufe 3	65	M	4	43	(80)
Getriebestufe 4	66	M	4	44	(80)
Getriebestufe 5	67	M	4	45	(80)
Spindel-Drehzahl	68	S	4	-1	(79)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 5</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	69	M	5	3	(81)
Spindel links	70	M	5	4	(81)
Spindel halt	71	M	5	5	(81)
Spindel positionieren	72	M	5	19	(81)
Achsbetrieb	73	M	5	70 <sup>2)</sup>	(81)
automatische Getriebestufe	74	M	5	40	(83)
Getriebestufe 1	75	M	5	41	(83)
Getriebestufe 2	76	M	5	42	(83)
Getriebestufe 3	77	M	5	43	(83)
Getriebestufe 4	78	M	5	44	(83)
Getriebestufe 5	79	M	5	45	(83)
Spindel-Drehzahl	80	S	5	-1	(82)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 6</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	81	M	6	3	(84)
Spindel links	82	M	6	4	(84)
Spindel halt	83	M	6	5	(84)
Spindel positionieren	84	M	6	19	(84)
Achsbetrieb	85	M	6	70 <sup>2)</sup>	(84)
automatische Getriebestufe	86	M	6	40	(86)
Getriebestufe 1	87	M	6	41	(86)
Getriebestufe 2	88	M	6	42	(86)
Getriebestufe 3	89	M	6	43	(86)
Getriebestufe 4	90	M	6	44	(86)
Getriebestufe 5	91	M	6	45	(86)
Spindel-Drehzahl	92	S	6	-1	(85)



<b>Spindel-spezifische Hilfsfunktionen, Spindel 7</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	93	M	7	3	(87)
Spindel links	94	M	7	4	(87)
Spindel halt	95	M	7	5	(87)
Spindel positionieren	96	M	7	19	(87)
Achsbetrieb	97	M	7	70 <sup>2)</sup>	(87)
automatische Getriebestufe	98	M	7	40	(89)
Getriebestufe 1	99	M	7	41	(89)
Getriebestufe 2	100	M	7	42	(89)
Getriebestufe 3	101	M	7	43	(89)
Getriebestufe 4	102	M	7	44	(89)
Getriebestufe 5	103	M	7	45	(89)
Spindel-Drehzahl	104	S	7	-1	(88)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 8</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	105	M	8	3	(90)
Spindel links	106	M	8	4	(90)
Spindel halt	107	M	8	5	(90)
Spindel positionieren	108	M	8	19	(90)
Achsbetrieb	109	M	8	70 <sup>2)</sup>	(90)
automatische Getriebestufe	110	M	8	40	(92)
Getriebestufe 1	111	M	8	41	(92)
Getriebestufe 2	112	M	8	42	(92)
Getriebestufe 3	113	M	8	43	(92)
Getriebestufe 4	114	M	8	44	(92)
Getriebestufe 5	115	M	8	45	(92)
Spindel-Drehzahl	116	S	8	-1	(91)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 9</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	117	M	9	3	(93)
Spindel links	118	M	9	4	(93)
Spindel halt	119	M	9	5	(93)
Spindel positionieren	120	M	9	19	(93)
Achsbetrieb	121	M	9	70 <sup>2)</sup>	(93)
automatische Getriebestufe	122	M	9	40	(95)
Getriebestufe 1	123	M	9	41	(95)
Getriebestufe 2	124	M	9	42	(95)
Getriebestufe 3	125	M	9	43	(95)
Getriebestufe 4	126	M	9	44	(95)
Getriebestufe 5	127	M	9	45	(95)
Spindel-Drehzahl	128	S	9	-1	(94)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 10</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	129	M	10	3	(96)
Spindel links	130	M	10	4	(96)
Spindel halt	131	M	10	5	(96)
Spindel positionieren	132	M	10	19	(96)
Achsbetrieb	133	M	10	70 <sup>2)</sup>	(96)
automatische Getriebestufe	134	M	10	40	(98)
Getriebestufe 1	135	M	10	41	(98)
Getriebestufe 2	136	M	10	42	(98)
Getriebestufe 3	137	M	10	43	(98)
Getriebestufe 4	138	M	10	44	(98)
Getriebestufe 5	139	M	10	45	(98)
Spindel-Drehzahl	140	S	10	-1	(97)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 11</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	141	M	11	3	(99)
Spindel links	142	M	11	4	(99)
Spindel halt	143	M	11	5	(99)
Spindel positionieren	144	M	11	19	(99)
Achsbetrieb	145	M	11	70 <sup>2)</sup>	(99)
automatische Getriebestufe	146	M	11	40	(101)
Getriebestufe 1	147	M	11	41	(101)
Getriebestufe 2	148	M	11	42	(101)
Getriebestufe 3	149	M	11	43	(101)
Getriebestufe 4	150	M	11	44	(101)
Getriebestufe 5	151	M	11	45	(101)
Spindel-Drehzahl	152	S	11	-1	(100)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 12</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	153	M	11	3	(102)
Spindel links	154	M	12	4	(102)
Spindel halt	155	M	12	5	(102)
Spindel positionieren	156	M	12	19	(102)
Achsbetrieb	157	M	12	70 <sup>2)</sup>	(102)
automatische Getriebestufe	158	M	12	40	(104)
Getriebestufe 1	159	M	12	41	(104)
Getriebestufe 2	160	M	12	42	(104)
Getriebestufe 3	161	M	12	43	(104)
Getriebestufe 4	162	M	12	44	(104)
Getriebestufe 5	163	M	12	45	(104)
Spindel-Drehzahl	164	S	12	-1	(103)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 13</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	165	M	13	3	(105)
Spindel links	166	M	13	4	(105)
Spindel halt	167	M	13	5	(105)
Spindel positionieren	168	M	13	19	(105)
Achsbetrieb	169	M	13	70 <sup>2)</sup>	(105)
automatische Getriebestufe	170	M	13	40	(107)
Getriebestufe 1	171	M	13	41	(107)
Getriebestufe 2	172	M	13	42	(107)
Getriebestufe 3	173	M	13	43	(107)
Getriebestufe 4	174	M	13	44	(107)
Getriebestufe 5	175	M	13	45	(107)
Spindel-Drehzahl	176	S	13	-1	(106)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 14</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	177	M	14	3	(108)
Spindel links	178	M	14	4	(108)
Spindel halt	179	M	14	5	(108)
Spindel positionieren	180	M	14	19	(108)
Achsbetrieb	181	M	14	70 <sup>2)</sup>	(108)
automatische Getriebestufe	182	M	14	40	(110)
Getriebestufe 1	183	M	14	41	(110)
Getriebestufe 2	184	M	14	42	(110)
Getriebestufe 3	185	M	14	43	(110)
Getriebestufe 4	186	M	14	44	(110)
Getriebestufe 5	187	M	14	45	(110)
Spindel-Drehzahl	188	S	14	-1	(109)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 15</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	189	M	15	3	(111)
Spindel links	190	M	15	4	(111)
Spindel halt	191	M	15	5	(111)
Spindel positionieren	192	M	15	19	(111)
Achsbetrieb	193	M	15	70 <sup>2)</sup>	(111)
automatische Getriebestufe	194	M	15	40	(113)
Getriebestufe 1	195	M	15	41	(113)
Getriebestufe 2	196	M	15	42	(113)
Getriebestufe 3	197	M	15	43	(113)
Getriebestufe 4	198	M	15	44	(113)
Getriebestufe 5	199	M	15	45	(113)
Spindel-Drehzahl	200	S	15	-1	(112)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 16</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	201	M	16	3	(114)
Spindel links	202	M	16	4	(114)
Spindel halt	203	M	16	5	(114)
Spindel positionieren	204	M	16	19	(114)
Achsbetrieb	205	M	16	70 <sup>2)</sup>	(114)
automatische Getriebestufe	206	M	16	40	(116)
Getriebestufe 1	207	M	16	41	(116)
Getriebestufe 2	208	M	16	42	(116)
Getriebestufe 3	209	M	16	43	(116)
Getriebestufe 4	210	M	16	44	(116)
Getriebestufe 5	211	M	16	45	(116)
Spindel-Drehzahl	212	S	16	-1	(115)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 17</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	213	M	17	3	(117)
Spindel links	214	M	17	4	(117)
Spindel halt	215	M	17	5	(117)
Spindel positionieren	216	M	17	19	(117)
Achsbetrieb	217	M	17	70 <sup>2)</sup>	(117)
automatische Getriebestufe	218	M	17	40	(119)
Getriebestufe 1	219	M	17	41	(119)
Getriebestufe 2	220	M	17	42	(119)
Getriebestufe 3	221	M	17	43	(119)
Getriebestufe 4	222	M	17	44	(119)
Getriebestufe 5	223	M	17	45	(119)
Spindel-Drehzahl	224	S	17	-1	(118)

<b>Spindel-spezifische Hilfsfunktionen, Spindel 18</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Spindel rechts	225	M	18	3	(120)
Spindel links	226	M	18	4	(120)
Spindel halt	227	M	18	5	(120)
Spindel positionieren	228	M	18	19	(120)
Achsbetrieb	229	M	18	70 <sup>2)</sup>	(120)
automatische Getriebestufe	230	M	18	40	(122)
Getriebestufe 1	231	M	18	41	(122)
Getriebestufe 2	232	M	18	42	(122)
Getriebestufe 3	233	M	18	43	(122)
Getriebestufe 4	234	M	18	44	(122)
Getriebestufe 5	235	M	18	45	(122)
Spindel-Drehzahl	236	S	18	-1	(121)

Spindel-spezifische Hilfsfunktionen, Spindel 19					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	237	M	19	3	(123)
Spindel links	238	M	19	4	(123)
Spindel halt	239	M	19	5	(123)
Spindel positionieren	240	M	19	19	(123)
Achsbetrieb	241	M	19	70 <sup>2)</sup>	(123)
automatische Getriebestufe	242	M	19	40	(125)
Getriebestufe 1	243	M	19	41	(125)
Getriebestufe 2	244	M	19	42	(125)
Getriebestufe 3	245	M	19	43	(125)
Getriebestufe 4	246	M	19	44	(125)
Getriebestufe 5	247	M	19	45	(125)
Spindel-Drehzahl	248	S	19	-1	(124)

Spindel-spezifische Hilfsfunktionen, Spindel 20					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	249	M	20	3	(126)
Spindel links	250	M	20	4	(126)
Spindel halt	251	M	20	5	(126)
Spindel positionieren	252	M	20	19	(126)
Achsbetrieb	253	M	20	70 <sup>2)</sup>	(126)
automatische Getriebestufe	254	M	20	40	(128)
Getriebestufe 1	255	M	20	41	(128)
Getriebestufe 2	256	M	20	42	(128)
Getriebestufe 3	257	M	20	43	(128)
Getriebestufe 4	258	M	20	44	(128)
Getriebestufe 5	259	M	20	45	(128)
Spindel-Drehzahl	260	S	20	-1	(127)

<b>Werkzeughalter-spezifische Hilfsfunktionen, T-Hilfsfunktionen</b>					
<b>Systemfunktion</b>	<b>Index &lt;n&gt;</b>	<b>Typ</b>	<b>Adresserw.</b>	<b>Wert</b>	<b>Gruppe</b>
Werkzeugwahl	261	T	1	-1	129
Werkzeugwahl	262	T	2	-1	130
Werkzeugwahl	263	T	3	-1	131
Werkzeugwahl	264	T	4	-1	132
Werkzeugwahl	265	T	5	-1	133
Werkzeugwahl	266	T	6	-1	134
Werkzeugwahl	267	T	7	-1	135
Werkzeugwahl	268	T	8	-1	136
Werkzeugwahl	269	T	9	-1	137
Werkzeugwahl	270	T	10	-1	138
Werkzeugwahl	271	T	11	-1	139
Werkzeugwahl	272	T	12	-1	140
Werkzeugwahl	273	T	13	-1	141
Werkzeugwahl	274	T	14	-1	142
Werkzeugwahl	275	T	15	-1	143
Werkzeugwahl	276	T	16	-1	144
Werkzeugwahl	277	T	17	-1	145
Werkzeugwahl	278	T	18	-1	146
Werkzeugwahl	279	T	19	-1	147
Werkzeugwahl	280	T	20	-1	148



Werkzeughalter-spezifische Hilfsfunktionen, M6-Hilfsfunktionen					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Werkzeugwechsel	281	M	1	6 <sup>1)</sup>	149
Werkzeugwechsel	282	M	2	6 <sup>1)</sup>	150
Werkzeugwechsel	283	M	3	6 <sup>1)</sup>	151
Werkzeugwechsel	284	M	4	6 <sup>1)</sup>	152
Werkzeugwechsel	285	M	5	6 <sup>1)</sup>	153
Werkzeugwechsel	286	M	6	6 <sup>1)</sup>	154
Werkzeugwechsel	287	M	7	6 <sup>1)</sup>	155
Werkzeugwechsel	288	M	8	6 <sup>1)</sup>	156
Werkzeugwechsel	289	M	9	6 <sup>1)</sup>	157
Werkzeugwechsel	290	M	10	6 <sup>1)</sup>	158
Werkzeugwechsel	291	M	11	6 <sup>1)</sup>	159
Werkzeugwechsel	292	M	12	6 <sup>1)</sup>	160
Werkzeugwechsel	293	M	13	6 <sup>1)</sup>	161
Werkzeugwechsel	294	M	14	6 <sup>1)</sup>	162
Werkzeugwechsel	295	M	15	6 <sup>1)</sup>	163
Werkzeugwechsel	296	M	16	6 <sup>1)</sup>	164
Werkzeugwechsel	297	M	17	6 <sup>1)</sup>	165
Werkzeugwechsel	298	M	18	6 <sup>1)</sup>	166
Werkzeugwechsel	299	M	19	6 <sup>1)</sup>	167
Werkzeugwechsel	300	M	20	6 <sup>1)</sup>	168

( ) Der Wert kann geändert werden.

- 1) Der Wert ist abhängig von Maschinendatum:  
MD22560 \$MC\_TOOL\_CHANGE\_M\_MODE (M-Funktion für Werkzeugwechsel)
- 2) Der Wert lässt sich über folgende Maschinendaten mit einem anderen Wert vorbesetzen:  
MD20095 \$MC\_EXTERN\_RIGID\_TAPPING\_M\_NR (M-Funktion für das Umschalten in den gesteuerten Achsbetrieb (Ext. mode))  
MD20094 \$MC\_SPIND\_RIGID\_TAPPING\_M\_NR (M-Funktion für das Umschalten in den gesteuerten Achsbetrieb)  
Hinweis  
An die PLC wird immer der Wert 70 ausgegeben.
- 3) Der Wert wird eingestellt über Maschinendatum:  
MD22254 \$MC\_AUXFU\_ASSOC\_M0\_VALUE (Zusätzliche M-Fkt. für Programm-Halt)
- 4) Der Wert wird eingestellt über Maschinendatum:  
MD22256 \$MC\_AUXFU\_ASSOC\_M1\_VALUE (Zusätzliche M-Fkt. für bedingten Halt)
- 5) Der Wert wird eingestellt über Maschinendatum:  
MD10714 \$MN\_M\_NO\_FCT\_EOP (M-Funktion für Spindel aktiv nach Reset)
- 6) Der Wert wird eingestellt über Maschinendatum:  
MD26008 \$MC\_NIBBLE\_PUNCH\_CODE (Festlegung der M-Funktionen)

### 7.2.2 Übersicht: Ausgabeverhalten

Bedeutung der in der nachfolgenden Tabelle aufgeführten Parameter:

Parameter	Bedeutung
Index <n>	Maschinendatenindex der Parameter einer Hilfsfunktion
Ausgabeverhalten	MD22080 \$MC_AUXFU_PREDEF_SPEC[<n>], Bit 0 ... 18 Bit 19 ... 31: reserviert

### Ausgabeverhalten der vordefinierten Hilfsfunktionen

Systemfunktion	Index <n>																		
	Ausgabeverhalten, Bit																		
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Halt	0	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
bedingter Halt	1	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
Unterprogramm Ende	2	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
	3	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
	4	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
Werkzeugwechsel	5	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel rechts	6	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel links	7	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel halt	8	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel positionieren	9	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Achsbetrieb	10	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
automatische Getriebestufe	11	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 1	12	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 2	13	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 3	14	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 4	15	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 5	16	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Spindel-Drehzahl	17	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Vorschub	18	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(1)	(0)	0	(1)	0	(1)
Schneidenwahl	19	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(1)
DL	20	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(1)
Werkzeugwahl	21	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(1)
Halt (assoziiert)	22	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
bedingter Halt (assoziiert)	23	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
Unterprogramm Ende	24	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	(0)	(1)
Nibbeln	25	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(1)
Nibbeln	26	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(1)

Systemfunktion	Index <n>																			
	Ausgabeverhalten, Bit																			
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Nibbeln	27	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	28	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	29	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	30	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	31	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	32	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)

( ) Der Wert kann geändert werden.

### Bedeutung der Bits

Bit	Bedeutung
0	<p>Quittierung "normal" nach einen OB1-Takt</p> <p>Eine Hilfsfunktion mit normaler Quittierung wird zu Beginn des OB1-Zyklus in die NC/PLC-Nahtstelle ausgegeben. Über das hilfsfunktionsspezifische Änderungssignal wird dem PLC-Anwenderprogramm angezeigt, dass die Hilfsfunktion gültig ist.</p> <p>Die Quittierung der Hilfsfunktion erfolgt, sobald der Organisationsbaustein OB1 einmal komplett durchlaufen wurde. Dies entspricht einem vollständigen PLC-Anwenderzyklus.</p> <p>Die Ausgabe einer Hilfsfunktion mit normaler Quittierung erfolgt synchron zum Teileprogrammsatz, in dem sie programmiert ist. Ist die Abarbeitung des Teileprogrammsatzes, z. B. Bahn- und/oder Positionierachsbewegungen, vor der Quittierung der Hilfsfunktion beendet, wird der Satzwechsel so lange verzögert, bis die Quittierung durch die PLC erfolgt.</p> <p>Bei Bahnsteuerbetrieb kann eine konstante Bahngeschwindigkeit im Zusammenhang mit Hilfsfunktion mit normaler Quittierung nur dann eingehalten werden, wenn die Hilfsfunktionsausgabe während der Verfahrbewegung erfolgt und vor dem Erreichen des Satzendes durch die PLC quittiert wurde.</p>
1	<p>Quittierung "quick" mit OB40</p> <p>Eine Hilfsfunktion mit schneller Quittierung wird vor dem nächsten OB1-Zyklus in die NC/PLC-Nahtstelle ausgegeben. Über das hilfsfunktionsspezifische Änderungssignal wird dem PLC-Anwenderprogramm angezeigt, dass die Hilfsfunktion gültig ist.</p> <p>Die Quittierung der Hilfsfunktion erfolgt sofort durch das PLC-Grundprogramm im nächsten OB40-Takt. Die Quittierung der Hilfsfunktion ist daher keine Bestätigung für die Ausführung der entsprechenden PLC-Anwenderfunktion. Die Bearbeitung der Hilfsfunktion erfolgt weiterhin im OB1-Zyklus. Die nächste Ausgabe von Hilfsfunktionen an die PLC kann daher erst nach dem vollständigen Durchlauf dieses OB1-Zyklus erfolgen. Dies macht sich bei Bahnsteuerbetrieb vor allem dann bemerkbar (Absenken der Bahngeschwindigkeit), wenn in mehreren aufeinander folgenden Teileprogrammsätzen Hilfsfunktionen schneller Quittierung ausgegeben werden.</p> <p>Bei Hilfsfunktionen mit schneller Quittung ist nicht gewährleistet, dass die Reaktion im PLC-Anwenderprogramm satzsynchron erfolgt.</p> <p>Hinweis Die Parametrierung des Ausgabeverhaltens von Hilfsfunktionen als "Schnelle Hilfsfunktionen" ist nur in Verbindung mit anwenderdefinierten Hilfsfunktionen möglich.</p>

7.2 Vordefinierte Hilfsfunktionen

Bit	Bedeutung
2	Keine vordefinierte Hilfsfunktion
	Mit dieser Einstellung wird eine vordefinierte Hilfsfunktion wie eine anwenderdefinierte Hilfsfunktion behandelt. Die Hilfsfunktion löst dann nicht mehr die entsprechende Systemfunktion aus, sondern wird nur noch an die PLC ausgegeben. Beispiel: Umprojektion der Hilfsfunktion "Spindel positionieren" (Index 9) zu einer anwenderdefinierten Hilfsfunktion mit normaler Quittierung und Ausgabe vor der Verfahrbewegung. MD22080 \$MC_AUXFU_PREDEF_SPEC [ 9 ] = 'H25' (100101B)
3	Keine Ausgabe an die PLC
	Die Hilfsfunktion wird nicht an die PLC ausgegeben.
4	Spindelreaktion nach der Quittung durch die PLC
	Die zugehörige Systemfunktion wird erst nach der Quittierung durch die PLC ausgeführt.
5	Ausgabe vor der Bewegung
	Die Ausgabe der Hilfsfunktion an die PLC erfolgt vor den im Teileprogrammsatz programmierten Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen).
6	Ausgabe während der Bewegung
	Die Ausgabe der Hilfsfunktion an die PLC erfolgt während der im Teileprogrammsatz programmierten Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen).
7	Ausgabe am Satzende
	Die Ausgabe der Hilfsfunktion an die PLC erfolgt nach Abschluss der im Teileprogrammsatz programmierten Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen).
8	Keine Ausgabe nach Satzsuchlauf Type 1, 2, 4
	Satzsuchlauf Typ 1, 2, 4: Die während des Satzsuchlaufs aufgesammelte Hilfsfunktion wird nicht ausgegeben.
9	Aufsammlung während Satzsuchlauf mit Programmtest (Type 5, SERUPRO)
	Die Hilfsfunktion wird bei Satzsuchlauf mit Programmtest gruppenspezifisch in folgenden Systemvariablen aufgesammelt: <ul style="list-style-type: none"> <li>• \$AC_AUXFU_M_VALUE[&lt;n&gt;]</li> <li>• \$AC_AUXFU_M_EXT[&lt;n&gt;]</li> <li>• \$AC_AUXFU_M_STATE[&lt;n&gt;]</li> </ul>
10	Keine Ausgabe während Satzsuchlauf mit Programmtest (Type 5, SERUPRO)
	Die Hilfsfunktion wird bei Satzsuchlauf mit Programmtest nicht an die PLC ausgegeben.
11	Kanalübergreifende Hilfsfunktion (SERUPRO)
	Die Hilfsfunktion wird bei Satzsuchlauf mit Programmtest (SERUPRO) kanalübergreifend in der globalen Liste der Hilfsfunktionen aufgesammelt. Hinweis Pro Hilfsfunktionsgruppe wird immer nur die letzte Hilfsfunktion der Gruppe aufgesammelte.
12	Ausgabe erfolgte über Synchronaktion (nur lesbar)
	Das Bit wird gesetzt wenn die Hilfsfunktion über eine Synchronaktion an die PLC ausgegeben wurde.
13	Implizite Hilfsfunktion(nur lesbar)
	Das Bit wird gesetzt wenn die Hilfsfunktion implizit an die PLC ausgegeben wurde.
14	Aktives M01(nur lesbar)
	Das Bit wird gesetzt wenn die Hilfsfunktion bei aktivem M01 an die PLC ausgegeben wurde.

Bit	Bedeutung
15	Keine Ausgabe während Einfahr-Testlauf Die Hilfsfunktion wird während des Einfahr-Testlaufs nicht an die PLC ausgegeben`.
16	Nibbeln aus
17	Nibbeln ein
18	Nibbeln

**Hinweis**

Bei Hilfsfunktionen, für die kein Ausgabeverhalten festgelegt ist, wird folgendes Standard-Ausgabeverhalten wirksam:

- Bit 0 = 1: Ausgabedauer einen OB1-Takt
- Bit 7 = 1: Ausgabe am Satzende

## 7.2.3 Parametrierung

### 7.2.3.1 Gruppenzuordnung

Über die Gruppenzuordnung einer Hilfsfunktion wird die Behandlung der Hilfsfunktion bei Satzsuchlauf festgelegt. Die 168 zur Verfügung stehenden Hilfsfunktionsgruppen sind in vordefinierte und anwenderdefinierbare Gruppen aufgeteilt:

vordefinierte Gruppen:	1 ... 4	10 ... 12	72 ... 168
anwenderdefinierbare Gruppen:		5 ... 9	13 ... 71

Jede vordefinierte Hilfsfunktion ist standardmäßig einer Hilfsfunktionsgruppe zugeordnet. Diese Zuordnung kann für die meisten vordefinierte Hilfsfunktion über folgendes Maschinendatum geändert werden:

MD22040 \$MC\_AUXFU\_PREDEF\_GROUP[<n>] (Gruppenzuordnung von vordefinierten Hilfsfunktionen)

Soll eine Hilfsfunktion keiner Gruppe zugeordnet sein, ist im Maschinendatum der Wert "0" einzutragen.

Für die vordefinierten Hilfsfunktionen mit folgenden Indizes <n> kann die Gruppenzuordnung nicht geändert werden: 0, 1, 2, 3, 4, 22, 23, 24

**Hinweis****1. Hilfsfunktionsgruppe und Satzsuchlauf**

Hilfsfunktionen der 1. Hilfsfunktionsgruppe werden bei Satzsuchlauf nur aufgesammelt, aber nicht ausgegeben.

### 7.2.3.2 Typ, Adresserweiterung und Wert

Über die Parameter Typ, Adresserweiterung und Wert erfolgt die Programmierung einer Hilfsfunktion (siehe Kapitel "Programmierung einer Hilfsfunktion (Seite 439)").

#### Typ

Über den "Typ" wird der Bezeichner einer Hilfsfunktion festgelegt, z. B.:

"M"	für Zusatzfunktion
"S"	für Spindelfunktion
"F"	für Vorschub

Die Einstellung erfolgt über das Maschinendatum:

MD22050 \$MC\_AUXFU\_PREDEF\_TYPE[<n>] (Typ von vordefinierten Hilfsfunktionen)

---

#### Hinweis

Bei vordefinierten Hilfsfunktionen kann der "Typ" nicht verändert werden.

---

#### Adresserweiterung

Die "Adresserweiterung" einer Hilfsfunktion dient zur Adressierung unterschiedlicher Komponenten des gleichen Typs. Bei vordefinierten Hilfsfunktionen entspricht der Wert der "Adresserweiterung" der Spindelnummer, auf die sich die Hilfsfunktion bezieht.

Die Einstellung erfolgt über das Maschinendatum:

MD22060 \$MC\_AUXFU\_PREDEF\_EXTENSION[<n>] (Adresserweiterung für vordefinierte Hilfsfunktionen)

#### Zusammenfassen von Hilfsfunktionen

Soll eine Hilfsfunktion für alle Spindeln eines Kanals der gleichen Hilfsfunktionsgruppe zugeordnet werden, ist für den Parameter "Adresserweiterung" ein Wert von "-1" einzugeben.

Beispiel:

Für alle Spindeln des Kanals wird die Hilfsfunktion M3 (Maschinendatenindex = 6) der 2. Hilfsfunktionsgruppe zugeordnet.

```
MD22040 $MC_AUXFU_PREDEF_GROUP[ 6 ]      = 2
MD22050 $MC_AUXFU_PREDEF_TYPE[ 6 ]       = "M"
MD22060 $MC_AUXFU_PREDEF_EXTENSION[ 6 ]  = -1
MD22070 $MC_AUXFU_PREDEF_VALUE[ 6 ]     = 3
```

## Wert

Die Parameter "Wert" und "Typ" definieren die Bedeutung einer Hilfsfunktion. D. h. die Systemfunktion, die aufgrund dieser Hilfsfunktion aktiviert wird.

Der "Wert" einer Hilfsfunktion ist festgelegt im Maschinendatum:

MD22070 \$MC\_AUXFU\_PREDEF\_VALUE[<n>] (Wert von vordefinierten Hilfsfunktionen)

---

### Hinweis

Bei vordefinierter Hilfsfunktion kann der "Wert" nicht verändert werden. Für einige vordefinierte Hilfsfunktionen kann der "Wert" über zusätzliche Maschinendaten umprojektiert werden (siehe Kapitel "Assoziierte Hilfsfunktionen (Seite 433)").

---

### 7.2.3.3 Ausgabeverhalten

Über den Parameter "Ausgabeverhalten" wird festgelegt, wann eine vordefinierte Hilfsfunktion an die NC/PLC-Nahtstelle ausgegeben und wann sie von der PLC quittiert wird.

Die Einstellung erfolgt über das Maschinendatum:

MD22080 \$MC\_AUXFU\_PREDEF\_SPEC[<n>] (Ausgabeverhalten von vordefinierten Hilfsfunktionen)

### Ausgabeverhalten bezüglich der Bewegung

#### Ausgabe vor der Bewegung

- Die Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen) des vorausgehenden Teileprogrammsatzes werden mit Genauhalt beendet.
- Die Ausgabe der Hilfsfunktionen erfolgt mit Beginn des aktuellen Teileprogrammsatzes.
- Die Verfahrbewegungen des aktuellen Teileprogrammsatzes (Bahn- und/oder Positionierachsbewegungen) werden erst nach Quittierung der Hilfsfunktionen durch die PLC gestartet:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung): nach einem OB1-Zyklus
  - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung): nach einem OB40-Zyklus

#### Ausgabe während der Bewegung

- Die Ausgabe der Hilfsfunktionen erfolgt mit dem Start der Verfahrbewegungen (Bahn- und/oder Positionierachsbewegungen).
- Die Bahngeschwindigkeit des aktuellen Teileprogrammsatzes wird so reduziert, dass die Zeit bis zum Satzende größer als die Zeit zum Quittieren der Hilfsfunktionen durch die PLC ist:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung): ein OB1-Zyklus
  - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung): ein OB40-Zyklus

### **Ausgabe nach der Bewegung**

- Die Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen) des aktuellen Teileprogrammsatzes werden mit Genauhalt beendet.
- Die Ausgabe der Hilfsfunktionen erfolgt nach dem Beenden der Verfahrbewegungen.
- Der Satzwechsel erfolgt nach Quittierung der Hilfsfunktionen durch die PLC:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung): nach einem OB1-Zyklus
  - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung): nach einem OB40-Zyklus

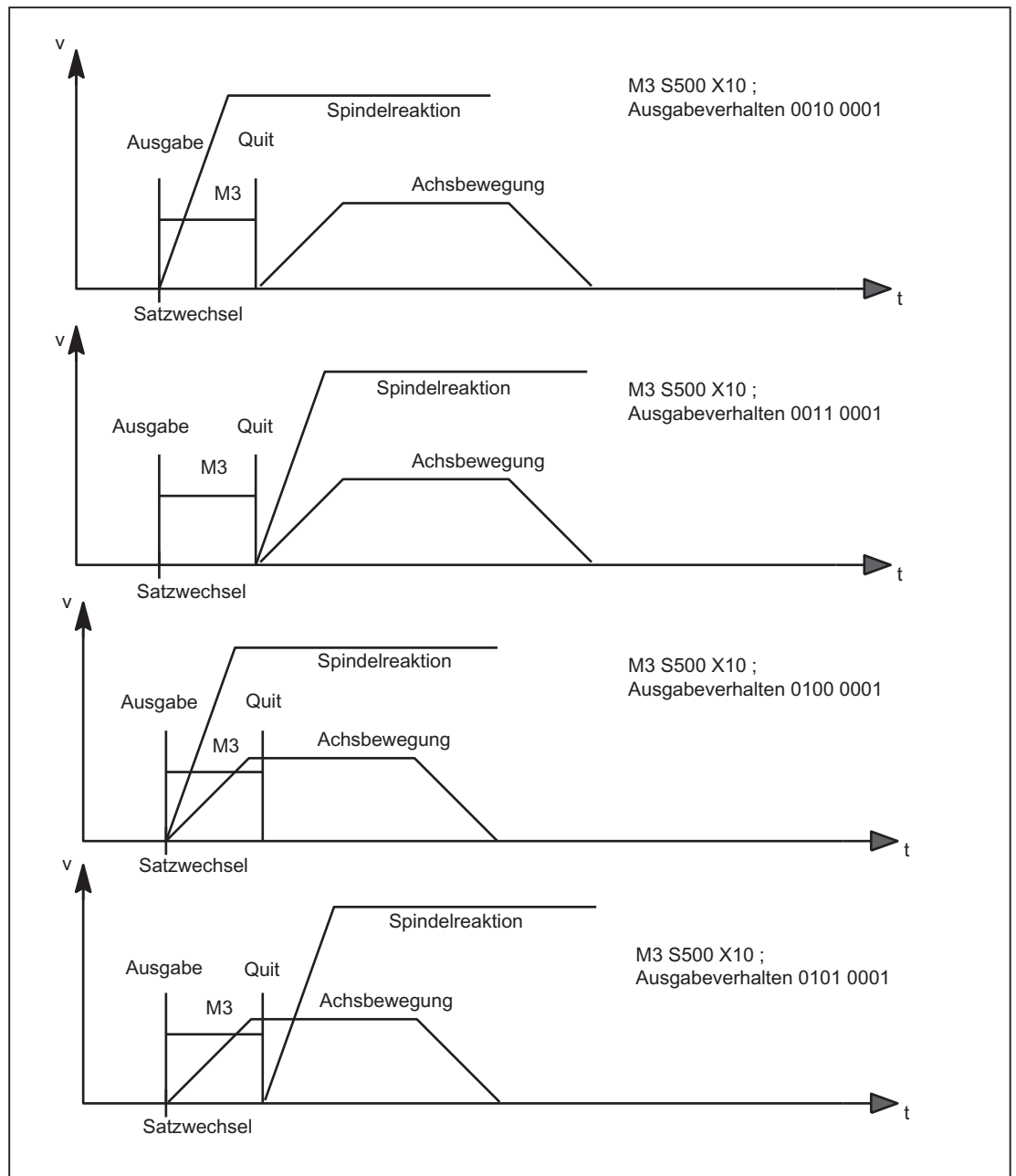
### **Beispiele für unterschiedliches Ausgabeverhalten**

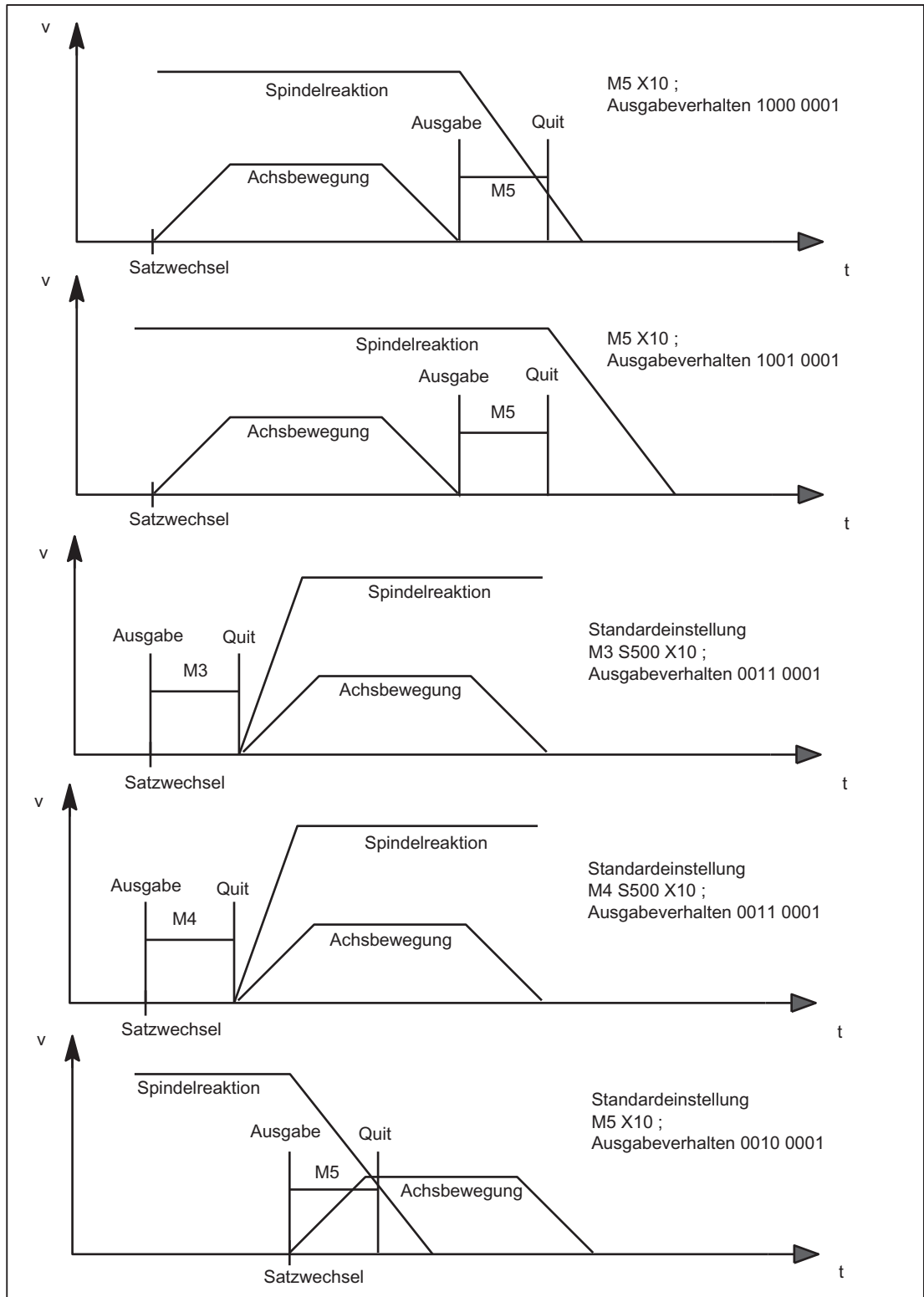
Die folgenden Abbildungen veranschaulichen das unterschiedliche Verhalten bezüglich:

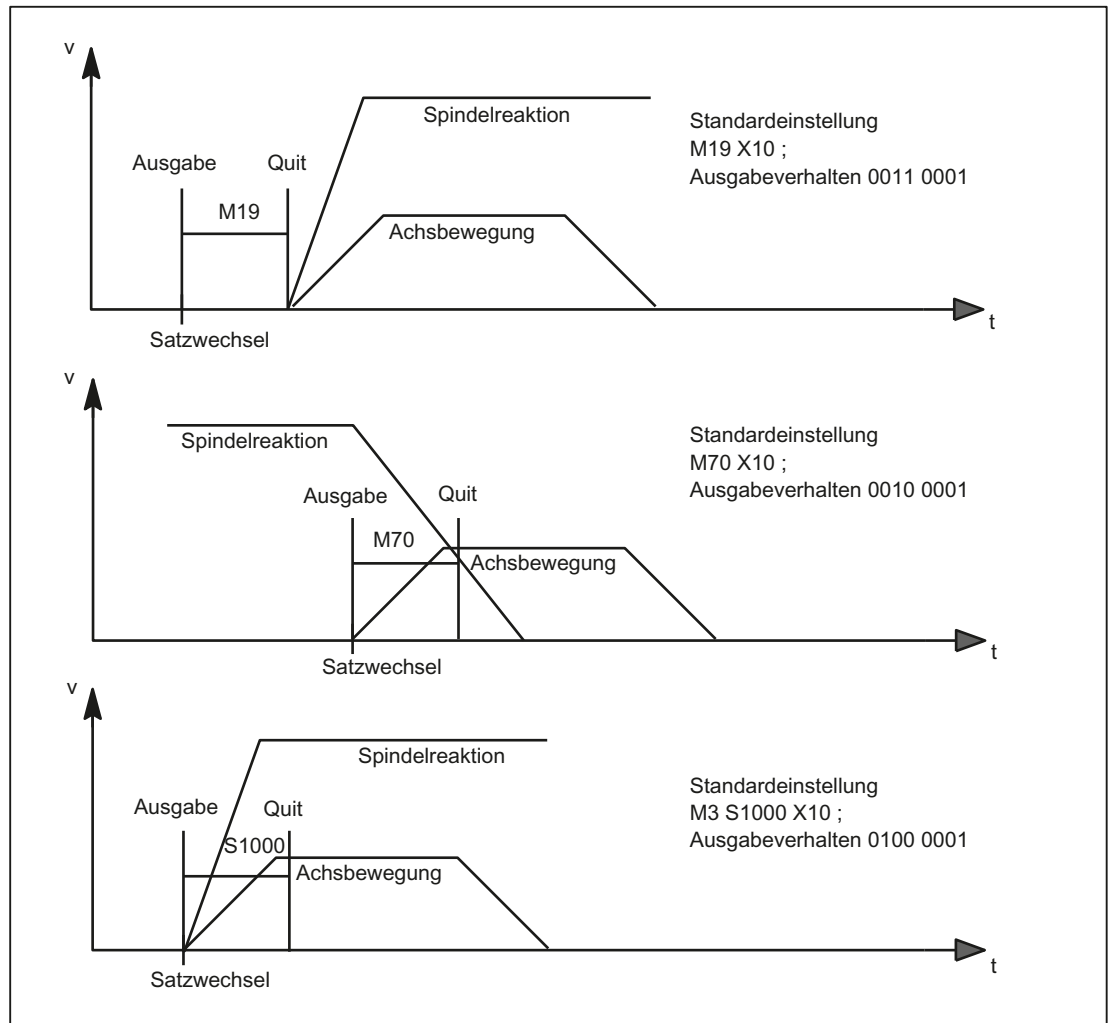
- Ausgabe und Quittierung der Hilfsfunktion
- Spindelreaktion (Drehzahländerung)
- Verfahrbewegung (Geschwindigkeitsänderung)

Die in den Abbildungen unter "Ausgabeverhalten" angegebenen binären Werte beziehen sich auf das parametrisierte Ausgabeverhalten (MD22080).









### 7.3 Anwenderdefinierte Hilfsfunktionen

Die Verwendung von anwenderdefinierten Hilfsfunktionen lässt sich in zwei Bereiche unterteilen:

- Erweiterung von vordefinierten Hilfsfunktionen
- Anwenderspezifische Hilfsfunktionen

#### Erweiterung von vordefinierten Hilfsfunktionen

Da die Maschinendaten der vordefinierten Hilfsfunktionen nur einmal vorhanden sind, kann darüber immer nur eine Spindel des Kanals adressiert werden. Zur Adressierung weiterer Spindeln müssen anwenderdefinierte Hilfsfunktionen zur Erweiterung der vordefinierten Hilfsfunktionen parametrisiert werden.

Die Erweiterung von vordefinierten Hilfsfunktionen bezieht sich ausschließlich auf den Parameter "Adresserweiterung". In den Parameter "Adresserweiterung" wird die Nummer der Spindel eingetragen, auf die sich die Hilfsfunktion bezieht.

Für folgende Systemfunktionen können die entsprechenden vordefinierten Hilfsfunktionen erweitert werden:

Systemfunktion	Typ		
		Adresserweiterung <sup>1)</sup>	
			Wert
Werkzeugwechsel	M	1	6
Spindel rechts	M	1	3
Spindel links	M	1	4
Spindel halt	M	1	5
Spindel positionieren	M	1	19
Achsbetrieb	M	1	70
automatische Getriebestufe	M	1	40
Getriebestufe 1	M	1	41
Getriebestufe 2	M	1	42
Getriebestufe 3	M	1	43
Getriebestufe 4	M	1	44
Getriebestufe 5	M	1	45
Spindel-Drehzahl	S	1	-1
Werkzeugwahl	T	1	-1

<sup>1)</sup> Adresserweiterung = 1 ist der in den Maschinendaten der vordefinierten Hilfsfunktionen verwendete Standardwert

**Beispiel:**

Erweiterung der vordefinierten Hilfsfunktion für die Systemfunktion "Spindel rechts" für die 2. und 3. Spindel des Kanals.

Hilfsfunktion "Spindel rechts" für die 2. Spindel des Kanals:

MD22010 \$MC_AUXFU_ASSIGN_TYPE [ n ]	= "M"
<b>MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [ n ]</b>	<b>= 2</b>
MD22030 \$MC_AUXFU_ASSIGN_VALUE [ n ]	= 3

Hilfsfunktion "Spindel rechts" für die 3. Spindel des Kanals:

MD22010 \$MC_AUXFU_ASSIGN_TYPE [ m ]	= "M"
<b>MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [ m ]</b>	<b>= 3</b>
MD22030 \$MC_AUXFU_ASSIGN_VALUE [ m ]	= 3

### Anwenderspezifische Hilfsfunktionen

Anwenderspezifische Hilfsfunktionen haben folgende Eigenschaften:

- Über anwenderspezifische Hilfsfunktionen werden ausschließlich Anwenderfunktionen aktiviert.
- Über anwenderspezifische Hilfsfunktionen können keine Systemfunktionen aktiviert werden.
- Eine anwenderspezifische Hilfsfunktion wird entsprechend dem parametrisierten Ausgabeverhalten an die PLC ausgegeben.
- Die Funktionalität einer anwenderspezifischen Hilfsfunktion wird durch den Maschinenhersteller/Anwender im PLC-Anwenderprogramm realisiert.

### 7.3.1 Parametrierung

#### 7.3.1.1 Maximale Anzahl anwenderdefinierter Hilfsfunktionen

Die maximale Anzahl von anwenderdefinierten Hilfsfunktionen pro Kanal ist parametrierbar über das Maschinendatum:

MD11100 \$MN\_AUXFU\_MAXNUM\_GROUP\_ASSIGN (Maximale Anzahl von anwenderdefinierten Hilfsfunktionen)

#### 7.3.1.2 Gruppenzuordnung

Über die Gruppenzuordnung einer Hilfsfunktion wird die Behandlung der Hilfsfunktion bei Satzsuchlauf festgelegt. Die 168 zur Verfügung stehenden Hilfsfunktionsgruppen sind in vordefinierte und anwenderdefinierbare Gruppen aufgeteilt:

vordefinierte Gruppen:	1 ... 4	10 ... 12	72 ... 168
anwenderdefinierbare Gruppen:		5 ... 9	13 ... 71

Jede anwenderdefinierte Hilfsfunktion ist standardmäßig der 1. Hilfsfunktionsgruppe zugeordnet. Die Zuordnung kann über das folgende Maschinendatum geändert werden:

MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP[<n>] (Gruppenzuordnung von anwenderdefinierten Hilfsfunktionen)

Soll eine Hilfsfunktion keiner Gruppe zugeordnet sein, ist im Maschinendatum der Wert "0" einzutragen.

---

#### Hinweis

##### 1. Hilfsfunktionsgruppe und Satzsuchlauf

Hilfsfunktionen der 1. Hilfsfunktionsgruppe werden bei Satzsuchlauf nur aufgesammelt, aber nicht ausgegeben.

---

### 7.3.1.3 Typ, Adresserweiterung und Wert

Über die Parameter Typ, Adresserweiterung und Wert erfolgt die Programmierung einer Hilfsfunktion (siehe Kapitel "Programmierung einer Hilfsfunktion (Seite 439)").

#### Typ

Über den "Typ" wird der Bezeichner einer Hilfsfunktion festgelegt.

Bezeichner für anwenderdefinierte Hilfsfunktionen sind:

Typ	Bezeichner	Bedeutung
"H"	Hilfsfunktion	Anwenderspezifische Hilfsfunktionen
"M"	Zusatzfunktion	Erweiterung von vordefinierten Hilfsfunktionen
"S"	Spindelfunktion	
"T"	Werkzeugnummer	

Die Einstellung erfolgt über das Maschinendatum:

MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE[<n>] (Typ von anwenderdefinierten Hilfsfunktionen)

#### Adresserweiterung

MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION[<n>] (Adresserweiterung für anwenderdefinierte Hilfsfunktionen)

Bei anwenderspezifischen Hilfsfunktionen ist die Funktionalität der Adresserweiterung nicht festgelegt. Sie dient allgemein der Unterscheidung von Hilfsfunktionen mit dem gleichen "Wert".

#### Zusammenfassen von Hilfsfunktionen

Sollen alle Hilfsfunktionen vom gleichen Typ und Wert der gleichen Hilfsfunktionsgruppe zugeordnet werden, ist für den Parameter "Adresserweiterung" ein Wert von "-1" einzugeben.

Beispiel:

Alle anwenderspezifischen Hilfsfunktionen mit Wert = 8 werden der 10. Hilfsfunktionsgruppe zugeordnet.

```
MD22000 $MC_AUXFU_ASSIGN_GROUP [ 1 ]           = 10
MD22010 $MC_AUXFU_ASSIGN_TYPE [ 1 ]           = "H"
MD22020 $MC_AUXFU_ASSIGN_EXTENSION [ 1 ]       = -1
MD22030 $MC_AUXFU_ASSIGN_VALUE [ 1 ]          = 8
```

### Wert

MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE[<n>] (Wert von anwenderdefinierten Hilfsfunktionen)

Bei anwenderspezifischen Hilfsfunktionen ist die Funktionalität des Parameters "Wert" nicht festgelegt. Allgemein wird über den Wert die entsprechende PLC-Anwenderfunktion aktiviert.

#### Zusammenfassen von Hilfsfunktionen

Sollen alle Hilfsfunktionen vom gleichen Typ und Adresserweiterung der gleichen Hilfsfunktionsgruppe zugeordnet werden, ist für den Parameter "Wert" ein Wert von "-1" einzugeben.

Beispiel:

Alle anwenderspezifischen Hilfsfunktionen mit Adresserweiterung = 2 werden der 11. Hilfsfunktionsgruppe zugeordnet.

MD22000 \$MC_AUXFU_ASSIGN_GROUP [ 2 ]	= 11
MD22010 \$MC_AUXFU_ASSIGN_TYPE [ 2 ]	= "H"
MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [ 2 ]	= 2
MD22030 \$MC_AUXFU_ASSIGN_VALUE [ 2 ]	= -1

### 7.3.1.4 Ausgabeverhalten

Das "Ausgabeverhalten" von anwenderdefinierten Hilfsfunktionen ist parametrierbar über das Maschinendatum:

MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC[<n>] (Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)

Zur Beschreibung der einzelnen Ausgabeparameter siehe Kapitel "Ausgabeverhalten (Seite 423)" der vordefinierten Hilfsfunktionen. Die dort gemachten Aussagen gelten sinngemäß auch für das Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen.



## 7.4 Assoziierte Hilfsfunktionen

### Funktion

Assoziierte Hilfsfunktionen sind anwenderdefinierte Hilfsfunktionen, die die gleiche Wirkung wie die entsprechenden vordefinierten Hilfsfunktionen haben. Für folgende vordefinierten Hilfsfunktionen können anwenderdefinierte Hilfsfunktionen assoziiert werden:

- M0 (Halt)
- M1 (bedingter Halt)

### Voraussetzungen

Voraussetzung für die Assoziierung einer anwenderdefinierten Hilfsfunktion zu einer der oben genannten vordefinierten Hilfsfunktionen ist die Parametrierung einer anwenderdefinierten Hilfsfunktion. Als Parameter "Typ" der anwenderdefinierten Hilfsfunktion ist dabei nur "M" erlaubt.

### Parametrierung

Die Assoziierung einer anwenderdefinierten Hilfsfunktion zu einer der oben genannten vordefinierten Hilfsfunktionen erfolgt in den Maschinendaten:

MD22254 \$MC\_AUXFU\_ASSOC\_M0\_VALUE (Zusätzliche M-Funktion für Programm-Halt)

MD22256 \$MC\_AUXFU\_ASSOC\_M1\_VALUE (Zusätzliche M-Funktion für bedingten Halt)

### Gruppenzuordnung

Als Gruppenzuordnung einer assoziierten anwenderdefinierten Hilfsfunktion wird fest die Gruppenzuordnung der entsprechenden vordefinierten Hilfsfunktion übernommen.

### Anwendung

Assoziierte Hilfsfunktionen dürfen angewendet werden in:

- Hauptprogramm
- Unterprogramm
- Zyklus

---

#### Hinweis

Assoziierte Hilfsfunktionen dürfen nicht in Synchronaktionen angewendet werden.

---

### NC/PLC-Nahtstellensignale

Bei einer assoziierten anwenderdefinierten Hilfsfunktion werden an die NC/PLC-Nahtstelle dieselben Signale ausgegeben wie bei der entsprechenden vordefinierten Hilfsfunktion. Zur Unterscheidung, welche Hilfsfunktion tatsächlich programmiert wurde, wird aber als Wert der Hilfsfunktion der Wert der anwenderdefinierten Hilfsfunktionen (Parameter "Wert") ausgegeben. Im PLC-Anwenderprogramm besteht damit die Möglichkeit, zwischen vordefinierter und anwenderdefinierter Hilfsfunktion zu unterscheiden.

---

#### Hinweis

Eine Änderung der Maschinendaten MD22254 und/oder MD22256 erfordert u. U. eine entsprechende Anpassung des PLC-Anwenderprogramms.

---

#### Spezifische NC/PLC-Nahtstellensignale

Folgende spezifische NC/PLC-Nahtstellensignale stehen zur Verfügung:

- DB21, ... DBX318.5 (Assoziiertes M00/M01 aktiv) Rückmeldesignal
- DB21, ... DBX30.5 (Assoziiertes M01 aktivieren) Aktivierungssignal

### Randbedingungen

Folgenden Randbedingungen sind zu beachten:

- Eine anwenderdefinierte Hilfsfunktion darf nicht mehrfach assoziiert werden.
- Vordefinierte Hilfsfunktionen (z. B. M3, M4, M5 etc.) dürfen nicht assoziiert werden.

### Beispiel

Assoziieren der anwenderdefinierten Hilfsfunktion M123 mit M0:

MD22254 \$MC\_AUXFU\_ASSOC\_M0\_VALUE = 123

Die anwenderdefinierte Hilfsfunktion M123 hat damit die gleiche Funktionalität wie M0.

## 7.5 Typ-spezifisches Ausgabeverhalten

### Funktion

Das Ausgabeverhalten der Hilfsfunktionen bezüglich einer im Teileprogrammsatz programmierten Verfahrbewegung kann Typ-spezifisch festgelegt werden.

### Parametrierung

Die Parametrierung des Typ-spezifischen Ausgabeverhaltens erfolgt über die Maschinendaten:

MD22200 \$MC\_AUXFU\_M\_SYNC\_TYPE (Ausgabezeitpunkt M-Funktionen)

MD22210 \$MC\_AUXFU\_S\_SYNC\_TYPE (Ausgabezeitpunkt S-Funktionen)

MD22220 \$MC\_AUXFU\_T\_SYNC\_TYPE (Ausgabezeitpunkt T-Funktionen)

MD22230 \$MC\_AUXFU\_H\_SYNC\_TYPE (Ausgabezeitpunkt H-Funktionen)

MD22240 \$MC\_AUXFU\_F\_SYNC\_TYPE (Ausgabezeitpunkt F-Funktionen)

MD22250 \$MC\_AUXFU\_D\_SYNC\_TYPE (Ausgabezeitpunkt D-Funktionen)

MD22252 \$MC\_AUXFU\_DL\_SYNC\_TYPE (Ausgabezeitpunkt DL-Funktionen)

Folgende Ausgabeverhalten können parametriert werden:

MD \$MC\_AUXFU\_XX\_SYNC\_TYPE = <Wert>

Wert	Ausgabeverhalten
0	Ausgabe vor der Bewegung
1	Ausgabe während der Bewegung
2	Ausgabe am Satzende
3	Keine Ausgabe an PLC
4	Ausgabe entsprechend des mit MD22080 festgelegten Ausgabeverhaltens

Zur Beschreibung der verschiedenen Ausgabeverhalten siehe Kapitel "Ausgabeverhalten (Seite 423)".

---

### Hinweis

Die für den jeweiligen Hilfsfunktionstyp einstellbaren Ausgabeverhalten sind dem Listenhandbuch "Ausführliche Maschinendaten-Beschreibung" zu entnehmen.

---

**Beispiel**

Ausgabe von Hilfsfunktionen mit unterschiedlichem Ausgabeverhalten in einem Teileprogrammsatz mit Verfahrbewegung.

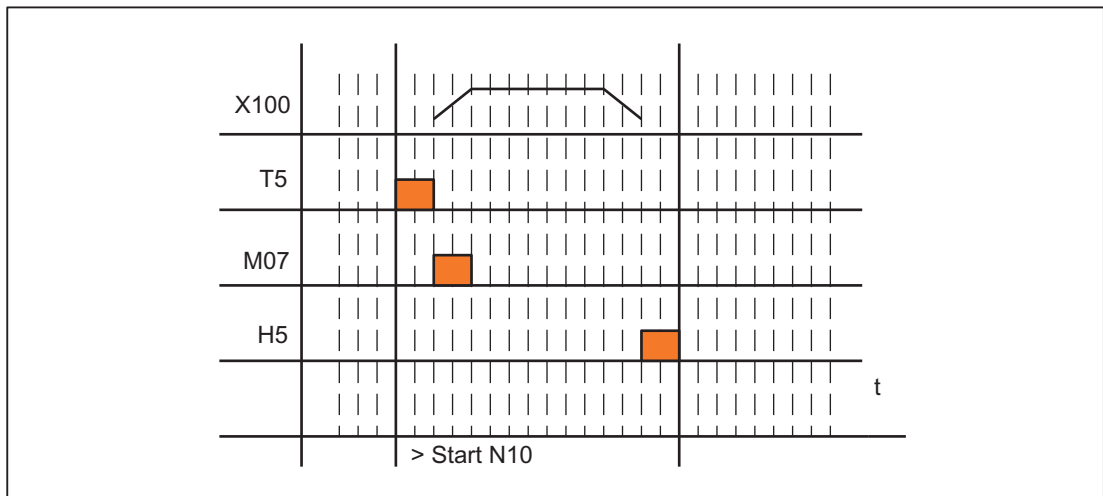
Parametriertes Ausgabeverhalten:

- MD22200 \$MC\_AUXFU\_M\_SYNC\_TYPE = 1      ⇒ M-Funktion:  
Ausgabe **während** der Bewegung
- MD22220 \$MC\_AUXFU\_T\_SYNC\_TYPE = 0      ⇒ T-Funktion:  
Ausgabe **vor** der Bewegung
- MD22230 \$MC\_AUXFU\_H\_SYNC\_TYPE = 2      ⇒ H-Funktion:  
Ausgabe **am Satzende**

Teileprogrammsatz:

```
Programmcode  
...  
N10 G01 X100 M07 H5 T5  
...  
...
```

Zeitlicher Ablauf der Hilfsfunktionsausgabe:



## 7.6 Prioritäten des parametrierten Ausgabeverhaltens

Die Prioritäten bezüglich des parametrierten Ausgabeverhaltens einer Hilfsfunktion müssen für folgende Kriterien getrennt beachtet werden:

- Ausgabedauer (normale / schnelle Quittierung)
- Ausgabe bezüglich der Bewegung (vor / während / nach der Bewegung)

Allgemein gilt, dass das parametrierte Ausgabeverhalten mit der niedrigeren Priorität immer dann wirksam wird, wenn kein höher priorisiertes Ausgabeverhalten parametriert wurde.

### Ausgabedauer

Für die Ausgabedauer gelten folgende Prioritäten:

Priorität	Ausgabeverhalten	Festlegung über:
Höchste	Hilfsfunktionsspezifisch	Teileprogrammanweisung: QU(...) (siehe Kapitel "Programmierbare Ausgabedauer (Seite 441)")
↓	Hilfsfunktionsspezifisch	MD22035 \$MC_AUXFU_ASSIGN_SYNC[<n>] MD22080 \$MC_AUXFU_PREDEF_SYNC[<n>]
↓	Gruppenspezifisch	MD11110 \$MC_AUXFU_GROUP_SPEC[<n>]
Niedrigste	Keine Festlegung	Standard-Ausgabeverhalten: Ausgabedauer einen OB1-Takt

### Ausgabe bezüglich der Bewegung

Für die Ausgabe bezüglich der Bewegung gelten folgende Prioritäten:

Priorität	Ausgabeverhalten	Festlegung über:
Höchste	Hilfsfunktionsspezifisch	MD22035 \$MC_AUXFU_ASSIGN_SYNC[<n> MD22080 \$MC_AUXFU_PREDEF_SYNC[<n>]
↓	Gruppenspezifisch	MD11110 \$MC_AUXFU_GROUP_SPEC[<n>]
↓	Typspezifisch	MD22200 \$MC_AUXFU_M_SYNC_TYPE MD22210 \$MC_AUXFU_S_SYNC_TYPE MD22220 \$MC_AUXFU_T_SYNC_TYPE MD22230 \$MC_AUXFU_H_SYNC_TYPE MD22240 \$MC_AUXFU_F_SYNC_TYPE MD22250 \$MC_AUXFU_D_SYNC_TYPE MD22252 \$MC_AUXFU_DL_SYNC_TYPE
Niedrigste	Keine Festlegung	Standard-Ausgabeverhalten: Ausgabe am Satzende

---

#### Hinweis

#### Teileprogrammsätze ohne Bahnbewegung

In einem Teileprogrammsatz ohne Bahnbewegung (also auch bei Positionierachsen und Spindeln) werden die Hilfsfunktionen sofort in einem Block ausgegeben.

---

## 7.7 Programmierung einer Hilfsfunktion

### Syntax

Die Programmierung einer Hilfsfunktion erfolgt in einem Teileprogrammsatz mit folgender Syntax:  
<Typ> [<Adresserweiterung>=] <Wert>

---

#### Hinweis

Wird keine Adresserweiterung programmiert, wird implizit die Adresserweiterung = 0 gesetzt.  
Vordefinierte Hilfsfunktionen mit der Adresserweiterung = 0 beziehen sich immer auf die Masterspindel des Kanals.

---

### Symbolische Adressierung

Die Werte für die Parameter "Adresserweiterung" und "Wert" können auch symbolisch angegeben werden. Der symbolische Name für die Adresserweiterung muss dann in eckigen Klammern angegeben werden.

Beispiel:

Symbolische Programmierung der Hilfsfunktion M3 (Spindel rechts) für die 1. Spindel:

Programmcode	Kommentar
DEF SPINDEL_NR=1	; 1.Spindel im Kanal
DEF DREHRICHTUNG=3	; Drehrichtung rechts
N100 M[SPINDEL_NR]=DREHRICHTUNG	; entsprechend: M1=3

---

#### Hinweis

Bei der Verwendung von symbolischen Namen zur Programmierung einer Hilfsfunktion wird bei der Ausgabe der Hilfsfunktion an die PLC nicht der symbolische Name übergeben, sondern der entsprechende Zahlenwert.

---

### Beispiele

#### Beispiel 1: Programmierung vordefinierter Hilfsfunktionen

Programmcode	Kommentar
N10 M3	; "Spindel rechts" für die Master-Spindel des Kanals.
N20 M0=3	; "Spindel rechts" für die Master-Spindel des Kanals.
N30 M1=3	; "Spindel rechts" für die 1.Spindel des Kanals.
N40 M2=3	; "Spindel rechts" für die 2.Spindel des Kanals.

#### Beispiel 2: Programmierbeispiele von Hilfsfunktionen mit den entsprechenden Werten zur Ausgabe an die PLC

Programmcode	Kommentar
DEF Kühlmittel=12	; Ausgabe an die PLC: - - -
DEF Schmiermittel=130	; Ausgabe an die PLC: - - -
H[Kühlmittel]=Schmiermittel	; Ausgabe an die PLC: H12=130
H=Kühlmittel	; Ausgabe an die PLC: H0=12
H5	; Ausgabe an die PLC: H0=5
H=5.379	; Ausgabe an die PLC: H0=5.379
H17=3.5	; Ausgabe an die PLC: H17=3.5
H[Kühlmittel]=13.8	; Ausgabe an die PLC: H12=13.8
H='HFF13'	; Ausgabe an die PLC: H0=65299
H='B1110'	; Ausgabe an die PLC: H0=14
H5.3=21	; Fehler



## 7.8 Programmierbare Ausgabedauer

### Funktion

Anwenderspezifischen Hilfsfunktionen, für die das Ausgabeverhalten "Ausgabedauer einen OB1-Zyklus (Langsame Quittierung)" parametrieren wurde, können für einzelne Ausgaben über die Teileprogrammanweisung `QU` (Quick) zu Hilfsfunktionen mit schneller Quittierung definiert werden.

### Syntax

Die Definition einer Hilfsfunktion mit schneller Quittierung erfolgt in einem Teileprogrammsatz mit folgender Syntax:  
`<Typ> [<Adresserweiterung>] =QU (<Wert>)`

### Beispiel

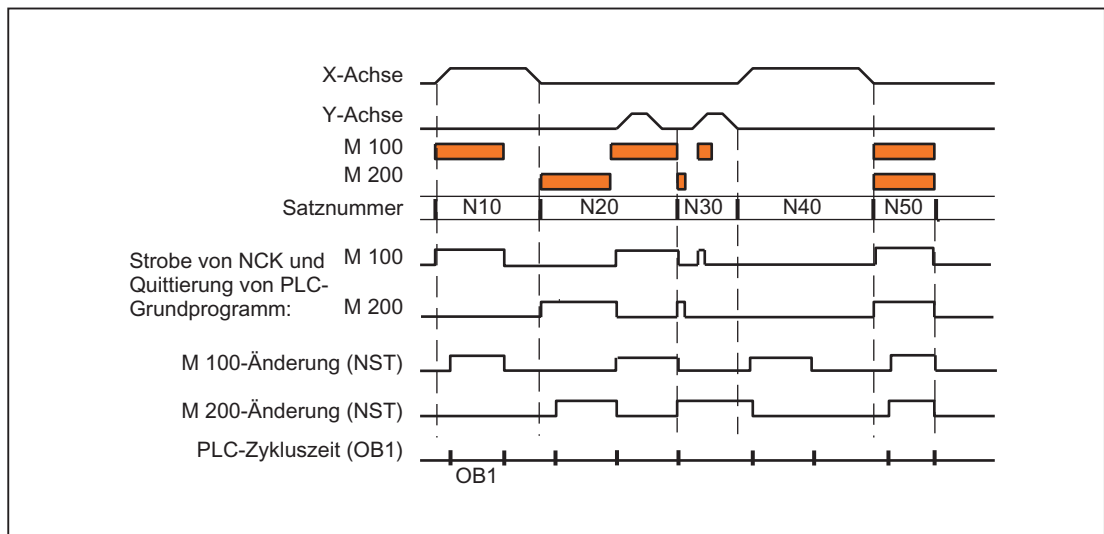
Unterschiedliches Verhalten bei der Ausgabe der Hilfsfunktionen M100 und M200 in einem Teileprogramm. Das Ausgabeverhalten der Hilfsfunktionen ist folgendermaßen parametrieren:

- M100
  - Ausgabedauer einen OB1-Zyklus (Langsame Quittierung)
  - Ausgabe während der Bewegung
- M200
  - Ausgabedauer einen OB1-Zyklus (Langsame Quittierung)
  - Ausgabe vor der Bewegung

Programmcode	Kommentar
N10 G94 G01 X50 M100	; Ausgabe von M100: während der Bewegung ; Quittierung: langsam
N20 Y5 M100 M200	; Ausgabe von M200: vor der Bewegung ; Ausgabe von M100: während der Bewegung ; Quittierung: langsam
N30 Y0 M=QU(100) M=QU(200)	; Ausgabe von M200: vor der Bewegung ; Ausgabe von M100: während der Bewegung ; Quittierung: schnell
N40 X0	
N50 M100 M200	; Ausgabe von M200: sofort 1) ; Ausgabe von M100: sofort 1) ; Quittierung: langsam
M17	

1) Ohne Verfahrbewegung erfolgt die Ausgabe von Hilfsfunktionen an die PLC immer sofort.

Das folgende Bild zeigt den zeitlichen Ablauf des Teileprogramms. Besonders zu beachten ist der zeitliche Unterschied bei der Abarbeitung der Teileprogrammätze N20 und N30.



## 7.9 Hilfsfunktionsausgabe an die PLC

### Funktion

Bei der Ausgabe einer Hilfsfunktion an die PLC werden folgende Signale und Werte an die NC/PLC-Nahtstelle übergeben:

- Änderungssignale
- Parameter "Adresserweiterung"
- Parameter "Wert"

### Datenbereiche in der NC/PLC-Nahtstelle

Die Änderungssignale und Werte der Hilfsfunktionen liegen in der NC/PLC-Nahtstelle in folgenden Datenbereichen:

- Änderungssignale bei Hilfsfunktionsübergabe von NC-Kanal:  
DB21, ... DBB58 - DBB67
- Übergebene M-/ und S-Funktionen:  
DB21, ... DBB68 - DBB112
- Übergebene T-/ D-/ DL-Funktionen:  
DB21, ... DBB116 - DBB136
- Übergebene H-/ F-Funktionen:  
DB21, ... DBB140 - DBB190
- Dekodierte M-Signale (M0 - M99):  
DB21, ... DBB194 - DBB206 (dynamische M-Funktionen)

Für Informationen der Zugriffsverfahren zur NC/PLC-Nahtstelle siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)".

Die Ausführliche Beschreibung der o. g. Datenbereiche in der NC/PLC-Nahtstelle findet sich in:

#### Literatur:

Listenhandbuch Listen, Buch 2; PLC-Anwendernahtstellen,  
Kapitel: Kanalspezifische Signale (DB 21 – DB 30)"

## 7.10 Hilfsfunktionen ohne Satzwechselferzögerung

### Funktion

Auch bei Hilfsfunktionen mit einem parametrisierten und/oder programmierten Ausgabeverhalten:

- "Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung)"
- "Ausgabe vor der Bewegung" oder "Ausgabe während der Bewegung"

kann es bei Bahnsteuerbetrieb (kurze Verfahrswege und hohen Geschwindigkeiten) zu Geschwindigkeitseinbrüchen kommen, weil zum Satzende hin auf die Quittierung der Hilfsfunktion durch die PLC gewartet werden muss. Um derartige Geschwindigkeitseinbrüche zu vermeiden, kann der Satzwechsel unabhängig bezüglich der Quittierung derartiger Hilfsfunktionen gemacht werden.

### Parametrierung

Das Unterdrücken der Satzwechselferzögerung bei schnellen Hilfsfunktionen wird eingestellt über das Maschinendatum:

MD22100 \$MC\_AUXFU\_QUICK\_BLOCKCHANGE (Satzwechselferzögerung bei schnellen Hilfsfunktionen)

Wert	Bedeutung
0	Bei der schnellen Hilfsfunktionsausgabe an die PLC wird der Satzwechsel bis zur Quittierung durch die PLC (OB40) verzögert.
1	Bei der schnellen Hilfsfunktionsausgabe an die PLC wird der Satzwechsel nicht verzögert.

### Randbedingungen

Die Synchronität von Hilfsfunktionen, die ohne Satzwechselferzögerung ausgegeben werden, ist bezüglich des Teileprogrammumsatzes, in dem sie programmiert sind, nicht mehr gewährleistet. Im "Worst-Case"-Szenario erfolgt die Quittierung einen OB40-Takt und die Bearbeitung der Hilfsfunktion einen OB1-Takt nach dem Satzwechsel zum nächsten Teileprogrammumsatz.

## 7.11 M-Funktion mit implizitem Vorlaufstopp

### Funktion

Soll im Zusammenhang mit einer Hilfsfunktion ein Vorlaufstopp ausgelöst werden, kann dieser explizit über den Teileprogrammbehehl `STOPRE` programmiert werden. Soll bei Programmierung der M-Funktion immer ein Vorlaufstopp ausgelöst werden, kann dieses Verhalten M-Funktionsspezifisch über folgendes Maschinendatum parametrierl werden:

MD10713 \$MN\_M\_NO\_FCT\_STOPRE[<n>] (M-Funktion mit Vorlaufstopp)

### Beispiel

Die anwenderdefinierte M-Funktion M88 soll einen Vorlaufstopp auslösen.

#### Parametrierung:

MD10713 \$MN\_M\_NO\_FCT\_STOPRE [ 0 ] = 88

#### Anwendung:

Teileprogramm (Auszug)

Programmcode	Kommentar
...	
N100 G0 X10 M88	; Verfabrbewegung und impliziter Vorlaufstopp durch M88.
N110 Y=R1	; N110 wird erst nach Abschluss der Verfabrbewegung und Quittierung der M-Funktion interpretiert.
...	

### Randbedingungen

Wird in einem Teileprogramm ein Unterprogramm durch eine der beiden folgenden Möglichkeiten indirekt über eine M-Funktion aufgerufen, erfolgt dabei kein Vorlaufstopp:

- MD10715 \$MN\_M\_NO\_FCT\_CYCLE (Durch Unterprogramm zu ersetzende M-Funktion)
- M98 (ISO-Dialekt-T / ISO-Dialekt-M)

## 7.12 Verhalten bei Überspeichern

### Überspeichern

Über die SINUMERIK-Bedienoberfläche können vor dem Start folgender Funktionen:

- NC-START eines Teileprogramms
- NC-START zur Wiederaufnahme eines unterbrochenen Teileprogramms

die Hilfsfunktionen, die mit dem Start ausgegeben werden, durch die Funktion "Überspeichern" verändert werden.

Mögliche Anwendungsfälle sind:

- Ergänzen von Hilfsfunktionen nach Satzsuchlauf
- Herstellen des Ausgangszustandes zum Einfahren eines Teileprogramms

### Überspeicherbare Hilfsfunktionstypen

Folgende Hilfsfunktionstypen können überspeichert werden:

- M (Zusatzfunktion)
- S (Spindeldrehzahl)
- T (Werkzeugnummer)
- H (Hilfsfunktion)
- D (Werkzeugkorrekturnummer)
- DL (Summenkorrektur)
- F (Vorschub)

### Gültigkeitsdauer

Eine überspeicherte Hilfsfunktion, z. B. M3 (Spindel rechts), ist so lange gültig, bis sie durch eine andere Hilfsfunktion der gleichen Hilfsfunktionsgruppe, durch erneute Überspeicherung oder durch Programmierung in einem Teileprogrammsatz überschrieben wird.

## 7.13 Verhalten bei Satzsuchlauf

### 7.13.1 Hilfsfunktionsausgabe bei Satzsuchlauf Typ 1,2 und 4

#### Ausgabeverhalten

Bei Satzsuchlauf Typ 1, 2 und 4 werden die Hilfsfunktionen gruppenspezifisch aufgesammelt. Die jeweils letzte Hilfsfunktion einer Hilfsfunktionsgruppe wird nach NC-START in einem eigenen Teileprogrammssatz vor dem eigentlichen Wiedereinstiegssatz mit folgendem Ausgabeverhalten ausgegeben:

- Ausgabedauer einen OB1-Zyklus (normale Quittierung)
- Ausgabe vor der Bewegung

#### Ausgabesteuerung

Ob eine Hilfsfunktion nach Satzsuchlauf an die PLC ausgegeben wird oder nicht, lässt sich projektieren über das Bit 8 der Maschinendaten:

- MD22080 \$MC\_AUXFU\_PREDEF\_SPEC[<n>]  
(Ausgabeverhalten von vordefinierten Hilfsfunktionen)  
mit <n> = Systemfunktionsindex (0 ... 32)
- MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC[<n>]  
(Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)  
mit <n> = Hilfsfunktionsindex (0 ... 254)
- MD11110 \$MN\_AUXFU\_GROUP\_SPEC[<n>]  
(Ausgabeverhalten der Hilfsfunktionen einer Gruppe)  
mit <n> = Gruppenindex (0 ... 63)

Bit	Wert	Bedeutung
10	0	Ausgabe während Satzsuchlauf Typ 1,2,4
	1	Keine Ausgabe während Satzsuchlauf Typ 1,2,4

Dieses Verhalten hat keinen Einfluss auf die Anzeige oder auf die Variablen \$AC\_AUXFU\_STATE[<n>], \$AC\_AUXFU\_VALUE[<n>] und \$AC\_AUXFU\_EXT[<n>]. Die Hilfsfunktionen gelten nach Satzsuchlauf immer noch als aufgesammelt, obwohl sie nicht an die PLC ausgegeben werden.

Eine nach Satzsuchlauf nicht ausgegebene Hilfsfunktion überschreibt beim Aufsammeln auch eine Hilfsfunktion, deren Bit 8 nicht gesetzt ist.

Der Anwender kann nach Satzsuchlauf die aufgesammelten Hilfsfunktionen abfragen und unter Umständen diese selber nochmals über das Teileprogramm oder über Synchronaktionen ausgeben.

---

#### Hinweis

Folgende Hilfsfunktionen werden nicht aufgesammelt:

- Hilfsfunktionen, die keiner Hilfsfunktionsgruppe zugeordnet sind.
  - Hilfsfunktionen, die der 1. Hilfsfunktionsgruppe zugeordnet sind.
- 

### Überspeichern von Hilfsfunktionen

Nach erfolgtem Satzsuchlauf werden mit dem nächsten NC-START die aufgesammelten Hilfsfunktionen ausgegeben. Besteht die Anforderung, zusätzliche Hilfsfunktionen auszugeben, können diese über die Funktion "Überspeichern" hinzugefügt werden (siehe Kapitel "Verhalten bei Überspeichern (Seite 446)").

### Verhalten bezüglich M19 (Spindel positionieren)

Nach Satzvorlauf wird immer die letzte mit M19 programmierte Spindelpositionierung durchgeführt, auch wenn vom Teileprogrammsatz mit M19 bis zum Zielsatz noch andere spindelspezifische Hilfsfunktionen programmiert sind. Das Setzen der erforderlichen Spindelfreigaben muss im PLC-Anwenderprogramm daher abgeleitet werden von den Nahtstellensignalen der Fahrbefehle:

DB31, ... DBX64.6 / 64.7 (Fahrbefehl minus / plus)

Die spindelspezifischen Hilfsfunktionen M3, M4, M5 eignen sich in diesem Fall nicht dazu, da sie unter Umständen erst nach der Spindelpositionierung an die PLC ausgegeben werden.

Für ausführliche Informationen zum Satzsuchlauf siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)".



## 7.13.2 Zuordnung einer Hilfsfunktion zu mehreren Gruppen

### Funktion

Über die Gruppenzuordnung (MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP) können anwenderdefinierte Hilfsfunktionen auch mehreren Gruppen zugeordnet werden. Beim Satzsuchlauf werden diese Hilfsfunktionen bezüglich aller projizierten Gruppen aufgesammelt.

---

### Hinweis

Vordefinierte Hilfsfunktionen können nur einer Gruppe zugeordnet werden.

---

### Beispiel

In der DIN sind folgende M-Befehle zur Kühlmittelausgabe vorgesehen:

- M7: Kühlmittel 2 EIN
- M8: Kühlmittel 1 EIN
- M9: Kühlmittel 1 und 2 AUS

Damit beide Kühlmittel auch gemeinsam aktiv sein können:

- Müssen M7 und M8 in zwei getrennten Gruppen aufgesammelt werden (z. B. Gruppe 5 und 6)
- Muss M9 diesen beiden Gruppen zugeordnet sein, z. B.:
  - Gruppe 5: M7, M9
  - Gruppe 6: M8, M9

**Parametrierung:**

MD11100 \$MN\_AUXFU\_MAXNUM\_GROUP\_ASSIGN = 4  
MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP [0] = 5  
MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP [1] = 5  
MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP [2] = 6  
MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP [3] = 6  
MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE [0] = M  
MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE [1] = M  
MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE [2] = M  
MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE [3] = M  
MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION [0] = 0  
MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION [1] = 0  
MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION [2] = 0  
MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION [3] = 0  
MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE [0] = 7  
MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE [1] = 9  
MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE [2] = 8  
MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE [3] = 9  
MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC [0] = 'H121'  
MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC [1] = 'H121'  
MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC [2] = 'H121'  
MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC [3] = 'H121'

**Teileprogramm (Ausschnitt):**

Programmcode
...
N10 ... M8
N20 ... M9
N30 ... M7
...

Beim Satzsuchlauf wird die Hilfsfunktion M9 bezüglich der Gruppen 5 und 6 aufgesammelt.

**Abfrage der aufgesammelten M-Hilfsfunktionen:**

M-Funktion der 5. Gruppe: \$AC\_AUXFU\_M\_VALUE [4] = 7

M-Funktion der 6. Gruppe: \$AC\_AUXFU\_M\_VALUE [5] = 9

### 7.13.3 Zeitstempel der aktiven M-Hilfsfunktion

Bei der Ausgabe aufgesammlter Hilfsfunktionen nach Satzsuchlauf muss die Reihenfolge beim Aufsammeln beachtet werden. Es wird deshalb zu jeder Gruppe ein Zeitstempel mitgeführt, der über die folgende Systemvariable gruppenspezifisch abgefragt werden kann:

`$AC_AUXFU_M_TICK[<n>]` (Zeitstempel der aktiven M-Hilfsfunktion)

### 7.13.4 Ermittlung der Ausgabe-Reihenfolge

#### Funktion

Um die Ermittlung der Ausgabe-Reihenfolge für M-Hilfsfunktionen für den Programmierer einfach zu machen, wird folgende vordefinierte Prozedur zur Verfügung gestellt:

```
AUXFUMSEQ(VAR INT _NUM_IN, VAR INT _M_IN[], VAR INT _EXT_IN[], VAR
INT _NUM_OUT, VAR INT _M_OUT[], VAR INT _EXT_OUT[])
```

#### Eingangsparameter:

<code>VAR INT _NUM_IN:</code>	Anzahl relevanter M-Befehle
<code>VAR INT _M_IN[]:</code>	Feld von relevanten M-Codes
<code>VAR INT _EXT_IN[]:</code>	Feld von relevanten M-Adresserweiterungen

#### Ausgangsparameter:

<code>VAR INT _NUM_OUT:</code>	Anzahl ermittelter M-Codes
<code>VAR INT _M_OUT[]:</code>	Feld von ermittelten M-Codes
<code>VAR INT _EXT_OUT[]:</code>	Feld von ermittelten M-Adresserweiterungen

Die Funktion ermittelt für die vorgegebenen M-Codes die Ausgabe-Reihenfolge der gruppenspezifisch aufgesammelten M-Hilfsfunktionen. Die Reihenfolge ergibt sich aus den Aufsammelzeitpunkten `$AC_AUXFU_M_TICK[<n>]` (siehe Kapitel "Zeitstempel der aktiven M-Hilfsfunktion (Seite 451)").

Ein bestimmter M-Code wird immer nur einmal berücksichtigt, auch wenn er mehreren Gruppen angehört. Ist die Anzahl der relevanten M-Befehle kleiner oder gleich 0, so werden alle aufgesammelten M-Codes ausgegeben. Die Anzahl der relevanten M-Befehle ist auf 64 begrenzt.

**Beispiel**

M-Befehle zur Kühlmittelausgabe:

- M7: Kühlmittel 2 EIN
- M8: Kühlmittel 1 EIN
- M9: Kühlmittel 1 und 2 AUS

Gruppenzuordnung:

- Gruppe 5: M7, M9
- Gruppe 6: M8, M9

Teileprogramm (Ausschnitt):

```
Programmcode
...
N10 ... M8
N20 ... M9
N30 ... M7
...
```

Bei Satzsuchlauf werden die Hilfsfunktionen gruppenspezifisch aufgesammelt. Die jeweils letzte Hilfsfunktion einer Hilfsfunktionsgruppe wird nach Satzsuchlauf an die PLC ausgegeben:

- Gruppe 5: M7
- Gruppe 6: M9

Werden diese in der Reihenfolge M7 → M9 ausgegeben, so ist anschließend kein Kühlmittel aktiv. Im Programmablauf wäre jedoch Kühlmittel 2 aktiv. Die korrekte Ausgabe-Reihenfolge für die M-Hilfsfunktionen wird deshalb mit einem ASUP ermittelt, dass die vordefinierte Prozedur `AUXFUMSEQ(...)` enthält:

```
Programmcode
DEF INT _I, _M_IN[3], _EXT_IN[3], _NUM_OUT, _M_OUT[2], _EXT_OUT[2]
_M_IN[0]=7 _EXT_IN[0]=0
_M_IN[1]=8 _EXT_IN[1]=0
_M_IN[2]=9 _EXT_IN[2]=0
AUXFUMSEQ(3, _M_IN, _EXT_IN, _NUM_OUT, _M_OUT, _EXT_OUT)
FOR _I = 0 TO _NUM_OUT-1
    M[_EXT_OUT[_I]] = _M_OUT[_I]
ENDFOR
```

## 7.13.5 Ausgabeunterdrückung von Spindel-spezifischen Hilfsfunktionen

### Funktion

In Verbindung mit bestimmten Situationen, z. B. einem Werkzeugwechsel, kann es erforderlich sein, die bei Satzsuchlauf aufgesammelten spindelspezifischen Hilfsfunktionen nicht in den Aktionssätzen, sondern erst zu einem späteren Zeitpunkt, z. B. nach einem Werkzeugwechsel, auszugeben. Die automatische Ausgabe der spindelspezifischen Hilfsfunktionen nach Satzsuchlauf kann dazu unterdrückt werden. Die Ausgabe kann dann zu einem späteren Zeitpunkt manuell durch Überspeichern oder durch ein ASUP erfolgen.

### Parametrierung

Die Unterdrückung der automatischen Ausgabe der spindelspezifischen Hilfsfunktionen nach Satzsuchlauf wird eingestellt über das Maschinendatum:

MD11450 \$MN\_SEARCH\_RUN\_MODE (Verhalten nach Satzsuchlauf)

Bit	Wert	Bedeutung
2	0	Die Ausgabe der spindelspezifischen Hilfsfunktionen erfolgt in den Aktionssätzen
	1	Die Ausgabe der Hilfsfunktionen in den Aktionssätzen wird unterdrückt.

### Systemvariablen

Die spindelspezifischen Hilfsfunktionen werden bei Satzsuchlauf, unabhängig von der oben genannten Parametrierung, immer in den folgenden Systemvariablen gespeichert:

Systemvariable	Beschreibung
\$P_SEARCH_S [<n> ]	Aufgesammelte Spindeldrehzahl
	Wertebereich: 0 ... Smax
\$P_SEARCH_SDIR [<n>]	Aufgesammelte Spindeldrehrichtung
	Wertebereich: 3, 4, 5, -5, -19, 70
\$P_SEARCH_SGEAR [<n>]	Aufgesammelte Spindelgetriebebestufe-M-Funktion
	Wertebereich: 40 ... 45
\$P_SEARCH_SPOS [<n>]	Aufgesammelte Spindelposition
	Wertebereich: 0 ... MD30330 \$MA_MODULO_RANGE (Größe des Modulbereichs)
	bzw.
	Aufgesammelter Verfahrenweg
	Wertebereich: -100.000.000 ... 100.000.000
\$P_SEARCH_SPOSMODE [<n>]	Aufgesammelter Positionsanfahrmodus
	Wertebereich: 0 ... 5

Zur späteren Ausgabe der spindelspezifischen Hilfsfunktionen können die Systemvariablen z. B. in einem ASUP gelesen und nach Ausgabe der Aktionssätze ausgegeben werden:

DB21, ... DBX32.6 = 1 (Letzter Aktionssatz aktiv)

---

**Hinweis**

Die Inhalte der Systemvariablen \$P\_S, \$P\_DIR und \$P\_SGEAR können nach Satzsuchlauf durch Synchronisationsvorgänge verloren gehen.

---

Für weiterführende Informationen zu ASUP, Satzsuchlauf und Aktionssätzen siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)".

**Beispiel**

Satzsuchlauf auf Kontur mit Unterdrückung der Ausgabe der spindelspezifischen Hilfsfunktionen und Start eines ASUP nach der Ausgabe der Aktionssätze.

Parametrierung: MD11450 \$MN\_SEARCH\_RUN\_MODE, Bit 2 = 1

Nach dem Satzsuchlauf auf N55 wird das ASUP gestartet.

Teileprogramm:

Programmcode	Kommentar
N05 M3 S200	; Spindel 1
N10 G4 F3	
N15 SPOS=111	; Spindel 1 wird im ASUP auf 111 Grad positioniert
N20 M2=4 S2=300	; Spindel 2
N25 G4 F3	
N30 SPOS[2]=IC(77)	; Spindel 2 verfährt inkrementell um 77 Grad
<b>N55 X10 G0</b>	; Zielsatz
N60 G4 F10	
N99 M30	

ASUP:

Programmcode	Kommentar
PROC ASUP_SAVE	
MSG ("Ausgabe der Spindelfunktionen")	
DEF INT SNR=1	
AUSG_SPI:	
M[SNR]=\$P_SEARCH_SGEAR[SNR]	; Getriebestufe ausgeben.
S[SNR]=\$P_SEARCH_S[SNR]	; Drehzahl ausgeben (bei M40 wird passende Getriebestufe ermittelt).
M[SNR]=\$P_SEARCH_SDIR[SNR]	; Drehrichtung, Positionierung oder Achsbetrieb ausgeben.
SNR=SNR+1	; Nächste Spindel.
REPEAT AUSG_SPI P=\$P_NUM_SPINDLES-1	; Für alle Spindeln.
MSG (" ")	
REPOSA	
RET	

### Erläuterungen zum Beispiel

Ist die Anzahl der Spindeln bekannt, können zur Reduzierung der Programmbearbeitungszeit gleichartige Ausgaben in einem Teileprogrammsatz geschrieben werden.

Die Ausgabe von \$P\_SEARCH\_SDIR sollte in einem separaten Teileprogrammsatz erfolgen, da die Spindelpositionierung bzw. die Umschaltung in den Achsbetrieb zusammen mit dem Getriebestufenwechsel zu einer Alarmmeldung führen kann.

Wird das gestartete ASUP mit REPOSA abgeschlossen, bleibt Spindel 1 auf der Position 111 Grad stehen, während Spindel 2 auf die Position 77 Grad repositioniert wird.

Ist ein anderes Verhalten erforderlich, muss die Programmsequenz von z. B. "N05 M3 S..." und "N30 SPOS[2]=IC(...)" für Satzsuchlauf besonders behandelt werden.

Ob Satzsuchlauf aktiv ist, kann im ASUP über die Systemvariablen \$P\_SEARCH ermittelt werden:

```
$P_SEARCH==1 ; Satzsuchlauf aktiv
```

Bei einer inkrementellen Positionierung nach Drehzahlsteuerbetrieb ist zwar der zu verfahrenende Weg definiert, jedoch ergibt sich die erreichte Endposition in einigen Fällen erst während des Positioniervorganges. Dies ist z. B. beim Positionsabgleich während des Überfahrens der Nullmarke oder beim Einschalten der Lageregelung der Fall. Aus diesem Grund wird als REPOS-Position (REPOSA im ASUP) der programmierte Weg ab der Position Null angenommen.

## Randbedingungen

### Aufgesammelte S-Werte

Die Bedeutung eines S-Werts im Teileprogramm ist abhängig vom aktuell aktiven Vorschubtyp:

G93, G94, G95, G97, G971:

S-Wert wird als Drehzahl interpretiert

G96, G961:

S-Wert wird als konstante Schnittgeschwindigkeit interpretiert

Wird die Vorschubart (z. B. für einen Werkzeugwechsel) vor der Ausgabe der Systemvariablen \$P\_SEARCH\_S geändert, muss, um zu vermeiden, dass ein falscher Vorschubtyp zugrunde gelegt wird, die Vorschubart wieder auf die ursprüngliche Einstellung aus dem Zielsatz des Teileprogramms restauriert werden.

### Aufgesammelte Drehrichtung

Bei der Ausgabe der Drehrichtung wird die Systemvariable \$P\_SEARCH\_SDIR zum Zeitpunkt des Satzsuchlauf-Starts mit dem Wert "-5" vorbesetzt. Dieser Wert ist bei der Ausgabe wirkungslos.

Damit wird sichergestellt, dass bei Satzsuchlauf über Bereiche, in denen Spindeln nicht mit einer Drehrichtung, Positionierung oder Achsbetrieb programmiert werden, der letzte Spindelbetriebsmode erhalten bleibt.

Die Programmierung von M19, SPOS und SPOSA wird als "M-19" (interner M19) in der Systemvariablen \$P\_SEARCH\_SDIR alternativ zu M3, M4, M5 und M70 aufgesammelt.

Bei der Ausgabe von "M-19" werden die Positionierdaten intern aus der Systemvariablen \$P\_SEARCH\_SPOS und \$P\_SEACH\_SPOSMODE gelesen. Beide Systemvariable sind auch beschreibbar, um z. B. Korrekturen vornehmen zu können.

---

### Hinweis

Die Werte "-5" und "19" bleiben dem Anwender aufgrund der oben genannten Zuweisungen (z. B. M[<n>] = \$P\_SEARCH\_SDIR[<n>]) prinzipiell verborgen und müssen nur bei spezieller Auswertung der Systemvariablen im ASUP beachtet werden.

---



## 7.13.6 Hilfsfunktionsausgabe bei Satzsuchlauf Typ 5 (SERUPRO)

### Ausgabeverhalten

Beim Satzsuchlauf Typ 5 (SERUPRO) kann eine Hilfsfunktion während des Satzsuchlaufs an die PLC ausgegeben und/oder in den folgenden Systemvariablen gruppenspezifisch aufgesammelt werden:

- \$AC\_AUXFU\_PREDEF\_INDEX[<n>] (Index einer vordefinierten Hilfsfunktion)
- \$AC\_AUXFU\_TYPE[<n>] (Typ der Hilfsfunktion)
- \$AC\_AUXFU\_STATE[<n>] (Ausgabestatus der Hilfsfunktion)
- \$AC\_AUXFU\_EXT[<n>] (Adresserweiterung der Hilfsfunktion)
- \$AC\_AUXFU\_VALUE[<n>] (Wert der Hilfsfunktion)

Zur Beschreibung der Systemvariablen siehe Kapitel "Abfrage von Systemvariablen (Seite 473)".

### Ausgabesteuerung

Ob eine Hilfsfunktion während Satzsuchlauf Typ 5 (SERUPRO) an die PLC ausgegeben und/oder in den genannten Systemvariablen gruppenspezifisch aufgesammelt wird, lässt sich projektieren über Bit 9 und 10 der Maschinendaten:

- MD22080 \$MC\_AUXFU\_PREDEF\_SPEC[<n>]  
(Ausgabeverhalten von vordefinierten Hilfsfunktionen)  
mit <n> = Systemfunktionsindex (0 ... 32)
- MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC[<n>]  
(Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)  
mit <n> = Hilfsfunktionsindex (0 ... 254)
- MD11110 \$MN\_AUXFU\_GROUP\_SPEC[<n>]  
(Ausgabeverhalten der Hilfsfunktionen einer Gruppe)  
mit <n> = Gruppenindex (0 ... 63)

Bit	Wert	Bedeutung
9	0	Keine Aufsammlung während Satzsuchlauf Typ 5 (SERUPRO)
	1	Aufsammlung während Satzsuchlauf Typ 5 (SERUPRO)
10	0	Ausgabe während Satzsuchlauf Typ 5 (SERUPRO)
	1	Keine Ausgabe während Satzsuchlauf Typ 5 (SERUPRO)

### Ausgabebezahlter

Der Anwender kann die aufgesammelten Hilfsfunktionen kanalweise im Satzsuchlauf-ASUP an die PLC ausgeben. Zum Zwecke der serialisierten Ausgabe über mehrere Kanäle werden die drei Ausgabebezahlter bei jeder Ausgabe einer Hilfsfunktion über alle Kanäle verändert:

\$AC\_AUXFU\_TICK[<n>,<m>] (Ausgabebezahlter der aktiven Hilfsfunktion)

<n>: Gruppenindex (0 ... 63)

<m>: Ausgabebezahlter (0 ... 2)

0: Ausgabesequenzbezahlter (Alle Ausgaben innerhalb eines IPO-Takts)

1: Paket-Zähler innerhalb einer Ausgabesequenz im IPO-Takt

2: Hilfsfunktionsbezahlter innerhalb eines Pakets

Erläuterung:

Ein Hilfsfunktionspaket besteht aus max. 10 Hilfsfunktionen.

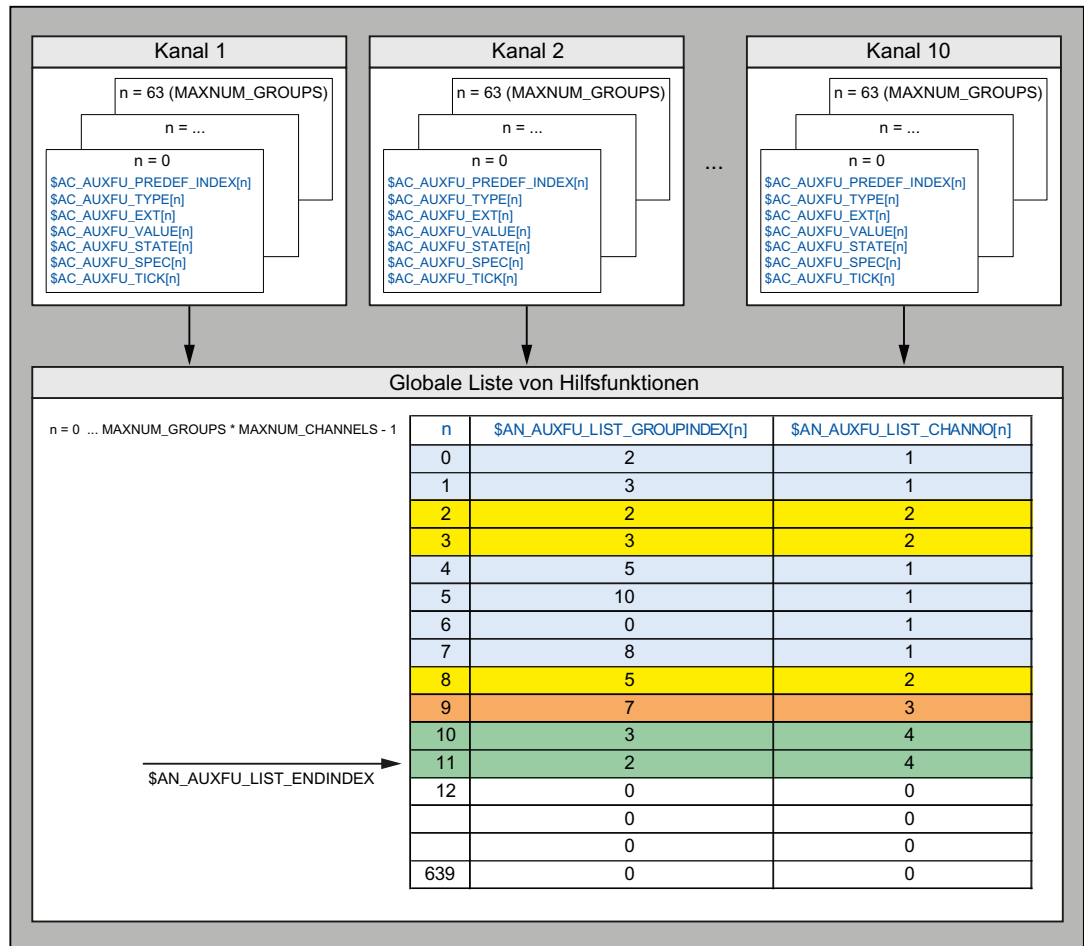
Pro Kanal können während SERUPRO zwei Pakete pro IPO-Takt abgearbeitet werden, da Synchronaktionen in diesem Takt bearbeitet werden.

Innerhalb eines IPO-Takts kann über alle Kanäle eine Ausgabesequenz von bis zu max. 20 Paketen (2 Pakete pro Kanal \* 10 Kanäle) abgearbeitet werden.

Über diese Kodierung ist festgelegt, wie viele Hilfsfunktionspakete und wie viele Hilfsfunktionen darin im gleichen IPO-Takt abgearbeitet wurden: Hilfsfunktionen, die in einem IPO-Takt aufgesammelt wurden, haben alle den gleichen Sequenzbezahlter. Hilfsfunktionen, die in einem Paket (Satz oder Synchronaktion) aufgesammelt wurden, haben alle den gleichen Paketbezahlter. Der Hilfsfunktionsbezahlter wird bei jeder aufgesammelten Hilfsfunktion inkrementiert.

### Globale Liste von Hilfsfunktionen

Am Ende von SERUPRO werden die in den einzelnen Kanälen gruppenspezifisch aufgesammelten Hilfsfunktionen entsprechend ihrem Zählerstand (\$AC\_AUXFU\_TICK[<n>,<m>]) in eine kanalübergreifende (globale) Liste mit Kanalnummer (\$AN\_AUXFU\_LIST\_CHANNO[<n>]) und Gruppenindex (\$AN\_AUXFU\_LIST\_GROUPINDEX[<n>]) eingetragen.



Die beiden Feldvariablen \$AN\_AUXFU\_LIST\_CHANNO[<n>] und \$AN\_AUXFU\_LIST\_GROUPINDEX[<n>] der globalen Liste sind schreib- und lesbar. Der Index <n> hat folgenden Wertebereich: 0 ... MAXNUM\_GROUPS \* MAXNUM\_CHANNELS - 1

Die globale Liste wird aufgebaut, nachdem das Suchziel gefunden wurde. Sie soll als Systemvorschlag für im nachfolgenden SERUPRO-Ende-ASUP auszugebende Hilfsfunktionen dienen. Soll eine Hilfsfunktion nicht ausgegeben werden, so ist der entsprechende Gruppenindex auf "0" zu setzen.

### Verhalten bezüglich Spindel-Hilfsfunktionen

Nach Suchlaufstart sammeln alle Kanäle die Hilfsfunktionen gruppenspezifisch in den Kanal-Variablen auf. Zur weitgehenden Restaurierung des Spindelzustandes im SERUPRO-Zielsatz durch aufgesammelte Hilfsfunktionen ist es notwendig, dass die letzte aktive Hilfsfunktion einer jeden Spindel-Hilfsfunktionsgruppe den Zustand der Spindel im Zielsatz repräsentiert. Bei Spindelzustandsübergängen werden dabei obsoleete Hilfsfunktionen gelöscht oder, wenn notwendig, implizite Hilfsfunktionen eingetragen.

Alle Spindel-Hilfsfunktionen der globalen Hilfsfunktionsliste müssen im Zielsatz den erreichten Spindelzuständen entsprechen, damit die Hilfsfunktionen bei Ausgabe der Liste abarbeitungsfähig sind und keine Alarmer oder ungewollte Spindelzustände angefordert werden, die eine Fortsetzung des Teileprogrammes verhindern können.

Betroffen davon sind die Hilfsfunktionsgruppen einer jeden im System projektierten Spindel, wobei die Spindelnummer der Adresserweiterung einer Hilfsfunktion entspricht.

Gruppe a: M3, M4, M5, M19, M70

Gruppe b: M40, M41, M42, M43, M44, M45

Gruppe c: S

#### Löschen obsoleter Hilfsfunktionen

Bei folgenden Funktionen werden für die betroffene Spindel die Hilfsfunktionen der Gruppe a gelöscht:

- Für die Folgespindel bei Einschalten einer generischen Kopplung wie COUPON, TRAILON, EGON, ...

#### Erzeugen von impliziten Hilfsfunktionen der Gruppe a

Bei folgenden Funktionen werden für die betroffene Spindel Hilfsfunktionen der Gruppe a implizit erzeugt:

- Für die Folgespindel beim Ausschalten der Synchronspindelkopplung
  - COUPOF erzeugt im Hauptlauf je nach Kopplungssituation M3, M4 und S oder M5.
  - COUPOF(S<n>, S<m>, POS) und COUPOFS(S<n>, S<m>, POS, POS) erzeugen M3, M4 und S.
  - COUPOFS erzeugt im Hauptlauf M5.
  - COUPOFS(S<n>, S<m>, POS) erzeugt im Hauptlauf M19.

Das implizite M19 (im ASUP "SPOS[<Adresserweiterung>] = IC(0)") schaltet den Positionierbetrieb ein ohne eine Verfahrbewegung.

- Bei einer Verfahrbewegung als Achse oder beim Übergang in den Achsbetrieb durch Anwahl einer Transformation, bei der die Spindel als Achse eingeht, wird M70 erzeugt.
- Bei SPCOF wird M5 erzeugt.

**Hinweis**

Die Hilfsfunktionen zur Spindelprogrammierung müssen in Zusammenhang mit den Funktionen "Achstausch" und "Achscontainer-Drehung" immer passend zum tatsächlichen Zustand (Motor) bei Tausch/Drehung mitgegeben werden. Dabei unterscheiden sich die Mechanismen von Achstausch und Achscontainer.

Beispiel für Achscontainer-Drehung:

Ein Achscontainer enthält 4 Spindeln. Jede Spindel ist jeweils einem Kanal (1 - 4) zugewiesen. Im Kanal 1 wird immer M3 S1000 programmiert und dann eine Achscontainer-Drehung ausgeführt. Die anderen Kanäle führen keine Spindelprogrammierung aus. Nach der 3. Achscontainer-Drehung und der 4. Spindelprogrammierung M3 drehen alle 4 Spindeln mit 1000 U/min rechts herum. Liegt das SERUPRO-Ende nun in diesem Bereich, so wird erwartet, dass jedes ASUP eines Kanals für die dort befindliche Spindel ein M3 S1000 enthält.

Beim Achstausch hingegen müssen die aufgesammelten Hilfsfunktionen nur dem Kanal zugeordnet werden, in dem sich die Spindel aktuell befindet.

---

**Kanalübergreifende Hilfsfunktion**

Eine Hilfsfunktion kann bei Satzsuchlauf Typ 5 (SERUPRO) auch kanalübergreifend in der globalen Hilfsfunktionsliste aufgesammelt werden. Es wird dann nur die letzte (höchster Zählerstand) aufgesammelte Hilfsfunktion dieser Gruppe in die globale Liste eingetragen.

Die Projektierung erfolgt über Bit 11 in den Maschinendaten:

- MD22080 \$MC\_AUXFU\_PREDEF\_SPEC[<n>]  
(Ausgabeverhalten von vordefinierten Hilfsfunktionen)  
mit <n> = Systemfunktionsindex (0 ... 32)
- MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC[<n>]  
(Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)  
mit <n> = Hilfsfunktionsindex (0 ... 254)
- MD11110 \$MN\_AUXFU\_GROUP\_SPEC[<n>]  
(Ausgabeverhalten der Hilfsfunktionen einer Gruppe)  
mit <n> = Gruppenindex (0 ... 63)

Bit	Wert	Bedeutung
11	0	Aufsammlung kanalspezifisch
	1	Aufsammlung kanalübergreifend

Die Spindel-Hilfsfunktionen werden vorab je nach Spindelzustand am Ende des Suchlaufs herausgefiltert. Die Kanaldaten werden entsprechend aktualisiert. Die globale Hilfsfunktionsliste kann in den SERUPRO-Ende-ASUPs der Reihe nach abgearbeitet und die sortierten Hilfsfunktionen kanalsynchron ausgegeben werden.

### Abfrage der letzten aufgesammelten Hilfsfunktion

Der Index der letzten aufgesammelten Hilfsfunktion in der globalen Liste kann mit der Systemvariablen \$AN\_AUXFU\_LIST\_ENDINDEX abgefragt werden.

## 7.13.7 SERUPRO-Ende-ASUP

### Funktion

Nach Abschluss des Satzsuchlaufs mit Programmtest (SERUPRO) müssen vor dem Start der nachfolgenden Bearbeitung die während des Suchlaufs aufgesammelten Hilfsfunktionen ausgegeben werden. Während des Satzsuchlaufs werden dazu die Hilfsfunktionen in einer globalen Liste aufgesammelten. Aus dieser Liste generieren kanalspezifisch die SERUPRO-Ende-ASUPs die entsprechenden Teileprogrammsätze. Damit wird sicher gestellt, dass die aufgesammelten Hilfsfunktionen sowohl kanalspezifisch als auch kanalübergreifend in zeitlich korrekter Reihenfolge ausgegeben werden. Ein vollständiges funktionsfähiges SERUPRO-Ende-ASUP ist Bestandteil der NCK-Software.

Das SERUPRO-Ende-ASUP kann vom Anwender/Maschinenhersteller verändert werden. Die nachfolgend beschriebenen Funktionen unterstützen die Bearbeitung der globalen Liste der Hilfsfunktionen und die Generierung der für die synchronisierte Hilfsfunktionsausgabe erforderlichen Teileprogrammsätze.

### Funktion AUXFUSYNC(...)

#### Funktion:

Die Funktion `AUXFUSYNC` generiert aus der globalen Liste der Hilfsfunktionen pro Aufruf einen vollständigen Teileprogrammsatz als String. Der Teileprogrammsatz enthält entweder Hilfsfunktionen oder Befehle zur Synchronisation der Hilfsfunktionsausgaben (`WAITM`, `G4`, etc.).

Die Funktion löst Vorlaufstopp aus.

#### Syntax:

```
PROC AUXFUSYNC (VAR INT <NUM>, VAR INT <GROUPINDEX> [10], VAR  
STRING [400] <ASSEMBLED>)
```

**Parameter:**

<NUM>:	Enthält Informationen über den im Parameter <ASSEMBLED> gelieferten Teileprogrammsatz bzw. den darin enthaltenen Hilfsfunktionen. Wertebereich: -1, 0, 1 ... 10
	<b>Wert Bedeutung</b>
	≥1 Anzahl der im Teileprogrammsatz enthaltenen Hilfsfunktionen
	0 Teileprogrammsatz ohne Hilfsfunktionen, z.B. WAITM, G4
	-1 Ende-Kennung. Die globale Liste der Hilfsfunktionen ist für den aktuellen Kanal vollständig bearbeitet.
<GROUPINDEX>:	Enthält die Indizes der im Teileprogrammsatz enthaltenen Hilfsfunktionsgruppen. Mit Index = Nummer der Hilfsfunktionsgruppe - 1
<ASSEMBLED>:	Enthält den vollständigen Teileprogrammsatz für das kanalspezifische SERUPRO-Ende-ASUP als String.

**Weitere Informationen:**

Wurden Hilfsfunktionen über eine Synchronaktion aufgesammelt, werden zwei NC-Sätze erzeugt. Ein NC-Satz zur Ausgabe der Hilfsfunktionen. Ein ausführbarer NC-Satz über den der NC-Satz zur Ausgabe der Hilfsfunktionen zum Hauptlauf transportiert wird:

1. Ausgabe der Hilfsfunktionen über Synchronaktion, z. B.: WHEN TRUE DO M100 M102
2. Ausführbaren NC-Satz, z. B.: G4 F0.001

**Funktion AUXFUDEL(...)****Funktion:**

Die Funktion AUXFUDEL löscht kanalspezifisch für den aufrufenden Kanal die angegebene Hilfsfunktion aus der globalen Liste der Hilfsfunktionen. Das Löschen erfolgt durch Setzen des entsprechenden Gruppenindex ...GROUPINDEX [n] auf 0.

Der Aufruf der Funktion muss vor dem Aufruf von AUXFUSYNC erfolgen.

Die Funktion löst Vorlaufstopp aus.

**Syntax:**

```
PROC AUXFUDEL (CHAR <TYPE>, INT <EXTENSION>, REAL <VALUE>, INT <GROUP>)
```

**Parameter:**

<TYPE>:	Typ der zu löschenden Hilfsfunktion
<EXTENSION>:	Adresserweiterung der zu löschenden Hilfsfunktion
<VALUE>:	Wert der zu löschenden Hilfsfunktion
<GROUP>:	Nummer der Hilfsfunktionsgruppe

### Funktion AUXFUDELG(...)

**Funktion:**

Die Funktion AUXFUDELG löscht kanalspezifisch für den aufrufenden Kanal alle Hilfsfunktionen der angegebenen Hilfsfunktionsgruppe aus der globalen Liste der Hilfsfunktionen. Das Löschen erfolgt durch Setzen des entsprechenden Gruppenindex ...GROUPINDEX [n] auf 0.

Der Aufruf der Funktion muss vor dem Aufruf von AUXFUSYNC erfolgen.

Die Funktion löst Vorlaufstopp aus.

**Syntax:**

PROC AUXFUDELG (INT <GROUP>)

**Parameter:**

<GROUP>:                    Nummer der Hilfsfunktionsgruppe

### Mehrkanaliger Satzsuchlauf

<b>VORSICHT</b>
<b>Mehrkanaliger Satzsuchlauf und AUXFUDEL / AUXFUDELG</b>
Werden bei einem mehrkanaligen Satzsuchlauf in den SERUPRO-Ende-ASUPs Hilfsfunktionen mit AUXFUDEL / AUXFUDELG aus der globalen Liste der Hilfsfunktionen gelöscht, muss vor dem Aufruf der Funktion AUXFUSYNC, eine Synchronisation der beteiligten Kanäle erfolgen. Durch die Synchronisation ist dann sicher gestellt, dass vor dem Aufruf der Funktion AUXFUSYNC alle Löschaufträge bearbeitet wurden und eine konsistente Liste vorliegt.



## Beispiele

Zwei Beispiele für den Aufbau eines anwenderspezifischen SERUPRO-Ende ASUPs.

### Beispiel 1: Löschen von Hilfsfunktionen und Erzeugen der Hilfsfunktionsausgabe mit AUXFUSYNC(...)

Programmcode	Kommentar
N10 DEF STRING[400] ASSEMBLED=""	
N20 DEF STRING[31] FILENAME="/_N_CST_DIR/_N_AUXFU_SPF"	
N30 DEF INT GROUPINDEX[10]	
N40 DEF INT NUM	
N60 DEF INT ERROR	
N90	
N140 AUXFUDEL("M",2,3,5)	; M2=3 (5. Hilfsfunktionsgruppe) löschen
N150	
N170 AUXFUDELG(6)	; die aufgesammelte Hilfsfunktion der ; 6. Gruppe löschen.
N180	
N190 IF ISFILE(FILENAME)	
N210 DELETE(ERROR,FILENAME)	; Datei FILENAME löschen
N220 IF (ERROR<>0)	; Fehlerauswertung
N230 SETAL(61000+ERROR)	
N240 ENDIF	
N250 ENDIF	
; VORSICHT!	
; Werden bei einem mehrkanaligen Satzsuchlauf Hilfsfunktionen mit AUXFUDEL/AUXFUDELG	
; aus der globalen Liste der Hilfsfunktionen gelöscht, muss vor der Schleife zum	
; Erzeugen des Unterprogramms FILENAME mit AUXFUSYNC eine Synchronisation der Kanäle	
; erfolgen. Durch die Synchronisation wird sicher gestellt, dass alle Löschaufträge	
; in allen Kanälen bearbeitet wurden und eine konsistente Liste vorliegt.	
; Beispiel: WAITM(99,1,2,3)	
N270 LOOP	
N300 AUXFUSYNC(NUM,GROUPINDEX,ASSEMBLED)	; Teileprogrammsatz erzeugen lassen
N310	
N320 IF (NUM== -1)	; Alle Hilfsfunktionen des Kanals ; sind abgearbeitet.
N340 GOTOF LABEL1	
N350 ENDIF	
N380 WRITE(ERROR,FILENAME,ASSEMBLED)	; Teileprogrammsatz in Datei FILENAME schreiben.
N390 IF (ERROR<>0)	; Fehlerauswertung
N400 SETAL(61000+ERROR)	
N410 ENDIF	
N430 ENDLOOP	
N440	
N450 LABEL1:	

Programmcode	Kommentar
N460	
N480 CALL FILENAME	; Erzeugtes Unterprogramm abarbeiten.
N490	
N510 DELETE(ERROR,FILENAME)	; Datei nach Ausführung wieder löschen.
N520 IF (ERROR<>0)	
N530 SETAL(61000+ERROR)	
N540 ENDIF	
N550	
N560 M17	

**Beispiel 2: Löschen von Hilfsfunktionen und Erzeugen der Hilfsfunktionsausgabe ohne AUXFUSYNC(...)**

Programmcode	Kommentar
N0610 DEF STRING[400] ASSEMBLED=""	
N0620 DEF STRING[31] FILENAME="/_N_CST_DIR/_N_AUXFU_SPF"	
N0630 DEF INT GROUPINDEX[10]	
N0640 DEF INT NUM	
N0650 DEF INT LAUF	
N0660 DEF INT ERROR	
N0670 DEF BOOL ISQUICK	
N0680 DEF BOOL ISSYNACT	
N0690 DEF BOOL ISIMPL	
...	
N0760 AUXFUDEL("M",2,3,5)	; M2=3 (5. Hilfsfunktionsgruppe) löschen
N0770	
N0790 AUXFUDELG(6)	; die aufgesammelte Hilfsfunktion der ; 6. Gruppe löschen.
N0800	
N0810 IF ISFILE(FILENAME)	
N0830 DELETE(ERROR,FILENAME)	; Datei ist bereits vorhanden und muss ; gelöscht werden.
N0840 IF (ERROR<>0)	
N0850 SETAL(61000+ERROR)	
N0860 ENDIF	
N0870 ENDIF	
N0880	
; VORSICHT!	
; Werden bei einem mehrkanaligen Satzsuchlauf Hilfsfunktionen mit AUXFUDEL/AUXFUDELG	
; aus der globalen Liste der Hilfsfunktionen gelöscht, muss vor der Schleife zum	
; Erzeugen des Unterprogramms FILENAME mit AUXFUSYNC eine Synchronisation der Kanäle	
; erfolgen. Durch die Synchronisation wird sicher gestellt, dass alle Löschaufträge	
; in allen Kanälen bearbeitet wurden und eine konsistente Liste vorliegt.	

Programmcode	Kommentar
; Beispiel: WAITM(99,1,2,3)	
N0890 LOOP	
N0920 AUXFUSYNC (NUM, GROUPINDEX, ASSEMBLED)	; Prozedur zur Erzeugung von ; Hilfsfunktionssätzen aus der globalen ; Hilfsfunktionsliste.
N0930	
N0940 IF (NUM== -1)	; Alle Hilfsfunktionen des Kanals sind ; abgearbeitet.
N0960 GOTO LABEL1	
N0970 ENDIF	
N0980	
N1000 IF (NUM > 0)	; Wenn Hilfsfunktionen zur Ausgabe kommen ; wird der Satz erzeugt.
N1010	
N1020 ASSEMBLED = ""	
N1030	
N1050 FOR LAUF = 0 TO NUM - 1	; Aufgesammelte Hilfsfunktionen für einen ; Satz.
N1060	
N1080 IF GROUPINDEX[LAUF] <> 0	; Aus der globalen Liste gelöschte ; Hilfsfunktionen haben den Gruppenindex 0.
N1090	
N1100 ISQUICK = \$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND 'H2'	
N1110	
N1120 ISSYNACT = \$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND 'H1000'	
N1130	
N1140 ISIMPL = \$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND 'H2000'	
N1150	
N1180 IF ISSYNACT	; Satz für die M-Hilfsfunktionsausgabe ; zusammenbauen
N1190 ASSEMBLED = ASSEMBLED << "WHEN TRUE DO "	
N1200 ENDIF	
N1210 ; Implizit erzeugtes M19 wird auf SPOS[SPI(<Spindelnr.>)] = IC(0) abgebildet.	
N1230 IF (ISIMPL AND (\$AC_AUXFU_VALUE[GROUPINDEX[LAUF]] == 19))	
N1240 ASSEMBLED = ASSEMBLED << "SPOS[SPI(" << \$AC_AUXFU_EXT[GROUPINDEX[LAUF]] << ") = IC(0) "	
N1260 ELSE	
N1270 ASSEMBLED = ASSEMBLED << "M[" << \$AC_AUXFU_EXT[GROUPINDEX[LAUF]] << "] = "	
N1280	
N1290 IF ISQUICK	
N1300 ASSEMBLED = ASSEMBLED << "QU("	
N1310 ENDIF	
N1320	
N1330 ASSEMBLED = ASSEMBLED << \$AC_AUXFU_VALUE[GROUPINDEX[LAUF]]	

Programmcode	Kommentar
N1340	
N1350 IF ISQUICK	
N1360 ASSEMBLED= ASSEMBLED << " ) "	
N1370 ENDIF	
N1380 ENDIF	
N1400 ENDIF	
N1420 ENDFOR	
N1430	
N1450 WRITE (ERROR, FILENAME, ASSEMBLED)	; Hilfsfunktionssatz in Datei schreiben.
N1460	
N1470 IF ISSYNACT	
N1480 ASSEMBLED="G4 FO.001"	
N1490 WRITE (ERROR, FILENAME, ASSEMBLED)	
N1500 ENDIF	
N1510	
N1520 ELSE	
N1540 WRITE (ERROR, FILENAME, ASSEMBLED)	; Hilfsfunktionssatz in Datei schreiben.
N1550 ENDIF	
N1560	
N1570 ENDLOOP	
N1580	
N1590 LABEL1:	
N1600	
N1620 CALL FILENAME	; Erzeugtes Unterprogramm abarbeiten.
N1630	
N1650 DELETE (ERROR, FILENAME)	; Datei nach Ausführung wieder löschen.
N1660 IF (ERROR<>0)	
N1670 SETAL (61000+ERROR)	
N1680 ENDIF	
N1690	
N1700 M17	

## 7.14 Implizit ausgegebene Hilfsfunktionen

### Funktion

Implizit ausgegebene Hilfsfunktionen sind Hilfsfunktionen, die nicht explizit programmiert wurden und zusätzlich von anderen Systemfunktionen (z. B. Transformationsanwahl, Werkzeuganwahl, etc.) ausgegeben werden. Diese impliziten Hilfsfunktionen führen zu keiner Systemfunktion, sondern die M-Codes werden entsprechend ihres parametrisierten Ausgabeverhaltens aufgesammelt und/oder an die PLC ausgegeben.

### Parametrierung

Die M-Codes für implizit auszugebende Hilfsfunktionen werden festgelegt mit den Maschinendaten:

- MD22530 \$MC\_TOCARR\_CHANGE\_M\_CODE (M-Code bei Werkzeugträgerwechsel)  
Der Betrag dieses Maschinendatums gibt die Nummer des M-Codes an, der bei Aktivierung eines Werkzeugträgers an der NC/PLC-Nahtstelle ausgegeben wird.  
Ist der Wert positiv, wird immer der unveränderte M-Code ausgegeben.  
Ist der Wert negativ, wird die Nummer des Werkzeugträgers zum Betrag des Maschinendatums addiert und diese Nummer ausgegeben.
- MD22532 \$MC\_GEOAX\_CHANGE\_M\_CODE (M-Code bei Umschaltung der Geometrieachsen)  
Nummer des M-Codes, der bei einer Umschaltung der Geometrieachsen an der NC/PLC-Nahtstelle ausgegeben wird.
- MD22534 \$MC\_TRAFO\_CHANGE\_M\_CODE (M-Code bei Transformationswechsel)  
Nummer des M-Codes, der bei einer Transformationsumschaltung der Geometrieachsen an der NC/PLC-Nahtstelle ausgegeben wird.

---

#### Hinweis

Hat die Nummer des auszugebenden M-Codes oder der Betrag von MD22530 / MD22532 / MD22534 selbst einen der Werte 0 bis 6, 17 oder 30, wird kein M-Code ausgegeben. Es wird nicht überwacht, ob ein derart erzeugter M-Code zu Konflikten mit anderen Funktionen führt.

---

### **Ausgabeverhalten**

Bei implizit ausgegebenen Hilfsfunktionen ist das Bit 13 im Maschinendatum MD22080 bzw. MD22035 (Ausgabeverhalten von vordefinierten bzw. anwenderdefinierten Hilfsfunktionen) gesetzt.

Über die Systemvariable \$AC\_AUXFU\_SPEC[<n>] kann dieses Bit abgefragt werden.

### **Implizit ausgegebene Hilfsfunktion M19**

Um eine Durchgängigkeit von M19 und SPOS bzw. SPOSA bezüglich des Verhaltens an der NC/PLC-Nahtstelle zu erreichen, kann bei SPOS und SPOSA die Hilfsfunktion M19 an die NC/PLC-Nahtstelle ausgegeben werden (siehe Kapitel "Allgemeine Funktionalität (Seite 1390)").

Die Implizit ausgegebene Hilfsfunktion M19 wird beim Satzsuchlauf aufgesammelt.

## 7.15 Informationsmöglichkeiten

Informationen zu Hilfsfunktionen (z. B. über den Ausgabestatus) sind möglich über:

- die gruppenspezifische modale M-Hilfsfunktionsanzeige an der Bedienoberfläche.
- die Abfrage von Systemvariablen in Teileprogrammen und Synchronaktionen.

### 7.15.1 Gruppenspezifische modale M-Hilfsfunktionsanzeige

#### Funktion

An der Bedienoberfläche kann der Ausgabe- und Quittungsstatus von M-Hilfsfunktionen gruppenspezifisch angezeigt werden.

#### Voraussetzungen

Vorraussetzung für die Realisierung einer funktionsorientierten Quittierung und Anzeige von M-Hilfsfunktionen ist, dass die Verwaltung der Hilfsfunktionen in der PLC, also im Anwenderprogramm selbst, erfolgen muss. Deshalb obliegt es den PLC-Programmierer, die Quittierung dieser Hilfsfunktionen selbst vorzunehmen. Er muss wissen, welche Hilfsfunktionen in welcher Gruppe sind, die er quittieren muss.

#### Standard

M-Hilfsfunktionen, die nicht über die PLC verwaltet werden, werden vom NCK an die PLC ausgegeben und als "übergeben" gekennzeichnet. Eine funktionale Quittierung gibt es für diese Hilfsfunktionen nicht. Alle nach einem Satzsuchlauf aufgesammelten M-Hilfsfunktionen werden zusätzlich zur Anzeige gebracht, damit der Bediener weiß, welche Hilfsfunktionen nach dem Start nach Satzsuchlauf ausgegeben werden.

#### PLC-Aktivitäten

Bei Hilfsfunktionsgruppen, die von der PLC selbst verwaltet werden, muss das PLC-Anwenderprogramm alle Hilfsfunktionen dieser Gruppe bei **Übernahme** und **Funktionsende** quittieren. Der PLC-Programmierer muss alle Hilfsfunktionen dieser Gruppen kennen.

### Sonstiges

Es werden nur die M-Hilfsfunktionen **gruppenspezifisch** angezeigt. Die satzweise Anzeige bleibt zusätzlich erhalten. Es können bis zu 15 Gruppen angezeigt werden, wobei je Gruppe immer **nur die letzte M-Funktion einer Gruppe**, die entweder aufgesammelt oder an die PLC ausgegeben wurde, angezeigt wird. Die M-Funktionen werden ihrem Status entsprechend in verschiedener Anzeigeart dargestellt:

Status	Anzeigeart
Hilfsfunktion ist aufgesammelt	Invertiert mit gelber Schrift
Hilfsfunktion ist von NCK an PLC ausgegeben	Invertiert
Hilfsfunktion ist von NCK an PLC übergeben worden und Transportquittierung ist erfolgt	Schwarze Schrift auf grauem Hintergrund
Hilfsfunktion wird von PLC verwaltet und ist von PLC direkt übernommen worden.	Schwarze Schrift auf grauem Hintergrund
Hilfsfunktion wird von PLC verwaltet und die Funktionsquittierung ist erfolgt.	Schwarze Schrift auf grauem Hintergrund

### Anzeigeaktualisierung

Die Anzeige wird so organisiert, dass zuerst immer die aufgesammelten Hilfsfunktionen vor den PLC-verwalteten und vor den NCK-verwalteten ausgegeben werden. Eine aufgesammelte Hilfsfunktion bleibt solange als aufgesammelt markiert, bis sie vom NCK an die PLC ausgegeben wurde. PLC-verwaltete Hilfsfunktionen bleiben solange bestehen, bis diese von einer anderen Hilfsfunktion abgelöst werden. Bei Reset werden nur die aufgesammelten und die NCK-verwalteten gelöscht.



## 7.15.2 Abfrage von Systemvariablen

## Funktion

Im Teileprogramm und über Synchronaktionen können Hilfsfunktionen gruppenspezifisch über Systemvariablen abgefragt werden:

\$AC\_AUXFU\_... [<n>] = <Wert>

Systemvariable	Bedeutung
\$AC_AUXFU_PREDEF_INDEX[<n>]	<Wert>: Index der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Suchlauf) oder ausgegebenen vordefinierten Hilfsfunktion
	Typ: INT
	Ist für die spezifizierte Gruppe noch keine Hilfsfunktion ausgegeben worden oder ist die Hilfsfunktion eine anwenderdefinierte Hilfsfunktion, so liefert die Variable den Wert "-1".
	<n>: Gruppenindex (0 ... 63)
<b>Hinweis:</b> Über diese Variable lässt sich eine vordefinierte Hilfsfunktion eindeutig identifizieren.	
\$AC_AUXFU_TYPE[<n>]	<Wert>: Typ der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Suchlauf) oder ausgegebenen Hilfsfunktion
	Typ: CHAR
	<n>: Gruppenindex (0 ... 63)
\$AC_AUXFU_EXT[<n>] bzw. M-Funktionsspezifisch: \$AC_AUXFU_M_EXT[<n>]	<Wert>: Adresserweiterung der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Suchlauf) oder ausgegebenen Hilfsfunktion
	Typ: INT
	<n>: Gruppenindex (0 ... 63)
\$AC_AUXFU_VALUE[<n>] bzw. M-Funktionsspezifisch: \$AC_AUXFU_M_VALUE[<n>]	<Wert>: Wert der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Suchlauf) oder ausgegebenen Hilfsfunktion
	Typ: REAL
	<n>: Gruppenindex (0 ... 63)
\$AC_AUXFU_SPEC[<n>]	Wert: Bit-kodiertes Ausgabeverhalten entsprechend MD22080/MD22035 (oder QU-Programmierung) der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Suchlauf) oder ausgegebenen Hilfsfunktion
	Typ: INT
	<n>: Gruppenindex (0 ... 63)
	<b>Hinweis:</b> Über diese Variable lässt sich ermitteln, ob die Hilfsfunktionsausgabe mit schneller Quittierung erfolgen soll.

Systemvariable	Bedeutung	
\$AC_AUXFU_STATE[<n> bzw. M-Funktionsspezifisch: \$AC_AUXFU_M_STATE[<n>]	<Wert>:	Ausgabestatus der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Suchlauf) oder ausgegebenen Hilfsfunktion
	Typ:	INT
	Wertebereich:	0 ... 5
	0:	Hilfsfunktion ist nicht vorhanden
	1:	M-Hilfsfunktion wurde per Suchlauf aufgesammelt
	2:	M-Hilfsfunktion wurde an die PLC ausgegeben
	3:	M-Hilfsfunktion wurde an die PLC ausgegeben und die Transportquittierung ist erfolgt
	4:	M-Hilfsfunktion wird von der PLC verwaltet und ist von der PLC übernommen worden
5:	M-Hilfsfunktion wird von der PLC verwaltet und die Funktionsquittierung ist erfolgt	
<n>:	Gruppenindex (0 ... 63)	

**Beispiel**

Alle M-Hilfsfunktionen der 1. Gruppe sollen der Reihe ihrer Ausgabe nach abgespeichert werden:

```
id=1 every $AC_AUXFU_M_STATE[0]==2 do
$AC_FIFO[0,0]=$AC_AUXFU_M_VALUE[0]
```

**Literatur**

Weitere Informationen zu den Systemvariablen siehe:

Listenhandbuch Systemvariablen

## 7.16 Randbedingungen

### 7.16.1 Allgemeine Randbedingungen

#### Spindeltausch

Da die Parametrierung der Hilfsfunktionen kanalspezifisch erfolgt, müssen bei Verwendung der Funktion: "Spindeltausch" die Spindel-spezifischen Hilfsfunktionen in allen Kanälen in denen die Spindel verwendet wird, gleich parametrieren werden.

#### Werkzeugverwaltung

Bei aktiver Werkzeugverwaltung gelten folgende Randbedingungen:

- T- und M<k>-Funktionen werden nicht an die PLC ausgegeben.  
Hinweis  
k ist der parametrierte Wert der Hilfsfunktion für den Werkzeugwechsel (Default: 6):  
MD22560 \$MC\_TOOL\_CHANGE\_M\_CODE (Hilfsfunktion für Werkzeugwechsel)
- Wird keine Adresserweiterung programmiert, bezieht sich die Hilfsfunktion auf die Master-Spindel bzw. des Master-Werkzeughalters des Kanals.

Definition der Master-Spindel:

- MD20090 \$MC\_SPIND\_DEF\_MASTER\_SPIND
- Teileprogrammanweisung: SETMS

Definition des Master-Werkzeughalters

- MD20124 \$MC\_TOOL\_MANAGEMENT\_TOOLHOLDER
- Teileprogrammanweisung: SETMTH

#### Maximale Anzahl von Hilfsfunktionen pro Teileprogrammsatz

In einem Teileprogrammsatz dürfen maximal 10 Hilfsfunktionen programmiert werden.

#### DL (Summenkorrektur)

Für die DL-Funktion gelten folgende Einschränkungen:

- Pro Teileprogrammsatz kann nur eine DL-Funktion programmiert werden.
- Bei Verwendung von DL-Funktionen in Synchronaktionen wird der Parameter: "Wert" nicht an die PLC ausgegeben.

## 7.16.2 Ausgabeverhalten

### Gewindeschneiden

Während aktivem Gewindeschneiden  $G_{33}$ ,  $G_{34}$  und  $G_{35}$  wirkt für die Spindel-spezifischen Hilfsfunktionen:

- $M_3$  (Spindel rechts)
- $M_4$  (Spindel links)

immer mit folgendes Ausgabeverhalten:

- Ausgabedauer einen OB40-Takt (Schnelle Quittierung)
- Ausgabe während der Bewegung

Die Spindel-spezifische Hilfsfunktion  $M_5$  (Spindel halt) wird immer am Satzende ausgegeben. Der Teileprogrammsatz, der  $M_5$  beinhaltet, wird immer d. h. auch bei aktivem Bahnsteuerbetrieb, mit Genauhalt beendet.

### Synchronaktionen

Bei der Hilfsfunktionsausgabe aus Synchronaktionen wird das parametrierte Ausgabeverhalten bis auf folgende Parameter ignoriert:

- Bit0: Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
- Bit1: Ausgabedauer einen OB40-Takt (Schnelle Quittierung)

### Hilfsfunktionen: M17 bzw. M2 / M30 (Unterprogrammende)

#### Allein in einem Teileprogrammsatz

Steht eine der Hilfsfunktionen  $M_{17}$ ,  $M_2$  oder  $M_{30}$  allein in einem Teileprogrammsatz und es ist noch eine Achse in Bewegung, erfolgt die Ausgabe der Hilfsfunktion an die PLC erst, nachdem die Achse steht.

#### Überlagerung des parametrierten Ausgabeverhaltens

Das parametrierte Ausgabeverhalten der Hilfsfunktionen  $M_{17}$  bzw.  $M_2/M_{30}$  wird durch das im folgenden Maschinendatum festgelegte Ausgabeverhalten überlagert:

MD20800 \$MC\_SPF\_END\_TO\_VDI, Bit 0 (Unterprogrammende / Halt an PLC)

Bit	Wert	Bedeutung
0	0	Die Hilfsfunktionen $M_{17}$ bzw. $M_2/M_{30}$ (Unterprogrammende) werden nicht an die PLC ausgegeben. Bahnsteuerbetrieb wird am Unterprogrammende nicht unterbrochen.
	1	Die Hilfsfunktionen $M_{17}$ bzw. $M_2/M_{30}$ (Unterprogrammende) werden an die PLC ausgegeben.

**Hilfsfunktion: M1 (Bedingter Halt)****Überlagerung des parametrisierten Ausgabeverhaltens**

Das parametrisierte Ausgabeverhalten der Hilfsfunktion  $M_1$  wird durch das im folgenden Maschinendatum festgelegte Ausgabeverhalten überlagert:

MD20800 \$MC\_SPF\_END\_TO\_VDI, Bit 1 (Unterprogrammende / Halt an PLC)

Bit	Wert	Bedeutung
1	0	Die Hilfsfunktionen $M_{01}$ (Bedingter Halt) wird immer an PLC ausgegeben. Eine schnelle Quittierung ist unwirksam, da $M_{01}$ fest der 1. Hilfsfunktionsgruppe zugeordnet ist und damit immer am Satzende ausgegeben wird.
	1	Die Hilfsfunktionen $M_{01}$ (Bedingter Halt) wird nur dann an PLC ausgegeben, wenn die Funktion: "Programmierter Halt" über die HMI-Bedienoberfläche aktiviert ist. Bei schneller Quittierung wird die $M_1$ während der Bewegung an die PLC ausgegeben. Solange die Funktion nicht aktiv ist, wird dadurch ein Bahnsteuerbetrieb nicht unterbrochen.

**Teileprogrammsätze ohne Verfahrbewegung**

In einem Teileprogrammsatz ohne Verfahrbewegung werden alle Hilfsfunktionen, unabhängig von ihrem jeweiligen parametrisierten Ausgabeverhalten, sofort in einem Block ausgegeben.

**Spindelspezifische Hilfsfunktionsausgabe nur als Information für das PLC-Anwenderprogramm**

In bestimmten Steuerungssituationen, z.B. zum Abschluss eines Satzsuchlaufs, werden die aufgesammelten spindelspezifischen Hilfsfunktionen (z.B.  $M_3$ ,  $M_4$ ,  $M_5$ ,  $M_{19}$ ,  $M_{40} \dots M_{45}$ ,  $M_{70}$ ) nur zur Information für das PLC-Anwenderprogramm an die NC/PLC-Nahtstelle ausgegeben. Die Steuerung erzeugt dazu einen Teileprogrammsatz (Aktionssatz) in welchem die aufgesammelten Hilfsfunktionen mit **negativer** Adresserweiterung eingetragen sind. Die entsprechenden Systemfunktionen werden dann **nicht** ausgeführt.

Beispiel:  $M(-2) = 41$  Anforderung Getriebestufenwechsel für die 2. Spindel

## 7.17 Beispiele

### 7.17.1 Erweiterung von vordefinierten Hilfsfunktionen

#### Aufgabe

Parametrierung der Hilfsfunktionen M3, M4 und M5 für die 2. Spindel des Kanals.

#### Parametrierung: M3

##### Anforderungen:

- Maschinendaten-Index: 0 (erste anwenderdefinierte Hilfsfunktion)
- Hilfsfunktionsgruppe: 5
- Typ und Wert: M3 (Spindel rechts)
- Adresserweiterung: 2 entsprechend der 2. Spindel des Kanals
- Ausgabeverhalten:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
  - Ausgabe vor der Bewegung

##### Parametrierung:

```
MD22000 $MC_AUXFU_ASSIGN_GROUP [ 0 ]      = 5
MD22010 $MC_AUXFU_ASSIGN_TYPE [ 0 ]       = "M"
MD22020 $MC_AUXFU_ASSIGN_EXTENSION [ 0 ]  = 2
MD22030 $MC_AUXFU_ASSIGN_VALUE [ 0 ]     = 3
MD22035 $MC_AUXFU_ASSIGN_SPEC [ 0 ]      = 'H21'
```

#### Parametrierung: M4

##### Anforderungen:

- Maschinendaten-Index: 1 (zweite anwenderdefinierte Hilfsfunktion)
- Hilfsfunktionsgruppe: 5
- Typ und Wert: M4 (Spindel links)
- Adresserweiterung: 2 entsprechend der 2. Spindel des Kanals
- Ausgabeverhalten:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
  - Spindelreaktion nach der Quittierung
  - Ausgabe während der Bewegung

**Parametrierung:**

MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP [ 1 ] = 5  
MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE [ 1 ] = "M"  
MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION [ 1 ] = 2  
MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE [ 1 ] = 4  
MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC [ 1 ] = 'H51'

**Parametrierung: M5****Anforderungen:**

- Maschinendaten-Index: 2 (dritte anwenderdefinierte Hilfsfunktion)
- Hilfsfunktionsgruppe: 5
- Typ und Wert: M5 (Spindel halt)
- Adresserweiterung: 2 entsprechend der 2. Spindel des Kanals
- Ausgabeverhalten:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
  - Spindelreaktion nach der Quittierung
  - Ausgabe am Satzende

**Parametrierung:**

MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP [ 2 ] = 5  
MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE [ 2 ] = "M"  
MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION [ 2 ] = 2  
MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE [ 2 ] = 5  
MD22035 \$MC\_AUXFU\_ASSIGN\_SPEC [ 2 ] = 'H91'

## 7.17.2 Definition von Hilfsfunktionen

### Aufgabe

Parametrierung der Hilfsfunktionsspezifischen Maschinendaten für eine Maschine mit folgender Konfiguration:

#### Spindeln

- Spindel 1: Masterspindel
- Spindel 2: Zweite Spindel

#### Getriebestufen

- Spindel 1: 5 Getriebestufen
- Spindel 2: keine Getriebestufen

#### Schaltfunktionen für Kühlwasser Ein/Aus

- Spindel 1
  - "Ein" = M50
  - "Aus" = M51
- Spindel 2
  - "Ein" = M52
  - "Aus" = M53

### Anforderungen

#### Spindel 1 (Masterspindel)

---

#### Hinweis

#### Standardzuordnungen

- Die Hilfsfunktionen M3, M4, M5, M70 und M1=3, M1=4, M1=5, M1=70 der Spindel 1 (Masterspindel) sind standardmäßig der 2. Hilfsfunktionsgruppe zugeordnet.
  - Alle S- und S1-Werte der Spindel 1 (Masterspindel) sind standardmäßig der 3. Hilfsfunktionsgruppe zugeordnet.
-



- Nach Satzsuchlauf soll die zuletzt programmierte Getriebestufe ausgegeben werden. Die folgenden Hilfsfunktionen werden dazu der 9. Hilfsfunktionsgruppe zugeordnet:
  - M40, M41, M42, M43, M44, M45
  - M1=40, M1=41, M1=42, M1=43, M1=44, M1=45
- Die Hilfsfunktionen M3, M4, M5, M70 und M1=3, M1=4, M1=5, M1=70 (2. Hilfsfunktionsgruppe) sowie S- und S1-Werte (3. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
  - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung)
  - Ausgabe vor der Bewegung
- Die Hilfsfunktionen zum Getriebeumschalten M40 bis M45 und M1=40 bis M1=45 (9. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
  - Ausgabe vor der Bewegung

### Spindel 2

- In einem Satz darf nur eine M-Funktion zur Drehrichtungsumschaltung programmiert werden. Nach Satzsuchlauf soll die zuletzt programmierte Drehrichtung ausgegeben werden. Die folgenden Hilfsfunktionen werden dazu der 10. Hilfsfunktionsgruppe zugeordnet:
  - M2=3, M2=4, M2=5, M2=70
- Alle S2-Werte werden der 11. Hilfsfunktionsgruppe zugeordnet.
- Die Hilfsfunktionen M2=3, M2=4, M2=5, M2=70 (10. Hilfsfunktionsgruppe) sowie S2-Werte (11. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
  - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung)
  - Ausgabe vor der Bewegung

### Kühlwasser

- Ein- und Ausschalten in einem Teileprogrammsatz ist nicht zulässig. Nach Satzsuchlauf soll das Kühlwasser ein- bzw. ausgeschaltet werden. Die folgenden Hilfsfunktionen werden dazu z. B. der 12. bzw. 13. Hilfsfunktionsgruppe zugeordnet:
  - 12. Hilfsfunktionsgruppe: M50, M51
  - 13. Hilfsfunktionsgruppe: M52, M53
- Die Hilfsfunktionen M50, M51 (12. Hilfsfunktionsgruppe) und M52, M53 (13. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
  - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
  - Ausgabe vor der Bewegung

### Parametrierung der Maschinendaten

Die Parametrierung der Maschinendaten erfolgt über eine entsprechende Programmierung innerhalb eines Teileprogramms:

Programmcode	Kommentar
\$MN_AUXFU_MAXNUM_GROUP_ASSIGN=21	; Anzahl anwenderdefinierter Hilfsfunktionen pro Kanal
\$MN_AUXFU_GROUP_SPEC [1]='H22'	; Ausgabeverhalten der 2.Hilfsfunktionsgruppe
\$MN_AUXFU_GROUP_SPEC [2]='H22'	; Ausgabeverhalten der 3.Hilfsfunktionsgruppe
\$MN_AUXFU_GROUP_SPEC [8]='H21'	; Ausgabeverhalten der 9.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE [0]="M"	; Beschreibung der 1.Hilfsfunktion: M40
\$MC_AUXFU_ASSIGN_EXTENSION [0]=0	
\$MC_AUXFU_ASSIGN_VALUE [0]=40	
\$MC_AUXFU_ASSIGN_GROUP [0]=9	
	; ... (entsprechend für 2.-5.Hilfsfunktion)
\$MC_AUXFU_ASSIGN_TYPE [5]="M"	; Beschreibung der 6.Hilfsfunktion: M45
\$MC_AUXFU_ASSIGN_EXTENSION [5]=0	
\$MC_AUXFU_ASSIGN_VALUE [5]=45	
\$MC_AUXFU_ASSIGN_GROUP [5]=9	
\$MC_AUXFU_ASSIGN_TYPE [6]="M"	; Beschreibung der 7.Hilfsfunktion: M1=40
\$MC_AUXFU_ASSIGN_EXTENSION [6]=1	
\$MC_AUXFU_ASSIGN_VALUE [6]=40	
\$MC_AUXFU_ASSIGN_GROUP [6]=9	
	; . . . (entsprechend für 8.-11.Hilfsfunktion)
\$MC_AUXFU_ASSIGN_TYPE [11]="M"	; Beschreibung der 12.Hilfsfunktion: M1=45
\$MC_AUXFU_ASSIGN_EXTENSION [11]=1	
\$MC_AUXFU_ASSIGN_VALUE [11]=45	
\$MC_AUXFU_ASSIGN_GROUP [11]=9	
\$MN_AUXFU_GROUP_SPEC [9] = 'H22'	; Ausgabeverhalten der 10.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE [12]="M"	; Beschreibung der 13.Hilfsfunktion: M2=3
\$MC_AUXFU_ASSIGN_EXTENSION [12]=2	
\$MC_AUXFU_ASSIGN_VALUE [12]=3	
\$MC_AUXFU_ASSIGN_GROUP [12]=10	
\$MC_AUXFU_ASSIGN_TYPE [13]="M"	; Beschreibung der 14.Hilfsfunktion: M2=4
\$MC_AUXFU_ASSIGN_EXTENSION [13]=2	
\$MC_AUXFU_ASSIGN_VALUE [13]=4	
\$MC_AUXFU_ASSIGN_GROUP [13]=10	

Programmcode	Kommentar
\$MC_AUXFU_ASSIGN_TYPE[14]="M"	; Beschreibung der 15.Hilfsfunktion: M2=5
\$MC_AUXFU_ASSIGN_EXTENSION[14]=2	
\$MC_AUXFU_ASSIGN_VALUE[14]=5	
\$MC_AUXFU_ASSIGN_GROUP[14]=10	
\$MC_AUXFU_ASSIGN_TYPE[15]="M"	; Beschreibung der 16.Hilfsfunktion: M2=70
\$MC_AUXFU_ASSIGN_EXTENSION[15]=2	
\$MC_AUXFU_ASSIGN_VALUE[15]=70	
\$MC_AUXFU_ASSIGN_GROUP[15]=10	
\$MN_AUXFU_GROUP_SPEC[10]='H22'	; Spezifikation der 11.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[16]="S"	; Beschreibung der 17.Hilfsfunktion: S2=<alle Werte>
\$MC_AUXFU_ASSIGN_EXTENSION[16]=2	
\$MC_AUXFU_ASSIGN_VALUE[16]=-1	
\$MC_AUXFU_ASSIGN_GROUP[16]=11	
\$MN_AUXFU_GROUP_SPEC[11]='H21'	; Spezifikation der 12.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[17]="M"	; Beschreibung der 18.Hilfsfunktion: M50
\$MC_AUXFU_ASSIGN_EXTENSION[17]=0	
\$MC_AUXFU_ASSIGN_VALUE[17]=50	
\$MC_AUXFU_ASSIGN_GROUP[17]=12	
\$MC_AUXFU_ASSIGN_TYPE[18]="M"	; Beschreibung der 19.Hilfsfunktion: M51
\$MC_AUXFU_ASSIGN_EXTENSION[18]=0	
\$MC_AUXFU_ASSIGN_VALUE[18]=51	
\$MC_AUXFU_ASSIGN_GROUP[18]=12	
\$MN_AUXFU_GROUP_SPEC[12]='H21'	; Spezifikation der 13.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[19]="M"	; Beschreibung der 20.Hilfsfunktion: M52
\$MC_AUXFU_ASSIGN_EXTENSION[19]=0	
\$MC_AUXFU_ASSIGN_VALUE[19]=52	
\$MC_AUXFU_ASSIGN_GROUP[19]=13	
\$MC_AUXFU_ASSIGN_TYPE[20]="M"	; Beschreibung der 21.Hilfsfunktion: M53
\$MC_AUXFU_ASSIGN_EXTENSION[20]=0	
\$MC_AUXFU_ASSIGN_VALUE[20]=53	
\$MC_AUXFU_ASSIGN_GROUP[20]=13	

## 7.18 Datenlisten

### 7.18.1 Maschinendaten

#### 7.18.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10713	M_NO_FCT_STOPRE	M-Funktion mit Vorlaufstopp
10714	M_NO_FCT_EOP	M-Funktion für Spindel aktiv nach NC-RESET
10715	M_NO_FCT_CYCLE	Durch Unterprogramm zu ersetzende M-Funktion
11100	AUXFU_MAXNUM_GROUP_ASSIGN	Maximale Anzahl anwenderdefinierbarer Hilfsfunktionen pro Kanal
11110	AUXFU_GROUP_SPEC	Gruppenspezifisches Ausgabeverhalten
11450	SEARCH_RUN_MODE	Verhalten nach Satzsuchlauf

#### 7.18.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungsgrundstellung nach Teileprogramm-Start
20112	START_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach Hochlauf und bei Reset bzw. Teileprogrammende
20270	CUTTING_EDGE_DEFAULT	Grundstellung der Werkzeugschneide ohne Programmierung
20800	SPF_END_TO_VDI	Unterprogrammende / Halt an PLC
22000	AUXFU_ASSIGN_GROUP	Gruppenzuordnung von anwenderdefinierten Hilfsfunktionen
22010	AUXFU_ASSIGN_TYPE	Typ von anwenderdefinierten Hilfsfunktionen
22020	AUXFU_ASSIGN_EXTENSION	Adresserweiterung für anwenderdefinierte Hilfsfunktionen
22030	AUXFU_ASSIGN_VALUE	Wert von anwenderdefinierten Hilfsfunktionen
22035	AUXFU_ASSIGN_SPEC	Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen
22040	AUXFU_PREDEF_GROUP	Gruppenzuordnung von vordefinierten Hilfsfunktionen
22050	AUXFU_PREDEF_TYPE	Typ von vordefinierten Hilfsfunktionen
22060	AUXFU_PREDEF_EXTENSION	Adresserweiterung für vordefinierte Hilfsfunktionen
22070	AUXFU_PREDEF_VALUE	Wert von vordefinierten Hilfsfunktionen
22080	AUXFU_PREDEF_SPEC	Ausgabeverhalten von vordefinierten Hilfsfunktionen

Nummer	Bezeichner: \$MC_	Beschreibung
22100	AUXFU_QUICK_BLOCKCHANGE	Satzwechselverzögerung bei schnellen Hilfsfunktionen
22110	AUXFU_H_TYPE_INT	Typ von H-Hilfsfunktionen
22200	AUXFU_M_SYNC_TYPE	Ausgabezeitpunkt M-Funktionen
22210	AUXFU_S_SYNC_TYPE	Ausgabezeitpunkt S-Funktionen
22220	AUXFU_T_SYNC_TYPE	Ausgabezeitpunkt T-Funktionen
22230	AUXFU_H_SYNC_TYPE	Ausgabezeitpunkt H-Funktionen
22240	AUXFU_F_SYNC_TYPE	Ausgabezeitpunkt F-Funktionen
22250	AUXFU_D_SYNC_TYPE	Ausgabezeitpunkt D-Funktionen
22252	AUXFU_DL_SYNC_TYPE	Ausgabezeitpunkt DL-Funktionen
22254	AUXFU_ASSOC_M0_VALUE	Zusätzliche M-Funktion für Programm-Halt
22256	AUXFU_ASSOC_M1_VALUE	Zusätzliche M-Funktion für bedingten Halt
22530	TOCARR_CHANGE_M_CODE	M-Code bei Werkzeugträgerwechsel
22532	GEOAX_CHANGE_M_CODE	M-Code bei Umschaltung der Geometrieachsen
22534	TRAFO_CHANGE_M_CODE	M-Code bei Transformationswechsel
22560	TOOL_CHANGE_M_CODE	Hilfsfunktion für Werkzeugwechsel

## 7.18.2 Signale

### 7.18.2.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Assoziiertes M01 aktivieren	DB21, ... .DBX30.5	DB3200.DBX14.5

### 7.18.2.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
M-Funktion 1 - 5 Änderung	DB21, ... .DBX58.0-4	DB2500.DBX4.0-4
M-Funktion 1 - 5 nicht decodiert	DB21, ... .DBX59.0-4	-
S-Funktion 1 - 3 Änderung	DB21, ... .DBX60.0-2	DB2500.DBX6.0
S-Funktion 1 - 3 Quick	DB21, ... .DBX60.4-6	-
T-Funktion 1 - 3 Änderung	DB21, ... .DBX61.0-2	DB2500.DBX8.0
T-Funktion 1 - 3 Quick	DB21, ... .DBX61.4-6	-
D-Funktion 1 - 3 Änderung	DB21, ... .DBX62.0-2	DB2500.DBX10.0
D-Funktion 1 - 3 Quick	DB21, ... .DBX62.4-6	-
DL-Funktion Änderung	DB21, ... .DBX63.0	-
DL-Funktion Quick	DB21, ... .DBX63.4	-
H-Funktion 1 - 3 Änderung	DB21, ... .DBX64.0-2	DB2500.DBX12.0-2

7.18 Datenlisten

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
H-Funktion 1 - 3 Quick	DB21, ... .DBX64.4-6	-
F-Funktion 1 - 6 Änderung	DB21, ... .DBX65.0-5	-
M-Funktion 1 - 5 Quick	DB21, ... .DBX66.0-4	-
F-Funktion 1 - 6 Quick	DB21, ... .DBX67.0-5	-
erweiterte Adresse M-Funktion 1 (16 Bit-Int)	DB21, ... .DBB68-69	DB2500.DBB3004
M-Funktion 1 (DInt)	DB21, ... .DBB70-73	DB2500.DBD3000
erweiterte Adresse M-Funktion 2 (16 Bit-Int)	DB21, ... .DBB74-75	DB2500.DBB3012
M-Funktion 2 (DInt)	DB21, ... .DBB76-79	DB2500.DBD3008
erweiterte Adresse M-Funktion 3 (16 Bit-Int)	DB21, ... .DBB80-81	DB2500.DBB3020
M-Funktion 3 (DInt)	DB21, ... .DBB82-85	DB2500.DBD3016
erweiterte Adresse M-Funktion 4 (16 Bit-Int)	DB21, ... .DBB86-87	DB2500.DBB3028
M-Funktion 4 (DInt)	DB21, ... .DBB88-91	DB2500.DBD3024
erweiterte Adresse M-Funktion 5 (16 Bit-Int)	DB21, ... .DBB92-93	DB2500.DBB3036
M-Funktion 5 (DInt)	DB21, ... .DBB94-97	DB2500.DBD3032
erweiterte Adresse S-Funktion 1 (16 Bit-Int)	DB21, ... .DBB98-99	DB2500.DBB4004
S-Funktion 1 (Real)	DB21, ... .DBB100-103	DB2500.DBD4000
erweiterte Adresse S-Funktion 2 (16 Bit-Int)	DB21, ... .DBB104-105	DB2500.DBB4012
S-Funktion 2 (Real)	DB21, ... .DBB106-109	DB2500.DBD4008
erweiterte Adresse S-Funktion 3 (16 Bit-Int)	DB21, ... .DBB110-111	DB2500.DBB4020
S-Funktion 3 (Real)	DB21, ... .DBB112-115	DB2500.DBD4016
erweiterte Adresse T-Funktion 1 (16 Bit-Int)	DB21, ... .DBB116-117	DB2500.DBB2004
T-Funktion 1 (Int)	DB21, ... .DBB118-119	DB2500.DBD2000
erweiterte Adresse T-Funktion 2 (16 Bit-Int)	DB21, ... .DBB120-121	-
T-Funktion 2 (Int)	DB21, ... .DBB122-123	-
erweiterte Adresse T-Funktion 3 (16 Bit-Int)	DB21, ... .DBB124-125	-
T-Funktion 3 (Int)	DB21, ... .DBB126-127	-
erweiterte Adresse D-Funktion 1 (8 Bit-Int)	DB21, ... .DBB128	DB2500.DBB5004
D-Funktion 1 (8 Bit-Int)	DB21, ... .DBB129	DB2500.DBD5000
erweiterte Adresse D-Funktion 2 (8 Bit-Int)	DB21, ... .DBB130	-
D-Funktion 2 (8 Bit-Int)	DB21, ... .DBB131	-
erweiterte Adresse D-Funktion 3 (8 Bit-Int)	DB21, ... .DBB132	-
D-Funktion 3 (8 Bit-Int)	DB21, ... .DBB133	-
erweiterte Adresse DL-Funktion (8 Bit-Int)	DB21, ... .DBB134	-
DL-Funktion (Real)	DB21, ... .DBB136	-
erweiterte Adresse H-Funktion 1 (16 Bit-Int)	DB21, ... .DBB140-141	DB2500.DBB6004
H-Funktion 1 (Real oder DInt)	DB21, ... .DBB142-145	DB2500.DBD6000
erweiterte Adresse H-Funktion 2 (16 Bit-Int)	DB21, ... .DBB146-147	DB2500.DBB6012
H-Funktion 2 (REAL oder DInt)	DB21, ... .DBB148-151	DB2500.DBD6008
erweiterte Adresse H-Funktion 3 (16 Bit-Int)	DB21, ... .DBB152-153	DB2500.DBB6020
H-Funktion 3 (Real oder DInt)	DB21, ... .DBB154-157	DB2500.DBD6016

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
erweiterte Adresse F-Funktion 1 (16 Bit-Int)	DB21, ... .DBB158-159	-
F-Funktion 1 (Real)	DB21, ... .DBB160-163	-
erweiterte Adresse F-Funktion 2 (16 Bit-Int)	DB21, ... .DBB164-165	-
F-Funktion 2 (Real)	DB21, ... .DBB166-169	-
erweiterte Adresse F-Funktion 3 (16 Bit-Int)	DB21, ... .DBB170-171	-
F-Funktion 3 (Real)	DB21, ... .DBB172-175	-
erweiterte Adresse F-Funktion 4 (16 Bit-Int)	DB21, ... .DBB176-177	-
F-Funktion 4 (Real)	DB21, ... .DBB178-181	-
erweiterte Adresse F-Funktion 5 (16 Bit-Int)	DB21, ... .DBB182-183	-
F-Funktion 5 (Real)	DB21, ... .DBB184-187	-
erweiterte Adresse F-Funktion 6 (16 Bit-Int)	DB21, ... .DBB188-189	-
F-Funktion 6 (Real)	DB21, ... .DBB190-193	-
Dynamische M-Funktion: M00 - M07	DB21, ... .DBB194	DB2500.DBB1000
Dynamische M-Funktion: M08 - M15	DB21, ... .DBB195	DB2500.DBB1001
Dynamische M-Funktion: M16 - M23	DB21, ... .DBB196	DB2500.DBB1002
Dynamische M-Funktion: M24 - M31	DB21, ... .DBB197	DB2500.DBB1003
Dynamische M-Funktion: M32 - M39	DB21, ... .DBB198	DB2500.DBB1004
Dynamische M-Funktion: M40 - M47	DB21, ... .DBB199	DB2500.DBB1005
Dynamische M-Funktion: M48 - M55	DB21, ... .DBB200	DB2500.DBB1006
Dynamische M-Funktion: M56 - M63	DB21, ... .DBB201	DB2500.DBB1007
Dynamische M-Funktion: M64 - M71	DB21, ... .DBB202	DB2500.DBB1008
Dynamische M-Funktion: M72 - M79	DB21, ... .DBB203	DB2500.DBB1009
Dynamische M-Funktion: M80 - M87	DB21, ... .DBB204	DB2500.DBB1010
Dynamische M-Funktion: M88 - M95	DB21, ... .DBB205	DB2500.DBB1011
Dynamische M-Funktion: M96 - M99	DB21, ... .DBX206.0-3	DB2500.DBB1012.0-3
Assoziiertes M00/M01 aktiv	DB21, ... .DBX318.5	DB3300.DBX4002.5

### 7.18.2.3 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
F-Funktion für Positionierachse (REAL)	DB31, ... .DBB78-81	-

### 7.18.2.4 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
M-Funktion für Spindel (Int)	DB21, ... .DBB86-87	DB370x.DBD0000
S-Funktion für Spindel (Real)	DB21, ... .DBB88-91	DB370x.DBD0004





# K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten

## 8.1 Kurzbeschreibung

### Kanal

Ein Kanal der NC stellt die kleinste Einheit für das manuelle Verfahren von Achsen und die automatische Abarbeitung von Teileprogrammen dar. Ein Kanal befindet sich zu einem Zeitpunkt immer in einer bestimmten Betriebsart, z. B. AUTOMATIK, MDA oder JOG. Ein Kanal kann als eigenständige NC betrachtet werden.

### Betriebsartengruppe (BAG)

Ein Kanal gehört immer einer Betriebsartengruppe an. Eine Betriebsartengruppe kann auch aus mehreren Kanälen bestehen.

Das Kennzeichen einer Betriebsartengruppe ist, dass sich alle Kanäle der Betriebsartengruppe zu einem Zeitpunkt immer in der gleichen Betriebsart, z. B. AUTOMATIK, MDA oder JOG, befinden. Dies wird durch die NC-interne Betriebsartenlogik sichergestellt.

Eine Betriebsartengruppe kann als eigenständige mehrkanalige NC betrachtet werden.

### Kanallücken

Bei der Konfiguration der Kanäle können Platzhalter-Kanäle vorgesehen werden, um die Konfiguration über Maschinen einer Serie möglichst einheitlich anzulegen und nur die tatsächlich benutzten Kanäle zu aktivieren.

### Programmtest

Zum Testen oder Einfahren eines neuen Teileprogramms gibt es folgende Möglichkeiten:

- Programmbearbeitung ohne Sollwertausgaben
- Programmbearbeitung im Einzelsatzbetrieb
- Programmbearbeitung mit Probelaufvorschub
- Teileprogrammätze ausblenden
- Satzsuchlauf mit oder ohne Berechnung

## Satzsuchlauf

Über Satzsuchlauf gibt es folgende Programmsimulationen zum Suchen bestimmter Programmstellen:

- Typ 1 ohne Berechnung an Kontur
- Typ 2 mit Berechnung an Kontur
- Typ 4 mit Berechnung an Satzendpunkt
- Typ 5 Selbsttätiger Start der angewählten Programmstelle mit Berechnung aller erforderlichen Daten aus der Vorgeschichte
- Automatischer Start eines ASUPs nach Satzsuchlauf
- Kaskadierter Satzsuchlauf
- Kanalübergreifender Satzsuchlauf im Modus "Programmtest"

## Programmbetrieb

Programmbetrieb liegt dann vor, wenn in der Betriebsart AUTOMATIK oder MDA Teileprogramme bzw. Teileprogrammsätze abgearbeitet werden. Während der Abarbeitung kann dabei der Programmablauf durch PLC-Nahtstellensignale und Kommandos beeinflusst werden.

Für jeden Kanal können Grundstellungen über kanalspezifische Maschinendaten vorgegeben werden. Diese Grundstellungen wirken u. a. auf die G-Gruppen sowie auf die Hilfsfunktionsausgabe.

Die Anwahl eines Teileprogramms kann nur erfolgen, wenn sich der betreffende Kanal im Reset-Zustand befindet.

Außerdem werden alle weiteren Programmabläufe mit PLC-Nahtstellensignalen und den entsprechenden Kommandos behandelt.

- Starten des Teileprogramms bzw. des Teileprogrammsatzes
- Teileprogrammrechnung und Programmbeeinflussung
- RESET-Kommando, Programmzustand und Kanalzustand
- Reaktionen auf Bedienungs- und Programmaktionen
- Ereignisgesteuerte Programmaufrufe

## Asynchrone Unterprogramme (ASUP), Interruptroutinen

Mit Hilfe von Interrupteingängen ist die NC in der Lage, die aktuelle Teileprogrammbearbeitung zu unterbrechen und auf hochpriorie Ereignisse in Interruptroutinen/ASUPs zu reagieren.

## **Einzelsatz**

In der Funktion Einzelsatz kann der Anwender das Teileprogramm satzweise abarbeiten.

Es gibt 3 Einstellungsarten der Funktion Einzelsatz:

- SLB1: = IPO-Einzelsatz
- SLB2: = Decodiereinzelsatz
- SLB3: = Stopp im Zyklus

## **Basis-Satzanzeige**

Zur bestehenden Satzanzeige können über eine zweite, der Basis-Satzanzeige, alle Sätze angezeigt werden, die eine Aktion an der Maschine bewirken werden.

Die tatsächlich angefahrenen Endpositionen werden als Absolutposition dargestellt. Die Positionswerte beziehen sich wahlweise auf das Werkstückkoordinatensystem (WKS) oder auf das Einstellbare Nullpunkt-System (ENS).

## **Programm von Extern abarbeiten**

Bei der Bearbeitung komplexer Werkstücke kommt es vor, dass der Speicher der NC für die Programme nicht ausreicht. Mit der Funktion "Abarbeiten von Extern" können Unterprogramme von einem externen Speicher (z. B. von der Festplatte) aufgerufen (`EXTCALL`) und abgearbeitet werden.

## **Verhalten nach POWER ON, Reset, ...**

Das Verhalten der Steuerung nach:

- Hochlauf (POWER ON)
- Reset/Teileprogramm-Ende
- Teileprogramm-Start

kann für Funktionen wie z. B. G-Codes, Werkzeuglängenkorrektur, Transformation, Mitschleppverbände, Tangentiale Nachführung, Programmierbare Synchronspindel, für bestimmte Systemeinstellungen durch Maschinendaten verändert werden.

### Unterprogrammaufruf mit M-, T- bzw. D-Funktionen

Für bestimmte Anwendungen kann es vorteilhaft sein, M- T- bzw. D-Funktionen sowie einige NC-Sprachbefehle ( $S_{POS}$ ,  $S_{POSA}$ ) durch einen Unterprogrammaufruf zu ersetzen. Dies kann z. B. für den Aufruf der Werkzeugwechselroutine genutzt werden.

Mit entsprechenden Maschinendaten können Unterprogramme mit M- T- bzw. D-Funktionen z. B. für einen Getriebestufenwechsel definiert und beeinflusst werden.

### Programmlaufzeit / Werkstückzähler

Zur Unterstützung des Bearbeiters an der Werkzeugmaschine werden Informationen zur Programmlaufzeit und zur Werkstückzählung bereitgestellt.

Die dabei definierte Funktionalität ist **nicht identisch mit Funktionen der Werkzeugverwaltung** und besonders für NC-Systeme ohne Werkzeugverwaltung vorgesehen.

## 8.2 Betriebsartengruppe (BAG)

### Betriebsartengruppe

Eine Betriebsartengruppe fasst NC-Kanäle mit Achsen und Spindeln zu einer Bearbeitungseinheit zusammen.

Eine Betriebsartengruppe enthält die Kanäle, die vom Bearbeitungsablauf her immer gleichzeitig in der gleichen Betriebsart laufen müssen.

Mit der Konfiguration einer Betriebsartengruppe wird festgelegt, welche Kanäle zu einer Gruppe zusammengefasst werden.

---

#### Hinweis

In dieser Beschreibung wird von einer BAG und einem Kanal ausgegangen.

Funktionen, die mehrere Kanäle benötigen (z. B. Funktion "Achstausch"), sind beschrieben in:

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; BAGs, Kanäle, Achstausch (K5)

---

### Zuordnung: Kanal - Betriebsartengruppe

Einem Kanal werden Achsen und/oder Spindeln zugeordnet.

Der Kanal wiederum wird über das folgende Maschinendatum einer BAG zugeordnet:

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP (Kanal gültig in Betriebsartengruppe)

Wird in mehreren Kanälen die gleiche BAG adressiert, so bilden diese zusammen eine BAG.

---

#### Hinweis

Die Steuerung kennt keine BAG-spezifischen Daten. Es ist aber möglich, einige kanalspezifische Einstellungen, die die BAG betreffen, vorzunehmen.

---

### Kanalspezifische Zuordnungen

Achsen können mehreren Kanälen zugeordnet werden, die sich wiederum in verschiedenen BAGs befinden. Die Achsen können dann auch zwischen diesen Kanälen getauscht werden (Achstausch). Der Achstausch ist unabhängig von der BAG.

Einem Kanal werden Maschinenachsen bzw. Spindeln zugeordnet und wie folgt unterschieden:

- Geometrieachsen können im Bahnverband betrieben werden.  
Die Masterspindel kann Funktionen wie G96, G961, G331, G332 etc. durchführen.
- Kanalachsen, die nicht als Geometrieachsen definiert sind, können als Bahnachsen, Synchronachsen, Positionierachsen, PLC-Achsen und Kommando-Achsen verfahren werden.
- Zusatzachsen haben keinen geometrischen Zusammenhang zwischen den Achsen.
- Masterspindel-Geometrieachsen können mit der Masterspindel Funktionen verrichten.
- Hilfsspindeln sind neben der Masterspindel alle weiteren Spindeln im Kanal.

Durch den Programmbefehl `GEOAX` kann zugeordnet werden, welche Kanalachse die wievielte Geometrieachse sein soll. Welche Spindel im Kanal die Masterspindel sein soll, wird mit `SETMS` festgelegt.

Es kann jede im Kanal vorhandene Achse als Spindel projektiert werden. Die Anzahl der Achsen pro Kanal ist abhängig von der Ausprägung der Steuerung. Damit die verfügbare Leistungsbandbreite optimal genutzt werden kann, sind je nach Hardware nur bestimmte Konfigurationen von Kanälen und dazugehörigen Achsen erlaubt.

Bei SINUMERIK 840D sl sind abhängig von der Ausprägung der HW/SW zulässig:

- Pro Kanal bis zu 12 Achsen / Spindeln
- Pro NCU maximal 31 Achsen oder maximal 20 Spindeln

Informationen zu weiteren Achskonfigurationen wie Achscontainer, Link-, Pendel-, Hauptlauf-, Rund-, Linear-, Leit- und Folge-Achsen und zu den verschiedenen Ausprägungen siehe Kapitel K2: Achsen, Koordinatensysteme, Frames (Seite 721) und S1: Spindeln (Seite 1383).

## BAG-spezifische Nahtstellensignale

Der Signalaustausch BAG-spezifischer Signale an/von Betriebsartengruppe wird in der Anwendernahtstelle in den DB11 übertragen. Dadurch kann die Betriebsartengruppe von der PLC bzw. der NCK aus überwacht bzw. beeinflusst werden.

Die nachfolgenden Tabellen stellen alle BAG-spezifischen Nahtstellensignale dar:

Signale von PLC an NCK
BAG-Reset
BAG-Stop Achsen plus Spindel
BAG-Stop
Betriebsartenwechsel
Betriebsart: JOG, MDA, AUTOM.
Einzelersatz: Typ A, Typ B
Maschinenfunktion: REF, REPOS, TEACH IN,

Signale von NCK an PLC
Betriebsart-Strobe: JOG, MDA, AUTOMATIK
Maschinenfunktion Strobe: REF, REPOS, TEACH IN
Alle Kanäle (1 bis maximal 10) im Reset-Zusand
BAG betriebsbereit
aktive Betriebsart: JOG, MDA, AUTOMATIK
Digitalisieren
aktive Maschinenfunktion: REF, REPOS, TEACH IN var. INC, 10000 INC ..... 1 INC

## Änderung der Betriebsartengruppe

Eine Konfigurationsänderung der Betriebsartengruppe hinsichtlich ihrer zugeteilten Kanäle ist nur mit anschließendem POWER ON möglich.

Die Änderung erfolgt über das Maschinendatum:

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP

BAG-Nummern müssen lückenlos ab 1 vergeben werden.

## Maschinendaten

Es existieren keine BAG-spezifischen Maschinendaten.

## Kanallücken

Kanäle, denen mit MD10010 eine Betriebsartengruppe zugeordnet ist, gelten als aktiviert.

Kanälen kann anstelle einer BAG-Nummer die Nummer 0 zugeordnet werden. Damit wird Folgendes erreicht:

- Der nicht aktivierte Kanal benötigt keinen Speicherplatz in der Steuerung.
- Bauähnliche Serienmaschinen können bei der Konfiguration weitgehend gleich gehalten werden, wobei jeweils nur die Kanäle aktiviert werden, mit denen die vorliegende Maschine tatsächlich arbeiten kann (BAG-Nummer größer als 0).

### Sonderfall:

Kanal 1 muss immer vorhanden sein!

=> bei folgender Einstellung:

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP [0] = 0

setzt die Steuerung automatisch:

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP [0] = 1 (BAG 1)

### Beispielkonfigurationen:

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP[0] = 1

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP[1] = 2

...

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP[3] = 0 ; Lücke

...

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP[8] = 1

MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP[9] = 2

## 8.2.1 BAG-Stop

### Funktion

Über die folgenden NC/PLC-Nahtstellensignale werden in allen Kanälen der BAG die Verfahrbewegungen der Achsen bzw. Achsen und Spindeln angehalten, sowie die Teileprogrammabarbeitung unterbrochen:

DB11 DBX0.5 (BAG-Stop)

DB11 DBX0.6 (BAG-Stop Achsen plus Spindeln)



## **8.2.2 BAG-Reset**

### **Funktion**

Die Anforderung zum BAG-Reset erfolgt über das BAG-spezifische NC/PLC-Nahtstellensignal:

DB11 DBX0.7 = 1 (BAG-Reset)

### **Wirkung**

Wirkung auf die Kanäle der BAG:

- Die Teileprogrammaufbereitung (Vorlauf) wird gestoppt.
- Alle Achsen und Spindeln werden entlang ihrer Beschleunigungskennlinien ohne Konturverletzung auf Stillstand abgebremst.
- Die noch nicht an die PLC ausgegebenen Hilfsfunktionen werden nicht mehr ausgegeben.
- Die Vorlaufzeiger werden auf die Unterbrechungsstelle und die Satzzeiger auf den Anfang der jeweiligen Teileprogramme gesetzt.
- Alle Grundstellungen (z. B. der G-Funktionen) werden auf die parametrisierten Werte eingestellt.
- Alle Alarmer mit Löschkriterium "Kanal-Reset" werden gelöscht.

Wenn alle Kanäle der BAG im Reset-Zustand sind, dann:

- werden alle Alarmer mit Löschkriterium "BAG-Reset" gelöscht.
- werden der Abschluss des BAG-Resets und die Betriebsbereitschaft der BAG an der NC/PLC-Nahtstelle angezeigt:

DB11 DBX6.7 = 1 (alle Kanäle im Reset-Zustand)

DB11 DBX6.3 = 1 (BAG betriebsbereit)

## 8.3 Betriebsarten und Betriebsartenwechsel

### Eindeutige Betriebsart

Alle Kanäle einer Betriebsartengruppe (BAG) befinden sich immer in derselben Betriebsart:

- AUTOMATIK
- JOG
- MDA

Sind einzelne Kanäle unterschiedlichen Betriebsartengruppen zugeordnet, so erfolgt über eine Kanalumschaltung auch eine Umschaltung auf die entsprechende BAG. Auf diese Weise ist über eine Kanalumschaltung ein Betriebsartenwechsel möglich.

### Betriebsarten

Es stehen folgende Betriebsarten zur Verfügung:

- **AUTOMATIK**

Automatisches Abarbeiten von Teileprogrammen:

- Teileprogrammtest
- Alle Kanäle der BAG können gleichzeitig aktiv sein.

- **JOG in AUTOMATIK**

JOG in AUTOMATIK ist eine Erweiterung der Betriebsart AUTOMATIK mit dem Ziel der Bedienungsvereinfachung. JOG kann ausgeführt werden, ohne AUTOMATIK zu verlassen, wenn Randbedingungen dies zulassen.

- **JOG**

Manuelles Verfahren von Achsen über Verfahrtasten der Maschinensteuertafel oder über ein an der Maschinensteuertafel angeschlossenes Handrad:

- Kanalspezifische Signale und Verriegelungen werden bei Bewegungen, die per ASUP oder über statische Synchronaktionen ausgeführt werden, beachtet.
- Kopplungen werden berücksichtigt.
- Jeder Kanal in der BAG kann aktiv sein.

- **MDA**

Manual Data Automatic (Eingabe der Sätze erfolgt über die Bedienoberfläche):

- Eingeschränktes Abarbeiten von Teileprogrammen und Teileprogrammteilen.
- Teileprogrammtest
- Maximal 1 Kanal pro BAG kann aktiv sein (nur bei TEACH IN).
- Achsen können nur in unterlagerten Maschinenfunktionen wie JOG, REPOS oder TEACHIN manuell verfahren werden.

## Gültig für alle Betriebsarten

### Betriebsarten-übergreifende Synchronaktionen

Übergreifend für alle Betriebsarten, können modale Synchronaktionen per IDS für folgende Funktionen parallel zum Kanal abgearbeitet werden:

- Kommandoachsfunktionen
- Spindelfunktionen
- Technologiezyklen

## Anwahl

Über die Bedienoberfläche kann der Anwender die gewünschte Betriebsart mit Hilfe von Softkeys anwählen.

Diese Anwahl (AUTOMATIK, MDA oder JOG) wird an die NC/PLC-Nahtstelle zur PLC weitergeleitet, aber noch nicht aktiviert:

DB11 DBX4.0, 0.1, 0.2 (Betriebsart-Strobe)

## Aktivierung und Prioritäten

Die Betriebsart der BAG wird aktiviert über die NC/PLC-Nahtstelle:

DB11 DBX0.0, 0.1, 0.2 (Betriebsart)

Werden mehrere Betriebsarten zur gleichen Zeit angewählt, besteht folgende Priorität:

Priorität	Betriebsart	BAG-Signal (NCK → PLC)
1. Priorität, hoch	JOG	DB11 DBX0.2
2. Priorität, mittel	MDA	DB11 DBX0.1
3. Priorität, niedrig	AUTOMATIK	DB11 DBX0.0

## Anzeige

Die aktuelle Betriebsart der BAG wird angezeigt über die NC/PLC-Nahtstelle:

DB11 DBX6.0, 0.1, 0.2 (aktive Betriebsart)

BAG-Signal (NCK → PLC)	Aktive Betriebsart
DB11 DBX6.2	JOG
DB11 DBX6.1	MDA
DB11 DBX6.0	AUTOMATIK

## Maschinenfunktionen

Innerhalb einer Betriebsart können Maschinenfunktionen angewählt werden, die ebenfalls innerhalb der BAG gelten:

- Maschinenfunktionen innerhalb der Betriebsart **JOG**
  - REF (Referenzpunktfahren)
  - REPOS (Repositionieren)
  - JOG-Retract (Rückzugsbewegung in Werkzeugrichtung)
- Maschinenfunktionen innerhalb der Betriebsart **MDA**
  - REF (Referenzpunktfahren)
  - REPOS (Repositionieren)
  - TEACHIN (Ein-Teachen von Achspositionen)

### NC/PLC-Nahtstellensignale

- DB11 DBX5.0, 0.1, 0.2 (Maschinenfunktion-Strobe): Anforderung
- DB11 DBX1.0, 0.1, 0.2 (Maschinenfunktion): Aktivierung
- DB11 DBX7.0-2 (aktive Maschinenfunktion): Rückmeldung

## Kanal-Zustände

- **Kanal-Reset**

Die Maschine befindet sich im Grundzustand. Dieser wird über das PLC-Programm vom Maschinenhersteller z. B. nach dem Einschalten oder nach Programmende definiert.

- **Kanal aktiv**

Ein Programm ist gestartet, die Programmabarbeitung oder Referenzpunktfahren läuft.

- **Kanal unterbrochen**

Das laufende Programm oder Referenzpunktfahren sind unterbrochen worden.

## Funktionen innerhalb von Betriebsarten

Die Betriebsarten werden ergänzt durch anwenderspezifische Funktionen. Die Funktionen sind technologie- und maschinenunabhängig und können aus den drei Kanalzuständen "Kanal-Reset", "Kanal aktiv" oder "Kanal unterbrochen" gestartet und/oder bearbeitet werden.

## Randbedingung für die Unterbetriebsart TEACH IN

TEACH IN ist nicht zulässig für Führungs- oder Leitachsen eines aktiven Achsverbundes, z. B. bei:

- Gantry-Achsverbund oder ein Gantry-Achsenpaar
- Mitschleppverbund von Leitachse mit Folgeachse

## JOG in AUTOMATIK

JOG in Betriebsart AUTOMATIK ist zulässig, wenn die BAG im Zustand "RESET" und die Achse JOG-fähig ist. Zustand "RESET" für die BAG bedeutet:

- Alle Kanäle sind im Zustand "RESET"
- Alle Programme sind abgebrochen
- In keinem Kanal ist DRF aktiv

Eine Achse ist **JOG-fähig** wenn sie in keinem der folgenden Zustände ist:

- PLC-Achse als konkurrierende Positionierachse (Anforderung der Achse von der PLC)
- Kommandoachse (die Achse wurde von einer Synchronaktion programmiert und die Bewegung ist noch nicht zu Ende)
- Drehende Spindel (Spindel die über RESET hinaus dreht)
- Asynchrone Pendelachse

Hinweis: Die Eigenschaft "JOG-fähig" ist unabhängig von der Funktion "JOG in AUTOMATIK".

### Aktivierung

Die Funktion "JOG in AUTOMATIK" ist aktivierbar mit dem Maschinendatum:

MD10735 \$MN\_JOG\_MODE\_MASK

- Vor POWER ON muss folgendes Maschinendatum gesetzt sein:  
MD10735 \$MN\_JOG\_MODE\_MASK, Bit 0 = 1
- Der Anwender schaltet nach AUTO (PLC-Anwenderschnittstelle DB11 DBX0.0 = 0→1 Flanke). "JOG in AUTOMATIK" ist damit aktiv, wenn vorher der NCK in allen Kanälen der BAG den Kanalzustand "RESET" und den Programmzustand "Abgebrochen" hatte. Zudem muss die dann betroffene Achse "JOG-fähig" sein. DRF muss abgeschaltet sein oder werden.
- In allen Kanälen der BAG ohne den Kanalzustand "RESET" und den Programmzustand "Abgebrochen" wird RESET ausgelöst oder das laufende Programm kommt mit M30/M2 zu Ende.
- Die betroffene Achse wird automatisch "JOG-fähig" gemacht (z. B. Achstauch: PLC → NC).

Hinweis: In den meisten Anwendungsfällen sind die zu verfahrenen Achsen "JOG-fähig" und mit dem Umschalten nach AUTOMATIK ist damit auch "JOG in AUTOMATIK" aktiv.

### Merkmale

- Die Tasten +/- leiten eine JOG-Bewegung ein, die BAG wird **intern** nach JOG geschaltet. (Kurz "Intern-JOG").
- Die Bewegungen der Handräder leiten eine JOG-Bewegung ein, die BAG wird **intern** nach JOG geschaltet, außer es ist DRF aktiv.
- Eine begonnene JOG-Bewegung ist erst zu Ende, wenn die Endposition des Inkrements (falls dies eingestellt worden war) erreicht wurde, oder die Bewegung mit "Restweglöschen" abgebrochen wurde.

Damit kann ein Inkrement mit Stopp angehalten und mit Start wieder bis zum Ende gefahren werden. Solange bleibt der NCK im "Intern-JOG". Ein teilweise abgefahrenes Inkrement ist möglich, muss aber nicht mit Stopp unterbrochen sein. Es gibt einen Modus, bei dem eine Wegnahme der Fahrtaste zur Unterbrechung innerhalb eines Inkrements führt.

- Ohne eine JOG-Bewegung verhält sich "JOG in AUTOMATIK" wie "Automatik". Insbesondere startet die Start-Taste das angewählte Teileprogramm und der entsprechende Softkey des HMI löst einen Satzsuchlauf aus.
- Mit einer aktiven JOG-Bewegung ist der NCK intern in JOG und eine Satzsuchlaufanforderung wird abgelehnt und der Start startet das Teileprogramm nicht. Der Start startet ggf. das verbliebene Inkrement oder er bleibt ohne Wirkung.
- Solange eine Achse der BAG in JOG verfahren wird, bleibt die BAG intern in JOG. Bemerkung: Diese Phase kann mit der JOG-Bewegung einer Achse beginnen und mit dem Ende der JOG-Bewegung einer *anderen* Achse enden.
- Eine Achse mit aktiver JOG-Bewegung darf keinem Achstausch unterworfen werden. (Die Achse könnte die BAG wechseln). Die NCK verriegelt diesen Achstausch-Versuch.
- Die PLC-Anwenderschnittstelle zeigt die Betriebsart "Automatik" an:
  - DB11 DBX6.0, 6.1, 6.2 = 1
  - DB11 DBX7.0, 7.1, 7.2 = 0
- Die NC/PLC-Nahtstelle zeigt an, wenn in "JOG in AUTOMATIK" sich die BAG in "BAG-RESET" befindet.
  - DB11 DBX6.4 (BAG reseted, BAG 1)
  - DB11 DBX26.4 (BAG reseted, BAG 2)
  - DB11 DBX46.4 (BAG reseted, BAG 3)
- Die NC/PLC-Nahtstelle zeigt an, wenn in "JOG in AUTOMATIK" die NC intern nach "Intern-JOG" geschaltet hat.
  - DB11 DBX6.5 (NCK intern JOG aktiv, BAG 1)
  - DB11 DBX26.5 (NCK intern JOG aktiv, BAG 2)
  - DB11 DBX46.5 (NCK intern JOG aktiv, BAG 3)

### Randbedingungen

"JOG in AUTOMATIK" kann intern nur dann nach JOG wechseln, wenn die BAG im Zustand "BAG-RESET" ist. D. h. mitten in einem gestoppten Programm kann nicht unmittelbar gejoggt werden. Der Benutzer kann joggen, wenn er in dieser Situation die JOG-Taste oder die Reset-Taste *in allen Kanälen* der BAG drückt.

Mit der Anwahl von Automatik werden die INC-Tasten abgeschaltet und der Anwender kann/muss die INC-Tasten erneut drücken, um ein von ihm gewünschtes Inkrement anzuwählen. Schaltet der NCK nach "Intern-JOG", so bleibt das angewählte Inkrement erhalten.

Versucht der Anwender die Geo- oder Orientierungsachsen zu joggen, so wechselt der NCK auch nach "Intern-JOG" und die Bewegung wird ausgeführt. Dabei können mehrere Achsen physisch bewegt werden, die alle "JOG-fähig" sein müssen.

Nach der JOG-Bewegung schaltet der NCK "Intern-JOG" wieder ab und wählt damit AUTO wieder an. Der Interne Modewechsel wird verzögert nach dem Bewegungsende durchgeführt. Damit werden unnötig viele Schaltvorgänge vermieden, die z. B. beim Handrad auftreten könnten. Die PLC darf sich nur auf das PLC-Signal "Intern-JOG-Aktiv" verlassen.

Der NCK schaltet auch dann nach "Intern-JOG", wenn die Achse keine Freigabe hat.

### Siehe auch

R1: Referenzieren (Seite 1319)

## 8.3.1 Überwachungen und Verriegelungen der einzelnen Betriebsarten

### Kanalzustand bestimmt Überwachungen

#### Überwachungen in den Betriebsarten

In den einzelnen Betriebsarten sind unterschiedliche Überwachungen aktiv. Diese Überwachungen sind Technologie- und Maschinen-unabhängig.

In jeder Betriebsart, in Abhängigkeit des Betriebszustandes, ist ein Teil der Überwachungen aktiv. Welche Überwachungen in welcher Betriebsart und in welchem Betriebszustand aktiv sind, wird durch den Kanalzustand bestimmt.

#### Verriegelungen in den Betriebsarten

In den einzelnen Betriebsarten können unterschiedliche Verriegelungen aktiv sein. Diese Verriegelungen sind Technologie- und Maschinen-unabhängig.

In jeder Betriebsart, in Abhängigkeit des Betriebszustandes, können fast alle Verriegelungen aktiviert werden.

### 8.3.2 Betriebsartenwechsel

#### Einführung

Ein Betriebsartenwechsel wird über die BAG-Nahtstelle (DB11) angefordert und aktiviert. Eine Betriebsartengruppe befindet sich entweder in der Betriebsart AUTOMATIK, JOG oder MDA, d. h. mehrere Kanäle einer Betriebsartengruppe können gleichzeitig keine unterschiedlichen Betriebsarten annehmen.

Ob die angeforderte Betriebsart erreichbar ist und wie diese durchgeführt wird, ist maschinenspezifisch über das PLC-Programm projektierbar.

---

#### Hinweis

Die Betriebsart wird erst dann steuerungsintern gewechselt, wenn der "Kanalzustand aktiv" nicht mehr vorliegt. Zum fehlerfreien Betriebsartenwechsel müssen aber alle Kanäle einen zulässigen Betriebszustand eingenommen haben.

---

#### Mögliche Betriebsartenwechsel

Die möglichen Betriebsartenwechsel können Sie der folgenden Tabelle für einen Kanal entnehmen:

	AUTOMATIK		JOG			MDA			
				AUTO	MDA	Handfahren		AUTO	
	Reset	unterbr.	Reset	unterbr.	unterbr.	Reset	unterbr.	aktiv	unterbr.
<b>AUTOMATIK</b>			X	X		X			
<b>JOG</b>	X	X				X	X		X
<b>MDA</b>	X	X	X		X				

Die mit "X" gekennzeichneten Positionen sind mögliche Betriebsartenwechsel.



## Sonderfälle

- **Fehler bei Betriebsartwechsel**

Wenn eine Betriebsartenwechsel-Anforderung vom System abgewiesen wurde, erfolgt die Fehlermeldung "BA-Wechsel erst nach NC-Stop möglich". Diese Fehlermeldung kann gelöscht werden, ohne den Kanalzustand zu ändern.

- **Betriebsartenwechsel-Sperre**

Mit Hilfe des Nahtstellensignals:  
DB11 DBX0.4 (Betriebsartenwechsel-Sperre)  
kann ein Wechseln der Betriebsart verhindert werden.

Es wird dabei schon die Betriebsartenwechsel-Anforderung unterdrückt.

Der Anwender muss eine Meldung über die aktive Sperre für den Bediener selber projektieren. Vom System wird dafür nichts abgesetzt.

- **Betriebsartenwechsel von MDA nach JOG**

Sind nach einem Betriebsartenwechsel von MDA nach JOG alle Kanäle der BAG im Reset-Zustand, so wechselt die NC von JOG nach AUTO. In diesem Zustand können die Teileprogrammbefehle `START` oder `INIT` durchgeführt werden.

Ist nach dem Betriebsartenwechsel ein Kanal der BAG nicht mehr im Reset-Zustand, so wird der Teileprogrammbefehl `START` in dieser Situation mit dem Alarm 16952 abgelehnt.

## 8.4 Kanal

### Zuordnung Teileprogramm - Kanal

Teileprogramme sind Kanälen zugeordnet.

Teileprogramme verschiedener Kanäle sind weitgehend voneinander unabhängig.

### Eigenschaften des Kanals

Ein Kanal stellt eine "NC" dar, in der zu einer Zeit ein Teileprogramm bearbeitet werden kann. Maschinenachsen, Geometrieachsen und Positionierachsen werden entsprechend der Maschinenkonfiguration und dem aktuellen Programmzustand (Achstausch, Geoachstausch, SETMS) den Kanälen zugeordnet.

Jedem Kanal wird vom System ein eigener Bahninterpolator mit zugehöriger Programmverarbeitung zugeordnet. Jeder Kanal kann, über die PLC gesteuert, sein eigenes Bearbeitungsprogramm abfahren.

Die Unabhängigkeit der Teileprogrammbearbeitung zwischen den Kanälen wird durch folgende kanalspezifische Funktionen sichergestellt:

- Eigener NC-Start, NC-Stop, RESET pro Kanal.
- Eine Vorschub- und eine Eilgangkorrektur pro Kanal.
- Eigener Interpreter pro Kanal.
- Eigener Bahninterpolator pro Kanal, der die Bahnpunkte so ermittelt, dass von Bahnachsen alle Bearbeitungsachsen dieses Kanals zeitlich gleich geführt werden.
- An- und Abwahl von Werkzeugschneiden und deren Längen- und Radiuskorrektur für ein Werkzeug in einen bestimmten Kanal.

Weitere Informationen zur Werkzeugkorrektur siehe Kapitel "W1: Werkzeugkorrektur (Seite 1567)".

- Kanalspezifische Frames und im Kanal wirksame Frames zum Überführen geschlossener Rechenvorschriften in kartesische Koordinatensysteme. In einem Frame werden Verschiebungen, Drehungen, Skalierungen oder Spiegelungen für Geometrieachsen und Zusatzachsen programmiert.

Weitere Informationen zu den Frames siehe Kapitel "Externe Nullpunktverschiebung über Systemframe (Seite 840)".

- Anzeige von kanalspezifischen Alarmreaktionen.
- Anzeige des momentanen Bearbeitungsablaufes (Achspeditionen, aktuelle G-Funktionen, aktuelle Hilfsfunktionen, aktueller Programmsatz) pro Kanal.
- Separate Programmbeeinflussung pro Kanal.

Diese Funktionen (außer den Anzeigefunktionen) werden von der PLC über Nahtstellensignale gesteuert und kontrolliert.

Kanäle, die in der gleichen Betriebsartengruppe zusammengefasst sind, können immer nur mit der gleichen Betriebsart betrieben werden (AUTOMATIK, JOG, MDA).

## Kanalkonfiguration

Kanäle können über folgendes Maschinendatum mit einem eigenen Kanalnamen belegt werden:

MD20000 \$MC\_CHAN\_NAME (Kanalname)

Die verschiedenen Achsen werden per Maschinendatum den vorhandenen Kanälen zugeordnet. Für eine Achse/Spindel kann es zu einer Zeit immer nur einen Sollwertgebenden Kanal geben. Der Istwert der Achse/Spindel kann von mehreren Kanälen gleichzeitig gelesen werden. Die Achse/Spindel muss dem jeweiligen Kanal bekannt sein.

Des Weiteren können noch folgende kanalspezifische Einstellungen über Maschinendaten definiert werden:

- Löschstellungen bzw. Programmiergrundstellungen von G-Gruppen über das Maschinendatum:  
MD20150 \$MC\_GCODE\_RESET\_VALUES (Löschstellung der G-Gruppen)
- Hilfsfunktionsgruppen bezüglich der Zusammenstellung und des Ausgabezeitpunkts
- Transformationsbedingungen zwischen den Maschinenachsen und den Geometrieachsen
- Weitere Einstellungen für die Abarbeitung eines Teileprogramms

## Änderung der Kanalzuordnung

Eine Änderung der Kanalkonfiguration kann online nicht auf programmtechnischem Wege in einem Teileprogramm oder über das PLC-Anwenderprogramm erfolgen. Änderungen in der Konfiguration müssen über die Maschinendaten erfolgen. Die Änderungen sind erst nach einem erneuten POWER ON wirksam.

## Containerachsen und Linkachsen

Ein Achscontainer fasst eine Gruppe von Achsen eines Containers zusammen. Diese Achsen werden als Containerachsen bezeichnet. Hierbei wird einer Kanalachse ein Zeiger auf einem Containerslot (Ringpuffer-Platz innerhalb des entsprechenden Containers) zugeordnet. In diesem Slot befindet sich zwischenzeitlich eine der in diesem Container zusammengefassten Achsen.

Maschinenachsen im Achscontainer müssen zu jedem Zeitpunkt jeweils genau **einer** Kanalachse zugewiesen sein.

Link-Achsen können fest einem Kanal oder dynamischen (Achscontainer-Switch) mehreren Kanälen der lokalen, oder der anderen NCU zugeordnet werden. Sie sind aus Sicht einer der Kanäle, an deren NCU die Achse nicht physikalisch angeschlossen ist, eine nicht lokale Achse.

## 8.4 Kanal

Die herbeigeführte Zuordnung der Linkachsen zu einem Kanal erfolgt:

- Für feste Zuordnung über Maschinendatum:  
Direktes logisches Maschinenachsbild auf Linkachsen zeigen lassen.
- Für dynamische Zuordnung:  
Achcontainerslot - Maschinendatum auf Linkachsen zeigen lassen.

Weitere Informationen zu Link- und Containerachsen finden Sie in:

### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3)

## Nahtstellensignale

Die Signale des 1. Kanals liegen in der NC/PLC-Nahtstelle im DB21, die vom Kanal 2 im DB22. Der oder die Kanäle können von der PLC bzw. NCK überwacht bzw. beeinflusst werden.

## Kanalspezifische Technologie-Angabe

Für jeden Kanal kann die verwendete Technologie angegeben werden:

MD27800 \$MC\_TECHNOLOGY\_MODE

Im Auslieferungszustand sind standardmäßig Maschinendaten für die Technologie Fräsen aktiv.

## Spindelfunktionen über PLC

Spindelfunktionen können neben dem Funktionsbaustein FC18 auch über axiale NC/PLC-Nahtstellensignale parallel zu laufenden Teileprogrammen gestartet und gestoppt werden.

Voraussetzungen:

- Kanalzustand: "Unterbrochen" oder "RESET"
- Programmzustand: "Unterbrochen" oder "Abgebrochen"

Folgende Funktionen können von der PLC über Nahtstellensignale gesteuert werden:

- Stopp (entspricht M5)
- Start mit Drehrichtung rechts (entspricht M3)
- Start mit Drehrichtung links (entspricht M4)
- Getriebestufe auswählen
- Positionieren (entspricht M19)

Bei mehreren Kanälen wird die von der PLC gestartete Spindel in dem Kanal aktiv, dem sie zum Startzeitpunkt zugeordnet ist.

Weitere Informationen zur speziellen Spindelschnittstelle siehe Kapitel "S1: Spindeln (Seite 1383)".

## PLC-kontrollierte Einzelachsvorgänge

Statt von einem Kanal, kann eine Achse auch von der PLC kontrolliert werden. Dazu fordert die PLC über die NC/PLC-Nahtstelle die Achse von der NC an:

DB31, ... DBX28.7 = 1 (PLC kontrolliert Achse)

Folgende Funktionen können vom PLC aus kontrolliert und beeinflusst werden:

- Ablauf von Achse/Spindel abbrechen (entspricht Restweg löschen)
- Achse/Spindel stoppen bzw. unterbrechen
- Achse/Spindel weiterfahren (Bewegungsablauf fortsetzen)
- Achse/Spindel in den Grundzustand zurücksetzen

Weitere Informationen zum kanalspezifischen Signalaustausch (PLC → NCK) siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)".

Die genaue Funktionalität von autarken Einzelachsvorgängen ist beschrieben in:

### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

## 8.4.1 Globale Startsperrung für Kanal

### Bedienung/PLC

Durch Bedienung über HMI oder von der PLC kann eine globale Startsperrung für den angewählten Kanal gesetzt werden.

### Funktion

Wenn Startsperrung gesetzt ist, werden für den angewählten Kanal keine Starts neuer Programme angenommen. Startversuche werden intern gezählt.

Wenn noch vor dem Eintreffen der globalen Startsperrung durch HMI im NCK ein Start durch PLC ausgeführt wurde, wird das Programm durch die Startsperrung nicht stillgesetzt, sondern dieser Zustand an HMI übermittelt.

NC-Start-Sperre und globale Start-Sperre wirken gleichartig auf den internen Zähler für eingetretene, aber nicht ausgeführte Starts. (BTSS-Variable startRejectCount).

### Umgehung der globalen Startsperr

Das Nahtstellensignal:

DB21, ... DBX7.5 (PLC → NCK)

erlaubt der PLC eine globale Startsperr temporär zu durchbrechen.

0: Globale Startsperr ist wirksam

1: Globale Startsperr ist temporär aufgehoben

### Meldungen

Ein Startversuch unter globaler Startsperr kann auf Wunsch gemeldet werden.

Die Steuerung erfolgt über das Maschinendatum:

MD11411 \$MN\_ENABLE\_ALARM\_MASK Bit 6

1: Alarm 16956 erscheint: Kanal %1, Programm %2 kann wg. "Globaler Startsperr" nicht gestartet werden.

0: Startversuche bei gesetzter globaler Startsperr werden nicht durch Alarm gemeldet.

## 8.5 Programmtest

Zum Testen bzw. Einfahren eines neuen Teileprogramms gibt es mehrere Steuerungsfunktionen. Durch die Verwendung dieser Funktionen wird eine Gefährdung der Maschine während der Testphase bzw. der Zeitaufwand dafür stark verringert. Es ist möglich, mehrere Programmtestfunktionen gleichzeitig zu aktivieren, um ein besseres Ergebnis zu bekommen.

### Testmöglichkeiten

Es werden folgende Testmöglichkeiten beschrieben:

- Programmbearbeitung ohne Sollwertausgaben
- Programmbearbeitung im Einzelsatzbetrieb
- Programmbearbeitung mit Probelaufvorschub
- Teileprogrammsätze ausblenden
- Satzsuchlauf mit oder ohne Berechnung

### 8.5.1 Programmbearbeitung ohne Sollwertausgaben

#### Funktion

Im Zustand "Programmtest" wird ein Teileprogramm abgearbeitet, ohne dass Achs- oder Spindel-Sollwerte ausgegeben werden.

Damit kann der Anwender die programmierten Achspositionen sowie die Hilfsfunktionsausgaben eines Teileprogramms kontrollieren. Außerdem kann diese Programmsimulation als erweiterter Syntax-Check verwendet werden.

#### Anwahl

Die Auswahl der Funktion erfolgt über die Bedienoberfläche im Menü "Programmbeeinflussungen".

Mit der Auswahl wird folgendes Nahtstellensignal gesetzt:

DB21, ... DBX25.7 (Programmtest angewählt)

Die Funktion wird damit noch nicht aktiviert.

### Aktivierung

Die Aktivierung der Funktion erfolgt über das Nahtstellensignal:

DB21, ... DBX1.7 (Programmtest aktivieren)

### Anzeige

Als Rückmeldung des aktiven Programmtests wird auf der Bedienoberfläche das entsprechende Feld invers geschaltet und in der PLC das folgende Nahtstellensignal gesetzt:

DB21, ... DBX33.7 (Programmtest aktiv)

### Programmstart und Programmablauf

Das Teileprogramm kann bei aktiver Programmtest-Funktion gestartet und abgearbeitet (inkl. Hilfsfunktionsausgaben, Wartezeiten, G-Funktionsausgaben etc) werden über das Nahtstellensignal:

DB21, ... DBX7.1 (NC-Start)

Sicherheitsfunktionen wie Softwareendschalter, Arbeitsfeldbegrenzungen sind weiterhin gültig.

Der einzige Unterschied zum normalen Programmablauf besteht darin, dass für alle Achsen (einschließlich Spindeln) intern **Achsensperre** gegeben ist. Die Maschinenachsen bewegen sich also nicht, die Istwerte werden intern aus den nicht ausgegebenen Sollwerten generiert. Die programmierten Geschwindigkeiten bleiben unverändert. Das bedeutet, dass die Positions- und Geschwindigkeitsangaben auf der Bedienoberfläche genau denen einer normalen Teileprogrammbearbeitung entsprechen. Die Lageregelung wird dabei nicht unterbrochen, so dass nach Abschalten der Funktion die Achsen nicht referiert werden müssen.

#### VORSICHT

Die Signale für Genauhalt:  
DB31, ... DBX60.6/60.7 (Genauhalt grob/fein)  
spiegeln den tatsächlichen Zustand an der Maschine wieder.

Während des Programmtests würden sie nur weggenommen, wenn die Achse aus ihrer (während des Programmtests konstanten) Sollposition weggedrückt würde.

Mit dem Signal:  
DB21, ... DBX33.7 (Programmtest aktiv)  
kann sowohl das PLC-Programm als auch das Teileprogramm mit der Variablen  
\$P\_ISTEST entscheiden, wie während des Tests auf diese Signale zu reagieren oder zu  
verzweigen ist.



---

**Hinweis**

**Probelaufvorschub**

Die "Programmbearbeitung ohne Achsbewegungen" kann auch zusammen mit der Funktion "Probelaufvorschub" aktiviert werden. Dadurch werden Teileprogrammabschnitte mit einem kleinen programmierten Vorschubwert in einem kürzeren Zeitabschnitt abgearbeitet.

---

**Hinweis**

**Werkzeugverwaltung**

Aufgrund der Achsensperre wird die Belegung eines Werkzeug-Magazins beim Programmtest nicht verändert. Über eine PLC-Applikation muss sichergestellt werden, dass die Konsistenz zwischen den Daten der Werkzeugverwaltung und dem Magazin nicht verloren geht. Auf den Toolbox-Disketten finden Sie dazu beim PLC-Grundprogramm ein Beispiel.

---

## 8.5.2 Programmbearbeitung im Einzelsatzbetrieb

### Funktion

Bei der "Programmbearbeitung im Einzelsatzbetrieb" stoppt die Teileprogrammbearbeitung nach jedem Programmsatz. Falls eine Fräser- oder Schneiden-Radiuskorrektur angewählt ist, so stoppt die Bearbeitung nach jedem von der Steuerung eingefügten Zwischensatz.

Der Programmzustand wechselt auf "Programmzustand angehalten".

Der Kanalzustand bleibt auf aktiv.

Mit NC-Start gelangt der nächste Teileprogrammsatz zur Abarbeitung.

### Anwendung

Der Anwender kann damit ein Teileprogramm Satz für Satz abarbeiten und die einzelnen Bearbeitungsschritte kontrollieren. Wenn er den abgearbeiteten Teileprogrammsatz für korrekt befunden hat, kann er den nächsten Satz anfordern.

## Einzelsatztypen

Es wird zwischen folgenden Einzelsatztypen unterschieden:

- Dekodier-Einzelsatz

Bei diesem Einzelsatztyp werden alle Sätze des Teileprogramms (auch die reinen Rechensätze ohne Verfahrbewegungen) nacheinander durch "NC-Start" abgearbeitet.

- Aktions-Einzelsatz (Grundeinstellung)

Bei diesem Einzelsatztyp werden nur die Sätze einzeln abgearbeitet, die Aktionen (Verfahrbewegungen, Hilfsfunktionsausgaben usw.) auslösen.

Auch die bei der Dekodierung zusätzlich generierten Sätze (z. B. bei Fräser-Radiuskorrektur an spitzen Winkeln) werden einzeln abgefahren.

Bei Rechensätzen wird dagegen die Abarbeitung nicht angehalten, da diese keine Aktionen auslösen.

Die Festlegung des Einzelsatztyps erfolgt über die Bedienoberfläche im Menü "Programmbeeinflussungen".

### VORSICHT

Bei einer Serie von G33/G34/G35-Sätzen ist Einzelsatz nur dann wirksam, wenn "Probelaufvorschub" angewählt ist.

Rechensätze werden nicht im Einzelschritt bearbeitet (nur beim Dekodier-Einzelsatz).

SBL2 ist auch bei G33/G34/G35 unwirksam.

## Anwahl

Die Anwahl des Einzelsatzbetriebs ist möglich:

- Über die Maschinensteuertafel (Taste "Single Block")
- Über die Bedienoberfläche

Zur genauen Vorgehensweise siehe:

### Literatur:

Bedienungshandbuch der installierten HMI-Applikation

## Aktivierung

Die Aktivierung der Funktion erfolgt durch das PLC-Grundprogramm über das Nahtstellensignal:

DB21, ... DBX0.4 (Einzelsatz aktivieren)

## Anzeige

Als Rückmeldung des aktiven Einzelsatzbetriebs wird auf der Bedienoberfläche das entsprechende Feld in der Statuszeile invers geschaltet.

Sobald die Teileprogrammbearbeitung wegen des Einzelsatzbetriebs einen Teileprogrammabsatz abgearbeitet hat, wird das folgende Nahtstellensignal gesetzt:

DB21, ... DBX35.3 (Programmzustand unterbrochen)

## Abarbeiten ohne Einzelsatzstopp

Für bestimmte Programmabläufe kann trotz angewählten Einzelsatzbetriebs ein Abarbeiten ohne Einzelsatzstopp eingestellt werden, z. B. für:

- Interne ASUPs
- Anwender-ASUPs
- Zwischensätze
- Satzsuchlaufsammelsätze (Aktionssätze)
- Init-Sätze
- Unterprogramme mit `DISPLOF`
- Nicht-reorganisierbare Sätze
- Nicht-repositionierbare Sätze
- Wiederanfahrsatz ohne Fahrhinformation
- Werkzeuganfahrsatz

Die Einstellung erfolgt über das Maschinendatum:

MD10702 \$MN\_IGNORE\_SINGLEBLOCK\_MASK (Einzelsatzstopp verhindern)

### Literatur:

Listenhandbuch Ausführliche Beschreibung der Maschinendaten

## 8.5.3 Programmbearbeitung mit Probelaufvorschub

### Funktion

Bei der "Programmbearbeitung mit Probelaufvorschub" werden die Verfahrensgeschwindigkeiten, die in Verbindung mit G01, G02, G03, G33, G34 und G35 programmiert sind, durch einen parametrisierten Vorschubwert ersetzt:

SD42100 \$SC\_DRY\_RUN\_FEED (Probelaufvorschub)

Der Probelaufvorschubwert gilt auch anstelle des programmierten Umdrehungsvorschubs in Programmsätzen mit G95.

Die genaue Wirkungsweise des parametrisierten Probelaufvorschubs (SD42100) ist von der Einstellung eines weiteren Settingdatums abhängig:


SD42101 \$SC\_DRY\_RUN\_FEED\_MODE (Modus für Testlauf-Geschwindigkeit)

Wert	Bedeutung
0	Als Probelaufvorschub wirkt das Maximum von SD42100 und dem programmierten Vorschub. (Grundeinstellung!) ⇒ SD42100 wird nur wirksam, wenn der hinterlegte Wert größer ist als der programmierte Vorschub.
1	Als Probelaufvorschub wirkt das Minimum von SD42100 und dem programmierten Vorschub. ⇒ SD42100 wird nur wirksam, wenn der hinterlegte Wert kleiner ist als der programmierte Vorschub.
2	Als Probelaufvorschub wirkt der Wert in SD42100, unabhängig vom programmierten Vorschub.
10	Wie bei "0" außer Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Diese Funktionen werden wie programmiert ausgeführt.
11	Wie bei "1" außer Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Diese Funktionen werden wie programmiert ausgeführt.
12	Wie bei "2" außer Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Diese Funktionen werden wie programmiert ausgeführt.

Ein Probelaufvorschub kann in den Automatikbetriebsarten angewählt und bei einer Automatikunterbrechung oder am Ende eines Satzes aktiviert werden.

Weitere Informationen zur Vorschubbeeinflussung siehe Kapitel "V1: Vorschübe (Seite 1507)".

## Anwendung

 <b>GEFAHR</b>
Bei aktiver Funktion "Probelaufvorschub" darf keine Werkstückbearbeitung erfolgen, da durch die geänderten Vorschubwerte die Schnittgeschwindigkeiten der Werkzeuge überschritten bzw. das Werkstück oder die Werkzeugmaschine zerstört werden könnte.

## Anwahl

Die Anwahl der Funktion erfolgt über die Bedienoberfläche im Menü "Programmbeeinflussungen".

Mit der Anwahl wird folgendes Nahtstellensignal gesetzt:

DB21, ... DBX24.6 (Probelaufvorschub angewählt)

Die Funktion wird damit noch nicht aktiviert.

## Aktivierung

Die Aktivierung der Funktion erfolgt über das Nahtstellensignal:

DB21, ... DBX0.6 (Probelaufvorschub aktivieren)

## Anzeige

Als Rückmeldung des aktiven Probelaufvorschubs wird auf der Bedienoberfläche das entsprechende Feld in der Statuszeile invers geschaltet.

## 8.5.4 Teileprogrammsätze ausblenden

### Funktion

Beim Testen bzw. Einfahren neuer Programme ist es hilfreich, wenn bestimmte Teileprogrammsätze für die Programmabarbeitung gesperrt bzw. ausgeblendet werden. Dazu müssen die betreffenden Sätze mit Schrägstrich gekennzeichnet werden.

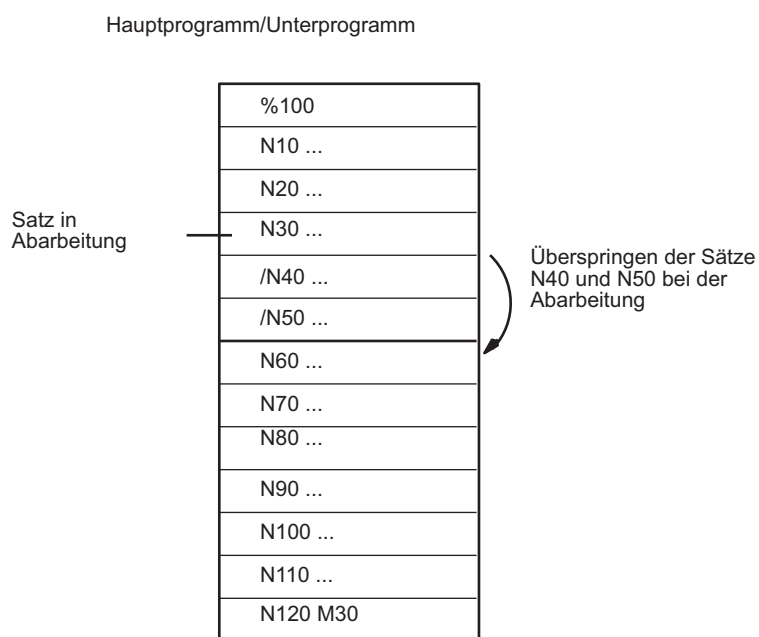


Bild 8-1 Ausblenden von Teileprogrammsätzen

### **Anwahl**

Die Auswahl der Funktion erfolgt über die Bedienoberfläche im Menü "Programmbeeinflussungen".

Mit der Auswahl wird folgendes Nahtstellensignal gesetzt:

DB21, ... DBX26.0 (Satz ausblenden angewählt)

Die Funktion wird damit noch nicht aktiviert.

### **Aktivierung**

Die Aktivierung dieser Funktion erfolgt über das Nahtstellensignal:

DB21, ... DBX2.0 (Satz ausblenden aktivieren)

---

#### **Hinweis**

Die Funktion "Teileprogrammsätze ausblenden" ist auch bei einem Satzsuchlauf aktiv.

---

### **Anzeige**

Als Rückmeldung der aktivierten Funktion "Teileprogrammsätze ausblenden" wird auf der Bedienoberfläche das entsprechende Feld in der Statuszeile invers geschaltet.

## 8.6 Werkstücksimulation

### Funktion

In der Werkstücksimulation wird das aktuelle Teileprogramm vollständig berechnet und das Ergebnis in der Bedienoberfläche grafisch dargestellt. Ohne die Maschinenachsen zu verfahren, wird so das Ergebnis der Programmierung kontrolliert. Falsch programmierte Bearbeitungsschritte werden frühzeitig erkannt und Fehlbearbeitungen am Werkstück verhindert.

### Simulations-NCK

Die Simulation nutzt eine eigene NCK-Instanz (Simulations-NCK). Deshalb muss vor einem Simulationsstart ein Abgleich des realen NCK zum Simulations-NCK erfolgen. Bei diesem Abgleich werden unter anderem sämtliche aktive Maschinendaten aus dem NCK ausgelesen und in den Simulations-NCK eingelesen. Unter den aktiven Maschinendaten befinden sich die NCK- und Zyklusmaschinendaten.

### Compile-Zyklen in Simulation (nur 840D sl)

Bis SW 4.4 werden keine, ab SW 4.4 nur ausgesuchte Compile-Zyklen (CC) bei der Werkstücksimulation unterstützt. Die Maschinendaten der unterstützten Compile-Zyklen werden einmalig nach Steuerungshochlauf abgeglichen. Ein Abgleich mit "Simulationsstart" findet nicht statt!

---

#### Hinweis

In Teileprogrammen können CC-spezifische Sprachbefehle und Maschinendaten von nicht unterstützten CCs **nicht** benutzt werden (siehe auch Absatz "CC-Befehle im Teileprogramm").

Sonderbewegungen von unterstützten CCs (OEM-Transformationen) werden unter Umständen falsch dargestellt.

---

### CC-Befehle im Teileprogramm

Sprachbefehle im Teileprogramm von nicht unterstützten Compile-Zyklen (OMA1 ... OMA5, OEMIPO1/2, G810 ... G829, eigene Prozeduren und Funktionen) führen ohne individuelle Behandlung zur Alarmmeldung und zum Abbruch der Simulation.

**Lösung:** Die fehlenden CC-spezifischen Sprachelemente im Teileprogramm individuell behandeln (\$P\_SIM Abfrage). Beispiel:

Programmcode	Kommentar
N1 G01 X200 F500	
IF (1== \$P_SIM)	
N5 X300	; Bei Simulation CC nicht aktiv.
ELSE	
N5 X300 OMA1=10	
ENDIF	



## 8.7 Satzsuchlauf

### Funktion

Satzsuchlauf bietet die Möglichkeit, die Abarbeitung eines Teileprogramms von einem nahezu beliebigen Teileprogrammsatz aus zu beginnen.

Beim Satzsuchlauf erfolgt ein NC-interner Schnelldurchlauf ohne Verfahrbewegungen durch das Teileprogramm bis zum gewählten Zielsatz. Dabei wird versucht, möglichst exakt den Steuerungszustand zu erzielen, wie er sich am Zielsatz bei der normalen Teileprogrammabarbeitung (z. B. bezüglich Achspositionen, Spindeldrehzahlen, eingewechselte Werkzeuge, NC/PLC-Nahtstellensignale, Variablenwerte) ergibt, um mit möglichst wenigen manuellen Eingriffen ab dem Zielsatz mit der automatischen Teileprogrammabarbeitung fortfahren zu können.

### Satzsuchlauftypen

- **Typ 1: Satzsuchlauf ohne Berechnung**

Satzsuchlauf ohne Berechnung dient zum schnellstmöglichen Auffinden eines Teileprogrammsatzes. Es wird dabei keinerlei Berechnung durchgeführt. Der Steuerungszustand am Zielsatz entspricht unverändert dem vor dem Start des Satzsuchlaufs.

- **Typ 2: Satzsuchlauf mit Berechnung an Kontur**

Satzsuchlauf mit Berechnung an Kontur dient dazu, um in beliebigen Situationen an die programmierte Kontur anfahren zu können. Mit NC-START wird die Anfangsposition des Zielsatzes bzw. die Endposition des Satzes vor dem Zielsatz angefahren. Dieser wird bis zur Endposition abgefahren. Die Abarbeitung erfolgt konturtreu.

- **Typ 4: Satzsuchlauf mit Berechnung an Satzendpunkt**

Satzsuchlauf mit Berechnung an Satzendpunkt dient dazu, um in beliebigen Situationen eine Zielposition (z. B. Werkzeugwechselposition) anfahren zu können. Angefahren wird der Endpunkt des Zielsatzes bzw. die nächste programmierte Position unter Verwendung der im Zielsatz gültigen Interpolationsart. Dies geschieht nicht konturtreu.

Es werden nur die im Zielsatz programmierten Achsen gefahren. Gegebenenfalls muss an der Maschine vor dem Start der weiteren automatischen Teileprogrammbearbeitung manuell über die Betriebsart "JOG-REPOS" eine kollisionsfreie Ausgangssituation geschaffen werden.

- **Typ 5: Satzsuchlauf mit Berechnung im Modus "Programmtest" (SERUPRO)**

SERUPRO (**search run by programtest**) ist ein Kanal-übergreifender Satzsuchlauf mit Berechnung. Die NC startet dabei das angewählte Teileprogramm im Modus "Programmtest". Mit Erreichen des Zielsatzes wird der Programmtest automatisch wieder abgewählt. Bei diesem Satzsuchlauftyp sind auch Interaktionen zwischen dem Kanal, in dem der Suchlauf stattfindet, und Synchronaktionen und anderen Kanälen der NC möglich.

---

**Hinweis**

Weitere Erläuterungen zum Satzsuchlauf siehe Kapitel "Verhalten bei Satzsuchlauf (Seite 447)".

---

## Folgeaktionen

Nach dem Abschluss eines Satzsuchlaufs können folgende Folgeaktionen erfolgen:

- Typ1 - Typ 5: Automatischer Start eines ASUPS

Mit dem Einwechseln des letzten Aktionssatzes kann ein Anwenderprogramm als ASUP gestartet werden.

- Typ1 - Typ 4: Kaskadierter Satzsuchlauf

Aus dem Zustand "Suchziel gefunden" kann ein weiterer Suchlauf mit geänderter Suchzielangabe gestartet werden.

## 8.7.1 Ablauf für Satzsuchlauf vom Typ 1, 2 und 4

### Zeitlicher Ablauf

Der Satzsuchlauf der Typen 1, 2 und 4 läuft wie folgt ab:

1. Aktivierung über die Bedienoberfläche
2. Suchziel gefunden oder Alarm wenn kein Suchziel gefunden wurde
3. NC-Start für Ausgabe der Aktionssätze
4. NC-Start für Programmfortsetzung

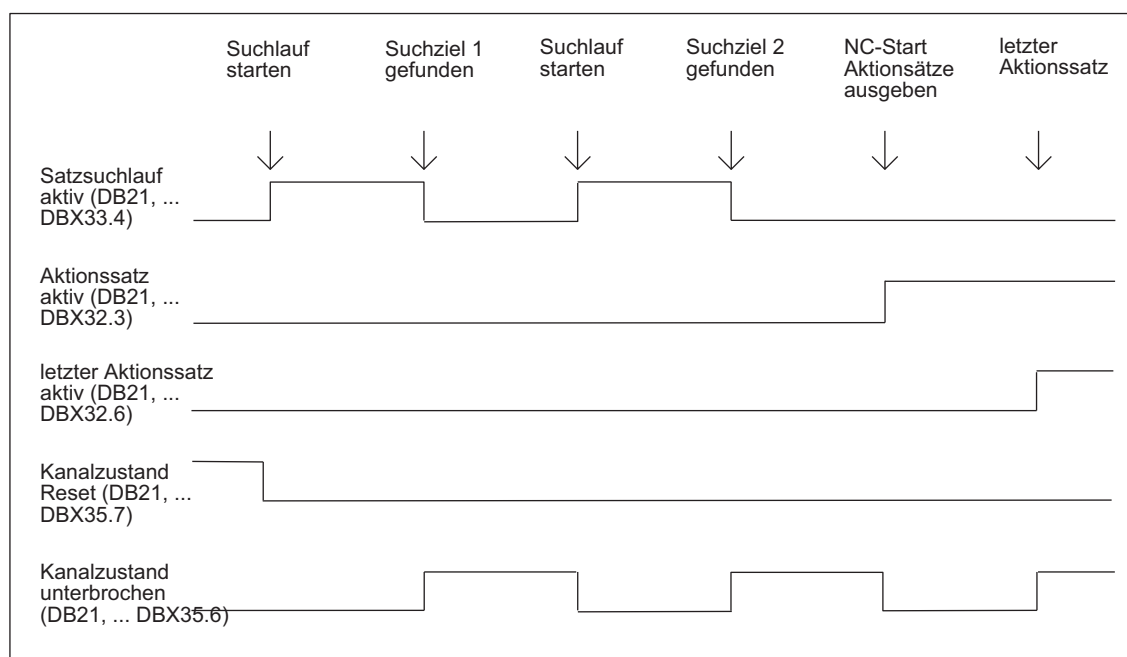


Bild 8-2 Zeitlicher Ablauf der Nahtstellensignale

### Nahtstellensignale

In der PLC werden nach dem im Bild dargestellten zeitlichen Ablauf die folgenden Nahtstellensignale gesetzt:

- DB21, ... DBX33.4 (Satzsuchlauf aktiv)
- DB21, ... DBX32.3 (Aktionssatz aktiv)
- DB21, ... DBX32.4 (Anfahrsatz aktiv)
- DB21, ... DBX32.6 (letzter Aktionssatz aktiv)
- DB21, ... DBX1.6 (PLC-Aktion beendet)

### Fortsetzung und Aufsetzmodus nach Satzsuchlauf

Nach dem Satzvorlauf kann das Programm zwecks Fortsetzung über das Nahtstellensignal: DB21, ... DBX7.1 (NC-Start) gestartet werden.

Erfolgt die erste Programmierung einer Achse nach "Satzsuchlauf mit Berechnung an Satzendpunkt", so kann mit Settingdatum SD42444 \$SC\_TARGET\_BLOCK\_INCR\_PROG der inkrementelle Wert auf den bis Suchziel aufgesammelten Wert addiert werden.

### Aktionssätze

Aktionssätze beinhalten die während "Satzsuchlauf mit Berechnung" aufgesammelten Aktionen, wie z. B. Hilfsfunktionsausgaben, Werkzeug- (T, D), Spindel- (S), Vorschub-Programmierung. Während "Satzsuchlauf mit Berechnung" (Kontur oder Satzendpunkt) werden Aktionen wie z. B. M-Funktionsausgaben in so genannten Aktionssätzen aufgesammelt. Mit NC-Start nach "Suchziel gefunden" werden diese Sätze ausgegeben.

---

#### Hinweis

Mit den Aktionssätzen wird auch die aufgesammelte Spindelprogrammierung (S-Wert, M3/M4/M5/M19, SPOS) aktiv.

Das PLC-Anwenderprogramm muss sicherstellen, dass das Werkzeug betrieben werden kann und ggf. die Spindelprogrammierung über PLC-Signal: DB31, ... DBX2.2 (Spindelreset) zurückgesetzt bzw. die Spindelprogrammierung nicht ausgegeben wird.

---

Einzelatzbearbeitung: MD10702 \$MN\_IGNORE\_SINGLEBLOCK\_MASK ()

Durch Setzen von Bit 3 = 1 kann verhindert werden, dass im Einzelatz nach jedem Aktionssatz angehalten wird.

### Randbedingungen Anfahratz/Zielsatz

#### Satzsuchlauf Typ 2

Das Nahtstellensignal: DB21, ... DBX32.4 (Anfahratz aktiv) wird **nur** bei "Satzsuchlauf mit Berechnung an Kontur" gesetzt, da bei "Satzsuchlauf mit Berechnung an Satzendpunkt" kein eigener Anfahratz erzeugt wird (Anfahratz ist gleich Zielsatz).

#### Satzsuchlauf Typ 4

Die Anfahrbewegung "Suchlauf mit Berechnung an Satzendpunkt" wird in der im Zielsatz gültigen Interpolationsart durchgeführt. Sinnvollerweise sollte dies G0 oder G1 sein. Bei anderen Interpolationsarten kann die Anfahrbewegung mit Alarm abgebrochen werden (z. B. Kreisendpunktfehler bei G2/G3).

## 8.7.2 Satzsuchlauf im Zusammenhang mit weiteren NCK-Funktionen

### 8.7.2.1 ASUP nach und bei Satzsuchlauf

#### Synchronisation der Kanalachsen

Mit dem Start eines ASUPs nach "Satzsuchlauf mit Berechnung" werden im Vorlauf die Istpositionen aller Kanalachsen synchronisiert.

Auswirkungen:

- Systemvariable: \$P\_EP (Programmierte Endposition)

Im ASUP liefert die Systemvariable: \$P\_EP (Programmierte Endposition) die aktuelle Istposition einer Kanalachse im Werkstück-Koordinatensystem.

\$P\_EP == "aktuelle Istposition der Kanalachse (WKS)"

- Systemvariable: \$AC\_RETPOINT (Wiederanfahrpunkt im ASUP)

Im ASUP liefert die Systemvariable: \$AC\_RETPOINT (Wiederanfahrpunkt im ASUP) die mit Satzsuchlauf aufgesammelte Istposition einer Kanalachse im Werkstück-Koordinatensystem.

\$AC\_RETPOINT == "aufgesammelte Suchlaufposition der Kanalachse (WKS)"

#### Satzsuchlauf Typ 2

Bei Satzsuchlauf Typ 2 (Satzsuchlauf mit Berechnung an Kontur) muss zum Abschluss des ASUPs folgender Teileprogrammbefehl programmiert werden:

REPOSA (Wiederanfahren an die Kontur; linear; alle Kanalachsen)

Auswirkung:

- Alle Kanalachsen werden auf ihre im Satzsuchlauf aufgesammelte Suchlaufposition verfahren.
- \$P\_EP == "aufgesammelte Suchlaufposition der Kanalachse (WKS)"

#### Satzsuchlauf Typ 4 und Teileprogrammbefehl REPOS

Nach Satzsuchlauf Typ 4 (Satzsuchlauf mit Berechnung an Satzendpunkt) wird während des folgenden Zeitraums durch den Teileprogrammbefehl REPOS kein automatisches Repositionieren ausgelöst:

- Beginn: NC/PLC-Nahtstellensignals: DB21,... DBB32, Bit6 (letzter Aktionssatz aktiv) == 1
- Ende: Fortsetzen der Teileprogrammbearbeitung per NC-START,.

Der Startpunkt der Anfahrbewegung sind die aktuellen Achspositionen der Kanalachsen zum Zeitpunkt des NC-START Kommandos. Der Endpunkt ergibt sich durch die weiteren im Teileprogramm programmierten Verfahrbewegungen.

Bei Satzsuchlauf Typ 4 wird durch die NC keine Anfahrbewegung erzeugt.

Auswirkung:

- Die Systemvariable \$P\_EP liefert damit nach Verlassen des ASUPs die Istposition, auf die die Kanalachsen vom ASUP oder manuell (Betriebsart: JOG) positioniert wurden.  
 \$P\_EP == "aktuelle Istposition der Kanalachse (WKS)"

### 8.7.2.2 PLC-Aktionen nach Satzsuchlauf

Um das Aktivieren von PLC-Aktionen (Starten von ASUPs, Aufruf von PLC-Funktionen) nach dem Ende des Satzsuchlaufs an einer definierten Stelle zu ermöglichen, gibt es das NCK/PLC-Nahtstellensignal:

DB21... DB32.6 (letzter Aktionssatz aktiv) == 1

Es hat die Bedeutung, dass alle Aktionssätze abgearbeitet sind und Aktionen seitens der PLC (ASUP, FC) oder des Bedieners (Überspeichern, Betriebsartenwechsel nach JOG/REPOS) möglich sind. So kann über PLC z. B. vor dem Start der Verfahrbewegung noch einen Werkzeugwechsel durchführen.

Defaultmäßig wird zu diesem Zeitpunkt der Alarm 10208 ausgegeben, um den Bediener darauf hinzuweisen, dass zur Fortsetzung der Programmbearbeitung noch NC-START erforderlich ist.

Zusammen mit dem Alarm 10208 werden folgende Nahtstellensignale gesetzt:

DB21, ... DBX36.7(NCK-Alarm mit Bearbeitungsstillstand)

DB21, ... DBX36.6 (NCK-Alarm kanalspezifisch steht an)

### PLC-gesteuerte Alarmausgabe

Die Einstellung, dass der Alarm 10208 erst nach dem Beenden der PLC-Aktion ausgegeben wird, erfolgt über Maschinendatum:

MD11450 \$MN\_SEARCH\_RUN\_MODE, Bit 0 = 1

Bit	Wert	Bedeutung
0	1	Mit dem Einwechseln des letzten Aktionssatzes nach Satzsuchlauf erfolgt: <ul style="list-style-type: none"> <li>Die Bearbeitung des Teileprogramms wird gestoppt</li> <li>DB21, ... DBB32.6 (letzter Aktionssatz aktiv) = 1</li> <li>Alarmanzeige: Alarm 10208 erst, wenn gilt: DB21, ... DBX1.6 (PLC-Aktion beendet) == 1</li> </ul>

### 8.7.2.3 Spindelfunktionen nach Satzsuchlauf

#### Steuerungsverhalten und Ausgabe

Das Verhalten bezüglich der Spindelfunktionen nach Beendigung des Satzsuchlaufs ist einstellbar über Maschinendatum:

MD11450 \$MN\_SEARCH\_RUN\_MODE, Bit 2

Bit	Wert	Bedeutung
2	0	Ausgabe der Spindelhilfsfunktionen (M3, M4, M5, M19, M70) erfolgt in den Aktionssätzen.
	1	Die Ausgabe der Hilfsfunktionen in den Aktionssätzen wird unterdrückt. Die bei Satzsuchlauf aufgesammelten Spindelprogrammierungen können zu einem späteren Zeitpunkt (z.B. durch ASUP) ausgegeben werden. Die Programmdateien werden dazu in folgenden Systemvariablen gespeichert: <ul style="list-style-type: none"> <li>• \$P_SEARCH_S</li> <li>• \$P_SEARCH_SDIR</li> <li>• \$P_SEARCH_SGEAR</li> <li>• \$P_SEARCH_SPOS</li> <li>• \$P_SEARCH_SPOSMODE</li> </ul>

#### Systemvariablen

Die Spindel-spezifischen Hilfsfunktionen werden bei Satzsuchlauf, unabhängig von der oben genannten Parametrierung, immer in den folgenden Systemvariablen gespeichert:

Systemvariable	Beschreibung
\$P_SEARCH_S [ n ]	Aufgesammelte Spindeldrehzahl, Wertebereich = { 0 ... Smax }
\$P_SEARCH_SDIR [ n ]	Aufgesammelte Spindeldrehrichtung, Wertebereich = { 3, 4, 5, -5, -19, 70 }
\$P_SEARCH_SGEAR [ n ]	Aufgesammelte Spindelgetriebest.-M-Funktion, Wertebereich = { 40 ... 45 }
\$P_SEARCH_SPOS [ n ]	Aufgesammelte Spindel-Position, Wertebereich = { 0 ... MD30330 \$MA_MODULO_RANGE } Aufgesammelter Verfahrensweg, Wertebereich = { -100.000.000 ... 100.000.000 }
\$P_SEARCH_SPOSMODE [ n ]	Aufgesammelter Positionsanfahrmode, Wertebereich = { 0 ... 5 }

8.7 Satzsuchlauf

Zur späteren Ausgabe der Spindel-spezifischen Hilfsfunktionen können die Systemvariablen z.B. in einem ASUP gelesen und nach Ausgabe der Aktionssätze ausgegeben werden: DB21, ... DBX32.6 == 1 (Letzter Aktionssatz aktiv)

**Hinweis**

Die Inhalte der Systemvariablen \$P\_S, \$P\_DIR und \$P\_SGEAR können nach Satzsuchlauf durch Synchronisationsvorgänge verloren gehen.

Für weiterführende Informationen zu ASUP, Satzsuchlauf und Aktionssätzen siehe Kapitel "Ausgabeunterdrückung von Spindel-spezifischen Hilfsfunktionen (Seite 453)" und "Programmtest (Seite 511)".

**8.7.2.4 Lesen von Systemvariablen bei Satzsuchlauf**

In der NC-Sprache stehen unterschiedliche Systemvariable zur Verfügung, um auf Werte der NC-Bereiche Vorlauf, Hauptlauf oder Servo/Antrieb zuzugreifen:

\$P_...	Systemvariable, die mit \$P beginnen, liefern den Vorlaufzustand (d. h. die programmierten Werte).
\$A_...	Systemvariable, die mit \$A beginnen, liefern den Hauptlaufzustand.
\$V_...	Systemvariable, die mit \$V beginnen, liefern Daten, die vom Servo/Antrieb kommen.

Da während eines Satzsuchlaufs vom Typ 2 und 4 keine Sätze in den Hauptlauf gelangen, ist zu beachten, dass Systemvariable die Hauptlauf- oder Servo-/Antriebszustände wiedergeben, durch den Suchlauf nicht verändert werden. Gegebenenfalls muss für diese Variable durch Abfrage der Bearbeitungsart mit \$P\_SEARCH im NC-Programm eine Sonderbehandlung für Satzsuchlauf durchgeführt werden.

Systemvariable, die mit \$P beginnen, können in allen Suchlaufarten bedenkenlos verwendet werden.

**8.7.3 Automatischer Start eines ASUP nach Satzsuchlauf**

**Parametrierung**

**Wirksamsetzen der Funktion**

Der automatische ASUP-Start nach Satzsuchlauf wird durch folgende MD-Einstellung aktiviert:

MD11450 \$MN\_SEARCH\_RUN\_MODE, Bit 1 = 1

**Zu aktivierendes Programm**

In der Grundstellung wird nach dem Satzsuchlauf mit dem Einwechseln des letzten Aktionssatzes das Programm **\_N\_PROG\_EVENT\_SPF** aus dem Verzeichnis **\_N\_CMA\_DIR** als ASUP aktiviert. Soll ein anderes Programm aktiviert werden, muss im folgenden Maschinendatum der Name dieses Anwenderprogramms eingetragen sein:

MD11620 \$MN\_PROG\_EVENT\_NAME



**Verhalten bei gesetzter Einzelsatzbearbeitung**

Über das folgende kanalspezifische Maschinendatum kann eingestellt werden, ob das aktivierte ASUP trotz gesetzter Einzelsatzbearbeitung ohne Unterbrechung abgearbeitet werden oder die Einzelsatzbearbeitung wirksam sein soll:

MD20106 \$MC\_PROG\_EVENT\_IGN\_SINGLEBLOCK

Bit	Wert	Bedeutung
4	0	Einzelsatzbearbeitung ist wirksam.
	1	Einzelsatzbearbeitung wird unterdrückt.

**Verhalten bei gesetzter Einlesesperre**

Über das folgende kanalspezifische Maschinendatum kann eingestellt werden, ob das ASUP trotz gesetzter Einlesesperre (DB21, ... DBX6.1 = 1) komplett abgearbeitet werden soll oder die Einlesesperre wirksam sein soll:

MD20107 \$MC\_PROG\_EVENT\_IGN\_INHIBIT

Bit	Wert	Bedeutung
4	0	Einlesesperre ist wirksam.
	1	Einlesesperre wird unterdrückt.

**Hinweis**

Weitere Informationen zur Parametrierung von MD11620, MD20108 und MD20107 siehe Kapitel "Parametrierung (Seite 603)".

**Programmierung**

Durch welches Ereignis das ASUP gestartet wurde, kann durch Abfrage der Systemvariablen \$P\_PROG\_EVENT festgestellt werden. Im Falle einer automatischen Aktivierung nach Satzsuchlauf liefert \$P\_PROG\_EVENT den Wert "5".

## Ablauf

Ablauf zum automatischen Start eines ASUP nach Satzsuchlauf:

1. Satzsuchlauf starten (mit/ohne Berechnung, an Kontur, an Satzendpunkt).
2. Stopp nach "Suchziel gefunden".
3. NC-Start für die Ausgabe der Aktionssätze.
4. Letzter Aktionssatz wird eingewechselt.
5. Automatischer Start von /\_N\_CMA\_DIR/\_N\_PROG\_EVENT\_SPF (Voreinstellung) als ASUP.
6. Mit dem Einwechseln des letzten ASUP-Satzes (`REPOSA`-Befehl) stoppt die NC und es wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB21, ... DBX32.6 (letzter Aktionssatz aktiv)

Alarm 10208 "Zur Programmfortsetzung NC-Start geben" wird ausgegeben.

---

### Hinweis

Wenn Bit 0 im MD11450 `$MN_SEARCH_RUN_MODE` auf "1" gesetzt ist, dann wird Alarm 10208 erst dann ausgegeben, wenn die PLC dies durch Setzen des folgenden NC/PLC-Nahtstellensignals anfordert:

DB21, ... DBX1.6 (PLC-Aktion beendet)

---

## 8.7.4 Kaskadierter Satzsuchlauf

### Funktionalität

Mit der Funktion "Kaskadierter Satzsuchlauf" ist es möglich aus dem Zustand "Suchziel gefunden" einen weiteren Suchlauf zu starten. Die Kaskadierung kann nach jedem gefundenen Suchziel beliebig oft fortgesetzt werden und ist für folgende Suchlauf-Funktionen anwendbar:

- Suchlauf Typ 1 ohne Berechnung
- Suchlauf Typ 2 mit Berechnung an Kontur
- Suchlauf Typ 3 mit Berechnung an Satzendpunkt

---

### Hinweis

Nur wenn das Suchziel gefunden wurde, kann aus der gestoppten Programmbearbeitung ein weiterer "Kaskadierter Satzsuchlauf" gestartet werden.

---

## Aktivierung

Der "Kaskadierte Satzsuchlauf" wird im bestehenden Maschinendatum projiziert:  
MD11450 \$MN\_SEARCH\_RUN\_MODE

- Mit Bit 3 = 0 (FALSE) wird der kaskadierte Satzsuchlauf frei geschaltet (d. h. mehrfache Suchzielvorgaben sind möglich).
- Aus Kompatibilitätsgründen kann der kaskadierte Satzsuchlauf mit Bit 3 = 1 (TRUE) gesperrt werden. Voreingestellt ist der kaskadierte Suchlauf Bit 3 = 0.

## Ablaufverhalten

### Suchziel gefunden, Suchlauf erneut starten

Mit dem Erreichen des Suchziels wird die Programmbearbeitung gestoppt und das Suchziel als aktueller Satz angezeigt. Nach jedem gefundenen Suchziel ist ein neuer Satzsuchlauf beliebig oft wiederholbar.

### Suchzielvorgaben ändern

Vor jeden Suchlaufstart können Suchzielangabe und Suchlauffunktion geändert werden.

**Beispiel: Ablauf einer Bearbeitungssequenz mit kaskadiertem Satzsuchlauf**

- RESET
- Satzsuchlauf bis Suchziel 1
- Satzsuchlauf bis Suchziel 2 → "Kaskadierter Satzsuchlauf"
- NC-Start für Ausgabe der Aktionssätze → Alarm 10208
- NC-Start → Fortsetzen der Programmbearbeitung

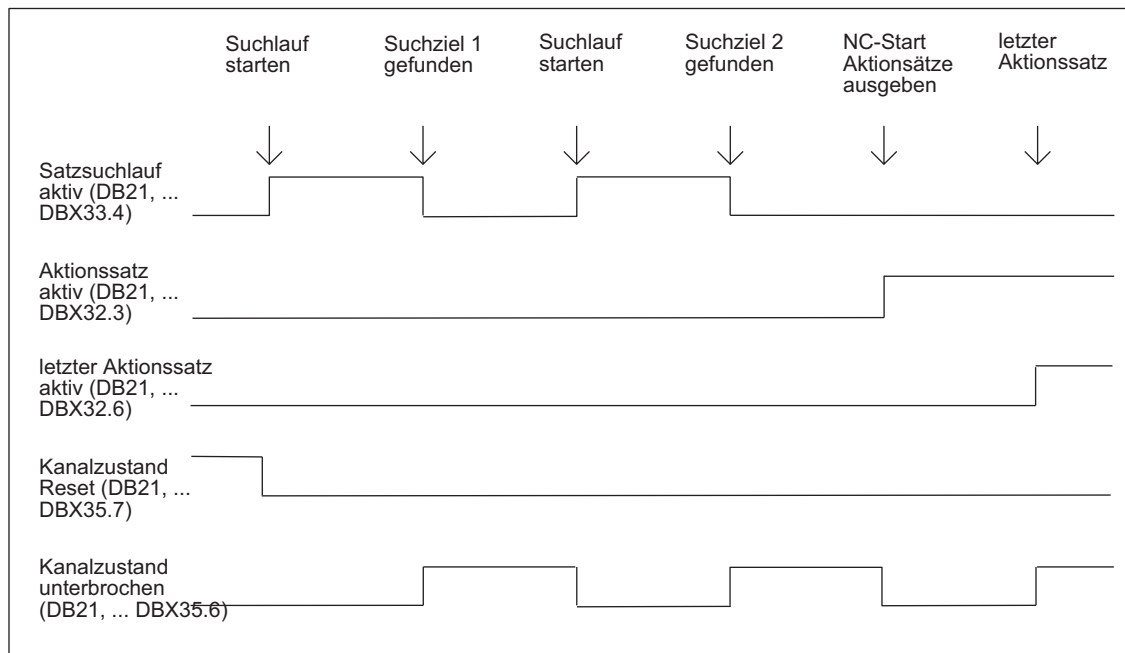


Bild 8-3 Zeitlicher Ablauf Nahtstellensignale

## 8.7.5 Beispiele zum Satzsuchlauf mit Berechnung

### Auswahl

Wählen Sie aus den folgenden Beispielen den Satzsuchlauftyp aus, der Ihrer Problemstellung am Besten entspricht.

### Satzsuchlauf Typ 4 mit Berechnung an Satzendpunkt

Beispiel mit automatischem Werkzeugwechsel nach Satzsuchlauf bei aktiver Werkzeugverwaltung:

1. Maschinendaten setzen:  
MD11450 \$MN\_SEARCH\_RUN\_MODE auf 1  
MD11602 \$MN\_ASUP\_START\_MASK Bit 0 = 1 (ASUP-Start aus gestopptem Zustand)
2. ASUP "SUCHLAUF\_ENDE" von PLC über FB4 anwählen (siehe auch Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)").
3. Teileprogramm "WERKSTUECK\_1" laden und anwählen.
4. Suchlauf auf Satzendpunkt Satznummer  $N_{220}$ .
5. HMI meldet "Suchziel gefunden".
6. NC-Start für die Ausgabe der Aktionssätze.
7. Mit dem PLC-Signal:  
DB21... DB32.6 (letzter Aktionssatz aktiv)  
startet die PLC über FC9 das ASUP "SUCHLAUF\_ENDE" (siehe auch Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)").
8. Nach ASUP-Ende (auswertbar z. B. über die zu definierende M-Funktion  $M_{90}$ , siehe Beispiel Satz  $N_{1110}$ ) setzt die PLC das Signal:  
DB21, ... DBX1.6 (PLC-Aktion beendet).

Alternativ kann auch das NC/PLC-Nahtstellensignal:  
DB21-DB30 DBB318 Bit 0 (ASUP ist angehalten)  
abgefragt werden.

Dadurch wird der Alarm 10208 angezeigt, d. h. es können jetzt weitere Aktionen seitens des Bedieners erfolgen.

9. Manuelle Eingriffe durch Bediener (JOG, JOG-REPOS, Überspeichern).

10. Teileprogramm mit NC-Start fortsetzen.

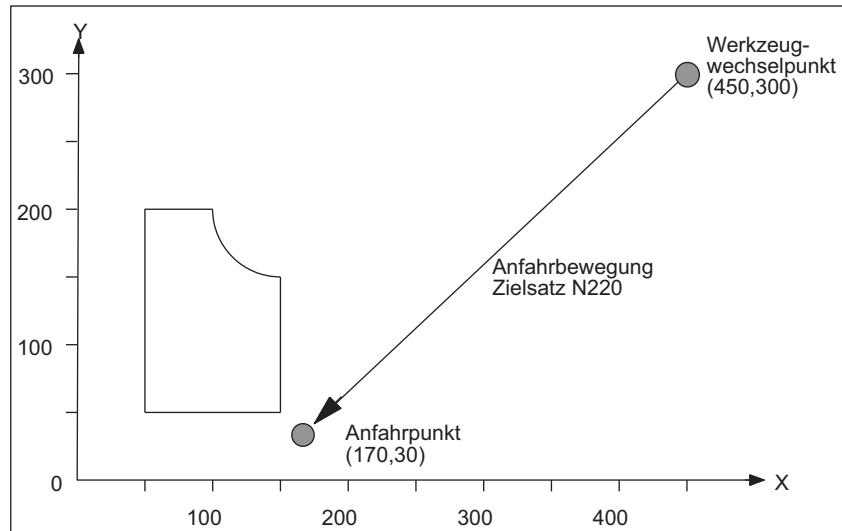


Bild 8-4 Anfahrbewegung bei Suchlauf auf Satzendpunkt (Zielsatz N220)

#### Hinweis

"Suchlauf an Kontur" mit Zielsatz N220 würde eine Anfahrbewegung zum Werkzeugwechsellpunkt (Startpunkt des Zielsatzes) erzeugen.

### Satzsuchlauf Typ 2 mit Berechnung an Kontur

Beispiel mit automatischem Werkzeugwechsel nach Satzsuchlauf bei aktiver Werkzeugverwaltung:

1. bis 3. wie Beispiel zum Satzsuchlauf Typ 4
4. Suchlauf an Kontur Satznummer N260
5. bis 10. wie Beispiel zum Satzsuchlauf Typ 4

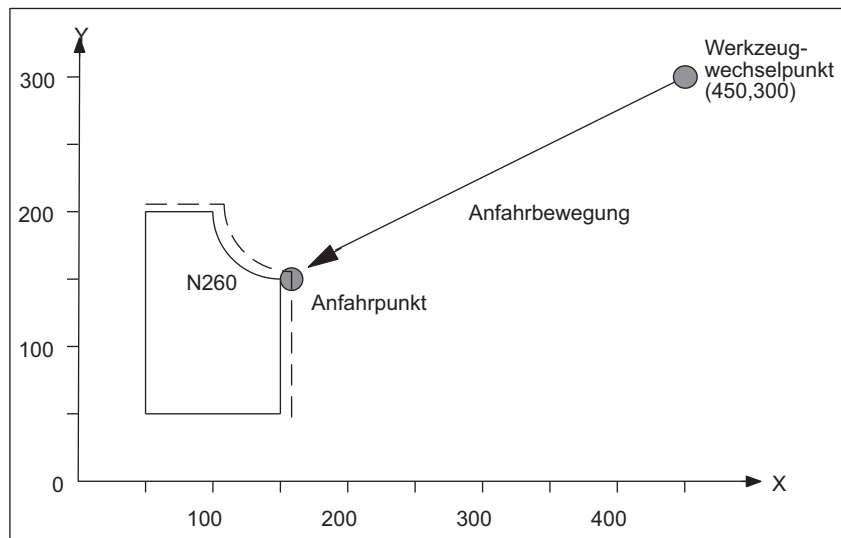


Bild 8-5 Anfahrbewegung bei Suchlauf an Kontur (Zielsatz N260)

**Hinweis**

"Suchlauf auf Satzendpunkt " mit Zielsatz N260 würde zu Alarm 14040 (Kreisendpunktfehler) führen.

**Teilprogramme für Typ 4 und Typ 2**

## PROC WERKSTUECK\_1

Programmcode	Kommentar
<b>;Hauptprogramm</b>	
...	
;Bearbeitung Konturabschnitt 1 mit Werkzeug "FRAESER_1"	
...	
N100 G0 G40 X200 Y200	; Abwahl Radiuskorrektur
N110 Z100 D0	; Abwahl Längenkorrektur
;Ende Konturabschnitt 1	
;	
;Bearbeitung Konturabschnitt 2 mit Werkzeug "FRAESER_2"	
N200 T="FRAESER_2"	; Werkzeug vorwählen
N210 WZW	; Werkzeugwechselroutine aufrufen
N220 G0 X170 Y30 Z10 S3000 M3 D1	; Anfahransatz Konturabschnitt 2
N230 Z-5	; Zustellung
N240 G1 G64 G42 F500 X150 Y50	; Startpunkt der Kontur
N250 Y150	
N260 G2 J50 X100 Y200	

8.7 Satzsuchlauf

Programmcode	Kommentar
N270 G1 X50	
N280 Y50	
N290 X150	
N300 G0 G40 G60 X170 Y30	; Abwahl Radiuskorrektur
N310 Z100 D0	; Abwahl Längenkorrektur
Ende Konturabschnitt 2	
...	
M30	
PROC WZW	
<b>;Werkzeugwechselroutine</b>	
N500 DEF INT TNR_AKTIV	; Variable für aktive T-Nummer
N510 DEF INT TNR_VORWAHL	; Variable für vorgewählte T-Nummer
N520 TNR_AKTIV = \$TC_MPP6[9998,1]	; T-Nummer des aktiven Werkzeugs lesen
N530 GETSELT(TNR_VORWAHL)	; T-Nummer des vorgewählten Werkzeugs lesen
;	
;Werkzeugwechsel nur ausführen, wenn Werkzeug noch nicht aktiv ist	
N540 IF TNR_AKTIV == TNR_VORWAHL GOTOF ENDE	
N550 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	; Werkzeugwechsellpunkt anfahren
N560 M6	; Werkzeugwechsel ausführen
;	
ENDE: M17	
PROC SUCHLAUF_ENDE SAVE	
<b>;ASUP</b> für den Aufruf der Werkzeugwechselroutine nach Satzsuchlauf	
N1000 DEF INT TNR_AKTIV	; Variable für aktive T-Nummer
N1010 DEF INT TNR_VORWAHL	; Variable für vorgewählte T-Nummer
N1020 DEF INT TNR_SUCHLAUF	; Variable für im Suchlauf ermittelt T-Nummer
N1030 TNR_AKTIV = \$TC_MPP6[9998,1]	; T-Nummer des aktiven Werkzeugs lesen
N1040 TNR_SUCHLAUF = \$P_TOOLNO	; durch Suchlauf ermittelte T-Nummer lesen
N1050 GETSELT(TNR_VORWAHL)	; T-Nummer des vorgewählten Werkzeugs lesen
N1060 IF TNR_AKTIV ==TNR_SUCHLAUF GOTOF ASUP_ENDE	
N1070 T = \$TC_TP2[TNR_SUCHLAUF]	; T-Anwahl über Werkzeugnamen
N1080 WZW	; Werkzeugwechselroutine aufrufen
N1090 IF TNR_VORWAHL == TNR_SUCHLAUF GOTOF ASUP_ENDE	
N1100 T = \$TC_TP2[TNR_VORWAHL]	; T-Vorwahl restaurieren über Werkzeugnamen
ASUP_ENDE:	
N1110 M90	; Rückmeldung an PLC
N1120 REPOSA	; ASUP Ende



## 8.8 Satzsuchlauf Typ 5 SERUPRO

### 8.8.1 Funktionsbeschreibung

Der Suchlauftyp 5, Satzsuchlauf mit Berechnung im Modus "Programmtest" (SERUPRO, "Search-Run by Programtest") ermöglicht einen kanalübergreifenden Suchlauf mit Berechnung zu einem wählbaren Unterbrechungspunkt. Dazu werden während SERUPRO, unter Beachtung vorhandener Programmkoordinierungsbefehle, alle zur Programmfortsetzung in den unterbrochenen Kanälen erforderlichen Zustandsdaten ermittelt und anschließend NC und PLC in einen Zustand versetzt, der die Programmfortsetzung erlaubt.

Vor dem Wiederaufstart an die Kontur mit anschließendem Fortsetzen der Programmbearbeitung, können über ein automatisch gestartetes anwenderspezifisches ASUP alle noch eventuell erforderlichen Ausgangszustände erzeugt werden.

### Kanäle

In Verbindung mit HMI ist SERUPRO für folgende Kanäle vorgesehen:

- Nur für den aktuellen SERUPRO-Kanal (1)
- Für alle Kanäle mit gleichem Werkstücknamen wie der SERUPRO-Kanal (2)
- Für alle Kanäle mit gleicher BAG wie der SERUPRO-Kanal (3)
- Für alle Kanäle der NCU (4)

Die Auswahl des Kanalumfangs für SERUPRO erfolgt über die Projektierungsdatei **maschine.ini**, in dem Abschnitt [BlockSearch]:

Section [BlockSearch]	Suchlauf-Funktion für HMI freigeben und Suchlauf-Konfiguration auswählen
SeruproEnabled=1	;SERUPRO Softkey bei HMI verfügbar. Standardwert ist (1)
SeruproConfig=1	;Nummer (1) bis (4) der oben genannten Kanalgruppierung. Standardwert ist (1)

Alle anderen mit SERUPRO gestarteten Kanäle werden im Modus "Self-Acting Serupro" betrieben. Nur der Kanal, in dem auch ein Zielsatz ausgewählt wurde, kann mit Satzsuchlauf im Modus SERUPRO gestartet werden.

### Aktivierung

Die Aktivierung von SERUPRO erfolgt über HMI. Mit den Softkey "Prog.Test Kontur" wird SERUPRO bedient.

SERUPRO verwendet REPOS, um den Zielsatz anzufahren.

### Zeitlicher Ablauf von SERUPRO

1. Über HMI wird Softkey "Pog. Test Kontur" und das Suchziel bedient.
2. Die NC startet jetzt selbsttätig das angewählte Programm im Modus "Programmtest".
  - Achsen werden dabei nicht verfahren.
  - Hilfsfunktionen \$A\_OUT und die direkte PLC-IO werden ausgegeben.
  - Die Hilfsfunktionen des Zielsatzes werden nicht ausgegeben.
3. Die NC stoppt am Beginn des Zielsatzes, wählt intern Programmtest ab, und zeigt die Stopp-Bedingung "Warten: Suchziel gefunden" an.
4. Liegt das anwenderspezifische ASUP "PROG\_EVENT.SPF" vor, wird es automatisch gestartet.
5. Mit dem nächsten NC-Start erfolgt ein Wiederanfahren an die Kontur (REPOS).  
Der REPOS-Vorgang erfolgt über ein System-ASUP und kann über die Funktion "Editierbares ASUP" erweitert werden.

### Randbedingungen für Satzsuchlauf SERUPRO

Die Funktion SERUPRO darf nur in der Betriebsart "Automatik" aktiviert und im Programmzustand (Kanalzustand RESET) abgebrochen werden.

Startet im Normalbetrieb **nur** der PLC gemeinsam mehrere Kanäle, so kann dies durch SERUPO in jedem Kanal simuliert werden.

Bei Maschinendatum-Einstellung:

MD10708 \$MN\_SERUPRO\_MASK, Bit 1 = 0

bricht der Alarm 16942 "Kanal %1 Start-Programmbefehl Aktion %2<ALNX> nicht möglich" die Simulation dann ab, wenn der Teileprogrammbefehl `START` verwendet wird.

Das Maschinendatum:

MD10707 \$MN\_PROG\_TEST\_MASK

erlaubt das Abschalten im gestoppten Zustand und hat keinen Einfluss auf den SERUPRO-Vorgang. Die Standardvorbelegung erlaubt das Abschalten nur im RESET-Zustand.

---

#### Hinweis

Nach dem Abschalten von Programmtest beginnt ein REPOS-Vorgang, für den dieselben Einschränkungen wie beim SERUPRO-Anfahren gelten. Auftretende Beeinträchtigungen lassen sich mit einem ASUP verhindern.

---

## SERUPRO-Verhalten beeinflussen

Für die nachfolgend beispielhaft aufgeführten Funktionen kann das Verhalten von SERUPRO NC-spezifisch vorgegeben werden:

- Programmierter Halt (<sub>M0</sub>)
- Programmkoordinierungsbefehl <sub>START</sub>
- Group-SERUPRO
- Kanalübergreifendes Beenden von SERUPRO
- Override

MD10708 \$MN\_SERUPRO\_MASK = <Verhalten bei SERUPRO>

## Kanalspezifische Grundstellungen für SERUPRO

Mit dem nachfolgenden Maschinendatum werden im Normalfall die kanalspezifischen Grundstellungen nach Teileprogrammstart festgelegt:

MD20112 \$MC\_START\_MODE\_MASK= <Grundstellungen>

Für SERUPRO können eigene Grundstellungen welche die Grundstellungen aus MD20112 ersetzen, vorgegeben werden:

MD22620 \$MC\_START\_MODE\_MASK\_PRT = <SERUPRO-Grundstellungen>

Die SERUPRO-Grundstellungen müssen explizit freigegeben werden über:

MD22621 \$MC\_ENABLE\_START\_MODE\_MASK\_PRT = 1

## NC/PLC-Nahtstellensignal "Satzsuchlauf via Programmtest ist aktiv"

Der Satzsuchlauf via Programmtest wird angezeigt über das NC/PLC-Nahtstellensignal: DB21, ... DBX318.1 == 1

Das Nahtstellensignal ist gesetzt vom Start des Satzsuchlaufs bis zum Einwechseln des Zielsatzes in den Hauptlauf.

## Zum benutzerdefinierten ASUP nach dem SERUPRO-Vorgang

Hinweis

Entschließt sich der Maschinenhersteller nach dem SERUPRO-Vorgang ein ASUP gemäß Punkt 7. zu starten, so muss Folgendes beachtet werden:

### Gestoppter Zustand nach Punkt 6.:

Die Maschinendaten:

MD11602 \$MN\_ASUP\_START\_MASK

und

MD11604 \$MN\_ASUP\_START\_PRIO\_LEVEL

erlauben es dem NCK, das ASUP aus dem gestoppten Zustand selbsttätig über FC9-Baustein zu starten.

**Quittierung vom FC9 erst nach Beendigung des REPOS-Satz:**

Das ASUP kann erst vom FC9-Baustein mit "Asup Done" als beendet gemeldet werden, wenn auch der REPOS-Satz beendet ist.

**Abwahl des vorgesehenen REPOS-Vorgangs nach Punkt 8.:**

Der Start des ASUPs wählt den vorgesehenen REPOS-Vorgang ab!

Daher sollte das ASUP mit REPOSA abgeschlossen werden, um den REPOS-Vorgang beizubehalten.

**Löschen eines unerwünschten REPOS-Vorgangs:**

Der unerwünschte REPOS-Vorgang wird gelöscht, indem das ASUP mit M17 oder RET beendet wird.

**Besondere Behandlung von ASUP:**

Grundsätzlich wird ein ASUP, das mit REPOS endet und aus einem gestoppten Zustand heraus gestartet wird, besonders behandelt.

Das ASUP stoppt selbsttätig vor dem REPOS-Satz und zeigt dies an über:

DB21, ... DBX318.0 (ASUP angehalten)

**Automatischer ASUP-Start**

Das unter dem Pfad:

/\_N\_CMA\_DIR/\_N\_PROG\_EVENT\_SPF

liegende ASUP, wird automatisch mit dem Maschinendatum:

MD11450 \$MN\_SEARCH\_RUN\_MODE, Bit1 = 1

im SERUPRO-Anfahren nach folgenden Ablauf gestartet:

1. Der SERUPRO-Vorgang ist komplett durchgeführt.
2. Der Anwender löst "NC-Start" aus.
3. Das ASUP wird gestartet.
4. Die NC stoppt vor dem Teileprogrammbefehl REPOS selbsttätig und die Meldung "Zur Programmfortsetzung NC-Start geben" erscheint.
5. Der Anwender drückt zum zweiten Mal "NC-Start".
6. Die NC führt die Wiederanfahrbewegung aus und setzt das Teileprogramm mit dem Zielsatz fort.

---

**Hinweis**

Der automatische ASUP-Start mit MD11450 benötigt **Starts**, um das Programm fortzusetzen.

Damit nähert sich das Ablaufverhalten an andere Suchlauftypen an.

---

## 8.8.2 REPOS

### MD11470

REPOS erfolgt abhängig vom Maschinendatum:

MD11470 \$MN\_REPOS\_MODE\_MASK

#### Fall A:

Der REPOS-Vorgang bewegt in einem Satz alle Achsen von der aktuellen Position zum Zielsatzanfangspunkt.

MD11470 \$MN\_REPOS\_MODE\_MASK Bit 3 = 1

#### Fall B:

Die Bahnachsen werden in einem Satz zusammen repositioniert. Im Restsatz werden die SPOS- und POS-Achsen repositioniert.

MD11470 \$MN\_REPOS\_MODE\_MASK Bit 3 = 0

### 8.8.2.1 Wiederaufsetzen nach SERUPRO-Suchziel gefunden

#### Benutzerhinweis zum REPOS-Vorgang


REPOS wird in der Regel eingesetzt, um eine begonnene Bearbeitung zu unterbrechen und danach die Bearbeitung fortzusetzen.

Im SERUPRO-Anfahren muss dagegen ein Programmstück "nachgeholt" werden.

Dies ist der Fall, wenn SERUPRO die Simulation beendet hat und wieder auf den Zielsatz gefahren werden soll. SERUPRO bezieht sich hierbei auf die bestehende Funktion REPOS, die vom Anwender entsprechend angepasst werden kann.

#### SERUPRO-Anfahren

Der Anwender kann das REPOS-Verhalten einzelner Achsen zu gezielten Zeitpunkten so verändern, dass bestimmte Achstypen entweder früher, später oder gar nicht repositionieren. Dies betrifft insbesondere SERUPRO-Anfahren. Wiederanfahrbewegungen einiger Achsen können beim REPOS-Vorgang auch unabhängig von SERUPRO-Anfahren beeinflusst werden.

 <b>VORSICHT</b>
<p>Der REPOS-Vorgang bewegt bei entsprechender Einstellung von Maschinendatum: MD11470 \$MN_REPOS_MODE_MASK Bit 3 in einem Satz alle Achsen von der aktuellen Position zum Zielsatzanfangspunkt.</p> <p>Dabei kann die NC mögliche Kollisionen mit der Maschine oder dem Werkstück nicht erkennen!</p> <p>Schutzbereiche und Softwarelimits werden überwacht.</p>

## REPOS Verhalten einstellen

Mit dem Maschinendatum:

MD11470 \$MN\_REPOS\_MODE\_MASK

kann durch Setzen der Bits das Verhalten der NC beim Repositionieren eingestellt werden:

- Bit 0 = 1     Im Restsatz des Repositionierens wird die Verweilzeit dort fortgesetzt, wo unterbrochen wurde.
- Bit 1 = 1     Reserviert
- Bit 2 = 1     Über NC/PLC-Nahtstellensignale das Repositionieren einzelner Achsen verhindern.
- Bit 3 = 1     Bei Satzsuchlauf über Programmtest (SERUPRO) Positionierachsen im Anfahrstanz repositionieren.
- Bit 4 = 1     Positionierachsen im Anfahrstanz bei jedem REPOS.
- Bit 5 = 1     Geänderte Vorschübe und Spindeldrehzahlen werden bereits im Restsatz gültig. Sonst erst im darauf folgenden Satz.
- Bit 6 = 1     Nach SERUPRO werden neutrale Achsen und positionierende Spindeln im Anfahrstanz als Kommando-Achsen repositioniert.
- Bit 7 = 1     Der Pegel von Nahtstellensignal:  
DB31, ... DBX10.0 (REPOSDELAY)  
wird gelesen, wenn REPOSA interpretiert wird.  
Achsen, die weder Geometrie- noch Orientierungsachsen sind, werden dann vom REPOS ausgeschlossen und auch nicht bewegt.

## Wiederanfahren mit gesteuertem REPOS

An einer beliebigen Stelle einer Bearbeitung wird ein Teileprogramm unterbrochen und ein ASUP mit einem REPOS begonnen.

Bei Bahnachsen kann der REPOS-Modus zum Wiederanfahren auf Kontur von PLC über NC/PLC-Nahtstellensignale gesteuert werden. Dieser Modus wird im Teileprogramm programmiert und bestimmt das Anfahrverhalten (siehe Kapitel "Wiederanfahren an Kontur mit gesteuertem REPOS (Seite 551)").

Das REPOS-Verhalten einzelner Achsen kann auch über NC/PLC-Nahtstellensignale gesteuert werden

und wird mit:

MD11470 \$MN\_REPOS\_MODE\_MASK BIT 2==1

freigegeben.

Bahnachsen können nicht einzeln beeinflusst werden. Bei allen weiteren Achsen, die keine Geometrieachsen sind, kann das Repositionieren einzelner Achsen vorübergehend verhindert und auch verschoben werden. Über NC/PLC-Nahtstellensignal werden die einzelnen Kanalachsen, die REPOS herausfahren möchte, zu einem späteren Zeitpunkt wieder freigegeben oder weiter gesperrt.

 **GEFAHR**

Durch das Signal:  
DB31, ... DBX2.2 (Restweg löschen, achsspezifisch)  
ergibt sich folgendes gefährliches Verhalten beim "Repositionieren einzelner Achsen verhindern" über:  
MD11470 \$MN\_REPOS\_MODE\_MASK (Bit 2==1).

Solange eine Achse nach der Unterbrechung inkrementell programmiert wird, fährt die NC andere Positionen an als ohne Unterbrechung (siehe folgendes Beispiel).

### Beispiel: Achse wird inkrementell programmiert

Die Achse A steht vor dem REPOS-Vorgang auf 11°, die Programmierung im Unterbrechungssatz (Zielsatz bei SERUPRO) legt 27° fest.

Beliebig viele Sätze später wird diese Achse inkrementell um 5° mit:

```
N1010 POS[A]=IC(5) FA[A]=1000
```

programmiert.

Mit dem Nahtstellensignal:

DB31, ... DBX10.0 (REPOSDELAY)

verfährt die Achse im REPOS-Vorgang nicht und wird mit N1010 nach 32° gefahren.

(Der Benutzer muss evtl. den Weg von 11° nach 27° bewusst quittieren).

**Vorsicht:**

Die Achse wird nach der Unterbrechung inkrementell programmiert.

Im Beispiel fährt die NC nach 16° (statt 32°).

#### A) Achsen einzeln starten

Das REPOS-Verhalten für das SERUPRO-Anfahren mit mehreren Achsen wird angewählt mit:

```
MD11470 $MN_REPOS_MODE_MASK BIT 3 == 1
```

Die NC beginnt SERUPRO-Anfahren mit einem Satz, der **alle** Positionierachsen auf das programmierte Ende und die Bahnachse auf den Zielsatz fährt.

Der Benutzer startet die einzelnen Achsen, indem er die Vorschubfreigaben entsprechend anwählt. Anschließend wird der Zielsatz abgefahren.

### B) Positionierachsen im Wiederanfahrtsatz repositionieren

Positionierachsen werden nicht im Restsatz, sondern im Wiederanfahrtsatz repositioniert und betreffen nicht nur den Satzsuchlauf über Programmtest beim SERUPRO-Anfahren:

MD11470 \$MN_REPOS_MODE_MASK	
Bit 3=1	für Satzsuchlauf über Programmtest (SERUPRO)
Bit 4=1	für jedes REPOS

---

#### Hinweis

Ist weder Bit 3 noch Bit 4 gesetzt, werden in dieser Phase "nicht Bahnachsen" im Restsatz repositioniert.

---

### REPOS vorziehen oder ignorieren

Weitere REPOS-Anpassungen können vorgenommen werden durch Setzen der Bits in:

MD11470 \$MN\_REPOS\_MODE\_MASK

- Bit 5 = 1      Geänderte Vorschübe und Spindeldrehzahlen werden bereits im Restsatz gültig und werden damit vorgezogen. Dieses Verhalten bezieht sich auf jeden REPOS-Vorgang.
- Bit 6 = 1      Neutrale Achsen und positionierende Spindeln werden nach SERUPRO repositioniert.  
Neutrale Achsen, die weiter nicht repositioniert werden dürfen, müssen beaufschlagt werden mit Nahtstellensignal:  
DB31, ... DBX10.0 (REPOSDELAY)  
Damit wird die REPOS-Bewegung gelöscht.
- Bit 7 = 1      Der Pegel von Nahtstellensignal:  
DB31, ... DBX10.0 (REPOSDELAY)  
wird gelesen, wenn REPOSA interpretiert wird.  
Achsen, die weder Geometrie- noch Orientierungsachsen sind, werden dann vom REPOS ausgeschlossen und auch nicht bewegt.  
Anmerkung: REPOSDELAY wurde von Flanken- auf Pegelauswertung umgestellt.




### Achse mit REPOS-Verschiebung verzögert anfahren

Mit dem axialen pegelgetriggerten NC/PLC-Nahtstellensignal Achs/Spindel (PLC→NCK):  
DB31, ... DBX10.0 (REPOSDELAY)

wird mit der Flanke von NST:  
DB21, ... DBX31.4 (REPOSMODEEDGE)  
die REPOS-Verschiebung für diese Achse erst mit ihrer nächsten Programmierung  
herausgefahren.

Ob diese Achse gerade einer REPOS-Verschiebung unterliegt, kann über Synchronaktionen  
mit \$AA\_REPOS\_DELAY abgefragt werden.

 <b>VORSICHT</b>
Das Nahtstellensignal: DB31, ... DBX10.0 (REPOSDELAY) hat auf Maschinenachsen, die eine Bahn bilden, keine Auswirkung.  Ob eine Achse eine Bahnachse ist, kann mit: DB31, ... DBX76.4 (Bahnachse) festgestellt werden.

### Übernahmezeitpunkt der REPOS NC/PLC-Nahtstellensignale

Mit der 0/1 Flanke vom kanalspezifischen NC/PLC-Nahtstellensignal (PLC→NCK):  
DB21, ... DBX31.4 (REPOSMODEEDGE)

werden die Pegelsignale von:  
DB21, ... DBX31.0-31.2 (REPOSPATHMODE0 bis 2)

und  
DB31, ... DBX10.0 (REPOSDELAY)

in der NC übernommen.

Die Pegel beziehen sich auf den aktuellen Satz im Hauptlauf. Es werden folgende zwei Fälle  
unterschieden:

#### Fall A:

Im Hauptlauf befindet sich **ein** Wiederanfahrtsatz eines gerade aktiven REPOS-Vorgangs.

Der laufende REPOS-Vorgang wird abgebrochen, neu aufgesetzt, und die REPOS-  
Verschiebungen werden durch die Signale:

DB21, ... DBX31.0-31.2 (REPOSPATHMODE0 bis 2)

und  
DB31, ... DBX10.0 (REPOSDELAY)  
beeinflusst.

**Fall B:**

Im Hauptlauf befindet sich **kein** Wiederanfahrtsatz eines gerade aktiven REPOS-Vorgangs.

Jeder zukünftiger REPOS-Vorgang, der an den aktuellen Hauptlaufsatz wieder anfahren möchte, wird durch die Pegel von Nahtstellensignal:

DB21, ... DBX31.0-31.2 (REPOSPATHMODE0 bis 2)

und

DB31, ... DBX10.0 (REPOSDELAY)

beeinflusst.

---

**Hinweis**

Im laufenden ASUP wirkt das NST:

DB21, ... DBX31.4 (REPOSMODEEDGE)

nicht auf das abschließende REPOS, außer man trifft mit diesem Signal die REPOS-Sätze.

Im Fall A ist das Signal nur im gestoppten Zustand erlaubt.

**Verhalten bei RESET:**

**NCK hat PLC-Signal quittiert**

Ist der Pegel der Signale:

DB21, ... DBX31.4 (REPOSMODEEDGE) = 1

und

DB21, ... DBX319.0 (REPOSMODEEDGEACKN) = 1

und

tritt in dieser Situation ein RESET auf, so wird vom NCK das Nahtstellensignal:

DB21, ... DBX319.1–319.3 (Repos Path Mode Quitt0 bis 2)

gelöscht.

**NCK hat PLC-Signal noch nicht quittiert:**

Ist der Pegel der Signale:

DB21, ... DBX31. (REPOSMODEEDGE 4) = 1

und

DB21, ... DBX319.0 (REPOSMODEEDGEACKN) = 0

und tritt in dieser Situation ein RESET auf, so wird vom NCK

DB21, ... DBX319.0 (REPOSMODEEDGEACKN) = 0

und

DB21, ... DBX319.1–319.3 (Repos Path Mode Quitt0 bis 2)

gelöscht.

---

## Mit NC/PLC-Nahtstellensignalen SERUPRO-Anfahren beeinflussen

Das SERUPRO-Anfahren kann mit Nahtstellensignal:

DB21, ... DBX31.4 (REPOSMODEEDGE)

und den dazugehörigen Signalen in den folgenden Phasen eingesetzt werden:

- Zwischen "Suchziel gefunden" und "Start des SERUPRO-ASUPs"
- Von "SERUPRO-ASUP stoppt selbsttätig vor dem REPOS" bis "Zielsatz wird abgearbeitet"

Während das SERUPRO-ASUP z. B. im Programmteil vor dem REPOS abgearbeitet wird, wirkt das Nahtstellensignal:

DB21, ... DBX31.4 (REPOSMODEEDGE)

nicht auf SERUPRO-Anfahren.

## REPOS Abläufe mit NC/PLC-Nahtstellensignalen

### REPOS mit NC/PLC-Nahtstellensignale steuern

REPOS-Verschiebungen lassen sich günstig mit den folgenden kanalspezifischen NC/PLC-Nahtstellensignalen **vom PLC** aus beeinflussen:

DB21, ... DBX31.0-31.2 (REPOSPATHMODE0 bis 2) \*kanalspezifisch

DB21, ... DBX31.4 (REPOSMODEEDGE) kanalspezifisch

DB31, ... DBX10.0 (REPOSDELAY) \*Achse/Spindel

(Dieses axiale NST wirkt **nicht** auf Maschinenachsen, die eine Bahn bilden.)

DB31, ... DBX72.0 (REPOSDELAY) Achse/Spindel

\* Diese Signale sind im jeweiligen DB vom HMI bzw. PLC verfügbar.

## REPOS-Quittierungssignale

Mit folgenden NC/PLC-Nahtstellensignalen werden Funktionen, die das REPOS-Verhalten über PLC beeinflussen, **von NCK** aus quittiert:

DB21, ... DBX319.0 (REPOSMODEEDGEACKN) kanalspezifisch

DB21, ... DBX319.1-319.3 (Repos Path Mode Quitt0 bis 2) kanalspezifisch

DB21, ... DBX319.5 (Repos DEFERRAL Chan) kanalspezifisch

DB31, ... DBX70.0 (Repos Verschiebung) Achse/Spindel

DB31, ... DBX70.1 (Repos Verschiebung gültig) Achse/Spindel

DB31, ... DBX70.2 (Repos Delay Quitt) Achse/Spindel

DB31, ... DBX76.4 (Bahnachse) Achse/Spindel

Weitere Informationen hierzu finden Sie unter "REPOS-Verschiebung in der Nahtstelle".

## REPOS-Quittierungsvorgänge

Mit dem kanalspezifischen NC/PLC-Nahtstellensignal:  
DB21, ... DBX319.0 (REPOSMODEEDGEACKN)

wird ein "Handshake" aufgebaut, indem das Nahtstellensignal:  
DB21, ... DBX31.4 (REPOSMODEEDGE)

von der NC erkannt und mit DB21, ... DBX319.0 zum PLC quittiert wird.

---

### Hinweis

Hat NCK das Nahtstellensignal:

DB21, ... DBX31.4 (REPOSMODEEDGE)

noch nicht mit dem Nahtstellensignal:

DB21, ... DBX319.0 (REPOSMODEEDGEACKN)

quittiert, so führte ein RESET in dieser Situation zum Programmabbruch, und das REPOS, mit dem der REPOSPATHMODE beeinflusst werden soll, kann nicht mehr stattfinden.

---

Ein vom PLC vorgegebener REPOSMODE wird vom NCK mit den Nahtstellensignalen:  
DB21, ... DBX319.1-319.3 (Repos Path Mode Quitt0 bis 2)

und

DB31, ... DBX10.0 (Repos Delay)

mit:

DB31, ... DBX70.2 (Repos Delay Quitt)

in folgender Weise quittiert:

Ein Teileprogramm wird auf  $N20$  gestoppt (→ Zeitpunkt (2) im Bild). Der NCK hält nach der Bremsrampe an. Nachdem der PLC den REPOSPATHMODE vorgegeben hat, übernimmt im → Zeitpunkt (3) der NCK den REPOSPATHMODE mit der 0/1-Flanke von REPOSMODEEDGE Repos Path Mode Quitt bleibt solange gesetzt bis das ASUP ausgelöst wird (→ Zeitpunkt (4)). Der Befehl REPOS wird im ASUP begonnen (→ Zeitpunkt (5)). Der RESET-Satz des ASUPs wird wieder eingewechselt (→ Zeitpunkt (6)):

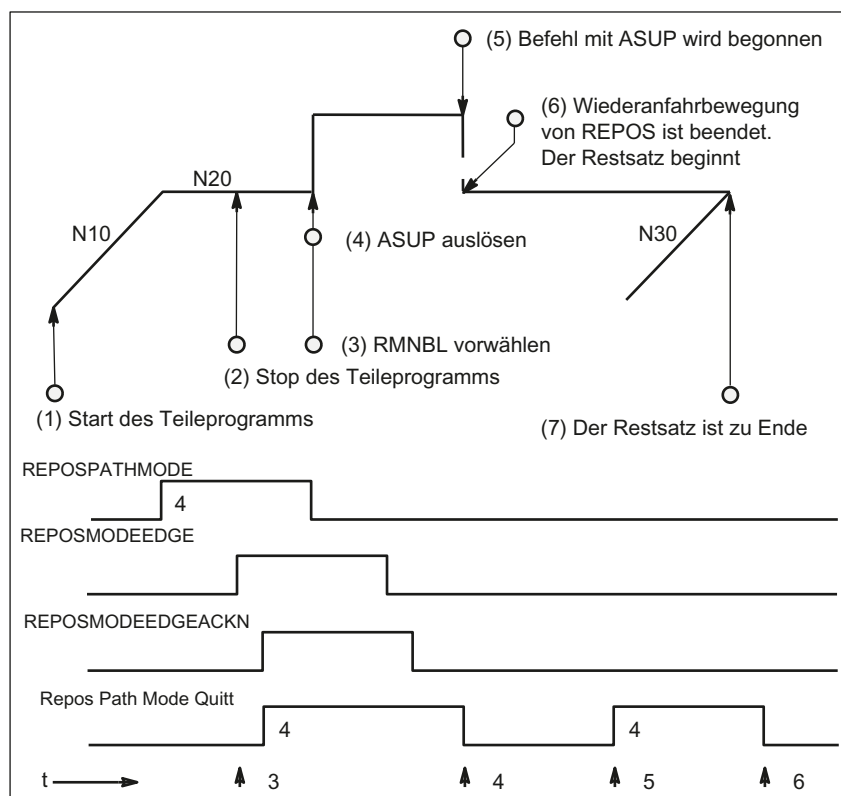


Bild 8-6 REPOS Ablauf im Teileprogramm mit zeitlichen Quittierungssignalen von NCK

### NCK setzt Quittierung erneut

Phase, in der REPOSPATHMODE weiterhin wirkt (Restsatz des im → Zeitpunkt (2) gestoppten Programms ist noch nicht zu Ende ausgeführt).

Sobald die REPOS-Wiederanfahrbewegung des ASUP's bearbeitet wird, setzt der NCK den "Repos Path Mode Quitt" erneut (→ Zeitpunkt (5)). Sollte kein REPOSPATHMODE über NC/PLC-Nahtstellensignal vorgewählt worden sein, wird der programmierte REPOS-Mode angezeigt.

"Repos Path Mode Quitt" wird mit dem Einwechseln des Restsatzes (→ Zeitpunkt (6)) zurückgenommen. Der dem im → Zeitpunkt (2) folgende Teileprogrammsatz  $N_{30}$  wird fortgesetzt.

Das Nahtstellensignal:

DB31, ... DBX70.2 (Repos Delay Quitt) ist analog definiert.

DB31, ... DBX70.1 (Repos Verschiebung gültig) = 1, wenn:

DB21, ... DBX319.1-319.3 (Repos Path Mode Quittt0 bis 2) = 4 (RMNBL).

### Gültige REPOS-Verschiebung

Mit dem Ende des SERUPRO-Vorgangs kann der Anwender die REPOS-Verschiebung über das NC/PLC-Nahtstellensignal Achse/Spindel (NCK→PLC):  
DB31, ... DBX70.0 (REPOS Verschiebung) auslesen.

Dieses Signal hat folgende Auswirkung für diese Achse:

Wert 0: Keine REPOS-Verschiebung wird herausgefahren werden.

Wert 1: Eine REPOS-Verschiebung wird herausgefahren werden.

### Gültigkeitsbereich

Das Nahtstellensignal:  
DB31, ... DBX70.0 (REPOS Verschiebung)  
wird mit dem Ende des SERUPRO-Vorganges versorgt.

Mit dem Start eines SERUPRO-ASUP oder dem automatischen ASUP-Start wird die REPOS-Verschiebung ungültig.

### REPOS-Verschiebung im Gültigkeitsbereich aktualisieren

Zwischen SERUPRO-Ende und -Start kann mit Moduswechsel die Achse in JOG verfahren werden.

Der Anwender fährt die REPOS-Verschiebung händisch mit JOG heraus, um das NST DB31, ... DBX70.0 (REPOS Verschiebung) auf den Wert 0 zu setzen.

Im Gültigkeitsbereich kann die Achse auch über FC18 verfahren werden, wobei das NST DB31, ... DBX70.0 (REPOS Verschiebung) ständig aktualisiert wird.

### Gültigkeitsbereich anzeigen

Der Gültigkeitsbereich von der REPOS-Verschiebung wird angezeigt mit dem Nahtstellensignal:

DB31, ... DBX70.1 (REPOS Verschiebung gültig)

Es wird angegeben, ob eine gültige Berechnung vorliegt:

Wert 0: REPOS-Verschiebung dieser Achse ist korrekt berechnet.

Wert 1: REPOS-Verschiebung dieser Achse ist nicht berechenbar, das REPOS liegt in der Zukunft, z. B. am ASUP-Ende, oder kein REPOS aktiv.

## REPOS-Verschiebung nach einem Achstausch

Mit dem Sammelsignal:  
DB21, ... DBX319.5 (Repos DEFERRAL Chan)  
kann festgestellt werden, ob eine gültige REPOS-Verschiebung stattgefunden hat:

Wert 0: Alle Achsen, die von diesem Kanal aktuell kontrolliert werden, haben entweder keine REPOS-Verschiebung oder ihre REPOS-Verschiebungen sind ungültig.  
Wert 1: Sonstige.

## REPOS-Verschiebung bei synchronisierter Synchronspindelkopplung

Beim Wiederanfahren mit SERUPRO wird an die Unterbrechungsstelle wieder vorgelaufen. War eine Synchronspindelkopplung bereits synchronisiert, dann existiert keine REPOS-Verschiebung der Folgespindel und es steht auch kein Synchronisationsweg an. Die Synchronisationssignale bleiben gesetzt.

## Suchziel gefunden beim Satzwechsel

Das axiale NC/PLC-Nahtstellensignal:  
DB31, ... DBX76.4 (Bahnachse) ist 1,  
wenn die Achse Teil des Bahnverbundes ist.

Dieses Signal zeigt den Zustand des aktuell zu bearbeitenden Satzes beim Satzwechsel an. Spätere Zustandsänderungen werden nicht berücksichtigt.

Wenn der SERUPRO-Vorgang mit "Suchziel gefunden" beendet ist, bezieht sich das NST DB31, ... DBX76.4 (Bahnachse) auf den Zielsatz.

### 8.8.2.2 Wiederanfahren an Kontur mit gesteuertem REPOS

#### Anfahrmodi

##### Bahnachsen einzeln beeinflussen

Beim SERUPRO-Anfahren wird ein REPOS-Vorgang ausgelöst, um wieder an die Kontur zu fahren. Dabei werden häufig sehr viele Achsen bewegt, die der Anwender über Nahtstellensignale beeinflussen kann. Die BTSS-Schnittstelle liefert die Verschiebungen pro Kanalachse, die REPOS herausfahren möchte.

Wiederanfahren einzelner Bahnachsen kann vom PLC mit Nahtstellensignalen gesteuert werden und hat damit Vorrang gegenüber den eigentlichen Befehlen `RMIBL`, `RMBBL` und `RMEBL` im Teileprogramm.

`RMIBL` Wiederanfahren an den Unterbrechungspunkt

`RMBBL` Wiederanfahren an den Satzanfang

`RMEBL` Wiederanfahren an Satzende

`RMNBL` Wiederanfahren an den nächstliegenden Bahnpunkt

### Wiederanfahren mit RMNBL

Analog zu  $RMIBL$ ,  $RMBBL$  und  $RMEBL$  wird für SERUPRO-Anfahren  $RMNBL$  (REPOS-Mode-Next) **neu definiert**. Nach einer Unterbrechung wird mit  $RMNBL$  der Wiederanfahrtsatz nicht noch einmal komplett begonnen, sondern nur vom nächstliegenden Bahnpunkt wie folgt abgearbeitet:

Zum Interpretationszeitpunkt von REPOSA wird die Position (B) herangezogen, um den Punkt C auf dem Unterbrechungssatz zu finden, der den kürzesten Abstand zu B ergibt. Der Wiederanfahrtsatz fährt von B nach C zur Endposition.

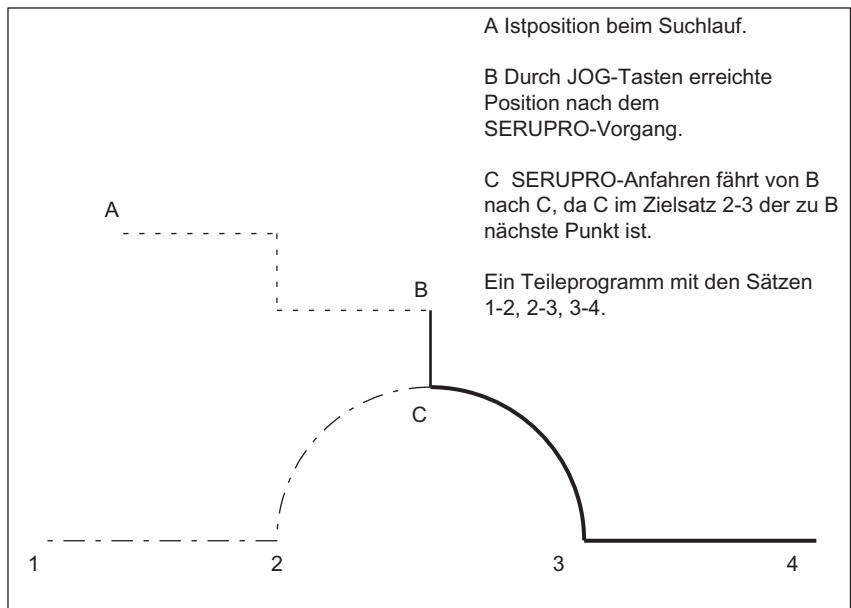


Bild 8-7 SERUPRO-Anfahren unter  $RMNBL$

### Anwendung und Vorgehensweise

SERUPRO-Anfahren unter  $RMNBL$  eröffnet gemäß Bild folgende Anwendung: Wird bei der Bearbeitung von 2 nach 3 an einer beliebigen Stelle ein Abbruch mit RESET erzwungen, so wird mit

- $RMNBL$  der kürzeste Weg zur Abbruchstelle angefahren, um anschließend nur den Restweg von C-3 und 3-4 abzarbeiten. Der Anwender startet ein SERUPRO-Vorgang auf den Unterbrechungssatz und positioniert mit den JOG-Tasten vor die schadhafte Stelle des Zielsatzes.
- $RMIBL$  und  $RMBBL$  immer B-2, 2-3, 3-4 angefahren, und damit der Zielsatz noch einmal vollständig wiederholt.



## REPOS-Mode auswählen

Mit dem kanalspezifischen NC/PLC-Nahtstellensignal (PLC→NCK) DB21, ... DBX31.0-31.2 (REPOSPATHMODE0-2) kann mit den 3 Bits die jeweilige Funktion **RM<sub>B</sub>BLL**, **RM<sub>I</sub>BLL**, **RM<sub>E</sub>BLL** oder **RM<sub>N</sub>BLL** ausgewählt werden.

### Wiederanfahrpunkt

**RM<sub>N</sub>OTDEF** REPOS-Mode nicht umdefiniert

**RM<sub>B</sub>BLL** Wiederanfahren Satz-Anfangspunkt bzw. letzten Endpunkt

**RM<sub>I</sub>BLL** Wiederanfahren Unterbrechungspunkt

**RM<sub>E</sub>BLL** Wiederanfahren Satzendpunktpunkt

**RM<sub>N</sub>BLL** Wiederanfahren an nächstliegenden Bahnpunkt

DB21, ... DBX31.0-31.2 (REPOSPATHMODE)=0 entspricht **RM<sub>N</sub>OTDEF**

DB21, ... DBX31.0-31.2 (REPOSPATHMODE)=1 entspricht **RM<sub>B</sub>BLL**

DB21, ... DBX31.0-31.2 (REPOSPATHMODE)=2 entspricht **RM<sub>I</sub>BLL**

DB21, ... DBX31.0-31.2 (REPOSPATHMODE)=3 entspricht **RM<sub>E</sub>BLL**

DB21, ... DBX31.0-31.2 (REPOSPATHMODE)=4 entspricht **RM<sub>N</sub>BLL**

Mit DB21, ... DBX31.0-31.2 (REPOSPATHMODE0 bis 2) = 0 wird nichts überschrieben und es gilt das aktuelle Programm. Das Nahtstellensignal reagiert auf den Pegel des entsprechenden Modes.

---

### Hinweis

**RM<sub>N</sub>BLL** ist eine allgemeine REPOS-Erweiterung und nicht nur auf SERUPRO beschränkt. Für SERUPRO ist **RM<sub>I</sub>BLL** und **RM<sub>B</sub>BLL** identisch.

Mit DB21, ... DBX31.0–31.2 (REPOSPATHMODE0 bis 2) wird die Bahn als ganzes beeinflusst. Die Bahnachsen können nicht einzeln verändert werden.

Das Verhalten der anderen Achstypen kann einzeln mit dem Nahtstellensignal DB31, ... DBX10.0 (REPOSDELAY) verändert werden. Diese REPOS-Verschiebung wird nicht sofort, sondern erst mit ihrer nächsten Programmierung herausgefahren.

Weitere Informationen zur Programmierung des Wiederanfahrpunkts siehe unter:

### Literatur:

Programmierhandbuch Arbeitsvorbereitung; Bahnverhalten, Kapitel: Wiederanfahren an Kontur

---

## REPOS-Mode in Synchronaktionen lesen

Der gültige REPOS-Mode des unterbrochenen Satzes kann gelesen werden über Synchronaktionen mit der Systemvariablen \$AC\_REPOS\_PATH\_MODE=

0: nicht definiert	Wiederanfahren nicht definiert
1: RMBBL	Wiederanfahren auf den Beginn
2: RMIBL	Wiederanfahren auf den Unterbrechungspunkt
3: RMEBL	Wiederanfahren auf den Satzendpunktpunkt
4: RMNBL	Wiederanfahren auf den geometrisch nächstliegenden Bahnpunkt des unterbrochenen Satzes

## 8.8.3 Beschleunigungsmaßnahmen über MD

### Maschinendateneinstellungen

Die Abarbeitungsgeschwindigkeit des gesamten SERUPRO-Vorgangs kann über die nachfolgenden Maschinendaten beschleunigt werden.

MD22600 \$MC\_SERUPRO\_SPEED\_MODE und

MD22601 \$MC\_SERUPRO\_SPEED\_FACTOR

Mit MD22600 \$MC\_SERUPRO\_SPEED\_MODE == 1 läuft der SERUPRO-Vorgang in der von "Probelaufvorschub" gewohnten Geschwindigkeit ab.

Durch MD22600 \$MC\_SERUPRO\_SPEED\_MODE == 0 wird

MD22601 \$MC\_SERUPRO\_SPEED\_FACTOR ausgewertet,

und erlaubt eine weitere Beschleunigung. In diesem Modus sind Dynamiküberwachungen ausgeschaltet.

### SPEED-Faktor bei Kanalachsen im Hochlauf

Das Maschinendatum MD22600 \$MC\_SERUPRO\_SPEED\_MODE wirkt für den gesamten SERUPRO-Vorgang auf folgende Kanalachsen im Hauptlauf:

- PLC-Achsen
- Kommando-Achsen
- Positionierachsen
- Pendelachsen

Die Funktionen vom MD22600 \$MC\_SERUPRO\_SPEED\_MODE sowie MD22601 \$MC\_SERUPRO\_SPEED\_FACTOR gelten nur bei SERUPRO und nicht im Programmtest. Hierbei werden **keine Achsen/Spindeln bewegt**.

#### VORSICHT

Die NC erzeugt als diskretes System eine Folge von Interpolations-Punkten.

Damit kann es vorkommen, dass eine Synchronaktion, die im Normalbetrieb auslöste, bei SERUPRO nicht mehr auslöst.

#### Wirkungsweise bei DryRUN

Ein aktiver SERUPRO SPEED-FACTOR wirkt sich auf DryRun wie folgt aus:

- Gleichzeitig wird DryRun aktiviert.  
Dadurch wird von G95/G96/G961/G97G971 auf G94 umgeschaltet, um wie gewünscht, G95/G96/G961/G97/G971 beschleunigt abzuarbeiten.
- Gewindebohren und Gewindeschneiden wird in der von DryRun gewohnten Geschwindigkeit abgearbeitet.

DryRun und SERUPRO wirkt mit folgenden G-Codes auf die Spindel/Achse:

- Mit G331/G332 wird die Spindel als Achse in einen Bahnverbund interpoliert. Beim Gewindebohren wird die Bohrtiefe (z. B. Achse X), die Gewindesteigung und Drehzahl (z. B. Spindel S) vorgegeben.

Im DryRun-Fall ist die Geschwindigkeit von X vorgegeben, die Drehzahl bleibt konstant und die Gewindesteigung wird angepasst.

Nach der Simulation durch SERUPRO ergibt sich für die Spindel S eine zum Normalbetrieb abweichende Position, weil die Spindel S in der Simulation weniger oft gedreht hat.

## 8.8.4 SERUPRO-ASUP

### SERUPRO-ASUP Besonderheiten

Während SERUPRO-ASUP sind Besonderheiten zu beachten bei:

- Referenzpunktfahren: Referenzieren per Teileprogramm G74
- Werkzeugverwaltung: Werkzeugwechsel und Magazindaten
- Spindelhochlauf: Beim Start eines SERUPRO-ASUP

### G74 Referenzpunkt fahren

Befindet sich zwischen dem Programmanfang und dem Suchziel die Anweisung G74 (Referenzpunktfahren), so wird dies von der NC ignoriert.

SERUPRO-Anfahren berücksichtigt diese Anweisung G74 nicht!

## Werkzeugverwaltung

Bei aktiver Werkzeugverwaltung (WZV) wird folgende Einstellung empfohlen:

MD18080 \$MA\_TOOL\_MANAGEMENT\_MASK das BIT 20 = 0 setzen.

Damit wird das während des SERUPRO-Vorgangs erzeugte WZV-Kommando **nicht** an die PLC ausgegeben!

Das WZV-Kommando wirkt sich wie folgt aus:

- Die NC quittiert die Kommandos selbst.
- Es werden keine Magazindaten verändert.
- Werkzeug-Daten werden nicht verändert.

Ausnahme:

Der Werkzeug-Zustand des im Testbetrieb aktivierten Werkzeuges kann den Zustand 'aktiv' annehmen. Damit kann nach dem SERUPRO-Vorgang das falsche Werkzeug auf der Spindel sein.

Abhilfe:

Der Anwender startet ein SERUPRO-ASUP, das real verfahren wird. Vor dem Start kann der Anwender ein ASUP starten, welches das korrekte Werkzeug einwechselt.

SERUPRO-Vorgang: Funktionalität: Im Ablauf vom Punkt 2. bis 6.

SERUPRO-ASUP: Funktionalität: Im Ablauf Punkt 7.

Außerdem muss im Maschinendatum

MD18080 \$MA\_TOOL\_MANAGEMENT\_MASK das Bit 11 = 1 gesetzt werden, da das ASUP ggf. eine T-Anwahl wiederholen muss.

Anlagen mit Werkzeugverwaltung und Nebenspindel unterstützt SERUPO nicht!

## Beispiel

### Werkzeugwechselunterprogramm

Programmcode	Kommentar
PROC L6	; Werkzeugwechselroutine
N500 DEF INT TNR_AKTUELL	; Variable für aktive T-Nummer
N510 DEF INT TNR_VORWAHL	; Variable für vorgewählte T-Nummer
	; Aktuelles Werkzeug ermitteln
N520 STOPRE	; Im Programmtest-Betrieb wird
N530 IF \$P_ISTEST	; aus dem Programmkontext das
N540 TNR_AKTUELL = \$P_TOOLNO	; "aktuelle" Werkzeug gelesen.
N550 ELSE	; Andernfalls wird das Werkzeug der Spindel ausgelesen.
N560 TNR_AKTUELL = \$TC_MPP6[9998,1]	; T-Nummer des Werkzeugs auf der Spindel lesen
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	; T-Nummer des vorgewählten Werkzeugs der Masterspindel lesen. Werkzeugwechsel nur ausführen, wenn Werkzeug noch nicht aktuell ist.
N590 IF TNR_AKTUELL <> TNR_VORWAHL	; Werkzeugwechsellpunkt anfahren
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	
N610 M206	; Werkzeugwechsel ausführen
N620 ENDIF	
N630 M17	

**ASUP** für den Aufruf der Werkzeugwechselroutine nach Satzsuchlauf-Typ 5

Programmcode	Kommentar
PROC ASUPWZV2	
N1000 DEF INT TNR_SPINDEL	; Variable für aktive T-Nummer
N1010 DEF INT TNR_VORWAHL	; Variable für vorgewählte T-Nummer
N1020 DEF INT TNR_SUCHLAUF	; Variable für im Suchlauf ermittelte T-Nummer
N1030 TNR_SPINDEL = \$TC_MPP6[9998,1]	; T-Nummer des Werkzeugs auf der Spindel lesen
N1040 TNR_SUCHLAUF = \$P_TOOLNO	; durch Suchlauf ermittelte T-Nummer lesen, d.h. dieses Werkzeug bestimmt die momentane Werkzeugkorrektur.
N1050 GETSELT(TNR_VORWAHL)	; T-Nummer des vorgewählten Werkzeugs lesen
N1060 IF TNR_SPINDEL ==TNR_SUCHLAUF GOTOF ASUP_ENDE1	;
N1070 T = \$TC_TP2[TNR_SUCHLAUF]	; T-Anwahl über Werkzeugnamen
N1080 L6	; Werkzeugwechselroutine aufrufen
	;
N1085 ASUP_ENDE1:	;
N1090 IF TNR_VORWAHL == TNR_SUCHLAUF GOTOF ASUP_ENDE	;
N1100 T = \$TC_TP2[TNR_VORWAHL]	; T-Vorwahl restaurieren über Werkzeugnamen
	;
N1110 ASUP_ENDE:	;
N1110 M90	; Rückmeldung an PLC
N1120 REPOSA	; ;ASUP Ende

In beiden Programmen PROC L6 und PROC ASUPWZV2 wird der Werkzeugwechsel mit M206 anstelle mit M6 programmiert.  
 Das ASUP-Programm "ASUPWZV2" benutzt verschiedene Systemvariablen, die einerseits den Programmfortschritt (\$P\_TOOLNO) kennen und andererseits den aktuellen Zustand der Maschine (\$TC\_MPP6[9998,1] ) darstellen.

## Spindelhochlauf

Beim Start des SERUPRO-ASUPs wird die Spindel nicht auf die im Programm vorgesehene Drehzahl hochgedreht, denn mit SERUPRO-ASUP soll nach dem Werkzeugwechsel, das neue Werkzeug auf die richtige Werkstückposition korrigiert werden.

Ein Spindelhochlauf wird mit SERUPRO-ASUP wie folgt durchgeführt:

- SERUPRO-Vorgang ist komplett beendet.
- Der Anwender startet über den FC-9 Baustein das SERUPRO-ASUP mit dem ggf. die Spindel hochgedreht wird.
- Der Start nach M0 im ASUP verändert den Spindelzustand nicht.
- SERUPRO-ASUP stoppt vor dem REPOS-Teileprogrammsatz selbsttätig.
- Der Anwender drückt START.
- Die Spindel dreht auf den Zielsatzzustand hoch, falls im ASUP die Spindel nicht anders programmiert worden war.

---

### Hinweis

Anpassungen für REPOS der Spindeln:

Bei Anpassungen für SERUPRO-Anfahren und der Spindelfunktionalität sind Übergänge von Drehzahlsteuerbetrieb und Positionierbetrieb zu beachten.

Weitere Informationen zu den Betriebsartenwechsel von Spindeln siehe Kapitel "Betriebsarten (Seite 1384)".

---

## 8.8.5 Self-Acting SERUPRO

### Self-Acting SERUPRO

Die kanalspezifische Funktion "Self-Acting SERUPRO" erlaubt einen SERUPRO-Ablauf **ohne** vorher ein Suchziel in einem Programm der abhängigen SERUPRO-Kanäle definiert zu haben.

Außerdem kann ein spezieller Kanal, der "serurpoMasterChan", für **jedes** "Self-Acting SERUPRO" definiert werden. In diesem kann ein Suchziel definiert werden.

Die Funktion "Self-Acting SERUPRO" unterstützt den kanalübergreifenden Suchlauf SERUPRO.

### Funktion

Mit den Vorgang "Self-Acting SERUPRO" kann kein Suchziel gefunden werden. Wird das Suchziel nicht erreicht, so wird auch kein Kanal angehalten. In bestimmten Situationen wird aber trotzdem der Kanal vorübergehend angehalten. Dabei wird der Kanal in der Regel auf einen anderen Kanal warten. Beispiele hierfür sind: Wait-Marken, Kopplungen oder Achstausch.

**Warte-Phase tritt auf:**

In dieser Warte-Phase prüft die NC den Kanal "seruproMasterChan", ob dieser ein Suchziel erreicht hat. Die Warte-Phase wird bei Nichterreichen eines Suchziels wieder verlassen.

Wird das Suchziel erreicht, so wird der SERUPRO-Vorgang auch in diesem Kanal beendet. Der Kanal "seruproMasterChan" muss im normalen SERUPRO-Modus gestartet worden sein.

**Keine Warte-Phase tritt auf:**

"Self-Acting SERUPRO" wird durch das M30 des Teileprogramms beendet.

Der Kanalzustand befindet sich danach wieder im RESET-Zustand.

Ein SERUPRO-Anfahren findet nicht statt.

**Start einer Gruppe von Kanälen**

Wird eine Gruppe von Kanälen nur mit "Self-Acting SERUPRO" gestartet, so werden alle Kanäle mit "RESET" beendet.

Ausnahmen:

Ein Kanal wartet auf einen Partnerkanal, der überhaupt nicht gestartet wurde.

Ein kanalübergreifender Suchlauf kann wie folgt durchgeführt werden:

- Der Anwender wählt über HMI die Kanäle aus, die zusammenarbeiten müssen (Kanalgruppe).
- Aus der Kanalgruppe wählt der Anwender einen besonders wichtigen Kanal aus, für den er ein Suchziel explizit anwählen möchte (Zielkanal).
- Der HMI startet dann SERUPRO auf dem Zielkanal und "Self-Acting SERUPRO" auf dem Rest der Kanalgruppe.

Der Vorgang ist beendet, wenn **jeder** betroffene Kanal "seruproActive" gelöscht hat.

"Self-Acting SERUPRO" akzeptiert keinen Masterkanal auf einer anderen NCU.

**Aktivierung**

Die Aktivierung von "Self-Acting SERUPRO" erfolgt über HMI als Satzsuchlauf Start für Suchlauf-Typ 5 für den Zielkanal "seruproMasterChan".

Für die vom Zielkanal gestarteten abhängigen Kanäle wird kein Suchziel angegeben.



## 8.8.6 Programmabschnitt für Wiederaufsetzen sperren

### Programmierter Unterbrechungszeiger

Ist aufgrund von fertigungs- und/oder prozesstechnischen Gründen abzusehen, dass bei einem Programmabbruch innerhalb eines bestimmten Programmabschnitts kein Wiederaufsetzen möglich ist, kann dieser Programmabschnitt für den Zielsatz eines eventuellen Satzsuchlaufs gesperrt werden.

Erfolgt nach einem Programmabbruch innerhalb des für das Wiederaufsetzen gesperrten Programmabschnitts ein Satzsuchlauf auf die Unterbrechungsstelle, wird von der Steuerung dazu der letzte ausführbare Satz (Hauptlaufsatz) vor dem Beginn des gesperrten Bereichs als Zielsatz (Haltesatz) verwendet.

### Programmierung

#### Syntax

```
IPTRLOCK()
```

#### Funktionalität

Markiert den Anfang des Programmabschnitts, ab dem Wiederaufsetzen gesperrt ist. Als Zielsatz für einen Suchlauf mit "Wiederaufsetzen an der Unterbrechungsstelle" wird ab jetzt, bis zur Freigabe mit `IPTRUNLOCK`, der nächste ausführbare Satz (Hauptlaufsatz) verwendet, in dem `IPTRLOCK` aktiv wird. Dieser Satz wird im weiteren Verlauf als **Haltesatz** bezeichnet.

Wirksamkeit: modal

#### Syntax

```
IPTRUNLOCK()
```

Markiert das Ende des für ein Wiederaufsetzen gesperrten Programmabschnitts. Als Zielsatz für einen Suchlauf mit "Wiederaufsetzen an der Unterbrechungsstelle" wird ab dem nächsten ausführbaren Satz (Hauptlaufsatz) in dem `IPTRLOCK` aktiv wird, wieder der aktuelle Satz verwendet. Dieser Satz wird im weiteren Verlauf als **Freigabesatz** bezeichnet.

Wirksamkeit: modal

#### Beispiel

Programmcode	Kommentar
...	
N010 IPTRLOCK()	; Gesperrter Bereich: Anfang
N020 R1=R1+1	
N030 G4 F1	; <b>Haltesatz</b>
...	; Gesperrter Bereich
N200 IPTRUNLOCK()	; Gesperrter Bereich: Ende
N220 R1=R1+1	
N230 G4 F1	; <b>Freigabesatz</b>
...	

**Randbedingungen**

- IPTRLOCK wirkt innerhalb eines Programms (\*.MPF, \*.SPF) maximal bis zum Programmende (M30, M17, RET). Mit dem Programmende wird implizit IPTRUNLOCK aktiv.
- Eine mehrfache Programmierung von IPTRLOCK innerhalb eines Programms wirkt nicht kumulativ. Mit der ersten Programmierung von IPTRUNLOCK innerhalb des Programms oder mit Erreichen des Programmendes, werden alle vorausgegangenen IPTRLOCK-Aufrufe abgeschlossen.
- Erfolgt ein Unterprogrammaufruf innerhalb eines gesperrten Bereiches, ist auch für diese und alle eventuell folgenden Unterprogrammebenen das Wiederaufsetzen gesperrt. Die Sperre kann innerhalb des aufgerufenen Unterprogramms auch durch explizites Programmieren von IPTRUNLOCK nicht aufgehoben werden.

**Beispiel: Verschachtelung gesperrter Programmabschnitte in zwei Programmebenen**

Durch die Aktivierung der Wiederaufsetzsperrung in PROG\_1, ist Wiederaufsetzen auch für PROG\_2 und alle eventuell folgenden Programmebenen gesperrt.

Programmcode	Kommentar
PROC PROG_1	; Programm 1
...	
N010 IPTRLOCK()	
N020 R1=R1+1	
N030 G4 F1	; Haltesatz
...	; Gesperrter Bereich: Anfang
N040 PROG_2	; Gesperrter Bereich
...	; Gesperrter Bereich: Ende
N050 IPTRUNLOCK()	
N060 R2=R2+2	
N070 G4 F1	; Freigabesatz
...	

Programmcode	Kommentar
PROC PROG_2	; Programm 2
N210 IPTRLOCK()	; Unwirksam aufgrund von Programm 1
...	
N250 IPTRUNLOCK()	; Unwirksam aufgrund von Programm 1
...	
N280 RET	; Unwirksam aufgrund von Programm 1

**Beispiel 3: Mehrfache Programmierung von IPTRLOCK**

Programmcode	Kommentar
PROC PROG_1	; Programm 1
...	
N010 IPTRLOCK()	
N020 R1=R1+1	
N030 G4 F1	; Haltesatz
...	; Gesperrter Bereich: Anfang
N150 IPTRLOCK()	; Gesperrter Bereich
...	; Gesperrter Bereich
N250 IPTRLOCK()	; Gesperrter Bereich
...	; Gesperrter Bereich: Ende
N360 IPTRUNLOCK()	
N370 R2=R2+2	
N380 G4 F1	; Freigabesatz
...	

**Systemvariable**

Über die Systemvariable \$P\_IPTRLOCK kann der Status des aktuellen Satzes ermittelt werden:

\$P_IPTRLOCK	Bedeutung
FALSE	Der aktuelle Satz befindet sich <b>nicht</b> innerhalb eines für das Wiederaufsetzen gesperrten Programmabschnitts
TRUE	Der aktuelle Satz befindet sich innerhalb eines für das Wiederaufsetzen gesperrten Programmabschnitts

**Automatische funktionspezifische Wiederaufsetzsperrung**

Für verschiedene Kopplungen kann das Ein/Ausschalten der Wiederaufsetzsperrung kanalspezifisch automatisch mit dem Ein/Ausschalten der Kopplung erfolgen:

MD22680 \$MC\_AUTO\_IPTR\_LOCK, Bit x

Bit	Wert	Bedeutung
0	Elektronisches Getriebe ( EGON / EGOF)	
	1	Automatischen Wiederaufsetzsperrung ist aktiv
	0	Automatischen Wiederaufsetzsperrung ist nicht aktiv
1	Axiale Leitwertkopplung ( LEADON / LEADOF)	
	1	Automatischen Wiederaufsetzsperrung ist aktiv
	0	Automatischen Wiederaufsetzsperrung ist nicht aktiv

Dieser Programmbereich beginnt mit dem letzten ausführbaren Satz **vor** dem Einschalten und endet mit dem Ausschalten.

Der automatische Unterbrechungszeiger wird bei Kopplungen, die über Synchronaktionen ein- bzw. ausgeschaltet wurden, nicht aktiv.

**Beispiel:** Axiale Leitwertkopplung automatisch suchunfähig erklären:

Programmcode	Kommentar
N100 G0 X100	
N200 EGON(Y, "NOC", X, 1, 1)	; Suchunfähiger Programmabschnitt beginnt.
N300 LEADON(A, B, 1)	
...	
N400 EGOFS(Y)	
...	
N500 LEADOF(A, B)	; Suchunfähiger Programmabschnitt endet.
N600 G0 X200	

Ein Programmabbruch im suchunfähigen Programmabschnitt (N200 - N500) versorgt den Unterbrechungszeiger stets mit N100.

**VORSICHT**

Bei einer Überschneidung der Funktionen "Programmierbarer Unterbrechungszeiger" und "Automatischer Unterbrechungszeiger" über Maschinendatum wählt die NC den größtmöglichen suchunfähigen Bereich.

Ein Programm kann eine Kopplung fast über die gesamte Laufzeit benötigen. Der automatische Unterbrechungszeiger würde damit immer auf den Programmanfang weisen und die Funktion SERUPRO würde de facto nutzlos.

## 8.8.7 Besonderheiten im Zielsatz des Teileprogramms

### 8.8.7.1 STOPRE im Zielsatz des Teileprogramms

#### STOPRE-Satz

Alle satzübergreifenden Einstellungen erhält der STOPRE-Satz aus dem vorangegangenen Satz und kann damit Bedingungen vor dem eigentlichen Satz für die folgenden Fälle berücksichtigen:

- Aktuell bearbeitete Programmzeile mit dem Hauptlauf synchronisieren.
- Satzübergreifende Einstellungen für SERUPRO ableiten, um z. B. beim Anfahren von SERUPRO diese REPOS-Bewegung zu beeinflussen.

#### Beispiel 1:

Durch Sollwertvorgabe einer X-Achse eine Z-Achse positionieren.

Wenn der Satz "G1 F100 Z=\$AA\_IM[X]" interpretiert wird, so sorgt der vorangestellte STOPRE-Satz für die Synchronisation mit dem Hauptlauf. Damit wird über \$AA\_IM der korrekte Sollwert der X-Achse gelesen, um die Z-Achse auf die gleiche Position zu fahren.

#### Beispiel 2:

Externe Nullpunktverschiebung lesen und richtig einrechnen.

Programmcode	Kommentar
N10 G1 X1000 F100	;
N20 G1 X1000 F500	;
N30 G1 X1000 F1000	;
N40 G1 X1000 F5000	;
N50 SUPA G1 F100 X200	; externe Nullpunktverschiebung nach 200 fahren
N60 G0 X1000	;
N70 ...	;

Durch ein implizites STOPRE vor N50 kann der NCK die aktuelle Nullpunktverschiebung lesen und richtig einrechnen.

Bei einem SERUPRO-Vorgang auf das Suchziel N50 wird im SERUPRO-Anfahren auf den impliziten STOPRE repositioniert und die Geschwindigkeit ermittelt sich aus N40 mit F5000.

### Impliziter Vorlaufstopp

Situationen, in denen vom Interpreter ein impliziter Vorlaufstopp abgesetzt wird:

1. In allen Sätzen in denen einer der folgenden Variablenzugriffe vorkommt: -  
- Programmierung einer Systemvariablen, die mit \$A... beginnt  
- redefinierte Variable mit den Attribut SYN/R/SYNRW
2. Bei folgenden Teileprogrammbeehlen:  
- Teileprogrammbeehl MEACALC, MEASURE  
- Programmierung von SUPA (Frames und Online-Korrekturen unterdrücken)  
- Programmierung CTABDEF (Beginn der Kurventabellendefinition)  
- Teileprogrammbeehl WRITE/DELETE (Datei schreiben/löschen)  
- vor dem ersten WRITE/DELETE-Befehl einer Sequenz solcher Befehle  
- Teileprogrammbeehl EXTCALL  
- Teileprogrammbeehl GETSELT, GETEXET  
- bei Werkzeugwechsel und aktiver Werkzeugfeinkorrektur FTOCON
3. Bei folgenden Kommando-Bearbeitungen:  
- Endbearbeitung von Suchlauf vom Typ 1 ("Suchlauf ohne Berechnung") und vom Typ 2 mit Berechnung ("Suchlauf an Konturendpunkt")  
**Hinweis:** Der Suchlauf Typ 2 "Suchlauf an Konturanfangspunkt" verhält sich analog.

### 8.8.7.2 SPOS im Zielsatz

#### SPOS

Ist eine Spindel mit M3/M4 programmiert und wird im Zielsatz nach SPOS umgeschaltet, so ist am Ende des SERUPRO-Vorganges (Suchziel gefunden) die Spindel nach SPOS umgeschaltet. Das VDI-Interface zeigt dies an.

### 8.8.8 Verhalten bei Power On, Betriebsartenwechsel und RESET

SERUPRO ist bei Power On inaktiv. Während SERUPRO ist der Betriebsartenwechsel erlaubt. RESET bricht SERUPRO ab, der intern gewählte Programmtest ist wieder abgewählt. SERUPRO ist nicht mit anderen Suchlauftypen kombinierbar.

## 8.8.9 Randbedingungen

### 8.8.9.1 Fahren auf Festanschlag (FXS)

Beim Wiederauffahren an die Kontur (REPOS) wird die Funktion "Fahren auf Festanschlag" ( $\text{FXS}$ ) automatisch wiederholt. Dabei wird jede Achse berücksichtigt. Als Moment wird das zuletzt vor dem Suchziel programmierte Moment angewendet.

#### Systemvariable

Die Systemvariablen für "Fahren auf Festanschlag" haben bei SERUPRO folgende Bedeutung:

- \$AA\_FXS: Fortschritt der Programmsimulation
- \$VA\_FXS: Realer Maschinenzustand

Die Systemvariablen haben **außerhalb** der Funktion SERUPRO immer dieselben Werte.

#### ASUP

Für SERUPRO kann ein anwenderspezifisches ASUP aktiviert werden.

## Literatur

Ausführliche Informationen zum Satzsuchlauf SERUPRO finden sich in Kapitel "Allgemeine Funktionalität (Seite 304)".

### 8.8.9.2 Fahren mit begrenztem Moment/Kraft (FOC)

Beim Wiederauffahren an die Kontur (REPOS) wird die Funktion "Fahren mit begrenztem Moment/Kraft" ( $\text{FOC}$ ) automatisch wiederholt. Dabei wird jede Achse berücksichtigt. Als Moment wird das zuletzt vor dem Suchziel programmierte Moment angewendet.

#### Systemvariable

Die Systemvariablen für "Fahren mit begrenztem Moment/Kraft" haben bei SERUPRO folgende Bedeutung:

- \$AA\_FOC: Fortschritt der Programmsimulation
- \$VA\_FOC: Realer Maschinenzustand

## Randbedingung

Ein wechselnder Momentenverlauf kann beim Wiederanfahren an die Kontur **nicht** realisiert werden.

### Beispiel

Ein Programm fährt eine Achse X von Position 0 nach 100 und schaltet alle 20 Incremente für jeweils 10 Incremente "Fahren mit begrenztem Moment/Kraft" (FOC) ein. Dieser Momentenverlauf wird in der Regel mit satzweisem FOC erzeugt und kann beim Wiederanfahren an die Kontur (REPOS) nicht nachvollzogen werden. Stattdessen wird gemäß der letzten Programmierung die Achse X von 0 nach 100 mit oder ohne begrenztem Moment/Kraft fahren.

## Literatur

Ausführliche Informationen zum Satzsuchlauf SERUPRO finden sich in Kapitel "Allgemeine Funktionalität (Seite 304)".

### 8.8.9.3 Synchronspindel

#### Synchronspindel ist simulierbar.

Der Synchronspindelbetrieb mit einer Leitspindel und beliebig vielen Folgespindeln ist in allen vorhandenen Kanälen mit SERUPRO simulierbar.

Weitere Informationen zu Synchronspindeln entnehmen Sie bitte:

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Synchronspindel (S3)

### 8.8.9.4 Kopplungen und Master-Slave

#### Soll- und Istwert-Kopplungen


Der SERUPRO-Vorgang ist eine Programmsimulation im Modus Programmtest mit dem Soll- und Istwertkopplungen simulierbar sind.



## Festlegungen für Simulation von EG

Für die Simulation von EG werden damit folgende Festlegungen getroffen:

1. Es wird immer mit Sollwertkopplung simuliert.
2. Sind nur einige Leitachsen, d. h. nicht alle Leitachsen unter SERUPRO, wird die Simulation mit Alarm 16952 "ResetClear/NoStart" abgebrochen. Dies kann bei kanalübergreifenden Kopplungen auftreten.
3. Achsen, die aus NCK-Sicht nur einen Geber haben und von Extern bewegt werden, können nicht korrekt simuliert werden. Insbesondere dürfen diese Achsen nicht in Kopplungen einfließen.

 <b>VORSICHT</b>
Um Kopplungen korrekt simulieren zu können, müssen die Kopplungen vorher abgeschaltet werden.
Dies kann mit Maschinendatum MD10708 \$MA_SERUPRO_MASK erfolgen.

## Festlegungen für Achskopplungen

Der SERUPRO-Vorgang simuliert Achskopplungen stets unter der Annahme von Sollwertkopplung. Damit werden Endpunkte für **alle** Achsen berechnet, die als Zielpunkte für SERUPRO-Anfahren benutzt werden. Zugleich ist die Kopplung bereits mit "Suchziel gefunden" aktiv. Der Weg vom aktuellen Punkt bis zum Endpunkt wird bei SERUPRO-Anfahren unter aktiver Kopplung durchgeführt.

### LEADON

Für die Simulation von Axialen Leitwertkopplungen ergeben sich damit folgende Festlegungen:

1. Es wird immer mit Sollwertkopplung simuliert.
2. SERUPRO-Anfahren erfolgt mit aktiver Kopplung und einer überlagerten Bewegung der Folgeachse, um den simulierten Zielpunkt zu erreichen.

Die allein durch die Kopplung bewegte Folgeachse kann den Zielpunkt nicht immer erreichen. Es wird im SERUPRO-Anfahren eine Überlagerte lineare Bewegung für die Folgeachse berechnet, um den simulierten Punkt anzufahren!

### Simulierten Zielpunkt für LEAD mit JOG erreichen

Zum Zeitpunkt "Suchziel gefunden" ist die Kopplung insbesondere für die JOG-Bewegungen bereits aktiv. Bei nicht erreichten Zielpunkt kann beim SERUPRO-Anfahren die Folgeachse mit aktiver Kopplung und einer überlagerten Bewegung auf den Zielpunkt verfahren werden.

---

### Hinweis

Weitere Informationen zum Wiederaufsetzen von Achskopplungen siehe Kapitel "Wiederaufsetzen nach SERUPRO-Suchziel gefunden (Seite 541)".

---

**Master-Slave**

Nach Abschluss des Satzsuchlaufs kann ein System ASUP automatisch gestartet werden. In diesem hat der Anwender die Möglichkeit den Kopplungszustand und die zugehörigen Achspositionen nachträglich zu beeinflussen. Die dazu erforderlichen Informationen werden über folgende Systemvariable bereitgestellt:

Systemvariable	Beschreibung
\$P_SEARCH_MASLD[<Slaveachse>]	Positionsversatz zwischen Slave- und Masterachse zum Schließzeitpunkt der Kopplung.
\$AA_MASL_STAT[<Slaveachse>]	Aktueller Zustand einer Master-Slave Kopplung
\$P_SEARCH_MASLC[<Slaveachse>]	Status: der Zustand der Kopplung wurde im Satzsuchlauf verändert
Die Systemvariablen werden beim Einschalten der Kopplung ( MASLON) gelöscht.	

**Hinweis**

Die gekoppelten Achsen müssen sich zum Zeitpunkt des Satzsuchlaufs im gleichen Kanal befinden.

Weitere Informationen zur Kopplung Master-Slave finden sich in:

**Literatur:**

Funktionshandbuch Sonderfunktionen; Drehzahl-/Drehmomentkopplung (TE3) Master-Slave

Der System ASUP wird progevent.spf genannt und muss im Verzeichnis /\_N\_CMA\_DIR vorhanden sein. Der Inhalt könnte wie folgt aussehen:

progevent.spf  
 X=Masterachse, Y=Slaveachse

```

Programmierung
N10 IF((($S_SEARCH_MASLC[Y]< >0) AND ($AA_MASL_STAT[Y]< >0))
N20 MASLOF(Y)
N30 SUPA Y=$AA_IM[X] -$P_SEARCH_MASLD[Y]
N40 MASLON(Y)
N50 ENDIF
N60 REPOSA
    
```

Damit das ASUP automatisch gestartet werden kann, müssen folgende Maschinendaten gesetzt werden:

- NC-spezifisch:
  - MD11602 \$MN\_ASUP\_START\_MASK = 'H01'
  - MD11604 \$MN\_ASUP\_START\_PRIO\_LEVEL = 100
  - MD11450 \$MN\_SEARCH\_RUN\_MODE = 'H02'
- Kanal-spezifisch für den Kanal in dem das ASUP gestartet wird, oder allgemein für alle Kanäle:
  - MD20105 \$MC\_PROG\_EVENT\_IGN\_REFP\_LOCK = 'H3F'
  - MD20115 \$MC\_IGNORE\_REFP\_LOCK\_ASUP = 'HFFFFFFF'

## Achskopplungen

---

### Hinweis

Bei einer Leitachse, deren Folgeachsen in einem anderen Kanal sind, wirkt die Einstellung zur Beschleunigung der Abarbeitungsgeschwindigkeit nicht:

MD22601 \$MC\_SERUPRO\_SPEED\_FACTOR > 0

---

### Mitschleppen

Die Funktion Mitschleppen (`TRAILON`) wird von SERUPRO unterstützt.

Weitere Informationen zu Mitschleppen mit `TRAILON`, `TRAILOF` finden sich in:

#### Literatur:

- Funktionshandbuch Sonderfunktionen; Achskopplungen (M3)
- Programmierhandbuch Arbeitvorbereitung; Achskopplungen

### Gantry-Achsen

Die Funktion Gantry-Achsen wird von SERUPRO unterstützt.

Weitere Informationen zur Funktionalität von Gantry-Achsen finden sich in:

#### Literatur:

Funktionshandbuch Sonderfunktionen; Gantry-Achsen (G1)

### Tangentialsteuerung

Die Funktion tangentiales Nachführung einzelner Achsen wird von SERUPRO unterstützt.

Weitere Informationen zur Tangentialsteuerung entnehmen Sie bitte:

#### Literatur:

Funktionshandbuch Sonderfunktionen; Tangentialsteuerung (T3)

### 8.8.9.5 Achsfunktionen

#### SERUPRO Bedingungen

Bei Achsfreigabe, Autarken Achsvorgängen und Achstausch müssen die besonderen Bedingungen für SERUPRO berücksichtigt werden.

#### Achsfreigabe

Das axiale NST DB31, ... DBX3.7 ("Programmtest Achs-/Spindel Freigabe") beeinflusst die Achsfreigaben, wenn an die Maschine keine Reglerfreigabe gegeben werden soll oder kann und wirkt nur während Programmtest oder SERUPRO aktiv ist.

Es besteht die Möglichkeit über das Nahtstellensignal PLC → NCK DB31, ... DBX3.7 (Programmtest Achs-/Spindel Freigabe) diese Freigabe zu geben. Fehlt bei Programmtest oder SERUPRO die reale Reglerfreigabe, so hat das folgenden Einfluss auf Achsen/Spindeln:

- Sobald der simulierte Programmablauf eine Achse/Spindel verfahren möchte, wird die Meldung "Warten auf Achsfreigabe" oder "Warten auf Spindelfreigabe" angezeigt und die Simulation stoppt.
- Wird während einer simulierten Verfahrbewegung das NC/PLC-Nahtstellensignal DB31, ... DBX3.7 (Programmtest Achs-/Spindel Freigabe) wieder weggenommen, wird der Alarm 21612: "Kanal %1 Achse %2 NC/PLC-Nahtstellensignal 'Reglerfreigabe' während der Bewegung zurückgesetzt" ausgelöst.

#### Autarke Achsvorgänge

Autarke Einzelachsvorgänge sind vom PLC kontrollierte Achsen, die bei SERUPRO auch simuliert werden. Damit wird während SERUPRO wie im normalen Ablauf, die PLC die Kontrolle einer Achse übernehmen oder abgeben. Diese Achse kann gegebenenfalls auch über FC18 verfahren werden. Die PLC übernimmt die Kontrolle der Achse **vor** dem Anfahransatz und ist für die Positionierung dieser Achse verantwortlich. Dies gilt für alle Satzsuchlauf Typen.

Weitere Informationen zu autarken Einzelachsvorgängen entnehmen Sie bitte:

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

#### Achstausch

Problem: Ein Programm verfährt eine Achse und gibt sie vor dem Zielsatz mit WAITP(X) ab. Damit unterliegt X nicht dem REPOS und die Achse wird beim SERUPRO-Anfahren nicht berücksichtigt.


Über das Maschinendatum MD11470 \$MN\_REPOS\_MODE\_MASK kann für SERUPRO-REPOS folgendes Verhalten erzielt werden:

Die neutralen Achsen werden als "Kommando-Achsen" im SERUPRO-REPOS verfahren. Die Achse interpoliert ohne Bahnzusammenhang, auch wenn sie zuletzt als Bahnachse programmiert worden war. In diesem Fall ergibt sich die Geschwindigkeit aus MD32060 \$MA\_POS\_AX\_VELO. Nach dem SERUPRO-Anfahren ist diese Achse wieder neutral.

Neutrale Achsen, die trotzdem nicht repositioniert werden dürfen, müssen mit dem axialen NC/PLC-Nahtstellensignal "REPOSDELAY" beaufschlagt werden. Damit wird ihre REPOS Bewegung gelöscht.

**Beispiel:**

Nach SERUPRO wird via Technologiezyklen in der Synchronaktion eine Achse bewusst bewegt. Die Kommando-Achsen werden immer im Anfahrstakt und nie im Zielsatz bewegt. Der Zielsatz kann erst eingewechselt werden, wenn alle Kommando-Achsen zu Ende gefahren sind.

 <b>VORSICHT</b>
Die vom PLC kontrollierte Achse wird nicht repositioniert!
Achsen, die mit RELEASE(X) vor dem Zielsatz freigegeben worden sind, werden nicht repositioniert.

### 8.8.9.6 Getriebestufenwechsel

#### Abläufe

Der Getriebestufenwechsel (GSW) erfordert von der NCK physikalische Bewegungen, um einen neuen Gang einlegen zu können.  
Im SERUPRO-Vorgang ist ein Getriebestufenwechsel nicht erforderlich und wird wie folgt durchgeführt:

Manche Getriebe können nur NC-geführt gewechselt werden, da entweder die Achse pendeln, oder vorher eine bestimmte Position angefahren werden muss.

Der Getriebestufenwechsel kann im MD35035 \$MA\_SPIND\_FUNCTION\_MASK mit den Bits 0 bis 2 selektiv für DryRun, Programmtest und SERUPRO unterdrückt werden.

Der GSW muss dann im REPOS nachgeholt werden und funktioniert auch, wenn die zugehörige Achse am Zielsatz im "Drehzahlsteuerbetrieb" sein soll. Andernfalls wird der automatische GWS mit einem Alarm abgelehnt, wenn im Teileprogramm zwischen dem GWS und dem Zielsatz die Achse u. a. in eine Transformation oder Kopplung aufgenommen worden war.

---

**Hinweis**

Weitere Informationen zum Getriebestufenwechsel bei DryRun, Programmtest und SERUPRO siehe Kapitel "S1: Spindeln (Seite 1383)".

---

### 8.8.9.7 Überlagerte Bewegung

#### Nur SERUPRO

Werden "Überlagerte Bewegungen" verwendet, so kann nur der Satzsuchlauf über Programmtest (SERUPRO) verwendet werden, da dabei die überlagerten Bewegungen im Hauptlauf entsprechend interpoliert werden. Dies gilt insbesondere für \$AA\_OFF.

#### Geschwindigkeitsprofil statt maximale Achsgeschwindigkeit

Im Programmtest muss ein Geschwindigkeitsprofil verwendet werden, welches es erlaubt "Überlagerte Bewegungen" im Hauptlauf zu interpolieren. Es kann somit nicht mit der maximalen Achsgeschwindigkeit interpoliert werden.

Die Achsgeschwindigkeit wird im Modus "Probelaufvorschub" über SD42100 \$SC\_DRY\_RUN\_FEED eingestellt.

Die Geschwindigkeit des SERUPRO-Vorgangs wird über MD22600 \$MC\_SERUPRO\_SPEED\_MODE ausgewählt.

### 8.8.9.8 NC/PLC-Nahtstellensignale

#### REPOS-Verschiebung vorhanden

Hat sich während SERUPRO für eine Achse eine REPOS-Verschiebung ergeben, wird dies mit Abschluss des SERUPRO-Vorganges über die axiale NC/PLC-Nahtstelle angezeigt:

DB31, ...DBX70.0 == 1 (REPOS-Verschiebung vorhanden)

#### Gültigkeitsbereich der REPOS-Verschiebung

Mit dem Start eines SERUPRO-ASUP oder NC-Start zum Fortsetzen der Bearbeitung, wird die REPOS-Verschiebung ungültig:

DB31, ... DBX70.1 == 1 (REPOS-Verschiebung ungültig)

Zwischen dem Abschluss des SERUPRO-Vorganges und NC-Start zum Fortsetzen der Bearbeitung, kann die Achse in der Betriebsart JOG manuell oder über das PLC-Anwenderprogramm mittels FC 18 verfahren werden. Wird dabei die REPOS-Verschiebung vollständig herausgefahren, wird das Nahtstellensignal zurückgesetzt.

### 8.8.9.9 Flexibilisierung der Grundeinstellung

#### Grundeinstellung / SERUPRO-Grundeinstellung

Mit dem Maschinendatum MD20112 \$MC\_START\_MODE\_MASK wird die Grundstellung der Steuerung bei Teileprogrammstart bezüglich der G-Codes (insbesondere die aktuelle Ebene und einstellbare Nullpunktverschiebung), Werkzeuglängenkorrektur, Transformation und Achskopplungen festgelegt. Speziell für den SERUPRO-Vorgang besteht die Möglichkeit mit MD22620 \$MC\_ENABLE\_START\_MODE\_MASK\_PRT

eine zum normalen Teileprogrammstart abweichende Grundeinstellung zu wählen. Die neue Einstellung muss dabei hinterlegt sein in:

MD22620 \$MC\_START\_MODE\_MASK\_PRT

Die Bedeutung der jeweiligen Bits von MD22620 ist identisch mit denen von:

MD20112 \$MC\_START\_MODE\_MASK.

#### Beispiel:

Beim Teileprogrammstart bleibt die Synchronspindelkopplung zu Beginn des SERUPRO Vorgangs erhalten.

\$MC_START_MODE_MASK = 'H400'	; nicht projektierte Synchronspindelkopplung
\$MC_START_MODE_MASK_PRT = 'H00'	; wird ausgeschaltet
\$MC_ENABLE_START_MODE_MASK_PRT = 'H01'	; bleibt aktiv
	; \$MC_START_MODE_MASK_PRT wird bei SERUPRO ausgewertet statt \$MC_START_MODE_MASK

## 8.8.10 Systemvariable

### Ermittlung des SERUPRO-Status

Über folgende Systemvariable kann der aktuelle SERUPRO-Status im Teileprogramm oder Synchronaktion ermittelt werden:

Systemvariable	Bedeutung
\$AC_ASUP, Bit 20	ASUP Aktivierungsgrund: System-ASUP, aufgrund Suchlauf-Typ SERUPRO hat das Suchziel erreicht
\$AC_SERUPRO	SERUPRO aktiv
\$P_ISTEST	Programmtest aktiv; \$P_ISTEST == 1 von SERUPRO Anfang bis Ende
\$P_SEARCHL	Zuletzt aktiver Suchlauf-Typ; \$P_SEARCHL == 5 von SERUPRO Anfang bis Reset oder Programmende

### \$AC\_SERUPRO und \$P\_ISTEST

---

#### Hinweis

Im Vorlauf wird bei Systemvariable \$P\_ISTEST und \$AC\_SERUPRO geprüft, ob der SERUPRO Zielsatz bereits gefunden wurde. Ist dies der Fall, wird vor der Auswertung der Systemvariablen ein impliziter Vorlaufstop eingefügt. Dies hat zur Folge, dass die Interpretation angehalten und erst wieder fortgesetzt wird, wenn auch im Hauptlauf SERUPRO ausgeschaltet ist. Die Entscheidung SERUPRO aktiv oder inaktiv wird dann richtig getroffen.

---



## 8.9 Programmbetrieb

### PLC, MD, Bedienung

Der Ablauf von Teileprogrammen kann auf vielerlei Art durch PLC-Vorgaben, Maschinendateneinstellungen und Bedienungen über HMI beeinflusst werden.

### Definition

Programmbetrieb liegt dann vor, wenn in der Betriebsart AUTOMATIK oder MDA Teileprogramme bzw. Teileprogrammsätze abgearbeitet werden.

### Beeinflussung

Jeder Kanal kann über Nahtstellensignale von der PLC beeinflusst werden. Die Beeinflussung geschieht über BAG-spezifische oder über kanalspezifische Nahtstellensignale. Eine Übersicht dieser Signale sehen Sie unter Datenlisten in dieser Funktionsbeschreibung.

#### Rückmeldungen der Beeinflussung

Jeder Kanal teilt der PLC über Nahtstellensignale seinen momentanen Programmbetriebsstatus mit. Diese Signale sind wieder in BAG- und kanalspezifische Signale aufgeteilt.

### 8.9.1 Grundstellungen

#### Maschinendaten

Über Maschinendaten können definierte Bedingungen für den Programmbetrieb oder bestimmte Ausprägungen des NC-Sprachumfangs eingestellt werden.

#### Grundstellungen

Für jeden Kanal können Grundstellungen im über kanalspezifische Maschinendaten vorgegeben werden. Diese Grundstellungen wirken u. a. auf die G-Gruppen sowie auf die Hilfsfunktionsausgabe.

#### Hilfsfunktionsausgabe

Die zeitliche Ausgabe der Hilfsfunktionen kann über die Maschinendaten AUXFU\_x\_SYNC\_TYPE (MD22200, 22210, 22220, 22230, 22240, 22250, 22260), (Ausgabezeitpunkt der M, S, T, H, F, D, E-Funktionen) vorbestimmt werden. Nähere Erläuterungen dazu siehe Kapitel "H2: Hilfsfunktionsausgaben an PLC (Seite 395)".

### G-Gruppen

Für jede der vorhandenen G-Gruppen kann über MD20150 \$MC\_GCODE\_RESET\_VALUES (Löschstellung der G-Gruppen) eine Programmiergrundstellung vorgegeben werden. Diese Programmiergrundstellung ist automatisch beim Programmstart bzw. im Reset-Zustand wirksam, bis sie durch einen G-Befehl derselben G-Gruppe abgewählt wird.

Über das MD22510 \$MC\_GCODE\_GROUPS\_TO\_PLC (G-Codes, die bei Satzwechsel/RESET an Nahtstelle NCK-PLC ausgegeben werden) kann die Ausgabe der G-Codes an die PLC-Nahtstelle aktiviert werden.

Eine Auflistung der G-Gruppen mit den zugehörigen G-Funktionen entnehmen Sie bitte:

#### Literatur:

Programmierhandbuch Grundlagen

### Grundkonfigurationen des NC-Sprachumfangs bei SINUMERIK solution line

Für SINUMERIK 840D sl können bestimmte Grundkonfigurationen des NC-Sprachumfangs über Maschinendaten projektiert generiert werden. Damit wird speziell für den Anwender unter Berücksichtigung seiner benötigten Optionen und Funktionen der NC-Sprachumfang einheitlich auf Ihm zugeschnitten konfiguriert.

### NC-Sprachumfang

Wie mit NC-Sprachbefehlen nicht aktiver Optionen und Funktionen verfahren werden soll, kann mit dem MD10711 \$MN\_NC\_LANGUAGE\_CONFIGURATION eingestellt werden:

**0:** Alle verfügbaren Sprachbefehle sind programmierbar. Erst bei der Ausführung wird erkannt, ob die benötigte Funktion aktiviert ist.

Wenn nur **bestimmte Optionen** freigegeben sind und **nicht alle Optionen** vorhanden sind:

**1:** Es sind **alle Sprachbefehle bekannt**. Sprachbefehle zu nicht freigegebenen Optionen werden bereits zu Beginn der Programminterpretation erkannt und führen zum Alarm 12553 "Option/Funktion ist nicht aktiv".

**2:** Nur die Sprachbefehle sind bekannt, die dem **aktuellen Umfang freigegebener Optionen** der NCK-Software entsprechen. Alle Befehle zu nicht freigegebene Optionen werden nicht erkannt und führen zum Alarm 12550 "Name nicht definiert oder Option/Funktion nicht vorhanden".

---

#### Hinweis

Optionsfreie Funktionen haben auch den Status "Option freigegeben"

---

Wenn nur **bestimmte Funktionen aktiviert** sind:

**3:** Es sind **alle NC-Sprachebefehle bekannt**. Nicht aktivierte Funktionen werden bereits zu Beginn der Programminterpretation erkannt und haben den Alarm 12553 "Option/Funktion ist nicht aktiv" zur Folge. Wird zum Beispiel für die Zylindermanteltransformation das Optionsdatum gesetzt, aber die Transformation im Maschinendatum MD24100 \$MC\_TRAOF\_TYPE\_1 nicht aktiviert, so führt bereits die Programmierung von TRACYL zum Alarm 12553.

**4:** Es sind nur die NC-Sprachbefehle bekannt, die dem **aktuellen Umfang aktiver Funktionen** der NCK-Software entsprechen. Alle Befehle zu nicht aktiven Funktionen werden nicht erkannt und führen zum Alarm 12550 "Name nicht definiert oder Option/Funktion nicht vorhanden". Ob der betreffende Befehl generell in der Siemens NC-Sprache oder nur auf der entsprechenden Anlage nicht vorhanden ist, kann in diesem Fall nicht unterschieden werden.

Ob der aktuelle NC-Sprachumfang freigegebener Optionen und aktiver Funktionen auch wirklich programmierbar ist kann mit dem Programmierbefehl STRINGIS geprüft werden, siehe Beispiel.

### Anwendungsbeispiel für NC-Sprachumfang auf Zylindermanteltransformation TRACYL prüfen

Die Zylindermanteltransformation ist eine Option die vorher freigegeben werden muss. Um dies zu überprüfen werden folgende Anfangsbedingungen angenommen:

Die Option Zylindermanteltransformation ist **nicht** freigegeben und das Maschinendatum \$MN\_NC\_LANGUAGE\_CONFIGURATION = 2; NC-Sprachbefehl TRACYL ist unbekannt

**Folgendes Programm wird gestartet**

Programmcode	Kommentar
N1 R1=STRINGIS("TRACYL")	; R1 ist 0 (TRACYL ist unbekannter Name)
N2 IF STRINGIS("TRACYL")==204	
N3 TRACYL(1, 2, 3)	; Satz wird nicht interpretiert
N4 ELSE	
N5 G00	
N6 ENDIF	
N7 M30	

**Beispiel ob STRINGIS Ergebnis programmierbar oder nicht programmierbar**

Ergebnis von STRINGIS = stellenkodierter Rückgabewert (dreistellig)

Stellenkodierung der Basisinformationen (1. Stelle von links):

000 Name ist unbekannt, Programmierung wird mit Alarm 12550 abgewiesen

100: Name ist bekannt aber nicht programmierbar führt zum Alarm 12533

200: Name/Symbol ist bekannt, aber es ist keine Interpretation möglich

2xx: Name/Symbol ist bekannt, der **Befehl kann programmiert** werden, wenn xx > 0

Definition für Name/Symbol:

Name: beliebiger STRING, der darauf geprüft wird, ob er in der vorhandenen NCK-Version bzw. Konfiguration Bestandteil der NC-Sprache ist.

Symbol: enthält die Beschreibung bzw. Bedeutung eines NC-Sprachbefehls, die für die NC-Programm- und Zykleninterpretation benötigt wird.

Abhängig vom Maschinendatum MD10711

\$MN\_NC\_LANGUAGE\_CONFIGURATION = (eingestellter Wert) ergeben sich folgende Interpretationen der Option und Funktion bezogen auf ihre Programmierfähigkeit 2xx:

Tabelle 8- 1 Einstellungsmöglichkeiten

MD10711 =		0	1	2	3	4
Option	Funktion	Rückgabewert als Basisinformation (1. Stelle von links)				
0	0	2	1	0	1	0
1	0	2	2	2	1	0
1	1	2	2	2	2	2
0	1	2	1	0	1	0

Definition für Option/Funktion:

0 entspricht Option ist nicht aktiviert bzw. Funktion ist ausgeschaltet

1 entspricht Option/ ist aktiviert bzw. Funktion ist eingeschaltet

Weitere Detailinformationen zu den Wertebereichen 2xx programmierbarer Funktionen siehe:

**Literatur:**

Programmierhandbuch Arbeitsvorbereitung; Weitere Funktionen, Kapitel: Vorhandenen NC-Sprachumfang prüfen (STRINGIS)

## 8.9.2 Anwahl und Starten des Teileprogramms bzw. Teileprogrammsatzes

### Reset-Zustand

#### Kanalzustand

Die Anwahl eines Teileprogrammes kann nur erfolgen, wenn sich der betreffende Kanal im Reset-Zustand befindet.

#### Startkommando, Kanalzustand

Als START-Kommando für die Abarbeitung bestehen zwei Möglichkeiten:

- Das kanalspezifische NST DB21, ... DBX7.1(NC-Start), das üblicherweise von der MSTT-Taste NC-Start beeinflusst wird, startet die Programmabarbeitung im gleichen Kanal.
- Über den NC-Anweisung START kann z. B vom zweiten Kanal aus die Programmabarbeitung im ersten Kanal gestartet werden.

Das START-Kommando wird nur in den Betriebsarten AUTOMATIK und MDA ausgeführt. Der betreffende Kanal muss dazu im Zustand:

DB21, ... DBX35.7(Kanalzustand Reset) bzw.

DB21, ... DBX35.6 (Kanalzustand unterbrochen) sein.

### Signale, Alarme

#### Notwendige Signalzustände

Das Teileprogramm kann nun, vorausgesetzt dass an der Maschine bestimmte Signalzustände vorhanden sind, mit dem START-Kommando zur Abarbeitung im Kanal freigegeben werden.

Folgende Freigabesignale sind an der NC/PLC-Nahtstelle relevant:

- DB11 DBX4.4 (BAG-betriebsbereit) muss anstehen
- DB11 DBX0.7 (BAG-Reset) darf nicht anstehen
- DB21, ... DBX1.7 (Programmtest aktivieren) darf nicht anstehen
- DB21, ... DBX7.0 (NC-Start-Sperre) darf nicht anstehen
- DB21, ... DBX7.2 (NC-Stop an Satzgrenze) darf nicht anstehen
- DB21, ... DBX7.3 (NC-Stop) darf nicht anstehen
- DB21, ... DBX7.4 (NC-Stop Achsen plus Spindel) darf nicht anstehen
- DB21, ...:DBX7. 7 (Reset) darf nicht anstehen
- DB10 DBX56.1 (Not-Halt) darf nicht anstehen
- Achs- oder NCK-Alarm darf nicht anstehen

Erläuterungen zu den einzelnen Signalen entnehmen Sie bitte Kapitel 5.

#### Ausführung des Kommandos

Das Teileprogramm bzw. der Teileprogrammsatz wird automatisch abgearbeitet und die folgenden Nahtstellensignale werden gesetzt:

DB21, ... DBX35.5 (Kanalzustand aktiv)

DB21, ... DBX35.0 (Programmzustand läuft)

Das Programm wird solange bearbeitet, bis das Programmende erreicht bzw. der Kanal durch ein STOP- oder RESET-Kommando unterbrochen bzw. abgebrochen wird.

#### Alarmer

Das START-Kommando ist unter bestimmten Umständen nicht wirksam und es tritt dann einer der folgenden Alarmer auf:

- 10200 "Kein NC-Start bei aktivem Alarm erlaubt"
- 10202 "Kein NC-Start bei aktivem Kommando erlaubt"
- 10203 "Kein NC-Start bei nicht referierten Achsen erlaubt"

#### Literatur:

Diagnosehandbuch Alarmer

### 8.9.3 Teileprogrammunterbrechung

#### Kanalzustand

Eine Teileprogrammunterbrechung wird nur ausgeführt, wenn Kanal und Programm aktiv sind:

- DB21, ... D35.5 == 1 ("Kanal aktiv")
- DB21, ... D35.0 == 1 ("Programm läuft")

#### STOP-Kommandos

Über folgende STOP-Kommandos kann die Teileprogrammbearbeitung unterbrochen werden:

- DB21, ... DBX7.2 ("NC-Stop an Satzgrenze")
- DB21, ... DBX7.3 ("NC-Stop")
- DB21, ... DBX7.4 ("NC-Stop Achsen plus Spindel")
- DB21, ... DBX2.0 ("Einzelsatz")
- Programmbefehl M00 oder M01

Anschließend befinden sich Kanal und Programm im Zustand "unterbrochen":

- DB21, ... D35.6 == 1 ("Kanal unterbrochen")
- DB21, ... D35.3 == 1 ("Programm unterbrochen")

#### Literatur

Eine ausführliche Beschreibung der Nahtstellensignale findet sich in:

Funktionshandbuch Grundfunktionen; NC/PLC-Nahtstellensignale (Z1)

## Ablauf

Nach einem STOP-Kommando werden folgende Aktionen durchgeführt:

- Unterbrechung der Teileprogrammabarbeitung an der nächsten Satzgrenze (bei "NC-Stop an Satzgrenze", M00, M01 bzw. Einzelsatz), bzw. sofort bei allen anderen STOP-Kommandos.
- Die verfahrenen Achsen des Kanals, werden über eine Bremsrampe stillgesetzt. Das Abbremsen der Achsen bis zum Stillstand kann sich über mehrere Sätze erstrecken.
- Der Satzzeiger zeigt den aktuellen Satz an der Unterbrechungsstelle an.
- Die bis zur Unterbrechungsstelle noch nicht ausgegebenen Hilfsfunktionen werden nicht mehr ausgegeben.

#### Möglichkeiten im unterbrochenen Zustand

Während einer Teileprogrammunterbrechung können diverse Funktionen im Kanal ausgeführt werden z.B.:

- Überspeicher  
**Literatur**  
Bedienhandbuch HMI-Advanced, Kapitel "Bedienbereich Maschine" > Betriebsart "Automatik" > "Überspeichern"
- Satzsuchlauf  
**Literatur**  
Funktionshandbuch Grundfunktionen, Kapitel "BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)" > "Satzsuchlauf" bzw. "Satzsuchlauf Typ 5 SERUPRO"
- Wiederanfahren an die Kontur (REPOS)  
**Literatur** Funktionshandbuch Grundfunktionen, Kapitel "BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)" > "Satzsuchlauf Typ 5 SERUPRO" > "REPOS" > "Wiederanfahren an Kontur mit gesteuertem REPOS"
- Orientierter Werkzeugrückzug  
**Literatur**
  - Programmierhandbuch Arbeitsvorbereitung, Kapitel "Werkzeugkorrekturen"
  - Funktionsbeschreibung Grundfunktionen, Kapitel "Werkzeugkorrekturen (W1)" > "Orientierbare Werkzeugträger" > ""
- Interruptroutine (siehe Kapitel "Asynchrone Unterprogramme (ASUPs), Interruptroutinen (Seite 616)")

- DRF-Funktion, Verschieben des Werkstücknullpunktes  
**Literatur**  
Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1)
- Starten des unterbrochenen Programms über:
  - Befehl `START` aus einem anderen Kanal  
**Literatur**  
Programmieranleitung Arbeitsvorbereitung, Kapitel "Flexible NC-Programmierung" > "Programmkoordinierung (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)"
  - NC/PLC-Nahtstelle:  
DB21, ... DBX7.1 (NC-Start)

## 8.9.4 RESET-Kommando

### Kommando-Priorität

#### Kanalzustand

Das RESET-Kommando kann in jedem Kanalzustand ausgeführt werden. Dieses Kommando wird von keinem anderen Kommando abgebrochen.

### Kommandos

#### RESET-Kommandos

Es stehen folgende Reset-Kommandos zur Verfügung:

- DB11, ... DBX0.7 ("BAG-Reset")
- DB21, ... DBX7.7 ("Reset")

Erläuterungen zu den einzelnen Nahtstellensignalen entnehmen Sie bitte

**Literatur:** /FB1/ Funktionshandbuch Grundfunktionen; NC/PLC-Nahtstellensignale (Z1),

Durch ein RESET-Kommando kann ein aktives Teileprogramm bzw. ein Teileprogrammsatz (in MDA) abgebrochen werden.

Nach Ausführung des Reset-Kommandos wird das NST DB21, ... DBX35.7 ("Kanalzustand Reset") gesetzt.

Das Teileprogramm kann an der Unterbrechungsstelle nicht mehr fortgesetzt werden. Alle Achsen im Kanal befinden sich im Genauhalt mit Ausnahme bei Nachführbetrieb. Das gleiche gilt auch für projektierte Spindeln im Kanal.



Es werden folgende Aktionen nach Auslösung des RESET-Kommandos durchgeführt:

- Die Teileprogrammaufbereitung wird sofort gestoppt.
- Achsen und falls vorhanden Spindeln des Kanals werden über eine eingestellte Bremsrampe abgebremst.
- Die zu diesem Zeitpunkt noch nicht ausgegebenen Hilfsfunktionen des aktuellen Satzes werden nicht mehr ausgegeben.
- Der Satzzeiger wird auf den Anfang des Teileprogrammes zurückgesetzt.
- Alle Reset-Alarme (kanal-, achs- und spindelspezifisch) werden aus der Anzeige gelöscht.

### 8.9.5 Programmzustand

Für jeden Kanal wird der Zustand des angewählten Programms in der Nahtstelle angezeigt. Aufgrund des Zustandes kann dann die PLC vom Hersteller projektierbare Reaktionen oder Verriegelungen auslösen.

Der Programmzustand wird nur in den Betriebsarten AUTOMATIK und MDA angezeigt. In allen anderen Betriebsarten ist der Programmzustand abgebrochen oder unterbrochen.

#### Programmzustände

An der NC/PLC-Nahtstelle (DB21, ...) werden folgende Programmzustände angezeigt:

- DB21, ... DBX35.4 ("abgebrochen")
- DB21, ... DBX35.3 ("unterbrochen")
- DB21, ... DBX35.2 ("angehalten")
- DB21, ... DBX35.1 ("warten")
- DB21, ... DBX35.0 ("läuft")

Eine ausführliche Beschreibung der Nahtstellensignale findet sich in Kapitel "Signale von Kanal (DB21, ...) (Seite 1867)".

**Auswirkungen von Kommandos und NC/PLC-Nahtstellensignalen**

Der Programmzustand eines aktiven Programms wird durch verschiedene Kommandos und NC/PLC-Nahtstellensignale beeinflusst. Die folgende Tabelle zeigt den jeweiligen Folgezustand.

Ausgangszustand: "läuft"					
Kommando oder NC/PLC-Nahtstellensignal (NST)	Folgezustand				
	abgebrochen	unterbrochen	angehalten	warten	läuft
NST "Reset"	x				
NST "NC-Stop"			x		
NST "NC-Stop an Satzgrenze"			x		
NST "NC-Stop Achsen u. Spindeln"			x		
NST "Einlesesperre"					x
NST "Vorschub Halt, Kanalsp."					x
NST "Vorschub Halt, Achssp."					x
Vorschuboverride = 0%					x
NST "Spindel Halt"					x
M02/M30 im Satz	x				
M00/M01 im Satz			x		
NST "Einzelsatz"			x		
NST "Restweg löschen"					x
Hilfsfunktion an PLC ausgegeben, aber noch nicht quittiert			x		
Wait-Anweisung im Programm				x	

**8.9.6 Kanalzustand**

Für jeden Kanal wird in allen Betriebsarten der aktuelle Kanalzustand an der NC/PLC-Nahtstelle (DB21, ...) angezeigt..

**Kanalzustände**

An der NC/PLC-Nahtstelle (DB21, ...) werden folgende Kanalzustände angezeigt:

- DB21, ... DBX35.7 ("Reset")
- DB21, ... DBX35.6 ("unterbrochen")
- DB21, ... DBX35.5 ("aktiv")

Eine ausführliche Beschreibung der Nahtstellensignale findet sich in Kapitel "Signale von Kanal (DB21, ...) (Seite 1867)".

**Auswirkungen von Kommandos und NC/PLC-Nahtstellensignalen**

Der Kanalzustand wird durch verschiedene Kommandos und NC/PLC-Nahtstellensignale beeinflusst. Die folgende Tabelle zeigt den jeweiligen Folgezustand.

Ausgangszustand: "aktiv"			
Kommando oder NC/PLC-Nahtstellensignal (NST)	Folgezustand		
	Reset	unterbrochen	Aktiv
NST "Reset"	x		
NST "NC-Stop"		x	
NST "NC-Stop an Satzgrenze"		x	
NST "NC-Stop Achsen u. Spindeln"		x	
NST "Einlesesperre"			x
NST "Vorschub Halt, Kanalsp."			x
NST "Vorschub Halt, Achssp."			x
Vorschuboverride = 0%			
NST "Spindel Halt"			x
M02/M30 im Satz	x		
M00/M01 im Satz		x	
NST "Einzelsatz"		x	
NST "Restweg löschen"			x
Hilfsfunktion an PLC ausgegeben, aber noch nicht quittiert			x
Wait-Anweisung im Programm			x

Der "Kanalzustand aktiv" wird erreicht, wenn ein Teileprogramm oder Teileprogrammsatz abgearbeitet wird oder wenn in der Betriebsart JOG Achsen verfahren werden.

## 8.9.7 Reaktionen auf Bedienungs- oder Programmaktionen

### Zustandsübergänge

Die folgende Tabelle zeigt die Kanal- und Programmzustände auf, die nach bestimmten Bediener- oder Programmaktionen auftreten.

Im linken Teil der Tabelle sind die Kanal-, Programmzustände sowie die Betriebsarten aufgeführt, unter denen man die Ausgangssituation aussuchen muss. Im rechten Tabellenteil sind bestimmte Bedien-/Programmaktionen aufgeführt; in Klammern steht bei jeder Aktion die Nummer der Situation nach der Aktionsausführung.

Tabelle 8- 2 Reaktionen auf Bedienungs- oder Programmaktionen

Situation	Kanalzustand			Programmzustand					aktive BA			Bediener- oder Programmaktion (Situation nach der Aktion)
	R	U	A	N	U	S	W	A	A	M	J	
1		x						x	x			RESET (4)
2		x						x		x		RESET (5)
3		x						x			x	RESET (6)
4	x			x					x			NC-Start (13); BA-Wechsel (5 bzw. 6)
5	x			x						x		NC-Start (14); BA-Wechsel (4 bzw. 6)
6	x			x							x	Richtungstaste (15); BA-Wechsel (4 bzw. 5)
7		x		x						x		NC-Start (14)
8		x		x							x	NC-Start (15)
9		x			x				x			NC-Start (13); BA-Wechsel (10 bzw. 11)
10		x			x					x		NC-Start (16); BA-Wechsel (9 bzw. 11)
11		x			x						x	Richtungstaste (17); BA-Wechsel (9 bzw. 10)
12		x				x			x			NC-Start (13); BA-Wechsel (10 bzw. 11)
13			x					x	x			NC-Stop (12)
14			x	x						x		NC-Stop (7); bei Satzende (5)
15			x	x							x	NC-Stop (8); bei JOG-Ende (6)
16			x		x					x		NC-Stop (10); bei Satzende (10)
17			x		x						x	NC-Stop (11); bei JOG-Ende (11)
18			x				x		x			RESET (4); warten auf anderen Kanal (18)

Kanalzustand

R --&gt; abgebrochen

U --&gt; unterbrochen

A --&gt; läuft

Programmzustand

N --&gt; abgebrochen

U --&gt; unterbrochen

S --&gt; angehalten

W --&gt; wartet

A --&gt; läuft

Betriebsarten

A --&gt; abgebrochen

M --&gt; abgebrochen

J --&gt; abgebrochen

## 8.9.8 Teileprogramm starten

### Start-Handhabung

Tabelle 8- 3 Typischer Programmablauf

Reihenfolge	Kommando	Randbedingungen (müssen vor dem Kommando erfüllt sein)	Bemerkungen
1	Programm laden (über die Bedienoberfläche oder über Teileprogramm)		
2	Anwahl der Betriebsart AUTOMATIK		
3	Programmvorwahl	Kanal vorgewählt vorgewählter Kanal im RESET-Zustand Benutzerkennung für Programmvorwahl ausreichend	
4	NC-Start für vorgewählten Kanal	NC-Startsperre nicht vorhanden Referenzpunkt in allen Achsen angefahren	
5			Programmabarbeitung
6	M02/M30/RESET	keine	Programmende

### 8.9.9 Zeitdiagramm-Beispiel für einen Programmablauf

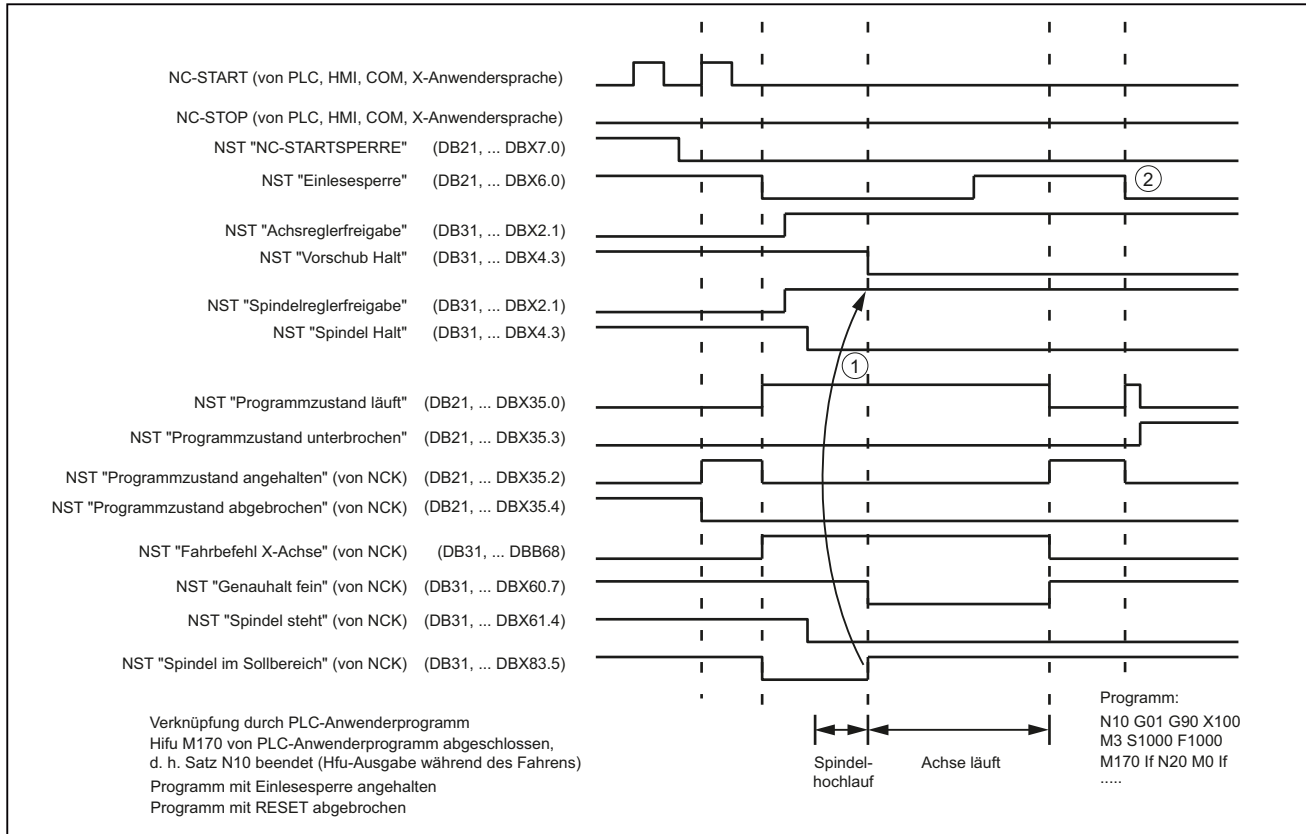


Bild 8-8 Beispiel für die Signale während des Programmablaufs

## 8.9.10 Programmsprünge

### 8.9.10.1 Rücksprung auf Programmanfang

#### Funktion

Mit der Funktion "Rücksprung auf den Programmanfang" kann aus einem Teileprogramm heraus an den Anfang des Programms zurückgesprungen werden. Das Programm wird daraufhin erneut abgearbeitet.

Im Vergleich zur Funktion "Programmsprünge auf Sprungmarken", mit der ebenfalls ein wiederholtes Abarbeiten eines Teileprogramms realisiert werden kann, bietet die Funktion "Rücksprung auf den Programmanfang" folgende Vorteile:

- Die Programmierung einer Sprungmarke am Programmanfang ist nicht notwendig.
- Der Programmneustart ist steuerbar durch das NC/PLC-Nahtstellensignal:  
DB21, ... DBX384.0 (Programmverzweigung steuern)
- Der Timer für die Programmlaufzeit kann beim Programmneustart auf "0" zurückgesetzt werden.
- Die Timer für die Werkstückzählung können beim Programmneustart um den Wert "1" erhöht werden.

#### Anwendung

Die Funktion findet Anwendung, wenn die Bearbeitung nachfolgender Werkstücke durch einen automatischen Programmneustart erfolgen soll, z. B. bei Drehmaschinen mit Stangenlader/-wechsler.

#### Wirksamkeit

Der Rücksprung erfolgt nur, wenn das folgende NC/PLC-Nahtstellensignal gesetzt ist:

DB21, ... DBX384.0 (Programmverzweigung steuern) = 1

Wenn das Signal auf "0" steht, dann wird kein Rücksprung ausgeführt und die Programmbearbeitung wird mit dem nächsten Teileprogrammsatz nach dem Funktionsaufruf fortgesetzt.

#### Parametrierung

##### Programmlaufzeit

Die Laufzeit des angewählten NC-Programms wird in der Systemvariablen \$AC\_CYCLE\_TIME gespeichert. Beim Start eines neuen Programms wird die Systemvariable automatisch auf "0" zurückgesetzt (siehe Kapitel " Programmlaufzeit (Seite 697) ")

Über das folgende Maschinendatum kann eingestellt werden, dass die Systemvariable \$AC\_CYCLE\_TIME auch bei einem Programmneustart durch die Funktion "Rücksprung auf den Programmanfang" auf "0" zurückgesetzt wird:

MD27860 \$MC\_PROCESSTIMER\_MODE (Aktivierung der Programmlaufzeit-Messung)

Bit	Wert	Bedeutung
8	0	\$AC_CYCLE_TIME wird durch die Funktion "Rücksprung auf den Programmanfang" <b>nicht</b> auf "0" zurückgesetzt.
	1	\$AC_CYCLE_TIME wird durch die Funktion "Rücksprung auf den Programmanfang" auf "0" zurückgesetzt.

**Hinweis**

Damit die Einstellung von Bit 8 wirksam werden kann, muss die Messung der aktuellen Programmlaufzeit aktiv sein (MD27860 Bit 1 = 1).

**Werkstückzählung**

Nachdem das Teileprogrammende (M02 / M30) erreicht ist, werden die aktivierten Werkstückzähler (\$AC\_TOTAL\_PARTS / \$AC\_ACTUAL\_PARTS / \$AC\_SPECIAL\_PARTS) um den Wert "1" erhöht (siehe Kapitel "Werkstückzähler (Seite 705)").

Über das folgende Maschinendatum kann eingestellt werden, dass die aktivierten Werkstückzähler auch bei einem Programmneustart durch die Funktion "Rücksprung auf den Programmanfang" hochgezählt werden:

MD27880 \$MC\_PART\_COUNTER (Aktivierung der Werkstück-Zähler)

Bit	Wert	Bedeutung
		Bei einem Programmneustart durch die Funktion "Rücksprung auf den Programmanfang" wird der Werkstückzähler:
7	0	\$AC_TOTAL_PARTS <b>nicht</b> hochgezählt.
	1	\$AC_TOTAL_PARTS hochgezählt.
11	0	\$AC_ACTUAL_PARTS <b>nicht</b> hochgezählt.
	1	\$AC_ACTUAL_PARTS hochgezählt.
15	0	\$AC_SPECIAL_PARTS <b>nicht</b> hochgezählt.
	1	\$AC_SPECIAL_PARTS hochgezählt.



## Programmierung

Die Funktion wird im Haupt- oder Unterprogramm über den Befehl `GOTOS` aufgerufen.

Syntax:            `GOTOS`

Parameter:        keine

Verwendung in Synchronaktionen:    nicht möglich

---

### Hinweis

`GOTOS` löst intern ein `STOPRE` (Vorlaufstopp) aus.

---

## Beispiel

Programmierung	Kommentar
N10 ...	; Programmanfang
...	
N90 GOTOS	; Sprung an den Programmanfang
...	

## 8.9.11 Programmteilwiederholungen

### 8.9.11.1 Übersicht

#### Funktion

Die Programmteilwiederholung ermöglicht die Wiederholung eines beliebigen durch Labels gekennzeichneten Bereich eines Teileprogramms.

Zu Labels siehe:

#### Literatur:

Programmierhandbuch Grundlagen; Programmsprünge und Programmwiederholungen

#### Definitionsmöglichkeiten von Teileprogrammgebieten

Die Programmteilwiederholung bietet verschiedene Möglichkeiten einen Teileprogrammgebiet der wiederholt werden sollen, zu definieren:

- Einen einzelnen Teileprogrammsatz
- Einen Teileprogrammgebiet ab einem Start-Label
- Einen Teileprogrammgebiet zwischen einem Start-Label und End-Label
- Einen Teileprogrammgebiet zwischen einem Start-Label und dem Schlüsselwort: ENDLABEL

#### Literatur:

Programmierhandbuch Arbeitsvorbereitung; Kapitel "Programmkoordinierung"

### 8.9.11.2 Einzelner Teileprogrammsatz

#### Funktionalität

Durch `REPEATB` (B=Block) in Teileprogrammsatz N150, verzweigt die Teileprogrammbearbeitung zum mit dem Label `START_1` gekennzeichnete Teileprogrammsatz N120. Dieser wird `n` mal wiederholt. Ist kein `P` angegeben, wird er genau einmal wiederholt. Nach der letzten Wiederholung wird das Teileprogramm mit dem auf die `REPEATB`-Anweisung folgenden Teileprogrammsatz N160 fortgesetzt.

```

:
N100 ...
N120 START_1: ... ; Label: START_1
N130 ...
N140 ...
N150 REPEATB START_1 P=n ; Wiederholung ab: START_1
N160 ...
:

```

---

#### Hinweis

##### Label-Suchrichtung

Der mit dem Label gekennzeichnete Teileprogrammsatz kann vor oder nach der `REPEATB`-Anweisung stehen. Gesucht wird zunächst in Richtung Programmanfang. Wird das Label nicht gefunden, wird in Richtung Programmende gesucht.

---

#### Programmierung

Syntax: `REPEATB <Label> [P=n]`

<code>Label</code>	Start-Label zu dem die Anweisung: <code>REPEAT</code> verzweigt Typ: String
<code>P</code>	Wiederholungszahl
<code>n</code>	Anzahl der Wiederholungen Typ: Integer

### 8.9.11.3 Teileprogrammabereich ab einem Start-Label

#### Funktionalität

Durch REPEAT in Teileprogrammsatz N150, verzweigt die Teileprogrammabearbeitung zum mit dem Label START\_1 gekennzeichneten Teileprogrammsatz N120. Dieser und alle bis zum Teileprogrammsatz der die REPEAT-Anweisung enthält (N150) folgenden Teileprogrammsätze (N130 und N140), werden n mal wiederholt. Ist kein P angegeben, wird der Teileprogrammabereich (N120 - N140) genau einmal wiederholt. Nach der letzten Wiederholung wird das Teileprogramm mit dem auf die REPEAT-Anweisung folgenden Teileprogrammsatz N160 fortgesetzt.

```
:  
N100 ...  
N120 START_1: ... ; Start-Label: START_1  
N130 ...  
N140 ...  
N150 REPEAT START_1 P=n ; Wiederholung ab: START_1  
N160 ...  
:
```

---

#### Hinweis

##### Label-Suchrichtung

Der mit dem Start-Label gekennzeichnete Teileprogrammsatz **muss** vor der REPEAT-Anweisung stehen.

---

#### Programmierung

Syntax: REPEAT <Label> [P=n]  
Label Start-Label zu dem die Anweisung: REPEAT verzweigt  
Typ: String  
P Wiederholungszahl  
n Anzahl der Wiederholungen  
Typ: Integer

### 8.9.11.4 Teileprogrammbereich zwischen einem Start-Label und End-Label

#### Funktionalität

Durch `REPEAT` in Teileprogrammsatz N160, verzweigt die Teileprogrammbearbeitung zum mit dem Start-Label `START_1` gekennzeichneten Teileprogrammsatz N120. Dieser und alle bis einschließlich des mit dem Ende-Label `END_1` gekennzeichneten Teileprogrammsatzes (N140) werden `n` mal wiederholt. Ist kein `P` angegeben, wird der Teileprogrammbereich (N120 - N140) genau einmal wiederholt. Nach der letzten Wiederholung wird das Teileprogramm mit dem auf die `REPEAT`-Anweisung folgenden Teileprogrammsatz N170 fortgesetzt.

```

:
N100 ...
N120 START_1: ... ; Start-Label: START_1
N130 ...
N140 END_1 ... ; Ende-Label: END_1
N150 ...
N160 REPEAT START_1 END_1 P=n ; Wiederholung: START_1 bis END_1
N170 ...
:

```

#### Hinweis

##### Label-Suchrichtung

Der mit Start- und Ende-Label gekennzeichnete Programmbereich kann vor oder nach der `REPEAT`-Anweisung stehen. Gesucht wird zunächst in Richtung Programm-anfang. Wird das Start-Label nicht gefunden, wird in Richtung Programm-ende gesucht.

Befindet sich die `REPEAT`-Anweisung zwischen dem Start- und Ende-Label, wird nur der Teileprogrammbereich von Start-Label bis `REPEAT`-Anweisung wiederholt.

`REPEAT`-Anweisung wiederholt.

#### Programmierung

<b>Syntax:</b>	<code>REPEAT &lt;Start_Label&gt; &lt;Ende_Label&gt; [P=n]</code>
<code>Start_Label</code>	Label zu dem die Anweisung: <code>REPEAT</code> verzweigt . Anfang des Teileprogrammbereichs der wiederholt wird. Typ: String
<code>Ende_Label</code>	Endes des Teileprogrammbereichs der wiederholt wird. Typ: String
<code>P</code>	Wiederholungszahl
<code>n</code>	Anzahl der Wiederholungen Typ: Integer

### 8.9.11.5 Teileprogrammabereich zwischen einem Start-Label und dem Schlüsselwort: ENDLABEL

#### Funktionalität

Durch REPEAT in Teileprogrammsatz N150, verzweigt die Teileprogrammabearbeitung zum mit dem Start-Label START\_1 gekennzeichneten Teileprogrammsatz N120. Dieser und alle bis einschließlich des mit dem Schlüsselwort ENDLABEL gekennzeichneten Teileprogrammsatzes (N140) werden n mal wiederholt. Ist kein P angegeben, wird der Teileprogrammabereich (N120 - N140) genau einmal wiederholt. Nach der letzten Wiederholung wird das Teileprogramm mit dem auf die REPEAT-Anweisung folgenden Teileprogrammsatz N170 fortgesetzt.

```
:  
N100 ...  
N120 START_1: ... ; Start-Label: START__1  
N130 ...  
N140 ENDLABEL: ... ; Ende-Label: Schlüsselwort ENDLABEL  
N150 ...  
N160 REPEAT START_1 END_1 P=n ; Wiederholung: START_1 bis END_1  
N170 ...  
:
```

---

#### Hinweis

##### Label-Suchrichtung

Der mit Start- und Ende-Label gekennzeichnete Programmabereich kann vor oder nach der REPEAT-Anweisung stehen. Gesucht wird zunächst in Richtung Programmabeginn. Wird das Start-Label nicht gefunden, wird in Richtung Programmende gesucht.

Wird zwischen dem Start-Label und der REPEAT-Anweisung kein Schlüsselwort ENDLABEL gefunden, wird der Teileprogrammabereich von Start-Label bis zur REPEAT-Anweisung wiederholt.

---

#### Programmierung

Syntax:	REPEAT <Label> [P=n]
	Label
	Label zu dem die Anweisung: REPEAT verzweigt . Anfang des Teileprogrammabereichs der wiederholt wird. Typ: String
	P
	Wiederholungszahl
	n
	Anzahl der Wiederholungen Typ: Integer

## 8.9.12 Ereignisgesteuerte Programmaufrufe

### 8.9.12.1 Funktion

#### Wozu dient die Funktion?

Die Funktion "Ereignisgesteuerte Programmaufrufe" bietet die Möglichkeit, bei bestimmten Ereignissen implizit ein Anwenderprogramm ablaufen zu lassen, z. B. um Grundeinstellungen von Funktionen oder Initialisierungen vorzunehmen.

#### Ereignisse

Auslösende Ereignisse können sein:

- Teileprogramm-Start
- Teileprogramm-Ende
- Bedientafel-Reset
- Hochlauf der NC-Steuerung

Die Auswahl der auslösenden Ereignisse erfolgt mit dem Maschinendatum MD20108 \$MC\_PROG\_EVENT\_MASK (siehe Kapitel "Parametrierung (Seite 603)").

#### Anwenderprogramm

In der Grundstellung wird nach Eintreten eines auslösenden Ereignisses das Programm `_N_PROG_EVENT_SPF` aktiviert. Soll ein anderes Anwenderprogramm aktiviert werden, muss es im Maschinendatum MD11620 \$MN\_PROG\_EVENT\_NAME eingetragen sein (siehe Kapitel "Parametrierung (Seite 603)").

Das durch das Ereignis aktivierte Anwenderprogramm wird grundsätzlich in dem Kanal abgearbeitet, in dem das jeweilige Ereignis aufgetreten ist.

Das Anwenderprogramm wird mit niedrigster Priorität ausgeführt und kann daher von einem Anwender-ASUP unterbrochen werden.

## Bearbeitungsablauf

### Ablauf bei Aktivierung durch Teileprogramm-Start

Ausgangszustand:

Kanal: im Reset-Zustand  
Betriebsart: AUTO  
AUTO + Überspeichern oder MDA  
TEACHIN

1. NC-Start
2. Initialisierungssequenz mit Auswertung von:  
MD20112 \$MC\_START\_MODE\_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START)
3. Impliziter Aufruf von \_N\_PROG\_EVENT\_SPF als Unterprogramm
4. Bearbeitung des Datenteils des Hauptprogramms
5. Bearbeitung des Programmteils des Hauptprogramms

### Ablauf bei Aktivierung durch Teileprogramm-Ende

Ausgangszustand:

Kanal: im Aktiv-Zustand  
Betriebsart: AUTO  
AUTO + Überspeichern oder MDA  
TEACHIN

1. Satz mit Teileprogramm-Ende wird eingewechselt.
2. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:  
MD \$MC\_RESET\_MODE\_MASK  
\$MC\_GCODE\_RESET\_VALUES  
\$MC\_GCODE\_RESET\_MODE
3. Impliziter Aufruf von \_N\_PROG\_EVENT\_SPF als ASUP
4. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:  
\$MC\_RESET\_MODE\_MASK  
\$MC\_GCODE\_RESET\_VALUES  
\$MC\_GCODE\_RESET\_MODE  
⇒ Die G-Code-Reset-Stellung wird weiterhin über Maschinendaten eingestellt!



### **Ablauf bei Aktivierung durch Bedientafel-Reset**

Ausgangszustand:

Kanal: beliebig

Betriebsart: beliebig

1. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:

MD \$MC\_RESET\_MODE\_MASK

\$MC\_GCODE\_RESET\_VALUES

\$MC\_GCODE\_RESET\_MODE

2. Impliziter Aufruf von `_N_PROG_EVENT_SPF` als ASUP

3. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:

\$MC\_RESET\_MODE\_MASK

\$MC\_GCODE\_RESET\_VALUES

\$MC\_GCODE\_RESET\_MODE

⇒ Die G-Code-Reset-Stellung wird weiterhin über Maschinendaten eingestellt!

### **Ablauf bei Aktivierung durch Hochlauf**

1. Steuerung aktiviert nach Hochlauf Reset-Sequenz mit Auswertung der Maschinendaten:

MD \$MC\_RESET\_MODE\_MASK

\$MC\_GCODE\_RESET\_VALUES

\$MC\_GCODE\_RESET\_MODE

2. Impliziter Aufruf von `_N_PROG_EVENT_SPF` als ASUP

3. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:

\$MC\_RESET\_MODE\_MASK

\$MC\_GCODE\_RESET\_VALUES

\$MC\_GCODE\_RESET\_MODE

⇒ Die G-Code-Reset-Stellung wird weiterhin über Maschinendaten eingestellt!

Signalverlauf

Die folgenden Diagramme zeigen die Signalverläufe der NC/PLC-Nahtstellensignale DB21, ... DBB35 ("Programmzustand" und "Kanalzustand") beim ereignisgesteuerten Programmaufruf:

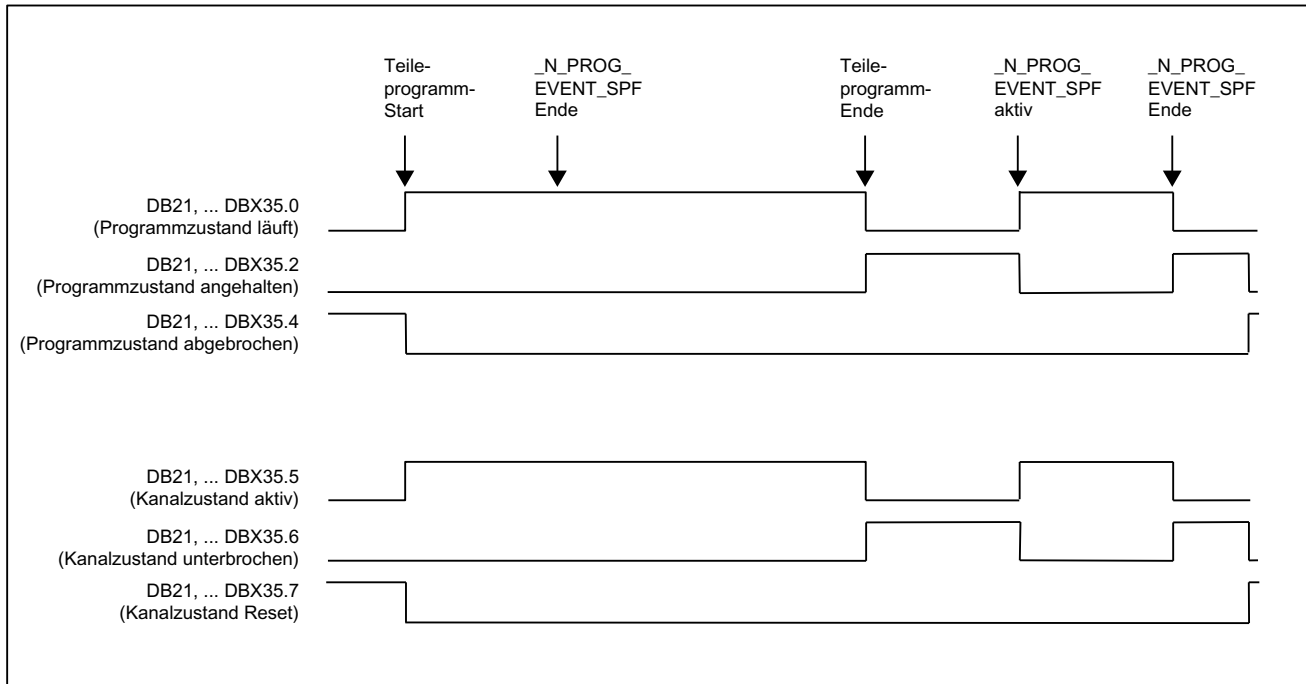


Bild 8-9 Signalverlauf bei Aktivierung durch Teileprogramm-Start und Teileprogramm-Ende

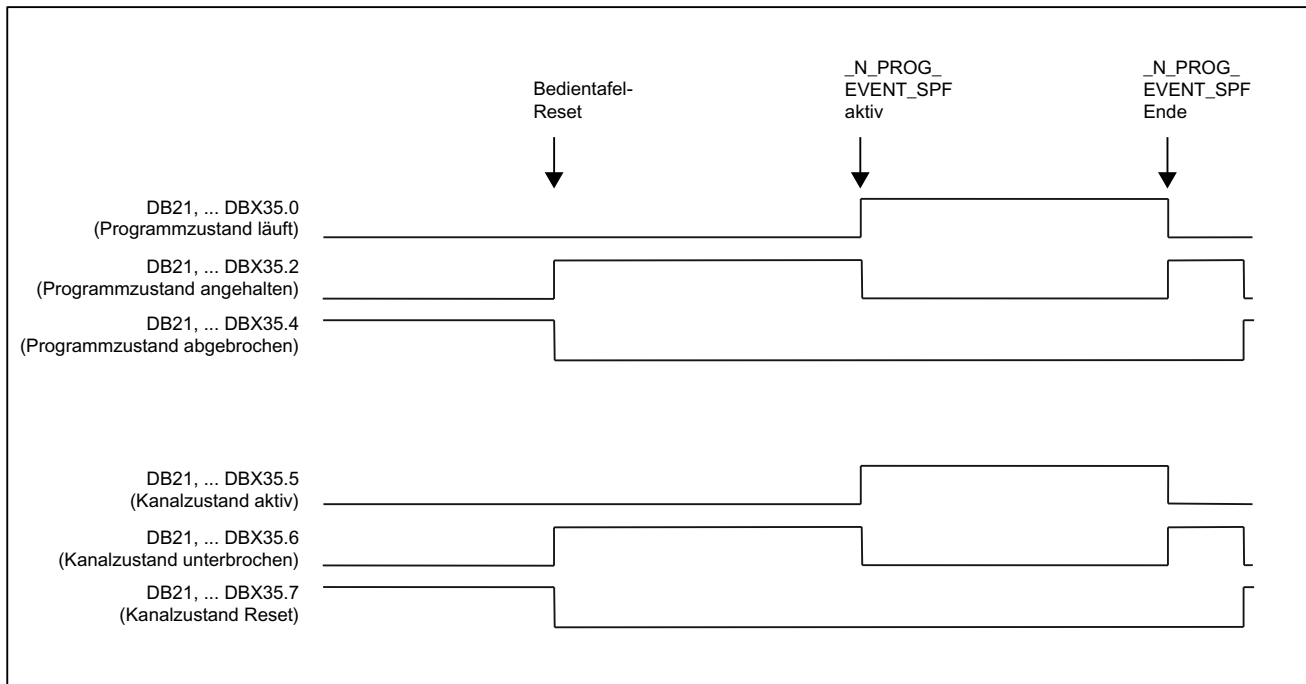


Bild 8-10 Signalverlauf bei Aktivierung durch Bedientafel-Reset

**Hinweis**

DB21, ... DBX35.4 (Programmzustand abgebrochen) und DB21, ... DBX35.7 (Kanalzustand Reset) werden erst dann eingenommen, wenn das ereignisgesteuerte Anwenderprogramm beendet ist. Zwischen Programmende und dem Start des ereignisgesteuerten Anwenderprogramms werden diese Zustände nicht eingenommen. Gleiches gilt zwischen Bedientafel-Reset und dem Start des Anwenderprogramms.

**Anzeige**

Die Information über das auslösende Ereignis wird über das NC/PLC-Nahtstellenbyte DB21, ... DBB376 der PLC zur Verfügung gestellt:

Bit	Wert	Bedeutung
0	1	Teileprogramm-Start aus dem Kanalzustand Reset
1	1	Teileprogramm-Ende
2	1	Bedientafel-Reset
3	1	Hochlauf
4	1	1. Start nach Satzsuchlauf (siehe "Automatischer Start eines ASUP nach Satzsuchlauf (Seite 528)")

Die Pauschalabfrage von DB21, ... DBB376 auf 0 ermöglicht die Feststellung, ob überhaupt ein ereignisgesteuertes Anwenderprogramm aktiv ist.

Wenn das ereignisgesteuerte Anwenderprogramm abgelaufen ist oder mit RESET abgebrochen wird, dann wird das zugehörige Anzeigebit in der Nahtstelle gelöscht. Bei einem sehr kurzen Programm bleibt das entsprechende Bit wenigstens für die Dauer eines kompletten PLC-Zyklus erhalten.

**8.9.12.2 Parametrierung****Auslösende Ereignisse**

Welche Ereignisse das Anwenderprogramm aktivieren sollen, wird kanalspezifisch eingestellt im Maschinendatum:

MD20108 \$MC\_PROG\_EVENT\_MASK (Ereignisgesteuerter Programmaufruf)

Bit	Wert	Bedeutung
0	1	Aktivierung durch <b>Teileprogramm-Start</b>
1	1	Aktivierung durch <b>Teileprogramm-Ende</b>
2	1	Aktivierung durch <b>Bedientafel-Reset</b>
3	1	Aktivierung durch <b>Hochlauf</b> der NC-Steuerung

Voraussetzung für die Aktivierung:

Das Anwenderprogramm (Voreinstellung: `_N_PROG_EVENT_SPF`) muss geladen und freigegeben sein.

---

**Hinweis**

MD20108 `$MC_PROG_EVENT_MASK` wird in der Simulation ignoriert.

---

## Anwenderprogramm

In der Grundstellung wird nach Eintreten eines mit MD20108 gesetzten Ereignisses das Programm `_N_PROG_EVENT_SPF` aus dem Verzeichnis `_N_CMA_DIR` aktiviert.

Soll ein anderes Anwenderprogramm aktiviert werden, dann muss im folgenden Maschinendatum der Name dieses Anwenderprogramms eingetragen sein:

MD11620 `$MN_PROG_EVENT_NAME`

Das angegebene Programm muss sich in einem der Zyklen-Verzeichnisse befinden.

Beim Eintreten eines mit MD20108 gesetzten Ereignisses wird folgender Suchpfad durchlaufen:

1. `/_N_CUS_DIR/` für Anwender-Zyklen
2. `/_N_CMA_DIR/` für Hersteller-Zyklen
3. `/_N_CST_DIR/` für Standard-Zyklen

Das zuerst gefundene Programm mit dem angegebenen Namen wird aufgerufen.

---

**Hinweis**

Der angegebene Name wird syntaktisch wie bei einem Unterprogrammbezeichner geprüft, d. h. die ersten beiden Zeichen müssen Buchstaben oder Unterstriche sein (keine Ziffern). Präfix (`_N_`) und Suffix (`_SPF`) des Programmnamens werden, wenn nicht angegeben, automatisch ergänzt.

---

---

**Hinweis**

Es greifen die Schutzmechanismen, wie sie allgemein für Zyklen aktivierbar sind (Schutzstufen für Schreiben, Lesen usw.).

---

### Verhalten beim Starten eines Anwender-ASUP

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" beim Starten eines Anwender-ASUP aus dem Kanalzustand Reset kann kanalspezifisch eingestellt werden mit dem Maschinendatum:

MD20109 \$MC\_PROG\_EVENT\_MASK\_PROPERTIES

Bit	Wert	Bedeutung
0	0	Das Eintreten eines mit MD20108 gesetzten Ereignisses (Teileprogramm-Start, Teileprogramm-Ende u./o. Bedientafel-Reset ) hat die Aktivierung des ereignisgesteuerten Anwenderprogramms zur Folge.
	1	Das Eintreten eines mit MD20108 gesetzten Ereignisses führt <b>nicht</b> zu einer Aktivierung des ereignisgesteuerten Anwenderprogramms.

### Verhalten bei gesetzter Einzelsatzbearbeitung

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" bei gesetzter Einzelsatzbearbeitung kann für jedes auslösende Ereignis kanalspezifisch eingestellt werden mit dem Maschinendatum:

MD20106 \$MC\_PROG\_EVENT\_IGN\_SINGLEBLOCK

Bit	Wert	Bedeutung
		Im ereignisgesteuerten Anwenderprogramm ist:
0		• Nach einer Aktivierung durch Teileprogramm-Start:
	0	Die Einzelsatzbearbeitung wirksam
	1	Die Einzelsatzbearbeitung unterdrückt
1		• Nach einer Aktivierung durch Teileprogramm-Ende:
	0	Die Einzelsatzbearbeitung wirksam
	1	Die Einzelsatzbearbeitung unterdrückt
2		• Nach einer Aktivierung durch Bedientafel-Reset:
	0	Die Einzelsatzbearbeitung wirksam
	1	Die Einzelsatzbearbeitung unterdrückt
3		• Nach einer Aktivierung durch Hochlauf:
	0	Die Einzelsatzbearbeitung wirksam
	1	Die Einzelsatzbearbeitung unterdrückt

Wenn die Einzelsatzbearbeitung unterdrückt ist, dann wird das ereignisgesteuerte Anwenderprogramm ohne Unterbrechung abgearbeitet.

**Hinweis**

MD20106 \$MC\_PROG\_EVENT\_IGN\_SINGLEBLOCK wirkt für alle Einzelsatzbearbeitungstypen.

**Hinweis**

Die Einzelsatzbearbeitung im ereignisgesteuerten Anwenderprogramm kann durch folgende Projektierung grundsätzlich ausgeschaltet sein:

MD10702 \$MN\_IGNORE\_SINGLEBLOCK\_MASK (Einzelsatzstopp verhindern) Bit 0 = 1

Die differenzierten Einstellungen im MD20106 \$MC\_PROG\_EVENT\_IGN\_SINGLEBLOCK sind dann unwirksam.

**Verhalten bei aktiver Einleesesperre**

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" bei gesetzter Einleesesperre (DB21, ... DBX6.1 = 1) kann für jedes auslösende Ereignis kanalspezifisch eingestellt werden mit dem Maschinendatum:

MD20107 \$MC\_PROG\_EVENT\_IGN\_INHIBIT

Bit	Wert	Bedeutung
		Im ereignisgesteuerten Anwenderprogramm ist:
0		• Nach einer Aktivierung durch Teileprogramm-Start:
	0	Die Einleesesperre wirksam
	1	Die Einleesesperre unterdrückt
1		• Nach einer Aktivierung durch Teileprogramm-Ende:
	0	Die Einleesesperre wirksam
	1	Die Einleesesperre unterdrückt
2		• Nach einer Aktivierung durch Bedientafel-Reset:
	0	Die Einleesesperre wirksam
	1	Die Einleesesperre unterdrückt
3		• Nach einer Aktivierung durch Hochlauf:
	0	Die Einleesesperre wirksam
	1	Die Einleesesperre unterdrückt

**Hinweis**

Für Bit 0 = 1 (Anwenderprogramm wird nach Teileprogramm-Start aktiviert) gilt folgende Einschränkung:

Wird das Anwenderprogramm mit dem Teileprogrammbehehl `RET` beendet, so führt `RET` immer zu einem ausführbaren Satz (analog zu `M17`).

Bei Bit 0 = 0 wird `RET` dagegen im Interpreter interpretiert und führt nicht zu einem ausführbaren Satz.

**Aktualisierung der Anzeige des Programm- und Kanalzustandes unterdrücken**

Um ein Flackern der Anzeige des Programm- und Kanalzustandes in der Bedienoberfläche zu vermeiden, kann für die Ausführung des i. d. R. sehr kurzen ereignisgesteuerten Anwenderprogramms die Aktualisierung der Anzeige unterdrückt werden. In der Anzeige bleibt dann der Programm- und Kanalzustand vor Aktivierung des ereignisgesteuerten Anwenderprogramms sichtbar (z. B. "Programmzustand abgebrochen" und "Kanalzustand Reset").

Die Parametrierung dieser Funktion erfolgt mit dem kanalspezifischen Maschinendatum:

MD20192 \$MC\_PROG\_EVENT\_IGN\_PROG\_STATE

Bit	Wert	Bedeutung
		Bei der Ausführung eines ereignisgesteuerten Anwenderprogramms wird die Aktualisierung der Anzeige des Programm- und Kanalzustandes:
1		• Bei einer Aktivierung durch Teileprogramm-Ende:
	0	Nicht unterdrückt
	1	Unterdrückt
2		• Bei einer Aktivierung durch Bedientafel-Reset:
	0	Nicht unterdrückt
	1	Unterdrückt
3		• Bei einer Aktivierung durch Hochlauf:
	0	Nicht unterdrückt
	1	Unterdrückt

**Hinweis**

Die Systemvariablen `$AC_STAT` und `$AC_PROG` bleiben von dieser Funktion unbeeinflusst, d. h. im laufenden ereignisgesteuerten Anwenderprogramm werden `$AC_STAT` auf "aktiv" und `$AC_PROG` auf "läuft" gesetzt sein.

Die NC/PLC-Nahtstellensignale DB21, ... DBX35.0-7 ("Programmzustand ..." und "Kanalzustand ...") bleiben ebenfalls unbeeinflusst.

**Verhalten bei NC-Stop**

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" bei NC-Stop (d. h. NC/PLC-Nahtstellensignal DB21, ... DBX7.2, 7.3 oder 7.4 ist gesetzt) kann für die auslösenden Ereignisse Teileprogramm-Ende, Bedientafel-Reset und Hochlauf kanalspezifisch eingestellt werden mit dem Maschinendatum:

MD20193 \$MC\_PROG\_EVENT\_IGN\_STOP

Bit	Wert	Bedeutung
		Das ereignisgesteuerte Anwenderprogramm wird:
1		• Bei einer Aktivierung durch Teileprogramm-Ende:
	0	Bei NC-Stop angehalten/verhindert
	1	Trotz NC-Stop komplett abgearbeitet
2		• Bei einer Aktivierung durch Bedientafel-Reset:
	0	Bei NC-Stop angehalten/verhindert
	1	Trotz NC-Stop komplett abgearbeitet
3		• Bei einer Aktivierung durch Hochlauf:
	0	Bei NC-Stop angehalten/verhindert
	1	Trotz NC-Stop komplett abgearbeitet

Dadurch kann z. B. ein vom Bediener durch Betätigen der NC-Stop-Taste bei Reset oder Hochlauf verursachter Flankenwechsel des Nahtstellensignals DB21, ... DBX7.3 (NC-Stop) bei der Ausführung des ereignisgesteuerten Anwenderprogramms ignoriert und ein nicht gewünschtes Stopp-Verhalten an der Maschine verhindert werden.

**Hinweis**

Eine Programmierung von DELAYFSTON/ DELAYFSTOF im ereignisgesteuerten Anwenderprogramm kann das mit MD20193 einstellbare Verhalten nicht bieten, da der NC-Stop vor Ausführung des ersten Befehls DELAYFSTON noch eine Unterbrechung bewirken kann.



### 8.9.12.3 Programmierung

#### Anwenderprogramm

##### Programmende

Soll das Anwenderprogramm durch Teileprogramm-Start aktiviert werden, muss folgendes beachtet werden:

- Das Anwenderprogramm muss mit `M17` bzw. `RET` beendet werden.
- Ein Rücksprung mittels `REPOS`-Befehl ist nicht zulässig und führt zum einem Alarm.

##### Satzanzeige

Die Anzeige in der aktuellen Satzanzeige kann durch das `DISPLOF`-Attribut in der `PROC`-Anweisung unterdrückt werden.

##### Bearbeitungsstatus

Über Anwender-M-Funktionen kann die PLC über den Bearbeitungsstatus des ereignisgesteuerten Anwenderprogramms informiert werden.

#### Abfrage des auslösenden Ereignisses

Das Ereignis, das die Aktivierung des Anwenderprogramms zur Folge hatte, kann im Anwenderprogramm mit folgender Systemvariablen abgefragt werden:

`$P_PROG_EVENT` (Ereignisgesteuerter Programmaufruf aktiv)

Wert	Bedeutung
1	Aktivierung durch <b>Teileprogramm-Start</b>
2	Aktivierung durch <b>Teileprogramm-Ende</b>
3	Aktivierung durch <b>Bedientafel-Reset</b>
4	Aktivierung durch <b>Hochlauf</b>
5	Aktivierung nach Ausgabe des letzten Aktionssatzes nach <b>Satzsuchlauf</b> (siehe "Automatischer Start eines ASUP nach Satzsuchlauf (Seite 528)")

#### Abfrage des aktuellen Kanals

Das Anwenderprogramm wird grundsätzlich in dem Kanal abgearbeitet, in dem das jeweilige Ereignis aufgetreten ist. Der aktuelle Kanal kann im Anwenderprogramm mit folgender Systemvariablen abgefragt werden:

`$P_CHANNO` (Abfrage der aktuellen Kanalnummer)

##### Hinweis

Hochlauf ist ein Ereignis in allen Kanälen.

### 8.9.12.4 Randbedingungen

#### Not-Halt / Fehlermeldung

Steht bei Bedientafel-Reset oder nach Hochlauf ein Not-Halt oder ein BAG/NCK-weiter Fehler an, dann wird das ereignisgesteuerte Anwenderprogramm erst nach der Not-Halt- bzw. Fehler-Quittierung in allen betroffenen Kanälen bearbeitet.

---

#### Hinweis

Das Ereignis Hochlauf tritt in allen Kanälen gleichzeitig auf.

---

### 8.9.12.5 Beispiele

#### Beispiel 1: Aufruf durch alle mit MD20108 gesetzten Ereignisse

Parametrierung:

MD20108 \$MC\_PROG\_EVENT\_MASK = 'H0F'

Aufruf von \_N\_PROG\_EVENT\_SPF bei:

- Teileprogramm-Start
- Teileprogramm-Ende
- Bedientafel-Reset
- Hochlauf

Programmierung:

Programmcode	Kommentar
PROC PROG_EVENT DISPLOF	
IF (\$P_PROG_EVENT==1)	; Bearbeitung für Teileprogramm-Start.
MY_GUD_VAR=0	; GUD-Variable initialisieren.
RET	
ENDIF	
IF (\$P_PROG_EVENT==2) OR (\$P_PROG_EVENT==3)	; Bearbeitung für Teileprogramm-Ende und Bedientafel-Reset.
DRFOF	; DRF-Verschiebungen ausschalten.
IF \$MC_CHAN_NAME=="CHAN1"	
CANCEL(2)	; Modale Synchronaktion 2 löschen.
ENDIF	
RET	
ENDIF	

Programmcode	Kommentar
IF (\$P_PROG_EVENT==4)	; Bearbeitung für Hochlauf.
IF \$MC_CHAN_NAME=="CHAN1"	
IDS=1 EVERY \$A_INA[1]>5.0 DO \$A_OUT[1]=1	
ENDIF	
RET	
ENDIF	
RET	

### Beispiel 2: Aufruf durch Bedientafel-Reset

Parametrierung:

MD20108 \$MC\_PROG\_EVENT\_MASK = 'H04'

Aufruf von \_N\_PROG\_EVENT\_SPF  
bei:

- Bedientafel-Reset

Programmierung:

Programmcode	Kommentar
PROC PROG_EVENT DISPLOF	
N10 DRFOF	; DRF-Verschiebungen ausschalten
N20 M17	

### Beispiel 3: Initialisierung der Funktion

Ausschnitt aus Inbetriebnahmedatei (\_N\_INITIAL\_INI):

Programmcode	Kommentar
...	
CHANDATA(3)	; Initialisierung für Kanal 3
\$MC_PROG_EVENT_IGN_INHIBIT='H04F'	
\$MC_PROG_EVENT_MASK='H04'	
...	

Bedeutung:

Das Teileprogramm \_N\_PROG\_EVENT\_SPF aus dem Verzeichnis \_N\_CMA\_DIR soll automatisch mit der RESET-Taste gestartet und bis zum Ende abgearbeitet werden, unabhängig ob die Einlesesperre ein- oder ausgeschaltet ist.

### 8.9.13 Beeinflussung von Stopp-Ereignissen durch Stopp-Delay-Bereiche

#### Stopp-Delay-Bereich

Die Reaktion auf ein Stopp-Ereignis kann durch einen bedingt unterbrechbaren Bereich im aktuellen Teileprogramm beeinflusst sein. Ein solcher Programmbereich wird als Stopp-Delay-Bereich bezeichnet.

Innerhalb von Stopp-Delay-Bereichen soll nicht angehalten und auch der Vorschub nicht verändert werden. Ein etwaiger Stopp soll erst wirken, nachdem der Programmabschnitt zu Ende bearbeitet worden ist (Beispiel: Herstellung eines Gewindes).

Die Definition eines Stopp-Delay-Bereichs erfolgt mit den Teileprogrammbefehlen:

DELAYFSTON                      Beginn eines Stopp-Delay-Bereichs  
 DELAYFSTOF                    Ende eines Stopp-Delay-Bereichs

**Literatur:**  
 Programmierhandbuch Arbeitsvorbereitung

#### Stopp-Ereignisse

Übersicht der NCK-Ereignisse, die einen Stopp zur Folge haben:

NCK-Ereignis	Reaktion	Stopp-Kriterien
Reset und BAG-Reset	immediate	NST: DB21, ... DBX7.7 und DB11 DBX20.7
PROG_END	Alarm 16954	NC-Prog: M30
Interrupt	delayed	NST: "FC-9" und ASUP DB10 DBB1
DELDISTOGO_SYNC	immediate	NST: "Restweg löschen" DB21, ... DBX6.2 und axial
PROGRESREPEAT	delayed	NST: "UP Durchlaufzahl löschen" DB21, ... DBX6.3
PROGCANCELSUB	delayed	NST: "Programmebenen Abbruch" DB21, ... DBX6.4
SINGLEBLOCKSTOP	delayed	Im Stopp-Delay-Bereich: NC stoppt am Ende des 1. Satzes außerhalb des Stopp-Delay-Bereichs. Einzelsatz bereits vor Stopp-Delay-Bereich aktiv: NST: "NC-Stop an der Satzgrenze" DB21, ... DBX7.2
SINGLEBLOCK_IPO	delayed	NST: "Einzelsatz Typ 1 einschalten" DB11 DBX21.7
SINGLEBLOCK_DECODIER	delayed	NST: "Einzelsatz Typ 2 einschalten" DB11 DBX21.6
STOPALL	immediate	NST: DB21, ... DBX7.4 und DB11 DBX20.6
STOPPROG	delayed	NST: DB21, ... DBX7.3 und DB11 DBX20.5
OVERSTORE_BUFFER_END_REACHED	Alarm 16954	NC-Prog: Stopp wegen leeren Overstore-Buffer
PREP_STOP	Alarm 16954	NC-Prog: STOPRE und alle impliziten Stopre's
PROG_STOP	Alarm 16954	NC-Prog: M0 und M1
STOPPROGATBLOCKEND	delayed	NST: "NC-Stop an Satzgrenze" DB21, ... DBX7.2
STOPPROGATSUPEND	Systemfehler	Unterprogramm-Ende sollte den Stopp-Delay-Bereich immer abwählen.
WAITM	Alarm 16954	NC-Prog: WAITM
WAITE	Alarm 16954	NC-Prog: WAITE

NCK-Ereignis	Reaktion	Stopp-Kriterien
INIT_SYNC	Alarm 16954	NC-Prog: INIT mit Parameter "S"
MMCCMD	Alarm 16954	NC-Prog: MMC( STRING, CHAR )
PROGMODESLASHON	delayed	NST: DB21, ... DBB26 Satz ausblenden ein/umschalten
PROGMODESLASHOFF	delayed	NST: DB21, ... DBB26 Satz ausblenden ausschalten
PROGMODEDRYRUNON	delayed	NST: DB21, ... DBX0.6 DryRun einschalten
PROGMODEDRYRUNOFF	delayed	NST: DB21, ... DBX0.6 DryRun ausschalten
BLOCKREADINHIBIT_ON	delayed	NST: DB21, ... DBX6.1 Einlesesperre einschalten
STOPATEND_ALARM	immediate	Alarm: Alarmprojektierung STOPATENDBYALARM
STOP_ALARM	immediate	Alarm: Alarmprojektierung STOPBYALARM
STOPATIOBUFFER_IEMPTY_ALARM	immediate	Intern: Stopp nach Alarm bei leerem Ipo-Buffer
STOPATIOBUF_EMPTY_ALARM_REORG	immediate	Intern: Stopp nach Alarm bei leerem Ipo-Buffer
RETREAT_MOVE_THREAD	Alarm 16954	NC-Prog: Alarm 16954 bei LFON (Stopp & Fastlift im G33 nicht möglich)
WAITMC	Alarm 16954	NC-Prog: WAITMC
NEWCONF_PREP_STOP	Alarm 16954	NC-Prog: NEWCONF
BLOCKSEARCHRUN_NEWCONF	Alarm 16954	NC-Prog: NEWCONF
SET_USER_DATA	delayed	BTSS: PI "_N_SETUDT"
ESR	delayed	Erweitertes Stillsetzen und Rückziehen
EXT_ZERO_POINT	delayed	Externe Nullpunktverschiebung
STOPRUN	Alarm 16955	BTSS: PI "_N_FINDST" STOPRUN

### Reaktion

Die Reaktion auf ein Stopp-Ereignis kann sein:

- **Immediate**  
Stoppt sofort, auch im Stopp-Delay-Bereich. Wird als "hartes Stopp-Ereignis" bezeichnet.
- **Delayed**  
Stopp (auch ein kurzfristiger) erfolgt erst nach dem Stopp-Delay-Bereich. Wird als "sanftes Stopp-Ereignis" bezeichnet.
- **Alarm 16954**  
Programm wird abgebrochen, da im Stopp-Delay-Bereich unerlaubte Programmbefehle verwendet worden sind.

- **Alarm 16955**

Programm wird fortgesetzt, im Stopp-Delay-Bereich hat eine unerlaubte Aktion stattgefunden.

- **Alarm 16957**

Der Programmbereich (Stopp-Delay-Bereich), der durch `DELAYFSTON` und `DELAYFSTOF` geklammert ist, konnte nicht aktiviert werden. Damit wirkt jeder Stopp sofort und wird nicht verzögert! Dies geschieht immer dann, wenn man in einen Stopp-Delay-Bereich hineinbremsst, d. h. vor den Stopp-Delay-Bereich beginnt ein Bremsvorgang, der erst im Stopp-Delay-Bereich endet. Betritt man den Stopp-Delay-Bereich mit Override 0, so kann der Stopp-Delay-Bereich ebenso nicht aktiviert werden (Beispiel: ein `G4` vor dem Stopp-Delay-Bereich erlaubt es dem Anwender, den Override auf 0 abzusenken. Der nächste Satz im Stopp-Delay-Bereich beginnt dann mit Override 0 und die beschriebene Alarmsituation tritt auf.)

---

**Hinweis**

MD11411 `$MN_ENABLE_ALARM_MASK` (Aktivierung von Warnungen) Bit 7 schaltet diesen Alarm erst ein.

---

**Stopp-Kriterien**

Ein Stopp-Ereignis kann ausgelöst werden durch:

- NC/PLC-Nahtstellensignale vom PLC → "hartes " Stopp-Ereignis
- Alarmer mit Reaktion `NOREADY` → "hartes " Stopp-Ereignis
- Stop-Taste → "sanftes" Stopp-Ereignis
- Einlesesperre → "sanftes" Stopp-Ereignis
- Einzelsatz → "sanftes" Stopp-Ereignis

---

**Hinweis**

Es gibt NCK-Ereignisse, die nur kurzfristig stoppen, um einen Umschaltvorgang durchzuführen und anschließend sofort wieder zu starten. Dazu zählt z. B. das ASUP, dass die Bahn kurzfristig stoppt, um anschließend sofort das ASUP-Programm zu starten. Auch diese Ereignisse sind im Stopp-Delay-Bereich zulässig, werden aber auf dessen Ende verschoben und gelten demnach als "sanftes" Stopp-Ereignis.

---

## Bedingungen

Solange ein Stopp-Delay-Bereich bearbeitet wird, gelten folgende Bedingungen:

- Eine Veränderung des **Vorschubs** wird während des Stopp-Delay-Bereichs ignoriert. Damit wirkt eine **Vorschubsperr**e erst nach dem Verlassen des Programmbereichs.
- Alle Hauptlaufachsen wie Kommandoachsen und Positionierachsen, die mit `POSA` verfahren, werden nicht gestoppt.
- Der Teileprogrammbehl `G4` ist im Stopp-Delay-Bereich zulässig.  
Andere Teileprogrammbehl, die zu einem zum vorübergehenden Stopp führen (z. B. `WAITM`), sind nicht zulässig und lösen den Alarm 16954 aus.
- Wird ein Stopp-Delay-Bereich mit Override 0% betreten, so wird der Stopp-Delay-Bereich nicht akzeptiert!

## 8.10 Asynchrone Unterprogramme (ASUPs), Interruptroutinen

### 8.10.1 Funktion

#### 8.10.1.1 Allgemeine Funktionalität

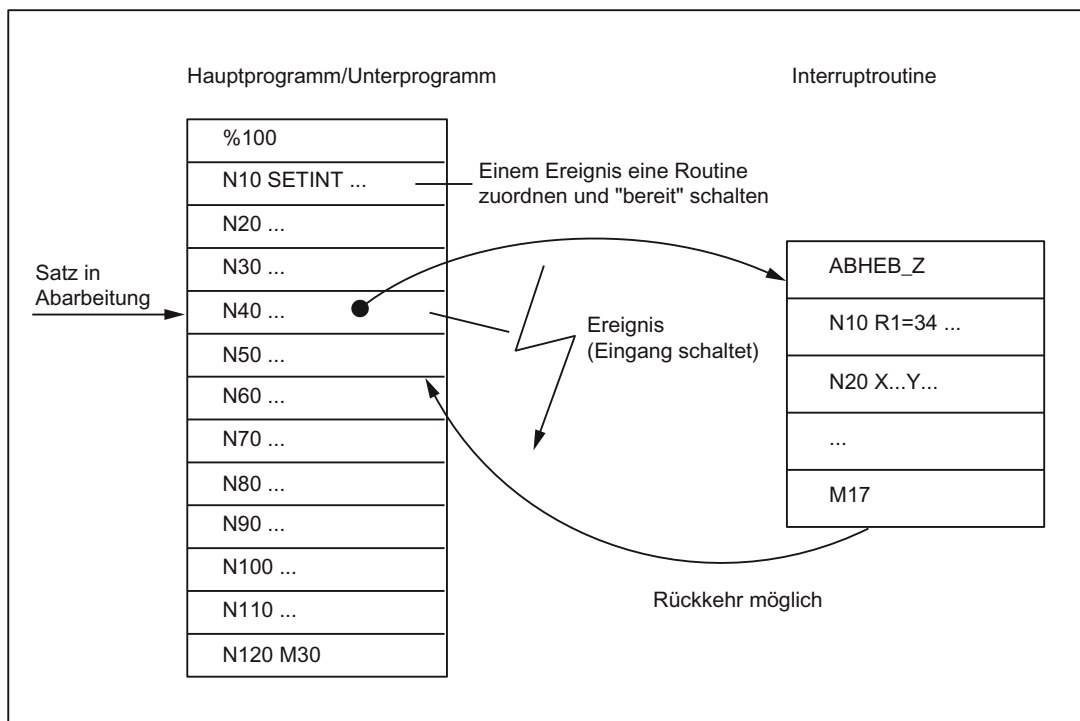
**Hinweis**

Die in der folgenden Beschreibung abwechselnd vorkommenden Begriffe "Asynchrones Unterprogramm (ASUP)" und "Interruptroutine" kennzeichnen die gleiche Funktionalität.

### Interruptroutinen

Interruptroutinen sind normale Teileprogramme, die als Reaktion auf Interruptereignisse (Interrupteingänge, Prozess- bzw. Maschinenzustand) vom Bearbeitungsprozess bzw. vom jeweiligen Maschinenzustand gestartet werden.

Ein in Abarbeitung befindlicher Teileprogrammabsatz wird hierbei abgebrochen (sofern er nicht durch entsprechende Deklaration gegen Unterbrechung verriegelt ist). Ein späteres Fortsetzen des Teileprogramms an der Unterbrechungsposition ist möglich.





## Definition von Interruptroutinen

Einem Teileprogramm, das als Interruptroutine wirksam sein soll, muss über den Befehl `SETINT` oder über den PI-Dienst "ASUP" ein Interruptsignal zugeordnet werden. Erst dadurch wird aus dem Teileprogramm eine Interruptroutine.

## Interruptsignale

- Es sind insgesamt 8 Interruptsignale(-eingänge) verfügbar.
- Alle Eingänge sind über die PLC beeinflussbar.
- Die ersten vier Interruptsignale werden zudem über die 4 schnellen NC-Eingänge der NCU-Baugruppe beeinflusst.
- Der Signalzustand der schnellen NC-Eingänge kann über die PLC-Nahtstelle (DB10) ausgelesen werden.
- Die Übertragung der schnellen NC-Eingangssignale auf die Interruptsignale kann über die PLC-Nahtstelle (DB10) einzeln gesperrt werden.

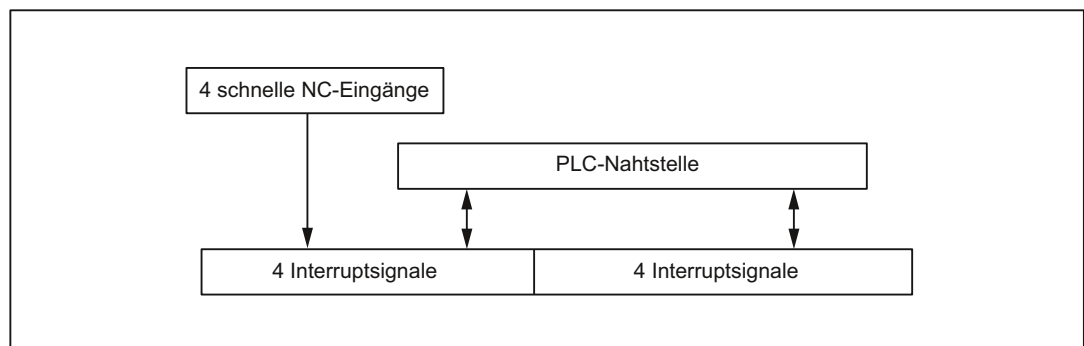


Bild 8-11 Interruptsignale

Weitere Informationen zur PLC-Beeinflussung der schnellen NC-Eingänge (Interruptsignale) siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)".

### Literatur:

Funktionshandbuch Erweiterungsfunktion; Digitale und analoge NCK-Peripherie (A4)

## Aufruf von Interruptroutinen

### Im Programmbetrieb

Der Aufruf von Interruptroutinen ist grundsätzlich möglich, wenn sich die Betriebsartengruppe im Programmbetrieb befindet. Das bedeutet, dass entweder in der Betriebsart AUTOMATIK oder MDA Teileprogrammsätze abgearbeitet werden.

### Außerhalb vom Programmbetrieb

Daneben können Interruptroutinen aber auch in folgenden ProgramMZuständen bzw. Betriebsarten aktiviert werden:

- JOG, JOG-REF
- MDA-Teach In, MDA-Teach In-REF, MDA-Teach In-JOG, MDA-REF, MDA-JOG
- AUTOMATIK, gestoppt, ready
- Nicht referenziert

Wird eine Interruptroutine während der Aktionen JOG oder REF aktiviert, so werden diese Aktionen von der Interruptroutine abgebrochen.

## Aktivierung

Die Aktivierung einer Interruptroutine kann erfolgen:

- Durch die 0/1-Flanke des Interruptsignals, hervorgerufen durch eine 0/1-Flanke am schnellen NC-Eingang
- Durch den Aufruf des "Function Call ASUP" (siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)")
- Dadurch, dass über Synchronaktion ein Ausgang gesetzt wird, der indirekt über Kurzschluss einen Interrupt-Eingang setzt (siehe "Beispiele (Seite 629)")

### Literatur:

Funktionshandbuch Synchronaktionen

## Anzeige

Die Aktivierung einer Interruptroutine wird mit dem folgenden NC/PLC-Nahtstellensignal angezeigt:

DB21, ... DBX378.0 (ASUP aktiv)

### 8.10.1.2 Ablauf einer Interruptroutine im Programmbetrieb

#### Abbremsen der Achsen

Nach Aktivierung einer Interruptroutine werden alle Maschinenachsen entlang der Beschleunigungsrampe (MD32300 \$MA\_MAX\_AX\_ACCEL) bis zum Stillstand abgebremst und die Achspositionen abgespeichert.

#### Reorganisation

Zusätzlich zum Abbremsen der Achsen werden die vordekodierten Rechensätze bis zum Unterbrechungssatz zurückgerechnet, d. h. alle Variablen, Frames und G-Codes bekommen den Wert zugeordnet, den sie ohne Vorausdekodierung des Teileprogramms an der Unterbrechungsstelle besitzen würden. Diese Werte werden auch wieder zwischengespeichert, um nach dem Ende der Interruptroutine wieder darauf zugreifen zu können.

Ausnahmen, wo keine Reorganisation möglich ist:

- Innerhalb von Gewindeschneidsätzen
- Bei komplexen Geometrien (z. B. Spline oder Radiuskorrektur)

#### Abarbeiten der Interruptroutine

Nach Beendigung der Reorganisation wird automatisch die Interruptroutine gestartet.

Die Interruptroutine wird vom System wie ein normales Unterprogramm behandelt (Schachtelungstiefe u. ä.).

#### Ende der Interruptroutine

Nachdem die Endkennung ( $M02$ ,  $M30$ ,  $M17$ ) der Interruptroutine bearbeitet wurde, wird standardmäßig auf die Endposition des auf den Unterbrechungssatz folgenden Teileprogrammsatzes gefahren.

Falls ein Rückpositionieren auf den Unterbrechungspunkt gewünscht wird, muss eine `REPOS-`Anweisung am Ende der Interruptroutine stehen, z. B.:

```
N104 REPOS L M17
```

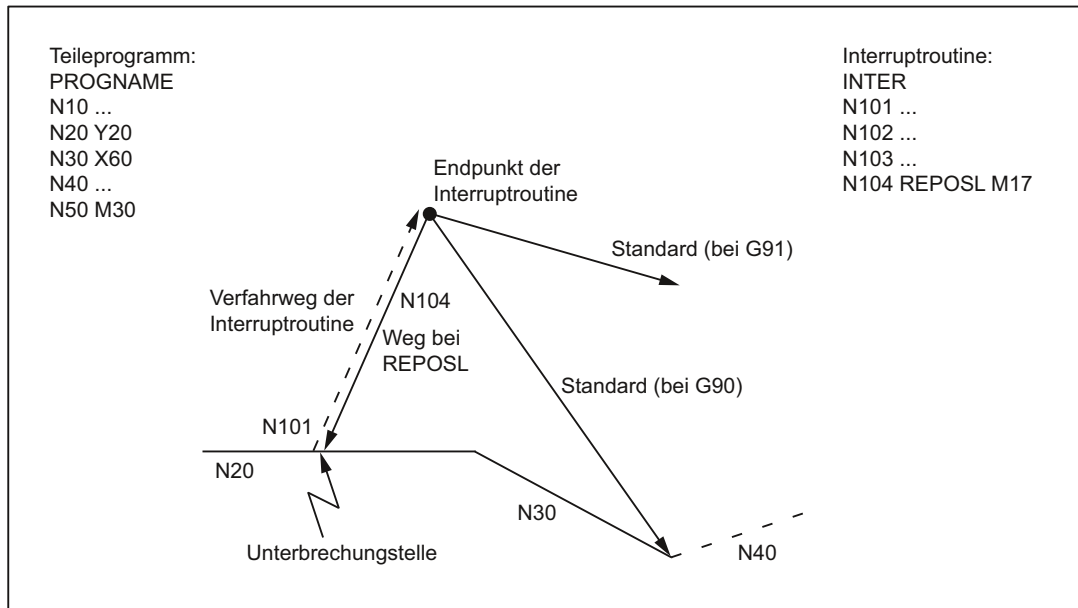


Bild 8-12 Ende der Interruptroutine

### 8.10.1.3 Interruptroutine mit REPOSA

Wird eine im Programmbetrieb im Kanalzustand "unterbrochen" durch die PLC (Baustein FC9) ausgelöste Interruptroutine mit REPOSA abgeschlossen, so ist folgender Ablauf typisch:

1. Vor dem Wiederaufstart an die Kontur hält die Steuerung an und geht in den Programmzustand "angehalten". Das folgende NC/PLC-Nahtstellensignal wird gesetzt:  
DB21, ... DBX318.0 (ASUP ist angehalten)
2. Der Bediener drückt die START-Taste. Daraufhin wird das Signal DB21, ... DBX318.0 zurückgesetzt und es startet die Wiederaufstartbewegung.
3. Mit dem Ende der Wiederaufstartbewegung wird das FC9-Signal "ASUP-Done" gesetzt und die Bahn des unterbrochenen Teilprogramms wird fortgesetzt.

---

#### Hinweis

Das NC/PLC-Nahtstellensignal DB21, ... DBX318.0 (ASUP ist angehalten) steht nur für folgenden Fall zur Verfügung: Interrupt im Programmbetrieb im Kanalzustand "unterbrochen".

---

#### Hinweis

Bei Interruptroutinen, die ohne REPOS abschließen, fallen die Signale "Asup-Done" und DB21, ... DBX318.0 (ASUP ist angehalten) zeitlich zusammen.

---

## 8.10.1.4 NC-Verhalten

Die unterschiedlichen Reaktionen der Steuerung auf eine aktivierte Interruptroutine in den verschiedenen Betriebszuständen beschreibt die folgende Tabelle:

Zustand der NC	ASUP-Start	Reaktion der Steuerung
Programm ist aktiv	Interrupt, (PLC)	<ol style="list-style-type: none"> <li>1. Schnellabheben oder Achsen stoppen</li> <li>2. Unterbrechung des Programms für die Dauer des ASUP</li> <li>3. Anfahren der Unterbrechungsstelle, wenn REPOS im ASUP</li> <li>4. Fortsetzung des Teileprogramms</li> </ol>
RESET	Interrupt, (PLC)	<p>Das ASUP läuft ab wie ein Hauptprogramm. Am ASUP-Ende wird wieder RESET (ohne M30) ausgeführt. Der nächste Zustand der Steuerung hängt von folgenden Maschinendaten ab:</p> <p>MD20110 \$MC_RESET_MODE_MASK MD20112 \$MC_START_MODE_MASK</p> <p><b>Literatur:</b> Funktionshandbuch Grundfunktionen; Achsen, Koordinatensysteme, Frames (K2), Kapitel: "Werkstücknahes Istwertsystem"</p>
Programmbetrieb (AUTOMATK oder MDA) + Kanal gestoppt	Interrupt, (PLC)	<p>ASUP läuft ab. An seinem Ende wird wieder der Stopp-Zustand eingenommen.</p> <p>Wenn REPOS im ASUP:</p> <ul style="list-style-type: none"> <li>• Die ASUP-Abarbeitung wird vor dem Anfahr Satz gestoppt.</li> <li>• Das Anfahren kann durch Start-Taste ausgelöst werden.</li> </ul>
	Start-Taste	Nach ASUP-Ablauf läuft auch das gestoppte Programm weiter ab.
Handbetrieb + Kanal gestoppt	Interrupt, (PLC)	Steuerung nimmt intern für den angesprochenen Kanal den Zustand "interne Programmbearbeitungsbetriebsart" ein (nach außen nicht merkbar) und aktiviert dann das ASUP. Die angewählte Betriebsart bleibt erhalten. Nach ASUP-Ende (M17) wird der ursprüngliche Zustand wieder eingenommen.
JOG AUTO-Teach-In AUTO-Teach-Referenzp.	Interrupt, (PLC)	Stopp der Bearbeitung, Auswerten von: MD11602 \$MN_ASUP_START_MASK MD11604 \$MN_ASUP_START_PRIO_LEVEL
MDA-JOG, MDA-Teach-In, MDA-Teach-Referenzp.	Interrupt, (PLC)	Ggf. internes Umschalten in "interne Programmbearbeitungsbetriebsart", Aktivierung des ASUP, Zustand vor ASUP-Start wieder herstellen. Ein zusammen mit SETINT definiertes LIFFFAST wird bei JOG nicht aktiviert.
Handbetrieb + Kanal läuft	Interrupt, (PLC)	Die gerade aktive Bewegung wird gestoppt. Der Restweg wird gelöscht. Der weitere Ablauf entspricht "Handbetrieb, Kanal gestoppt".
Abarbeitung von INITIAL.INI	<b>nicht möglich</b>	Es wird das Signal "Interruptanforderung nicht möglich" erzeugt.
Satzsuchlauf		
Alarm, der nicht durch NC- Start behoben werden kann		
Digitalisieren eingeschaltet		
Kanal im Fehlerzustand		

## 8.10.2 Parametrierung

### Wirkung der BAG-spezifischen Signale

Die Wirkung der BAG-spezifischen Signale (NC/PLC-Nahtstellensignale DB11) auf Kanäle der Betriebsartengruppe, die gerade Interruptroutinen bearbeiten, wird eingestellt im Maschinendatum:

MD11600 \$MN\_BAG\_MASK.Bit n = <Wert>

Bit	Wert	Bedeutung
0	0	Die BAG-spezifischen Signale (NC/PLC-Nahtstellensignale DB11) wirken.
	1	Die BAG-spezifischen Signale (NC/PLC-Nahtstellensignale DB11) wirken nicht.

MD11600 steuert auch, ob die interne Programmverarbeitungsart nur für den Kanal eingenommen wird, in dem die Interruptroutine aktiviert wurde, oder für alle Kanäle der BAG:

Bit	Wert	Bedeutung
1	0	Eine BAG-Umschaltung erfolgt in allen Kanälen der BAG.
	1	Eine interne BAG-Umschaltung erfolgt nur in dem Kanal, in dem eine Interruptroutine aktiv ist (nur möglich, bei Bit 0=1).

Wenn aufgrund der Maschinendateneinstellung der Kanal, in dem der Interrupt abgearbeitet wird, die BAG verlassen hat, wirken auf diesen Kanal die BAG-Signale "BAG-Reset", "BAG-Stop", etc. **nicht**. Damit wird die Interruptroutine durchlaufen, ohne von BAG-Signalen gestört zu werden.

### Parametrierbare Starfreigabe

In der Grundeinstellung wird der Start eines ASUP bei folgenden Zuständen im Kanal blockiert:

- Stop durch NC-Stop, M0 oder M01
- Nicht alle Achsen im Kanal sind referenziert
- Einleesperre ist aktiv (DB21, ... DBX6.1 = 1)

Über folgende Maschinendaten kann der Start Zustand-spezifisch freigegeben werden.

#### NC-Spezifische Startfreigabe für NC-Stop, M0, M01, Einleesperre

MD11602 \$MN\_ASUP\_START\_MASK

### Kanal-spezifische Startfreigabe bei nicht referenzierten Achsen im Kanal

- MD20105 \$MC\_PROG\_EVENT\_IGN\_REFP\_LOCK

Über das Maschinendatum kann für ereignisgesteuerte Programmaufrufe (ProgEvent) bei nicht referenzierter Achsen im Kanal, die Startfreigabe separat für folgenden Zustände freigegeben werden:

- Teileprogramm-Start aus Kanalzustand "Reset"
- Teileprogramm-Ende
- Reset
- Power On
- Erster Start nach Suchlauf

- MD20115 \$MC\_IGNORE\_REFP\_LOCK\_ASUP

Über das Maschinendatum kann für ASUP bei nicht referenzierter Achsen im Kanal, die Startfreigabe Interrupt-spezifisch erteilt werden.

### Manuelle Startfreigabe

Darf aufgrund der parametrisierten Startfreigaben ein ASUP nicht automatisch gestartet werden, kann der Anwender manuell, durch Auslösen von **NC-Start**, eine Startfreigabe erteilen.

---

#### Hinweis

Das ASUP für "Schnellabheben von der Kontur" (**LIFTFAST**) wird in jedem Fall gestartet.

---

### Manuelles Verfahren während Unterbrechung eines ASUP in der Betriebsart JOG

Während der Unterbrechung eines in der Betriebsart JOG automatisch gestarteten ASUP, können vom Bediener Achsen manuell über die Verfahrtasten verfahren werden. Die Freigabe der Funktion erfolgt über:

MD11602 \$MN\_ASUP\_START\_MASK.Bit 3

---

#### Hinweis

##### Mehrkanalige Systeme

Bei mehrkanaligen Systemen muss zusätzlich folgendes Maschinendatum gesetzt werden:

MD11600 \$MN\_BAG\_MASK.Bit 1 = 1

---

#### Fortsetzen des ASUP

Nach dem manuellen Verfahren der Achsen, ist vom Bediener NC-Start auszulösen. Dabei werden die Achsen automatisch an die Unterbrechungsstelle verfahren (REPOS). Anschließend wird das ASUP an der Unterbrechungsstelle fortgesetzt.

### Anwendungsbeispiel

Bei einer Einschlitzen-Drehmaschine wird in der Betriebsart JOG ein Abspannzyklus als ASUP gestartet und damit eine mehrere Meter lange Welle bearbeitet. Während der Bearbeitung wird es notwendig, die Schneide des Werkzeugs zu wechseln. Der Maschinenbediener stoppt das ASUP und verfährt die Achsen manuell über die Verfahrtasten zum Schneidenwechsel vom Werkstück weg. Nach dem Schneidenwechsel betätigt er NC-Start. Durch den REPOS-Vorgang erfolgt ein Wiederanfahren an die Unterbrechungsstelle. Anschließend wird das ASUP fortgesetzt.

### Randbedingung

Manuelles Verfahren von Achsen während der Unterbrechung eines ASUP in der Betriebsart JOG ist nur möglich, wenn das ASUP aus dem Programmzustand "abgebrochen", Kanalzustand "Reset" heraus aktiviert wurden.

### Wirksamkeit der parametrisierten Startfreigaben

Den anwenderspezifischen Interruptsignalen sind die Prioritäten 1 - 8, mit 1 = höchste Priorität zugewiesen (siehe Kapitel "Programmierung (Seite 626)"). Bis zu welcher Priorität, ausgehend von der höchsten, die parametrisierten Startfreigaben für die zugehörigen ASUP wirken, wird mit folgendem Maschinendatum festgelegt:

MD11604 \$MN\_ASUP\_START\_PRIO\_LEVEL = <Priorität>

### Verhalten bei Einlesesperre

Über das folgende kanalspezifische Maschinendatum kann für jedes Interruptsignal eingestellt werden, ob das zugeordnete ASUP trotz gesetzter Einlesesperre (DB21, ... DBX6.1 == 1) abgearbeitet wird oder die Einlesesperre wirksam sein soll:

MD20116 \$MC\_IGNORE\_INHIBIT\_ASUP.Bit 0 - Bit 31

Bit x ist dem Interruptsignal (x+1) zugeordnet.

### Randbedingungen

Die Einstellungen im MD20116 \$MC\_IGNORE\_INHIBIT\_ASUP sind unwirksam, wenn die Einlesesperre im ASUP grundsätzlich ignoriert werden soll:

MD11602 \$MN\_ASUP\_START\_MASK.Bit 2 = 1

### Verhalten bei gesetzter Einzelsatzbearbeitung

Über das folgende kanalspezifische Maschinendatum kann für jedes Interruptsignal eingestellt werden, ob das zugeordnete ASUP bei aktiver Einzelsatzbearbeitung ohne Unterbrechung abgearbeitet werden oder die Einzelsatzbearbeitung wirksam sein soll:

MD20117 \$MC\_IGNORE\_SINGLEBLOCK\_ASUP.Bit 0 - Bit 31

Bit x ist dem Interruptsignal (x+1) zugeordnet.



### Randbedingungen

- MD20117 \$MC\_IGNORE\_SINGLEBLOCK\_ASUP wirkt nur bei IPO-Einzelsatz (SBL1).
- Die Einstellungen im MD20117 \$MC\_IGNORE\_SINGLEBLOCK\_ASUP sind unwirksam, wenn die Einzelsatzbearbeitung in ASUP grundsätzlich ignoriert werden soll:  
MD10702 \$MN\_IGNORE\_SINGLEBLOCK\_MASK.Bit 1 = 1

### Aktualisierung der Anzeige "Programm- und Kanalzustand" unterdrücken

Um ein Flackern der Anzeige des Programm- und Kanalzustandes an der Bedienoberfläche bei der Ausführung von sehr kurzen ASUP zu vermeiden, kann die Aktualisierung der Anzeige unterdrückt werden. In der Anzeige wird dann der Programm- und Kanalzustand vor Aktivierung des ASUP angezeigt. Die Parametrierung erfolgt über das kanalspezifische Maschinendatum:

MD20191 \$MC\_IGN\_PROG\_STATE\_ASUP.Bit 0 - Bit 31

Bit x ist dem Interruptsignal (x+1) zugeordnet.

Bei Ausführung eines ASUP mit unterdrückter Anzeige wird folgendes NC/PLC-Nahtstellensignal gesetzt:

DB21, ... DBX378.1 == 1 (Stilles ASUP aktiv)

### Spezielle Systemvariable und NC/PLC-Nahtstellensignale

Die Systemvariablen und NC/PLC-Nahtstellensignale für Programm- und Kanalzustand werden durch die Unterdrückung der Anzeige während der Abarbeitung eines ASUP nicht beeinflusst:

- \$AC\_STAT (Kanalzustand)
- \$AC\_PROG (Programmzustand)
- DB21, ... DBX35.0 - 4 (Programmzustand)
- DB21, ... DBX35.5 - 7 (Kanalzustand)

### Schnellabheben von der Kontur (LIFTFAST)

Über das folgende Maschinendatum kann eingestellt werden, ob bei aktiver Funktion "Spiegeln" die Rückzugsrichtung beim "Schnellabheben von der Kontur" (LIFTFAST) gespiegelt werden soll:

MD21202 \$MC\_LIFTFAST\_WITH\_MIRROR (Schnellabheben mit Spiegeln)

Die Spiegelung der Rückzugsrichtung bezieht sich nur auf die Richtungskomponenten senkrecht zur Werkzeugrichtung.

### Literatur

Eine ausführliche Beschreibung der Maschinendaten findet sich in:  
Listenhandbuch "Ausführliche Beschreibung der Maschinen"

### 8.10.3 Programmierung

#### Zuordnung Interruptsignal ↔ Teileprogramm

Die Zuordnung Interruptsignal ↔ Teileprogramm erfolgt mit dem Befehl `SETINT`.

Beispiel:

Programmcode	Kommentar
...	
N20 SETINT(3) ABHEBEN_Z	; Wenn Eingang 3 schaltet, dann soll die Interruptroutine "ABHEBEN_Z" starten.
...	

Zusammen mit `SETINT` können zusätzlich folgende Anweisungen programmiert werden:

- `LIFTFAST`

Beim Eintreffen des Interruptsignals wird vor dem Start der Interruptroutine ein "Schnellabheben des Werkzeugs von der Kontur" durchgeführt. Die Bewegungsrichtung für das Schnellabheben wird durch die Programmanweisung `ALF` festgelegt.

- `BLSYNC`

Beim Eintreffen des Interruptsignals wird der laufende Programmsatz noch abgearbeitet und erst danach die Interruptroutine gestartet.

---

#### Hinweis

Die Zuordnung Interruptsignal ↔ Teileprogramm wird unter folgenden Bedingungen gelöscht:

- Reset-Zustand des Kanals
  - `CLRINT`-Anweisung im Teileprogramm
- 

#### Prioritäten

Falls im Teileprogramm mehrere `SETINT`-Befehle stehen und dadurch mehrere Signale gleichzeitig eintreffen können, müssen den zugeordneten Interruptroutinen Prioritätswerte zugewiesen werden, die die Reihenfolge bei der Abarbeitung festlegen:

`PRIO=<Wert>`

Es gibt die Prioritäten 1 bis 128. Priorität 1 entspricht der höchsten Priorität.

Beispiel:

Programmcode	Kommentar
...	
N20 SETINT(3) PRIO=2 ABHEBEN_Z	; Wenn Eingang 3 schaltet, dann soll die Interruptroutine "ABHEBEN_Z" starten.
N30 SETINT(2) PRIO=2 ABHEBEN_X	; Wenn Eingang 2 schaltet, dann soll die Interruptroutine "ABHEBEN_X" starten.
...	

Die Interruptroutinen werden in der Reihenfolge der Prioritätswerte nacheinander abgearbeitet, wenn die Eingänge gleichzeitig anstehen: zuerst "ABHEBEN\_Z", dann "ABHEBEN\_X".

## REPOS-Abfrage

Bei Interruptroutinen können Abläufe entstehen, für die es keine eindeutige Rückkehr zu einem Abbruchpunkt der Satzbearbeitung (REPOS) gibt.

Über die Systemvariable \$P\_REPINF kann im ASUP abgefragt werden, ob REPOS möglich ist:

Wert	Bedeutung
0	Repositionieren mit REPOS nicht möglich, weil: <ul style="list-style-type: none"> <li>• Nicht im ASUP aufgerufen</li> <li>• ASUP aus Reset-Zustand ablief</li> <li>• ASUP aus JOG ablief</li> </ul>
1	Repositionieren mit REPOS im ASUP möglich

## Ursache der Aktivierung feststellen

Mit der Systemvariablen \$AC\_ASUP wird die Ursache, die zur Aktivierung einer Interruptroutine geführt hat, bitcodiert angegeben und kann im Teileprogramm und in Synchronaktionen gelesen werden (siehe "Anwenderspezifisches ASUP für RET und REPOS > Programmierung (Seite 632)").

### Flexible Programmierung

Für die flexible Programmierung von Interruptroutinen dienen folgende Befehle:

Befehl	Bedeutung
SAVE	Wenn bei der Programmierung der Interruptroutine der <i>SAVE</i> -Befehl verwendet wird, dann werden die früheren G-Codes, Frames und Transformationen des unterbrochenen Teileprogramms wieder wirksam, sobald die Interruptroutine beendet ist.
DISABLE	Durch Verwendung des <i>DISABLE</i> -Befehls können Teileprogrammabschnitte vor dem Unterbrechen durch die Interruptroutine geschützt werden. Die Zuordnung Interruptsignal ↔ Teileprogramm bleibt dabei erhalten, es wird lediglich nicht mehr auf die 0/1-Flankenänderung des Interruptsignals reagiert.
ENABLE	Mit dem Befehl <i>ENABLE</i> kann der <i>DISABLE</i> -Befehl wieder zurückgesetzt werden. Die Interruptroutine wird erst bei der nächsten 0/1-Flankenänderung des Interruptsignals gestartet.
CLRINT	Zuordnung Interruptsignal ↔ Teileprogramm löschen.

### Literatur

Programmierhandbuch Arbeitsvorbereitung; Kapitel: "Flexible NC-Programmierung" > "Interruptroutine (ASUP)"

## 8.10.4 Randbedingungen

### Betriebsartenübergreifender Start von Interruptroutinen

#### Voraussetzung

MD11602 \$MN\_ASUP\_START\_MASK, mindestens Bit 0 = 1

Zusätzlich sind folgende Daten zu beachten:

- MD11600 \$MN\_BAG\_MASK
- MD11604 \$MN\_ASUP\_START\_PRIO\_LEVEL
- Die Priorität der Interruptzuordnung

### **Empfohlene Einstellungen**

NC-spezifische Maschinendaten:

- MD11600 \$MN\_BAG\_MASK = 'H11'
- MD11602 \$MN\_ASUP\_START\_MASK = 'H101'
- MD11604 \$MN\_ASUP\_START\_PRIO\_LEVEL = 7

Kanalspezifische Maschinendaten für den Kanal in dem das ASUP gestartet wird, oder allgemein für alle Kanäle:

- MD20105 \$MC\_PROG\_EVENT\_IGN\_REFP\_LOCK = 'H3F'
- MD20115 \$MC\_IGNORE\_REFP\_LOCK\_ASUP = 'HFFFFFFF'

## **8.10.5 Beispiele**

### **Aktivierung einer Interruptroutine über Synchronaktion**

1. Anzahl der aktiven digitalen Ein-/Ausgänge festlegen:  
MD10350 \$MN\_FASTIO\_DIG\_NUM\_INPUTS=3  
MD10360 \$MN\_FASTIO\_DIG\_NUM\_OUTPUTS=3
2. Mit folgenden MD-Einstellungen Kurzschluss erzeugen:  
MD10361 \$MN\_FASTIO\_DIG\_SHORT\_CIRCUIT[0]='H0102B102'  
MD10361 \$MN\_FASTIO\_DIG\_SHORT\_CIRCUIT[1]='H0202B202'
3. HW-Zuordnung des externen NC-Eingangsbytes für NC-Programm-Interrupts:  
MD21210 \$MC\_SETINT\_ASSIGN\_FASTIN=2 ; besser 1 Byte mehr als benötigt
4. Eingang als ASUP-Auslöser definieren:  
SETINT(1) PRIO=1 SYNCASUP
5. IDS=1 EVERY \$\$AC\_PATHN>=0.5 DO \$A\_OUT\_[9]=1

## 8.11 Anwenderspezifisches ASUP für RET und REPOS

### 8.11.1 Funktion

#### Funktion

Die ausgelieferte NCK-Software enthält fixe Abläufe (interne ASUP) zur Realisierung der Funktionen RET und REPOS. Diese können vom Maschinenhersteller durch ein selbst geschriebenes, anwenderspezifisches ASUP ersetzt werden.



Für den Inhalt der ASUP-Routine, welche die von SIEMENS ausgelieferten ASUP.SYF ersetzt, trägt der Maschinenhersteller die Verantwortung.

#### Installation

Im Hersteller-Verzeichnis `_N_CMA_DIR` oder im Anwender-Verzeichnis `_N_CUS_DIR` kann **eine** Routine mit dem Namen `"_N_ASUP_SPF"` geladen werden. Diese muss die vom Anwender gewünschten Aktionen für die Funktionen RET und REPOS realisieren.

### 8.11.2 Parametrierung

#### Aktivierung

Die Parameter bezüglich der Aktivierung der anwenderspezifischen Routine `"_N_ASUP_SPF"` werden eingestellt mit dem Maschinendatum:

MD11610 \$MN\_ASUP\_EDITABLE (Aktivierung eines anwenderspezifischen ASUP)

Mit Bit 0 und Bit 1 wird angegeben, welche der internen Systemroutinen durch das anwenderspezifische ASUP ersetzt werden sollen:

Binärwert	Bedeutung
0	<b>Weder</b> bei RET <b>noch</b> bei REPOS wird die anwenderspezifische Routine <code>_N_ASUP_SPF</code> aktiviert.
1	Bei <b>RET</b> wird die anwenderspezifische Routine aktiviert, bei REPOS wird die vom System bereitgestellte Routine aktiviert.
2	Bei <b>REPOS</b> wird die anwenderspezifische Routine aktiviert, bei RET wird die vom System bereitgestellte Routine aktiviert.
3	<b>Sowohl</b> bei RET <b>als auch</b> bei REPOS wird die anwenderspezifische Routine aktiviert.

Mit Bit 2 wird festgelegt, in welchem Verzeichnis die anwenderspezifische Routine im Falle der Aktivierung zuerst gesucht werden soll:

Bit	Wert	Bedeutung
2	0	Die anwenderspezifische Routine wird zuerst im Anwender-Verzeichnis <code>_N_CUS_DIR</code> gesucht.
	1	Die anwenderspezifische Routine wird zuerst im Hersteller-Verzeichnis <code>_N_CMA_DIR</code> gesucht.

### Festlegung einer Schutzstufe

Wenn ein anwenderspezifisches ASUP für RET und/oder REPOS benutzt werden soll, d. h. wenn:

MD11610 \$MN\_ASUP\_EDITABLE ≠ 0

dann kann für die anwenderspezifische Routine "`_N_ASUP_SPF`" eine Schutzstufe festgelegt werden. Die Schutzstufe kann Werte im Bereich 0 - 7 annehmen.

Die Einstellung erfolgt über das Maschinendatum:

MD11612 \$MN\_ASUP\_EDIT\_PROTECTION\_LEVEL (Schutzstufe des anwenderspezifischen ASUP)

Details zu den Schutzstufen finden Sie in:

**Literatur:**

Inbetriebnahmehandbuch; Schutzstufenkonzept

### Verhalten bei gesetzter Einzelsatzbearbeitung

Über das folgende Maschinendatum kann eingestellt werden, dass trotz gesetzter Einzelsatzbearbeitung die internen ASUP oder das anwenderspezifischen ASUP "`_N_ASUP_SPF`" ohne Unterbrechung abgearbeitet werden:

MD10702 \$MN\_IGNORE\_SINGLEBLOCK\_MASK (Einzelsatzstopp verhindern)

Bit	Wert	Bedeutung
0	0	Es wird in jedem ASUP-Satz angehalten.
	1	Das ASUP wird ohne Unterbrechung abgearbeitet.

### 8.11.3 Programmierung

#### Ermitteln der Ursache der ASUP-Aktivierung

Die Ursache, die zur Aktivierung des ASUP geführt hat, kann bitcodiert über die Systemvariable \$AC\_ASUP gelesen werden.

#### Fortsetzung

Bei Verwendung der System-ASUP ist das Verhalten für die Fortsetzung nach Abarbeitung der Aktionen innerhalb des ASUP fest vorgegeben:

- System-ASUP1 → Fortsetzung mit `RET` (Unterprogrammrücksprung)
- System-ASUP2 → Fortsetzung mit `REPOS` (Repositionieren)

In der Beschreibung der Systemvariablen ist unter "Fortsetzung durch" für jede Ursache das Verhalten bezüglich der System-ASUP angegeben.

---

#### Hinweis

##### Fortsetzung bei anwenderspezifischem ASUP

Es wird empfohlen bei anwenderspezifischen ASUP die entsprechende Fortsetzung der System-ASUP beizubehalten.

##### Ursache: Betriebsartenwechsel (\$AC\_ASUP, Bit 9 == 1)

Bei Betriebsartenwechsel ist die Fortsetzung abhängig vom Maschinendatum:

MD20114 \$MC\_MODESWITCH\_MASK (Unterbrechung MDA durch Betriebsartwechsel)

- Bit 0 == 0: System-ASUP1 → Fortsetzung mit `RET`
  - Bit 0 == 1: System-ASUP2 → Fortsetzung mit `REPOS`
- 

#### Literatur

Eine ausführliche Beschreibung der Systemvariable findet sich in:

Listenhandbuch Systemvariable



## 8.12 Einzelsatz

### Satzweise Bearbeitung

In der Funktion Einzelsatz kann der Anwender das Teileprogramm satzweise abarbeiten.

### Einzelsatz-Arten

Es gibt 3 Einstellungsarten der Funktion Einzelsatz:

- SBL1 := IPO-Einzelsatz  
Bei aktiver SLB1 Funktion erfolgt nach jedem Maschinenaktions-Satz (Ipo-Satz) ein Anhalten bzw. Stoppen der Bearbeitung.
- SBL2 := Decodiereinzelsatz  
Bei aktiver SLB2 Funktion erfolgt auf jeden Fall nach jedem Teileprogrammsatz ein Anhalten bzw. Stoppen der Bearbeitung. Erfolgt die Bearbeitung eines Teileprogrammsatzes in mehreren Ipo-Sätzen, erfolgt nach jedem Ipo-Satz das Anhalten. Eine Ausnahme bildet Gewindeschneiden.
- SBL3 := Decodiereinzelsatz  
Wie SLB2, jedoch wird auch in den Teileprogrammsätzen der Zyklen angehalten.

1. Das Anhalten nach jedem Satz ist in vielen Situationen bzw. bei einigen Sätzen nicht erwünscht.
  - 1. Beispiel:  
Wechsel nach Joggen, wenn nicht reorganisierbar bzw. repositionierbar ist, MD10702, Bit 6 und 7.  
Wird in einem Satz am Satzende angehalten, der nicht reorganisierbar bzw. repositionierbar ist, kann in dieser Situation die Betriebsart Joggen nicht angewählt werden.
  - 2. Beispiel:  
Wechsel nach JOG auf einem STOPRE-Satz, MD10702, Bit 6 und 7  
Wechselt man die Betriebsart AUTO nach Joggen, während ein STOPRE-Satz aktiv ist, kommen nach einem Fortsetzungsstart außer dem Systemasup2 ein Restsatz und ein bzw. evtl. (bei Decodiersingleblock) zwei weitere STOPRE-Sätze. Ein Logik, die in Einzelsatz immer einen Teileprogramm-Start auslöst und danach immer nach Joggen wechselt, bleibt ewig auf dem STOPRE-Satz stehen.
  - 3. Beispiel:  
DISPOF: Ausschalten der Satzanzeige, MD10702, Bit 6 und 7  
Wurde in einem Unterprogramm DISPOF programmiert, so erfolgt keine Satzanzeige. Der Bediener muss im Einzelsatz bis zum Ende des Unterprogrammes blind ständig Start drücken.

## 8.12 Einzelsatz

2. Nach dem Ausschalten von Einzelsatz erfolgt kein Stoppen am Satzende.
3. Bei Anzeige von STOPRE-Sätzen ist der Hauptlauf und der Vorlauf im Decodiereinzelsatz synchron.

Folgende Unterkapitel beschreiben Beeinflussen des Einzelsatzverhaltens und das Unterdrücken des Anhaltens in bestimmten Situationen.

### 8.12.1 Decodier-Einzelsatz SBL2 mit impliziten Vorlaufstop

#### Asynchronität

Durch die Vorausbearbeitung der Teileprogrammsätze kann der Bezug zwischen aktueller Satzanzeige, bezogen auf den Hauptlaufzustand des NCK's und der auf HMI angezeigten Variablenwerte verloren gehen. Dem Anwender werden dann nicht plausible Variablenwerte angezeigt.

#### Vorlaufstop je Satz

Mit den kanalspezifischen Settingdatum SD42200 \$SC\_SINGLEBLOCK2\_STOPRE (Debugmode für SBL2 aktivieren) wird bei aktivem SBL2 mit jedem Satz ein Vorlaufstopp ausgeführt. Dadurch wird die Vorausbearbeitung der Teileprogrammsätze unterdrückt und der Bezug zwischen aktueller Satzanzeige und Anzeige der Variablenwerte bleibt erhalten.

---

#### Hinweis

Diese Variante des SBL2 ist nicht konturtreu. Das bedeutet, dass bedingt durch den Vorlaufstopp möglicherweise ein anderer Konturverlauf generiert wird als ohne Einzelsatz oder mit SBL1.

Anwendung: Debug-Mode zum Austesten von Teileprogrammen.

---

### 8.12.2 Einzelsatzstopp: Unterdrückung über SBLOF

#### Single Block Aus

Mit dem Sprachbefehl `SBLOF` gekennzeichnete Programme werden bei jedem Einzelsatztyp wie ein Satz komplett abgearbeitet.

`SBLOF` gilt auch in den aufgerufenen Unterprogrammen.

## SBLOF

Beispiel für Unterprogramm ohne Stopp im Einzelsatz:

```

PROC BEISPIEL SBLOF
G1 X10
RET

```

Ob am Ende des Unterprogramms angehalten werden soll oder nicht, wird mit dem Rücksprung-Befehl entschieden:

Rücksprung mit M17

Stopp am Ende des Unterprogramms

Rücksprung mit RET

Kein Stopp am Ende des Unterprogramms

## SBLOF im Programm

**SBLOF** muss alleine im Satz stehen. Ab diesem Satz wird Einzelsatz-Stopp ausgeschaltet bis zum nächsten programmierten **SBLON** oder zum Ende der aktiven Unterprogrammebene.

Ist **SBLOF** aktiv, so gilt diese Festlegung auch in den aufgerufenen Unterprogrammen. **SBLON**

**Beispiel** für einen Bereich im Einzelsatzbetrieb

Der Bereich zwischen **N20** und **N60** wird im Einzelsatzbetrieb als ein Schritt bearbeitet.

```

N10 G1 X100 F1000
N20 SBLOF ; Einzelsatz ausschalten
N30 Y20
N40 M100
N50 R10=90
N60 SBLON ; Einzelsatz wieder einschalten
N70 M110
N80 ...

```

## Asynchrone Unterprogramme

Die systemintern bei REORG/REPOS gestarteten asynchronen Unterprogramme **ASUP1.SYF** und **ASUP2.SYF** können durch Programmierung von **SBLOF** das System-ASUP in einem Schritt abarbeiten.

**Beispiel: ASUP.SPF:**

```

N10 SBLOF
N20 IF $AC_ASUP == 'H200'
N30 RET ; kein REPOS bei BA-Wechsel
N40 ELSE
N50 REPOSA ; REPOS in allen übrigen Fällen
N60 ENDIF
N70 RET

```

### Randbedingungen

- Die aktuelle Satzanzeige kann in Zyklen unterdrückt werden mit `DISPLOF`.
- Wird `DISPLOF` zusammen mit `SBLOF` programmiert, so wird bei Einzelsatzstopps innerhalb des Zyklus nach wie vor der Zyklen-Aufruf angezeigt.
- Die durch das MD20117 `MC_IGNORE_SINGLEBLOCK_ASUP` (Interruptprogramm trotz Einzelsatz komplett abarbeiten) getroffene Voreinstellung für das Verhalten von asynchronen Unterprogrammen bei Einzelsatz kann programmspezifisch durch die Programmierung von `SBLOF` überschrieben werden.

### Zyklus

**Beispiel1:** Zyklus soll für den Anwender wie ein Befehl wirken.

Hauptprogramm:

```
N10 G1 X10 G90 F200  
N20 X-4 Y6  
N30 CYCLE1  
N40 G1 X0  
N50 M30
```

Programm cycle:1

```
N100 PROC CYCLE1 DISPLOF SBLOF ; Einzelsatz unterdrücken  
N110 R10=3*SIN(R20)+5  
N120 IF (R11 <= 0)  
N130 SETAL(61000)  
N140 ENDIF  
N150 G1 G91 Z=R10 F=R11  
N160 M17
```

Der Zyklus `CYCLE1` wird bei aktivem Einzelsatz abgearbeitet. D. h. es muss für die Bearbeitung von `CYCLE1` einmal die Start-Taste gedrückt werden.

**Beispiel 2:** Ein vom PLC gestartetes ASUP zum Aktivieren von geänderten Nullpunktverschiebung und Werkzeugkorrekturen soll nicht sichtbar sein.

```
N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF 0 GOTOF
_G500 1 GOTOF _G54 2 GOTOF _G55 3 GOTOF _G56 4
GOTOF _G57 DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET
```

### 8.12.3 Einzelsatzstopp: situationsabhängig verhindern

#### Anhalten fallweise unterdrücken

In Abhängigkeit des

MD10702 \$MN\_IGNORE\_SINGLEBLOCK\_MASK (Einzelsatzstopp verhindern)

kann durch Setzen der Bits 0 bis 12 = 1 das Anhalten am Satzende während den folgenden Bearbeitungsvorgängen unterdrückt werden.

Trotz gesetzter Einzelsatzbearbeitung soll nicht satzweise angehalten werden bei:

1. Während eines internen ASUPs
2. Während eines Anwender-ASUPs
3. Von Unterprogrammen mit dem Attribut DISPLOF
4. Von Zwischensätzen
5. Von Satzsuchlaufsammelsätzen
6. Von Init-Sätzen
7. Von Sätzen, die nicht reorganisierbar sind
8. Von Sätzen, die nicht repositionierbar sind
9. Eines Wiederanfahrtsatzes, der keine Verfahrinformation enthält
10. Einem Vorlauf/Hauptlauf Synchronisationssatz, aufgrund von REORG
11. An einem Werkzeuganwahlsatz
12. An einem GET-Satz
13. Während eines Einzelsatztyp 2

## Ablauf

Wird z.B. während des Einzelsatzes ein ASUP aktiviert, so erfolgt die ASUP-Bearbeitung geschlossen. Die Bremsbewegung erfolgt erst wieder nach dem ASUP-Ende bzw. im ersten IPO-Satz, in dem keine Einzelsatz-Unterdrückung aktiviert ist. Ist die Geschwindigkeit hier zu groß, als dass in diesem Satz gebremst werden kann (bei aktivem Bahnsteuerbetrieb <sup>G64</sup>), werden noch weitere Satzwechsel erlaubt.

Für Dekodiereinzelsatz wirkt das MD10702 nur bei "internes ASUP", "Anwender-ASUP" und "Unterprogramme mit dem Attribut DISPLOF". In diesen Fällen steht bereits zum Interpretationszeitpunkt fest, dass der Satz ein Satz des oben genannten Typs ist. In diesen Fällen können weitere Sätze erzeugt werden.

### SBLON im ASUP

Der mit MD10702 \$MN\_IGNORE\_SINGLEBLOCK\_MASK unterdrückte Einzelsatz-Stopp eines internen-ASUPs oder Anwender-ASUPs kann durch Programmierung von SBLON im ASUP wieder aktiviert werden.

Diese Funktionalität kann mit MD20117 \$MC\_IGNORE\_SINGLEBLOCK\_ASUP wiederum unterdrückt werden. Der Befehl SBLON wird damit unwirksam.

## Randbedingungen

Für Dekodiereinzelsatz SBL2 gibt es folgende Einschränkung:

- Satzsuchlauf-Anfahrtsätze
- Satz steht nicht in einem ASUP; DISPLOF, SBLOF
- Nicht reorganisierbare Sätze oder nicht repositionierbare Sätze
- Sätze, die nicht im Interpreter erzeugt werden, z. B. Zwischensätze

## 8.12.4 Einzelsatzverhalten in BAG mit Typ A/B

### Kanäle klassifizieren

Ein Kanal der BAG muss als Einzelsatz-Steuerkanal (KS), alle übrigen Kanäle der BAG müssen als abhängige Kanäle (KA) über Nahtstellensignal klassifiziert werden. Für die Kanäle KA kann das Einzelsatz-Verhalten bezüglich des Typ A oder Typ B gewählt werden.

Typ A legt Stopp (vergleichbar mit Stop-Taste) fest.

Typ B legt Stopp (vergleichbar mit NC-Stop an Satzgrenze) fest.

### Kanal Klassifizierung

Der Benutzer sollte in einer BAG in genau **einen** Kanal (**KS**) Einzelsatz (NST DB21 ... DBX0.4 (Einzelsatz aktivieren)) anwählen. Der Einzelsatz-Typ A oder B bezieht sich auf die **anderen** Kanäle (KA) der BAG.

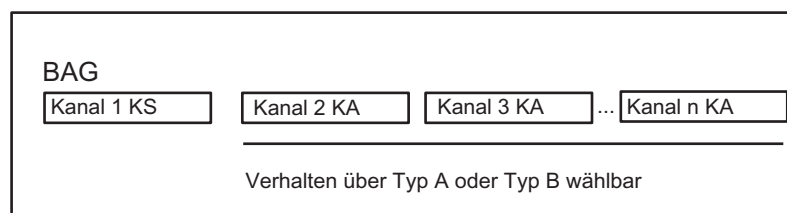


Bild 8-13 Kanal-Klassifizierung für Einzelsatz in der BAG 1

#### Typ A, NST DB11, ... DBX1.7=1 (Einzelsatz Typ A)

- Alle Kanäle sind gestoppt.
- Alle Kanäle erhalten einen Start (Start-Taste).
- Der Kanal KS stoppt am Satzende (wegen Einzelsatz)
- Die Kanäle KA erhalten eine **STOP**. (Vergleichbar mit der Stop-Taste).
- Alle Kanäle sind gestoppt. (Bremsphase von allen KAs)

#### Typ B, NST DB11, ... DBX1.6=1 (Einzelsatz Typ B)

- Alle Kanäle sind gestoppt.
- Alle Kanäle erhalten einen Start.
- Der Kanal KS stoppt am Satzende
- Die Kanäle KA erhalten eine **STOPATEND**.  
(Vergleichbar mit NST DB21, ... DBX7.2 (NC-Stop an Satzgrenze)).
- Alle Kanäle sind (irgendwann) an einer Satzgrenze gestoppt.

## 8.13 Programmbeeinflussung

### Möglichkeiten

1. Funktionenwahl über Bedienoberfläche oder über PLC
2. Aktivierung von Ausblendeebenen
3. Größenanpassung des Interpolationspuffers
4. Darstellungsweise der Programmanzeige über eine zusätzliche Basis-Satzanzeige
5. Abarbeiten von extern (Puffergröße und -anzahl)
6. Abarbeiten von externen Unterprogrammen

### 8.13.1 Funktionenwahl über Bedientafelfront oder über PLC

#### Bedienoberfläche oder PLC

Die Abarbeitungsart des Teileprogrammes kann der Anwender über die Bedientafelfront bzw. über die PLC beeinflussen.

#### Anwahl, Aktivierung, Rückmeldung

##### Anwahl

Unter dem Softkey Programmbeeinflussungen können bestimmte Funktionen in der Bedienoberfläche angewählt werden, wobei sich die Auswahl auf ein Nahtstellensignal der PLC auswirkt. Diese Nahtstellensignale sind als Auswahlssignale von der Bedienoberfläche zu verstehen und aktivieren die angewählte Funktion noch nicht.

##### Aktivierung

Damit die angewählten Funktionen wirksam werden, müssen diese Signalzustände auf einen anderen Bereich des Datenbausteins übertragen werden. Bei einer Beeinflussung seitens der PLC müssen diese Signale direkt gesetzt werden.

##### Rückmeldung

Die aktivierten Funktionen werden teilweise vom NCK an die PLC zurückgemeldet (siehe auch Kapitel "Z1: NC/PLC-Nahtstellensignale (Seite 1791)").



Tabelle 8-4 Programmbeeinflussung: Nahtstellen-Signale

Funktion	Anwahlsignal	Aktivierungssignal	Rückmeldesignal
SKP Ausblendsatz 0 bis 7	DB21, ... DBX26.0 - 26.7	DB21, ... DBX2.0 - 2.7	---
SKP Ausblendsatz 8 bis 9	DB21, ... DBX27.0 - 27.1	DB21, ... DBX31.6 - 31.7	---
DRY Probelauf Vorschub	DB21, ... DBX24.6	DB21, ... DBX0.6	DB21, ... DBX318.6
ROV Korrektur Eilgang	DB21, ... DBX25.3	DB21, ... DBX6.6	---
Einzelsatz:	Vorwahl von SBL1, SBL2 oder SBL3 über Programmbeeinflussungsanzeige von HMI		
SBL1: Aktions-Einzelsatz	Bedientafel HMI	DB21, ... DBX0.4	---
SBL2: Dekodier-Einzelsatz			
SBL3: Im Zyklus			
M01 (Programmierter Halt)	DB21, ... DBX24.5	DB21, ... DBX0.5	DB21, ... DBX32.5
Assoziiertes M01	DB21, ... DBX24.4	DB21, ... DBX30.5	DB21, ... DBX318.5
DRF-Anwahl	DB21, ... DBX24.3	DB21, ... DBX0.3	DB21, ... DBX33.3
PRT Programmtest	DB21, ... DBX25.7	DB21, ... DBX1.7	DB21, ... DBX33.7

**Literatur:**

Bedienungshandbuch HMI Advanced "Bedienbereich Maschine"

**8.13.2 Aktivierung von Ausblendeebenen****Funktion**

Sätze, die nicht bei jedem Programmlauf ausgeführt werden sollen, können ausgeblendet werden. Die Sätze, die ausgeblendet werden sollen, werden mit dem Zeichen "/" (Schrägstrich) vor der Satznummer gekennzeichnet.

Im Teileprogramm werden die Ausblendeebenen mit "/0" bis "/9" angegeben.

Pro Teileprogrammssatz kann nur eine Ausblendeebene angegeben werden.

**Parametrierung**

Die Anzahl der Ausblendeebenen wird festgelegt mit dem Maschinendatum:

MD51029 \$MM\_MAX\_SKP\_LEVEL (Max. Anzahl der Ausblendeebenen im NC-Programm)

## Programmierung

Sätze, die nicht bei jedem Programmlauf ausgeführt werden sollen (z. B. Programm-Einfahrsätze), können nach folgendem Schema ausgeblendet werden:

Programmcode	Kommentar
/N005	; Satz ausgeblendet, (DB21, ... DBX2.0) 1.Ausblendeebene
/0 N005	; Satz ausgeblendet, (DB21, ... DBX2.0) 1.Ausblendeebene
/1 N010	; Satz ausgeblendet, (DB21, ... DBX2.1) 2. Ausblendeebene
/2 N020	; Satz ausgeblendet, (DB21, ... DBX2.2) 3.Ausblendeebene
/3 N030	; Satz ausgeblendet, (DB21, ... DBX2.3) 4.Ausblendeebene
/4 N040	; Satz ausgeblendet, (DB21, ... DBX2.4) 5.Ausblendeebene
/5 N050	; Satz ausgeblendet, (DB21, ... DBX2.5) 6.Ausblendeebene
/6 N060	; Satz ausgeblendet, (DB21, ... DBX2.6), 7.Ausblendeebene
/7 N070	; Satz ausgeblendet, (DB21, ... DBX2.7) 8.Ausblendeebene
/8 N080	; Satz ausgeblendet, (DB21, ... DBX31.6) 9.Ausblendeebene
/9 N090	; Satz ausgeblendet, (DB21, ... DBX31.7) 10.Ausblendeebene

## Aktivierung

Die 10 Ausblendeebenen "/0" bis "/9" werden vom PLC durch Setzen der Nahtstellensignale PLC → NCK aktiviert.

Die Aktivierung der Funktion von HMI über das Menü "Programmbeeinflussung" im Bedienbereich "Maschine" erfolgt:

- Für die Ausblendeebenen "/0" bis "/7":  
über die Schnittstelle HMI → PLC DB21, ... DBB26 (Satzausblenden ausgewählt).
- Für die Ausblendeebenen "/8" bis "/9":  
über die Schnittstelle HMI → PLC DB21, ... DBX27.0 bis DBX27.1.

### Literatur:

Bedienhandbuch

---

### Hinweis

Änderungen der auszublenden Ebenen sind nur im STOP/RESET-Zustand der Steuerung möglich.

---

### 8.13.3 Größenanpassung des Interpolationspuffers

#### MD28060

Der kanalspezifische Interpolator arbeitet beim Teileprogrammablauf vorbereitete Sätze aus dem Interpolationspuffer ab. Die maximale Anzahl Sätze, die zu einem Zeitpunkt im Interpolationspuffer Platz finden sollen, wird durch das speicherkonfigurierende MD28060 \$MM\_IPO\_BUFFER\_SIZE (Anzahl der NC-Sätze im IPO-Puffer(DRAM)) festgelegt. Für einige Anwendungen kann es sinnvoll sein, diesen Puffer nicht voll auszunutzen, um den "Abstand" zwischen Vorbereitung und Interpolation klein zu halten.

#### SD42990

Mit dem Settingdatum SD42990 \$SC\_MAX\_BLOCKS\_IN\_IPOBUFFER (Max. Anzahl Sätze im IPO-Puffer) kann dynamisch die Zahl der Sätze im Interpolationspuffer auf einen **kleineren** Wert als in MD28060 \$MC\_MM\_IPO\_BUFFER\_SIZE (Anzahl der NC-Sätze im IPO-Puffer (DRAM)) begrenzt werden, minimal auf 2 Sätze.

**Werte des Settingdatums SD42990 \$SC\_MAX\_BLOCKS\_IN\_IPOBUFFER:**

Wert	Wirkung
< 0	Keine Begrenzung des Interpolationspuffers aktiv. Der max. mögliche IPO-Puffer entsprechend MD 28060: MM_IPO_BUFFER_SIZE wird aktiviert.
oder 1	Der minimal zulässige IPO-Puffer mit 2 Sätzen wird aktiv.
< MM_IPO_BUFFER_SIZE	Der IPO-Puffer wird maximal mit der angegebenen Anzahl Sätze aktiviert.
>= MM_IPO_BUFFER_SIZE	Der IPO-Puffer wird mit der in MD 28060: MM_IPO_BUFFER_SIZE angegebenen Anzahl Sätze aktiviert.

#### Hinweis

Wenn das SD42990 \$SC\_MAX\_BLOCKS\_IN\_IPOBUFFER im Teileprogramm gesetzt wird, wird die Begrenzung des Interpolationspuffers sofort wirksam, wenn der Satz mit dem SD vom Interpreter in der Vorbereitung abgearbeitet wird.

D.h. gegebenenfalls wird die Begrenzung des IPO-Puffers einige Sätze vor der beabsichtigten Begrenzung wirksam (siehe auch das MD28070 \$MC\_MM\_NUM\_BLOCKS\_IN\_PREP).

Will man das vermeiden, und die Begrenzung des IPO-Puffers soll satzsynchron wirksam werden, muss man vor dem Setzen des SD im Teileprogramm ein STOPRE (Vorlaufstopp) programmieren.

### Gültigkeit

Das SD42990 \$SC\_MAX\_BLOCK\_IN\_IPOBUFFER hat globale, kanalspezifische Gültigkeit und kann auch in einem Teileprogramm verändert werden. Dieser veränderte Wert wird bei Programmende beibehalten. Soll dieses Settingdatum bei definierten Ereignissen wieder zurückgesetzt werden, muss dafür ein so genanntes ereignisgesteuertes Programm eingerichtet werden. Man könnte z. B. dieses Settingdatum bei RESET immer auf einen vordefinierten Wert setzen.

### Anwendung

Die IPO-Pufferbegrenzung kann eingesetzt werden, wenn die Anzahl Sätze zwischen Satzaufbereitung und Interpolation möglichst klein sein soll, z. B. wenn im Teileprogramm Istpositionen gelesen und weiterverarbeitet werden sollen.

### Beispiel

```
N10 ...
N20 ...
.....
N100 $SC_MAX_BLOCKS_IN_IPOBUFFER = 5           ; Begrenzung des IPO-Buffers auf 5
                                                NC-Sätze
N110 ...
N120 ...
.....
N200 $SC_MAX_BLOCKS_IN_IPOBUFFER = -1         ; Aufhebung der Begrenzung des IPO-
                                                Puffers
N210 ...
.....
```

## 8.13.4 Darstellungsweise der Programmanzeige über eine zusätzliche Basis-Satzanzeige

### Basis-Satzanzeige (nur bei ShopMill/ShopTurn)

Zur bestehenden Satzanzeige können über eine zweite, der so genannten Basis-Satzanzeige, alle Sätze angezeigt werden, die eine **Aktion an der Maschine** bewirken werden.

### Vorausschauende Basis-Satzanzeige

Die tatsächlich angefahrenen Endpositionen werden als Absolutposition dargestellt. Die Positionswerte beziehen sich wahlweise auf das Werkstückkoordinatensystem (WKS) oder auf das Einstellbare Nullpunkt-System (ENS).

Die Anzahl der vorausschauend im Anzeigebuffer abgelegten Anzeigesätze hängt von der Anzahl der im jeweiligen Bearbeitungszustand aufbereiteten Sätze im Vorlauf des NCK's ab. Wird ein Vorlaufstopp bearbeitet, so geht die Anzahl der Anzeigesätze gegen Null und steigt nach der Quittierung des Vorlaufstopps wieder an. Bei REORG-Ereignissen (z. B. Betriebsartenwechsel, ASUP-Start) werden die vorausschauend angelegten Anzeigesätze gelöscht und danach wieder aufbereitet.

#### Aufbereitete Werte

In der Basis-Satzanzeige aufbereitete Werte stimmen überein mit:

- Angewählten Werkzeugen
- Vorschub und der Spindeldrehzahl
- Tatsächlich angefahrenen Positionswerten

Ausnahmen:

Bei aktiver Werkzeugradiuskorrektur können Abweichungen auftreten.

Bei Modulo-Achsen wird in der Basis-Satzanzeige der programmierte Wert angezeigt, der auch außerhalb des Modulo-Bereichs liegen kann.

---

#### Hinweis

Grundsätzlich werden die Positionen im WKS oder im ENS dargestellt.

Die Funktion der Basis-Satzanzeige kann mit dem Settingdatum

SD42750 \$SC\_ABSBLOCK\_ENABLE (Basissatzanzeige freigeben) ein- oder ausgeschaltet werden.

---

### 8.13.5 Basis-Satzanzeige bei ShopMill/ShopTurn

#### Basis-Satzanzeige konfigurieren

Die Basis-Satzanzeige ist über folgende Maschinendaten konfigurierbar:

NCK Maschinendaten für Basis-Satzanzeige	Bedeutung:
MD28400 \$MC_MM_ABSBLOCK	Basis-Satzanzeige aktivieren
MD28402 \$MC_MM_ABSBLOCK_BUFFER_CONF[2]	Größe des Anzeigepuffers
Anzeige Maschinendaten	einzustellende Positionswerte:
MD9004 \$MM_DISPLAY_RESOLUTION	für metrische Maßangabe
MD9011 \$MM_DISPLAY_RESOLUTION_INCH	für Inch Maßangabe
MD9010 \$MM_SPIND_DISPLAY_RESOLUTION	für Anzeigefeinheit Spindeln einzustellendes Koordinatensystem
MD9424 \$MM_MA_COORDINATE_SYSTEM	für Istwertanzeige WKS oder ENS

Diese Anzeigemaschinendaten werden in die NCK-Maschinendaten MD17200 \$MN\_GMMC\_INFO\_UNIT[0] (Globale HMI-Info) bis MD17200 \$MN\_GMMC\_INFO\_UNIT[3] kopiert. Damit kann vom NCK aus auf diese Anzeigemaschinendaten zugegriffen werden.

#### Aktivierung

Die Basis-Satzanzeige wird aktiviert durch das MD28400 \$MC\_MM\_ABSBLOCK (Satzanzeige mit Absolutwerten aktivieren) mittels Power On. Wird das MD28400 \$MC\_MM\_ABSBLOCK mit 1 beschrieben, so wird im Hochlauf ein kanalspezifischer Anzeigepuffer (FIFO) eingerichtet.

**Größe des Anzeigepuffer (FIFO)** = (MD28060 \$MC\_MM\_IPO\_BUFFER\_SIZE (Anzahl der NC-Sätze im IPO-Puffer) + MD28070 \$MC\_MM\_NUM\_BLOCKS\_IN\_PREP (Anzahl der Sätze für die Satzaufbereitung)) multipliziert mit 128 Byte. Bei Standard-Maschinendateneinstellung entspricht dies einer Größe von 6 KByte.

Größe des Anzeigepuffers optimieren:  
Der Speicherbedarf kann durch Angabe eines Wertes zwischen 128 und 512 optimiert werden. Die im Anzeigepuffer aufbereiteten Anzeigesätze werden über einen konfigurierbaren Upload-Puffer zum HMI transportiert.

**Die maximale Größe des Upload-Puffers** ergibt sich durch Multiplikation von (MD28402 \$MC\_MM\_ABSBLOCK\_BUFFER\_CONF[0] + MD28402 \$MC\_MM\_ABSBLOCK\_BUFFER\_CONF[1] + 1) mit der durch MD28400 \$MC\_MM\_ABSBLOCK projektierten Satzlänge.

Konfiguriert werden hier die Anzahl der Sätze **vor** dem aktuellen Satz im MD28402 \$MC\_MM\_ABSBLOCK\_BUFFER\_CONF[0] und **nach** dem aktuellen Satz im MD28402 \$MC\_MM\_ABSBLOCK\_BUFFER\_CONF[1].

### Randbedingungen

Bei Überschreiten der in MD28400 \$MC\_MM\_ABSBLOCK projizierten Länge eines Anzeigesatzes wird dieser Anzeigesatz entsprechend abgeschnitten. Um dies darzustellen wird am Satzende der String "..." angefügt.

Für vorübersetzte Zyklen (MD10700 \$MN\_PREPROCESSING\_LEVEL > 1 (Programmvorverarbeitungsstufe)) enthält der Anzeigesatz **nur** Achspositionen.

Weitere Randbedingungen für die Basis-Satzanzeige:

- Modale Synchronaktionssätze mit Absolutwerten werden nicht berücksichtigt.
- Die Basis-Satzanzeige ist während Satzsuchlauf mit oder ohne Berechnung deaktiviert.
- Polarkoordinatenprogrammierung wird nicht kartesisch dargestellt.

### Radius / Durchmesser-Werte

Auf der Basis-Satzanzeige und der Positionsanzeige dargestellte Durchmesserwerte können zur internen Berechnung als Radius benötigt werden. Es bestehen folgende Möglichkeiten diese Werte für Maßangaben in Radius/Durchmesser gemäß G-Code-Gruppe 29 zu beeinflussen:

- G-Code DIAMCYCOF (Erweiterung kanalspezifische Durchmesserprogrammierung)  
Dieser G-Code schaltet die kanalspezifische Durchmesserprogrammierung während der Zyklusbearbeitung aus. Im Zyklus können somit Berechnungen immer im Radius erfolgen. Die Positionsanzeige und die Basis-Satzanzeige erfolgt weiterhin entsprechend dem Zustand der Durchmesserprogrammierung vor DIAMCYCOF.  
In der Basis-Satzanzeige bleibt der zuletzt angezeigte Wert erhalten.
- G-Code DIACYCOFA[AX] (achs-spezifische Durchmesserprogrammierung)  
Dieser G-Code schaltet die achs-spezifische Durchmesserprogrammierung während der Zyklusbearbeitung aus. Im Zyklus können somit Berechnungen immer im Radius erfolgen. In der Positionsanzeige und in der Basis-Satzanzeige erfolgt weiterhin entsprechend dem Zustand bezüglich Durchmesserprogrammierung vor DIACYCOFA[AX].  
In der Basis-Satzanzeige bleibt der zuletzt angezeigte Wert erhalten.
- MD27100 \$MC\_ABSBLOCK\_FUNCTION\_MASK (Satzanzeige mit Absolutwerten parametrieren)

Bit0 = 1                      Sollwerte der Planachse werden in der Basis-Satzanzeige grundsätzlich als Durchmesserwerte angezeigt.

### Verhalten bei aktivem Kompressor

Bei aktivem Kompressor mit G-Code-Gruppe 30 ungleich COMPOF werden zwei Anzeigesätze generiert. Der

- Erste enthält den G-Code des aktiven Kompressors.
- Zweite enthält den String "..." als Zeichen dafür, dass Anzeigesätze fehlen.

Beispiel:

```
G0 X10 Y10 Z10      ; Satz der noch für die Basis-Satzanzeige aufbereitet wird
COMPCAD             ; Kompressor für optimierte Oberflächengüte (CAD-Prog.)
                    ; ein
...                 ; String als Zeichen dafür, das Anzeigesätze fehlen
```

Um Engpässe der NCK-Leistung zu vermeiden, wird die Basis-Satzanzeige automatisch abgeschaltet. Als Zeichen dafür, das Anzeigesätze fehlen, wird ein Anzeigesatz mit dem String "..." generiert.

Im Einzelsatz werden immer alle Anzeigesätze erzeugt.

### 8.13.6 Aufbau für einen DIN-Satz

#### Aufbau des Anzeigesatzes für einen DIN-Satz

Prinzipieller Aufbau des Anzeigesatzes für einen DIN-Satz

- Satznummer/Label
- G-Funktion der ersten G-Gruppe  
(nur bei Änderung gegenüber dem letzten Maschinenfunktionssatz).
- Achspositionen  
(Reihenfolge entsprechend MD20070 \$MC\_AXCONF\_MACHAX\_USED  
(Maschinenachsnummer gültig im Kanal)).
- Weitere modale G-Funktionen  
(nur bei Änderungen gegenüber den letzten Maschinenfunktionssatz).
- Weitere Adressen wie programmiert.

Der Anzeigesatz für die Basis-Satzanzeige wird direkt aus dem programmierten Teileprogrammssätzen nach folgenden Regeln abgeleitet:

- Makros werden expandiert.
- Ausblendkennungen und Kommentare entfallen.
- Satznummer und Labels werden aus dem Originalsatz übernommen, entfallen jedoch wenn DISPLOF aktiv ist.
- Die Anzahl der Nachkommastellen wird durch die Anzeigemaschinendaten MD9004, MD9010 und MD9011 über HMI festgelegt.



HMI-Anzeigemaschinendatum	Zugriff im NCK Maschinendatum
MD9004 \$MM_DISPLAY_RESOLUTION	MD17200 \$MN_GMMC_INFO_NO_UNIT[0]
MD9011 \$MM_DISPLAY_RESOLUTION_INCH	MD17200 \$MN_GMMC_INFO_NO_UNIT[1]
MD9010 \$MM_SPIND_DISPLAY_RESOLUTION	MD17200 \$MN_GMMC_INFO_NO_UNIT[2]
MD9424 \$MM_MA_COORDINATE_SYSTEM	MD17200 \$MN_GMMC_INFO_NO_UNIT[3]

- Programmierte Achs-Positionen werden als absolute Positionen in dem durch das MD9424 \$MM\_MA\_COORDINATE\_SYSTEM (Koordinatensystem für Istwertanzeige) vorgegebenen Koordinatensystem (WKS / ENS) dargestellt.

#### Hinweis

Bei Modulo-Achsen entfällt die Modulo-Korrektur. Dadurch sind Positionen außerhalb des Modulobereichs möglich und es ergibt sich zwangsläufig eine Differenz zur Positionsanzeige in der grundsätzlich modulo-gewandelt wird.

## Beispiele

Gegenüberstellungen von Anzeigesatz (Originalsatz) zur Basis-Satzanzeige:

- **Programmierte Positionen** werden absolut dargestellt.  
Die Adressen AP/RP werden mit ihren programmierten Werten dargestellt.

Originalsatz:	Anzeigesatz:
N10 G90 X10.123	N10 X10.123
N20 G91 X1	N20 X11.123

- **Zuweisungen von Adressen** (nicht DIN-Adressen) werden in der Form <adresse> = <konstante> dargestellt.

Originalsatz:	Anzeigesatz:
N110 R1 = -67.5 R2 = 7.5	
N130 Z = R1 RND = R2	N130 Z-67.5 RND = 7.5

- **Indizes von Adressen** (Adresserweiterungen) werden als Konstanten <adresse> [ <konstante> ] = <konstante> dargestellt.

Originalsatz:	Anzeigesatz:
N220 DEF AXIS AXIS_VAR = X	
N240 FA[ AXIS_VAR ] = R2	N240 FA[X] = 1000

- **DIN-Adressen ohne Adresserweiterung** werden in der Form <din\_adresse> <konstante> dargestellt.

Originalsatz:	Anzeigesatz:
N410 DEF REAL FEED = 1.5	
N420 F = FEED	N420 F1.5

Für **H-Funktionen** gilt: Unabhängig von der Ausgabeart zur PLC (MD22110 \$MC\_AUXFU\_H\_TYPE\_INT (Typ von H-Hilfsfunktionen ist integer)) wird der jeweils programmierte Wert angezeigt.

- Für die **Werkzeug-Anwahl über T-Befehl** wird eine Anzeigeinformation der Form T<wert> bzw. T=<string> generiert. Wurde eine Adresserweiterung programmiert, so wird diese auch aufgelöst.  
 Wurden mehrere Spindeln projektiert oder ist die Funktion "Werkzeugwechsel über Werkzeughalter" (MD20124 \$MC\_TOOL\_MANAGEMENT\_TOOLHOLDER (Werkzeughalternummer)) aktiv, so wird die T-Nummer grundsätzlich mit Adresserweiterung ausgegeben.  
 Wurde keine Adresserweiterung programmiert, wird die Nummer der Masterspindel bzw. des Mastertoolholders eingesetzt (T<spindel\_nummer/tool\_holder> = ).
- Für die **Spindelprogrammierung** über S, M3, M4, M5, M19, M40 - M45 und M70 (bzw. MD 20094 \$MC\_SPIND\_RIGID\_TAPPING\_M\_NR (M-Funktion für das Umschalten in den gesteuerten Achsbetrieben)) gilt bezüglich Adresserweiterung folgende Regelung: Wurde eine Adresserweiterung programmiert, so wird diese auch aufgelöst.  
 Wurden mehrere Spindeln projektiert, so wird grundsätzlich die Adresserweiterung mit ausgegeben.  
 Wurde keine Adresserweiterung programmiert, wird die Nummer der Masterspindel eingesetzt (S<spindel\_nummer>=).
- **Indirekte G-Codeprogrammierung** der Form G[ <gruppe> ] = <ausdruck> wird durch den entsprechenden G-Code ersetzt.

Originalsatz:	Anzeigesatz:
N510 R1=2	
N520 G[8]= R1	N520 G54

- **Modale G-Codes**, die keinen ausführbaren Satz erzeugen werden aufgesammelt und mit dem Anzeigesatz des nächsten ausführbaren Satz angezeigt, sofern dies von der Syntax her erlaubt ist (DIN-Satz). Ist dies nicht der Fall (z. B. vordefinierter Unterprogrammaufruf TRANSMIT) so wird ein eigener Anzeigesatz mit den geänderten G-Codes vorangestellt.

Originalsatz:	Anzeigesatz:
N610 G64	G64
N620 TRANSMIT	N620 TRANSMIT

- Für **Teileprogrammzeilen** in denen die Adressen **F** und **FA** vorkommen, wird immer ein Anzeigesatz generiert (auch bei MD22240 \$MC\_AUXFU\_F\_SYNC\_TYPE = 3 (Ausgabezeitpunkt der F-Funktionen)).

Originalsatz:	Anzeigesatz:
N630 F1000	N630 F1000
N640 X100	N640 X100

- Die für die **Satzanzeige generierten Anzeigesätze** werden **direkt** von den programmierten Teileprogrammsätzen abgeleitet. Werden durch die Konturaufbereitung Zwischensätze erzeugt (z.B. Werkzeugradiuskorrektur G41/G42, Radius/Fase RNDM, RND, CHF, CHR), so erhalten diese Sätze die Anzeigeinformation des der Bewegung zugrunde liegenden Teileprogrammsatzes.

Originalsatz:	Anzeigesatz:
N710 Y157.5 G42	N710 Y157.5 G42
N720 Z-67.5 RND=7.5	N720 Z-67.5 RND=7.5

- Beim **EXECTAB-Befehl** (Abarbeiten einer Tabelle von Konturelementen) wird im Anzeigesatz der durch EXECTAB generierte Satz angezeigt.

Originalsatz:	Anzeigesatz:
N810 EXECTAB (KTAB[5])	N810 G01 X46.147 Z-25.38

- Beim **EXECSTRING-Befehl** wird im Anzeigesatz der durch EXECSTRING generierte Satz angezeigt.

Originalsatz:
N910 DEF STRING[40] PROGSTRING = "N905 M3 S1000 G94 Z100 F1000 G55"
N920 EXECSTRING (PROGSTRING)

Originalsatz:
N905 Z100 G55 G94 M3 S1000 F1000

## 8.13.7 Abarbeiten von Extern

### Funktion

Mit der Funktion "Abarbeiten von Extern" können Programme, die wegen ihres Speicherbedarfs nicht mehr direkt im NC-Speicher ablegbar sind, von einem externen Programmspeicher abgearbeitet werden.

---

#### Hinweis

Geschützte Zyklen (\_CPF Dateien) können mit dieser Funktion nicht abgearbeitet werden.

---

### Externe Programmspeicher

Externe Programmspeicher können sich auf folgenden Datenträgern befinden:

- Lokales Laufwerk
- Netzlaufwerk
- USB-Laufwerk

---

#### Hinweis

##### Abarbeiten von Extern über USB-Schnittstelle

Sollen externe Programme von einem externen USB-Laufwerk über USB-Schnittstelle übertragen werden, so darf hierfür nur die Schnittstelle über X203 mit den Namen "TCU\_1" verwendet werden.

Ein USB-FlashDrive kann als persistentes Speichermedium nicht empfohlen werden.

---

### Anwendungen

- **Direktes Abarbeiten von externen Programmen**

Prinzipiell können alle Programme angewählt und abgearbeitet werden, die über die Verzeichnis-Struktur der Bedienoberfläche im HMI-Modus "Abarbeiten von Extern" erreichbar sind.

- **Abarbeiten von externen Unterprogrammen aus dem Teileprogramm**

Aufgerufen wird das "externe" Unterprogramm durch den Teileprogrammbefehl `EXTCALL` mit Angabe eines Aufrufpfades (optional) und des Unterprogrammbezeichners (→ siehe "Abarbeiten von externen Unterprogrammen (Seite 654)").

## Parametrierung

Für die Abarbeitung eines Programms im Modus "Abarbeiten von Extern" (Hauptprogramm oder Unterprogramm) muss im dynamischen NC-Speicher ein Nachladespeicher (FIFO-Puffer) reserviert werden.

### Größe des FIFO-Puffers

Die Größe des FIFO-Puffers wird eingestellt mit dem Maschinendatum:

MD18360 \$MN\_MM\_EXT\_PROG\_BUFFER\_SIZE (FIFO-Puffer-Größe für Abarbeiten von Extern)

Voreinstellung: 30 kByte

### Anzahl der FIFO-Puffer

Für alle Programme (Haupt- und Unterprogramme), die gleichzeitig im Modus "Abarbeiten von Extern" abgearbeitet werden, muss jeweils ein FIFO-Puffer zur Verfügung gestellt werden.

Die Anzahl der FIFO-Puffer wird eingestellt mit dem Maschinendatum:

MD18362 \$MN\_MM\_EXT\_PROG\_NUM (Anzahl der gleichzeitig von Extern abarbeitbaren Programmebenen)

## Verhalten bei RESET, POWER ON

Durch RESET und POWER ON werden externe Programmaufrufe abgebrochen und die jeweiligen FIFO-Puffer gelöscht.

Ein für "Abarbeiten von Extern" selektiertes Programm bleibt über RESET / Teileprogrammende weiterhin für "Abarbeiten von Extern" angewählt. Durch POWER ON geht die Anwahl verloren.

## 8.13.8 Abarbeiten von externen Unterprogrammen

### Funktion

Bei der Bearbeitung komplexer Werkstücke können sich für die einzelnen Bearbeitungsschritte Programmsequenzen ergeben, die wegen ihres Speicherbedarfs nicht mehr direkt im NC-Speicher ablegbar sind.

In solchen Fällen hat der Anwender die Möglichkeit, mit der Teileprogrammanweisung `EXTCALL` die Programmsequenzen als Unterprogramme von einem externen Programmspeicher im Modus "Abarbeiten von Extern" auszuführen.

### Voraussetzungen

Für das Abarbeiten externer Unterprogramme gelten folgende Voraussetzungen:

- Die Unterprogramme müssen über die Verzeichnis-Struktur der Bedienoberfläche erreichbar sein.
- Für jedes Unterprogramm muss im dynamischen NC-Speicher ein Nachladespeicher (FIFO-Puffer) reserviert sein.

### Parametrierung

Der Pfad zum externen Unterprogrammverzeichnis kann voreingestellt werden mit dem Settingdatum:

SD42700 \$SC\_EXT\_PROG\_PATH (Programmpfad für externen Unterprogrammaufruf EXTCALL)

Zusammen mit dem bei der Programmierung angegebenen Unterprogrammpfad bzw. -bezeichner ergibt sich daraus der Gesamtpfad des aufzurufenden Programms.

## Programmierung

Der Aufruf eines externen Unterprogramms erfolgt über den Teileprogrammbehehl `EXTCALL`.

**Syntax:** `EXTCALL("<Pfad/><Programmname>")`

### Parameter:

`<Pfad/>`: Absolute oder relative Pfadangabe (**optional**)  
Typ: STRING

`<Programmname>`: Der Programmname wird ohne Präfix "\_N\_" angegeben.  
Die Dateierweiterung ("MPF", "SPF") kann mit dem Zeichen "\_" oder "." am Programmnamen angefügt werden (**optional**).  
Typ: STRING

---

### Hinweis

#### Pfadangabe: Kurzbezeichnungen

Bei der Pfadangabe können folgende Kurzbezeichnungen verwendet werden:

- **LOCAL\_DRIVE**: für lokales Laufwerk
- **CF\_CARD**: für CompactFlash-Card
- **USB**: für USB Front-Anschluss

**CF\_CARD**: und **LOCAL\_DRIVE**: sind alternativ verwendbar.

---

#### EXTCALL-Aufruf mit absoluter Pfadangabe

Wenn das Unterprogramm unter dem angegebenen Pfad existiert, dann wird es nach dem `EXTCALL`-Aufruf ausgeführt. Wenn es nicht existiert, dann wird die Programmausführung abgebrochen.

### EXTCALL-Aufruf mit relativer Pfadangabe / ohne Pfadangabe

Bei einem EXTCALL-Aufruf mit relativer Pfadangabe bzw. ohne Pfadangabe werden die vorhandenen Programmspeicher nach folgendem Muster durchsucht:

- Wenn in SD42700 \$SC\_EXT\_PROG\_PATH eine Pfadangabe voreingestellt ist, dann wird zuerst ausgehend von diesem Pfad nach der Angabe im EXTCALL-Aufruf (Programmname ggf. mit relativer Pfadangabe) gesucht. Der absolute Pfad ergibt sich dann durch Zeichenverkettung aus:
  - der in SD42700 voreingestellten Pfadangabe
  - dem Zeichen "/" als Trennzeichen
  - dem bei EXTCALL angegebenen Unterprogrammpfad bzw. -bezeichner
- Wurde das aufgerufene Unterprogramm unter dem voreingestellten Pfad nicht gefunden, werden als nächstes die Verzeichnisse des Anwenderspeichers nach der Angabe im EXTCALL-Aufruf durchsucht.
- Die Suche endet, wenn das Unterprogramm erstmalig gefunden wurde. Sollte die Suche keinen Treffer ergeben, kommt es zum Programmabbruch.

### Beispiel

#### Abarbeiten von lokalem Laufwerk

Hauptprogramm:

```
Programmcode
-----
N010 PROC MAIN
N020 ...
N030 EXTCALL ("SCHRUPPEN")
N040 ...
N050 M30
```

Externes Unterprogramm:

```
Programmcode
-----
N010 PROC SCHRUPPEN
N020 G1 F1000
N030 X= ... Y= ... Z= ...
N040 ...
...
...
N999999 M17
```



Das Hauptprogramm "MAIN.MPF" befindet sich im NC-Speicher und ist zur Abarbeitung angewählt.

Das nachzuladende Unterprogramm "SCHRUPPEN.SPF" bzw. "SCHRUPPEN.MPF" befindet sich auf dem lokalen Laufwerk in dem Verzeichnis "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Der Pfad zu dem Unterprogramm ist im SD42700 voreingestellt:

SD42700 \$SC\_EXT\_PROG\_PATH = "LOCAL\_DRIVE:WKS.DIR/WST1.WPD"

---

#### **Hinweis**

Ohne Pfadangabe im SD42700 müsste die `EXTCALL`-Anweisung für dieses Beispiel wie folgt programmiert werden:

```
EXTCALL ("LOCAL_DRIVE:WKS.DIR/WST1.WPD/SCHRUPPEN")
```

---

## 8.14 Systemeinstellungen für Hochlauf, RESET/Teileprogrammende und Teileprogramm-Start

### Konzept

Das Verhalten der Steuerung kann über Maschinendaten für folgende Ereignissen eingestellt werden:

- Hochlauf (Power On)
- Reset / Teileprogrammende
- Teileprogrammstart

Verhalten der Steuerung nach:	Einstellbar mit:
Hochlauf (POWER ON) <sup>*)</sup>	MD20110 \$MC_RESET_MODE_MASK MD20144 \$MC_TRAFO_MODE_MASK MD20150 \$MC_GCODE_RESET_VALUES
RESET/Teileprogrammende	MD20110 \$MC_RESET_MODE_MASK MD20150 \$MC_GCODE_RESET_VALUES MD20152 \$MC_GCODE_RESET_MODE
Teileprogrammstart	MD20112 \$MC_START_MODE_MASK MD20110 \$MC_RESET_MODE_MASK

<sup>\*)</sup> siehe auch POWER ON (Seite 858)

### Systemeinstellungen nach Hochlauf

MD20110 \$MC\_RESET\_MODE\_MASK, Bit 0 = 0 oder 1

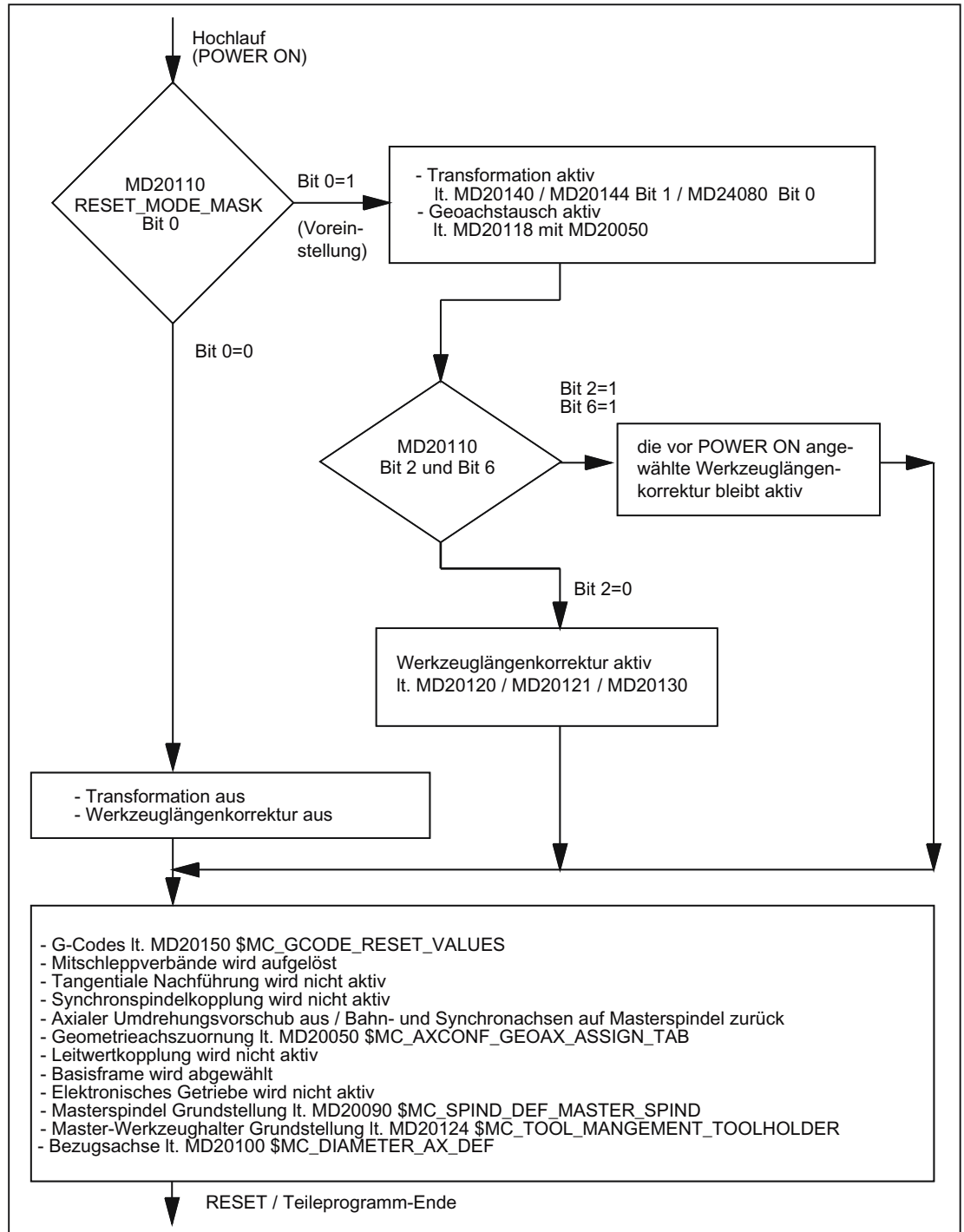


Bild 8-14 Systemeinstellungen nach Hochlauf

**Systemeinstellungen nach Reset / Teileprogramm-Ende und Teileprogramm-Start**

MD20110 \$MC\_RESET\_MODE\_MASK, Bit 0 = 0 oder 1

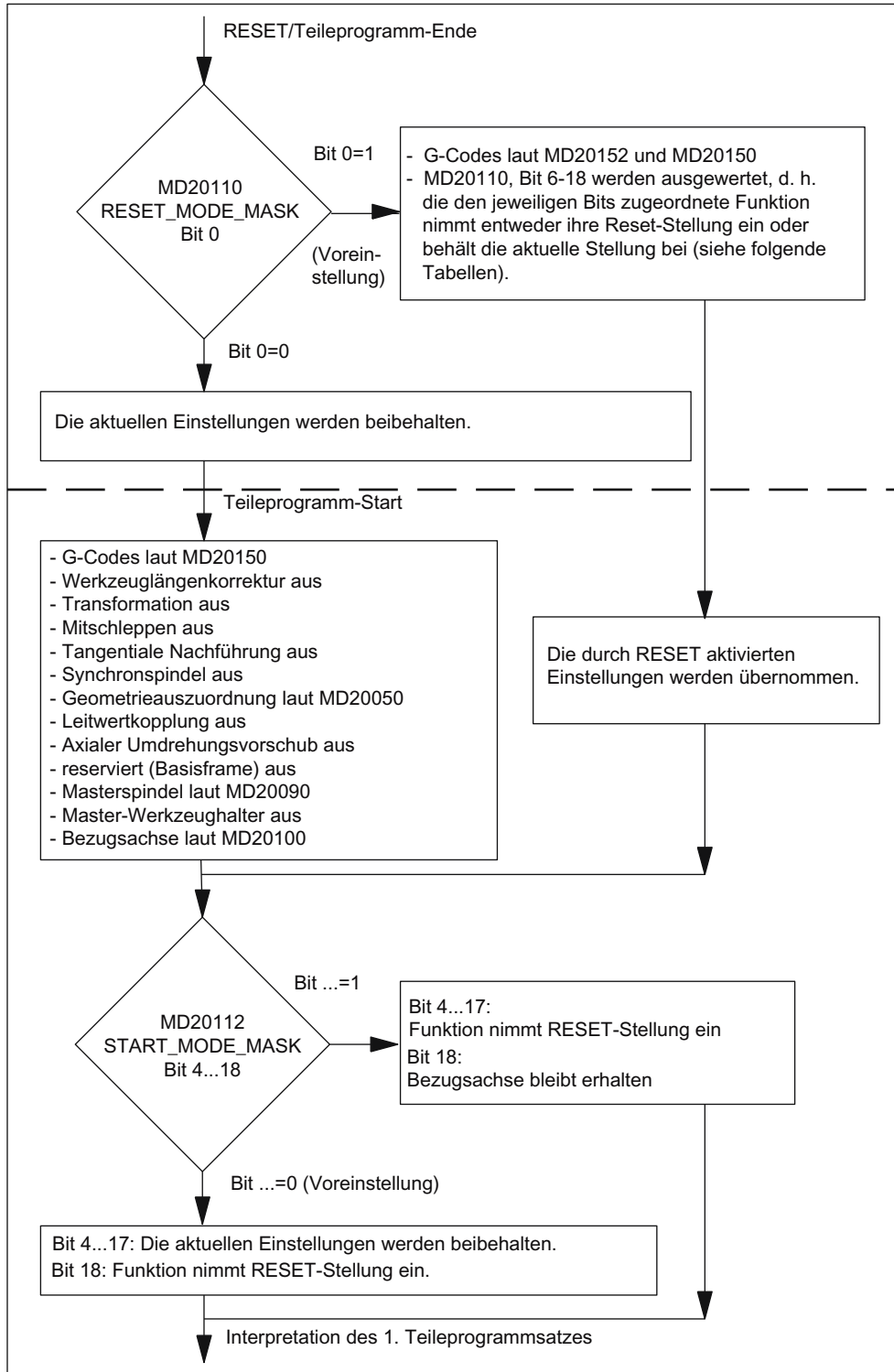


Bild 8-15 Systemeinstellungen nach RESET/Teileprogrammende und Teileprogrammstart

### Wirksamer G-Code nach Hochlauf und Reset / Teileprogrammende

Die Einstellung des nach Hochlauf (Power On) und Reset / Teileprogrammende in jeder G-Gruppe wirksamen G-Codes erfolgt über folgende Maschinendaten:

MD20150 \$MC\_GCODE\_RESET\_VALUES[<G-Gruppe>] = <Default-G-Code>

MD20152 \$MC\_GCODE\_RESET\_MODE[<G-Gruppe>] = <Wert>

Wert	Beschreibung: Pro G-Gruppe
0	Der Default-G-Code aus MD20150 \$MC_GCODE_RESET_VALUES wird wirksam.
1	Der letzte aktive / aktuelle G-Code ist wirksam

### Steuerungsgrundstellung nach Hochlauf, Reset / Teileprogrammende und Teileprogrammstart

Die Festlegung der Steuerungsgrundstellung nach Hochlauf, Reset / Teileprogrammende und Teileprogrammstart erfolgt über folgende Maschinendaten:

- MD20110 \$MC\_RESET\_MODE\_MASK (Festlegung der Steuerungs-Grundstellung nach **Hochlauf** und **Reset / Teileprogrammende**)
- MD20112 \$MC\_START\_MODE\_MASK (Festlegung der Grundstellung der Steuerung nach **Teileprogrammstart**)

#### Literatur

Ausführliche Maschinendaten-Beschreibung

### Relevante der Maschinendaten

Maschinendatum	Bedeutung
MD20120 \$MC_TOOL_RESET_VALUE	Werkzeug - Längenkorrektur im Hochlauf, Reset / Teileprogrammende
MD20121 \$MC_TOOL_PRESEL_RESET_VALUE	Vorgewähltes Werkzeug bei Reset
MD20130 \$MC_CUTTING_EDGE_RESET_VALUE	Werkzeugschneide - Längenkorrektur im Hochlauf
MD20140 \$MC_TRAFO_RESET_VALUE	Transformationsdatensatz Hochlauf
MD20144 \$MC_TRAFO_MODE_MASK	Funktionsanwahl der kinematischen Transformation
MD20150 \$MC_GCODE_RESET_VALUES	Löschstellung der G-Gruppen
MD20152 \$MC_GCODE_RESET_MODE	Resetverhalten der G-Gruppen
MD21330 \$MC_COUPLE_RESET_MODE_1	Kopplungs-Abbruch-Verhalten
MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrieachse zu Kanalachse
MD20118 \$MC_GEOAX_CHANGE_RESET	Automatischen Geometrieachswechsel erlauben

## Beispiel

### Reset-Stellung bei Reset aktivieren:

- MD20110, Bit 0 = 1
- MD20112 = 0

### Transformation bleibt bei Reset / Teileprogrammstart erhalten:

- MD20110, Bit 0 = 1
- MD20110, Bit 7 = 1
- MD20112 = 0

### Werkzeuglängenkorrektur bleibt über Reset / Teileprogrammstart erhalten:

- MD20110, Bit 4 = 1
- MD20110, Bit 6 = 1
- MD20112 = 0

### Aktive Ebene (Bit 4) und einstellbarer Frame (Bit 5) bleiben über Reset erhalten und werden bei Teileprogrammstart zurückgesetzt:

- MD20110, Bit 4 = 1
- MD20110, Bit 5 = 1
- MD20112, Bit 4 = 1
- MD20112, Bit 5 = 1

---

### Hinweis

#### MD20110 / MD20112, Bit 5 und Bit 6

Ist bei Teileprogrammstart in Betriebsart AUTOMATIK bzw. MDA durch entsprechende Parametrierung von MD20110 / MD20112 die Werkzeuglängenkorrektur oder ein Frame aktiv, muss wegen des Verfahrens der Verschiebung, die erste Programmierung der Achsen mit Absolutmaß erfolgen.

Ausnahme: Mit MD42442 / MD42440 wurde das Verfahren der Verschiebung bei G91 unterdrückt.

---

## 8.14.1 Werkzeugrückzug nach POWER ON mit Orientierungstransformation

### Funktion

Wurde ein Teileprogramm mit einer Bearbeitung mit Werkzeugorientierung durch Spannungsausfall oder Reset abgebrochen, besteht nach dem Hochlauf der Steuerung (Power On) die Möglichkeit, die zuvor aktive Transformation wieder anzuwählen und einen Frame in Richtung der Werkzeugachse zu generieren. Danach kann das Werkzeug in der Betriebsart JOG durch eine Rückzugsbewegung in Richtung der Werkzeugachse freigefahren werden.

### Voraussetzung

Für alle an der Transformation beteiligten Maschinenachsen, müssen die aktiven Messsysteme einen Maschinenbezug haben. Siehe Kapitel "Positions-Restaurierung bei POWER ON (Seite 1373)".

### Parametrierung

Damit die zuletzt aktive Transformation über POWER ON erhalten bleibt, sind folgende Maschinendaten einzustellen:

- MD20144 \$MC\_TRAFO\_MODE\_MASK, Bit 1 = 1
- MD20110 \$MC\_RESET\_MODE\_MASK, Bit 0 = 1
- MD20110 \$MC\_RESET\_MODE\_MASK, Bit 7 = 1

Siehe auch Kapitel "Systemeinstellungen für Hochlauf, RESET/Teileprogrammende und Teileprogramm-Start (Seite 658)".

### Programmierung

#### Warten auf Maschinenbezug WAITENC

Mit dem Befehl WAITENC wird in einem Programm kanalspezifisch gewartet, bis für alle aktiven Messsysteme der parametrisierten Achsen ein gültiger Maschinenbezug vorliegt. Siehe oben Abschnitt "Voraussetzung". Die Parametrierung der Achsen erfolgt über:

MD34800 \$MA\_WAIT\_ENC\_VALID = 1

#### Anwendung

In dem beim Hochlauf (Voraussetzung: MD20108 Bit 3 = 1) aufzurufenden ereignisgesteuerten Anwenderprogramm (.../\_N\_CMA\_DIR/\_N\_PROG\_EVENT\_SPF) muss mit dem Befehl `WAITENC` gewartet werden, bis gültige Achspositionen zur Verfügung stehen.

Danach kann durch den NC-Sprachbefehl `TOROTX/TOROTY/TOROTZ` ein Frame generiert werden, der die Werkzeugachse in Richtung der X-, Y- oder Z-Achse legt.

**Beispiel**

Orientierungstransformation und Orientierungsachsen mit inkrementellen Gebern.

Projektierung:	Bedeutung:
MD10720 \$MN_OPERATING_MODE_DEFAULT [ 0 ] = 6	Hochlauf in Betriebsart JOG
MD30240 \$MA_ENC_TYPE [ 0, <Achse>] = 1	Inkrementelles Messsystem
MD34210 \$MA_ENC_REFP_STATE [ 0, <Achse>] = 3	Restaurieren der Achspositionen für Inkrementalgeber freigeben.
MD20108 \$MC_PROG_EVENT_MASK = 'H9'	Ereignisgesteuertes Anwenderprogramm (PROG_EVENT) im Hochlauf und bei Teileprogrammstart aktivieren.
MD20152 \$MC_GCODE_RESET_MODE [ 52 ] = 1	TOFRAME über Reset erhalten.
MD20110 \$MC_RESET_MODE_MASK = 'HC1'	Transformation und WZK über Reset erhalten.
MD20144 \$MC_TRAFO_MODE_MASK = 'H02'	Transformation über POWER OFF erhalten.

**Ereignisgesteuertes Anwenderprogramm (.../\_N\_CMA\_DIR/\_N\_PROG\_EVENT\_SPF)**

Programmcode	Kommentar
; Beispiel mit Aktivierung des Frames, der das WKS in Werkzeugrichtung ausrichtet, im Hochlauf und Rücksetzen mit Teileprogrammstart.	
IF \$P_PROG_EVENT == 4	; Hochlauf.
IF \$P_TRAFO <> 0	; Transformation wurde angewählt.
WAITENC	; Warten auf gültige Achspositionen der Orientierungsachsen.
TOROTZ	; Z-Achse des WKS in Richtung der Werkzeugachse drehen.
ENDIF	
M17	
ENDIF	
IF \$P_PROG_EVENT == 1	; Teilprogrammstart.
TOROTOF	; Werkzeugframe zurücksetzen.
RET	
ENDIF	

Der Befehl `WAITENC` entspricht prinzipiell folgender Programmsequenz (Beispiel für 5-Achs-Maschine mit AB-Kinematik):

Programmcode	Kommentar
WHILE TRUE	; Warten auf Messsystem.
IF ((\$AA_ENC_ACTIVE[X]==TRUE) AND (\$AA_ENC_ACTIVE[Y]==TRUE) AND (\$AA_ENC_ACTIVE[Z]==TRUE) AND (\$AA_ENC_ACTIVE[A]==TRUE) AND (\$AA_ENC_ACTIVE[B]==TRUE)) GOTO GET_LABEL	
ENDIF	
G4 F0.5	; Wartezeit 0,5 s
ENDWHILE	
:Position synchronisieren	
GET_LABEL: GET(X,Y,Z,A,B,)	



## Bearbeitung fortsetzen

### Betriebsart AUTOMATIK

Zum automatischen Abarbeiten von Programmen in der Betriebsart AUTOMATIK müssen alle Maschinenachsen, deren Istposition des aktive Messsystems restauriert wurde, referenziert werden.

### Betriebsart MDA und Überspeichern

In der Betriebsart MDA und für das Überspeichern, kann die Bearbeitung auch ohne Referenzieren der Achsen, mit restaurierten Positionen erfolgen. Dazu muss kanalspezifisch NC-Start mit restaurierten Positionen explizit freigegeben werden:

```
MD20700 $MC_REFP_NC_START_LOCK = 2
```

## Randbedingung

### Achsen mit inkrementellen Gebern und ohne Istwert-Pufferung

Es wird davon ausgegangen, dass Achsen mit inkrementellen Gebern und **ohne** Istwert-Pufferung bei Spannungsausfall hinreichend schnell geklemmt werden, um ein Wegdriften von der letzten Sollposition zu verhindern.

## 8.14.2 Werkzeugrückzug in Unterbetriebsart JOG-Retract

### Funktion

Die Funktion "Rückzugsbewegung in Werkzeugrichtung in der Unterbetriebsart JOG-Retract", im Folgenden "JOG-Retract" genannt, unterstützt das manuelle Freifahren des Werkzeugs nach einem Programmabbruch in den Betriebsarten AUTOMATIK oder MDA durch Spannungsausfall oder Kanal-Reset.

### Wiederherzustellende Daten

Für die Rückzugsbewegung in Werkzeugrichtung müssen dazu folgende zuletzt vor dem Programmabbruch im Kanal aktiven Daten wiederhergestellt werden:

- Aktive Werkzeugkorrektur
- Aktive Bearbeitungsebene
- Aktive Toolcarrier
- Aktive Transformationsdatensatz mit Transformationsparametern
- Daten des Gewindeverbundes bei G33 bzw. G331/G332
- Positionen der Achsen, die an der Transformation beteiligt sind

Liegen nach einem Programmabbruch diese Daten vor, werden sie mit der Anwahl der Unterbetriebsart JOG-Retract im Kanal wirksam. Dabei wird das Werkzeugkoordinatensystem (WKS) automatisch so ausgerichtet, dass eine der Geometrieachsen in Richtung der Werkzeugachse liegt. Der Werkzeugrückzug kann dann durch das manuelle Verfahren dieser Geometrieachse vorgenommen werden.

---

### Hinweis

Durch JOG-Retract werden nur Verweise auf die oben genannten Daten gesichert, nicht die Daten (z. B. Werkzeugkorrekturwerte) selbst. Werden die Daten vor Anwahl von JOG-Retract geändert, wird die Funktion auf Basis der geänderten Daten durchgeführt.

---

### Anwendungsfälle

Freifahren des Werkzeugs nach Programmabbruch während:

- Bearbeitungen mit Werkzeugorientierung mit Schwenkzyklus oder Orientierungstransformation
- Gewindebohren mit G33 (mit Ausgleichsfutter und drehzahl geregelter Spindel mit Geber)
- Interpolatorisches Gewindebohren mit G331/G332 (ohne Ausgleichsfutter und lage geregelter Spindel)

### Ereignisgesteuerte Programmaufrufe

Folgende ereignisgesteuerten Programmaufrufe werden im Zusammenhang mit JOG-Retract nicht ausgeführt:

MD20108 \$MC_PROG_EVENT_MASK	
Bit	Ereignis
2	Bedientafel-Reset
3	Hochlauf
5	Safety-Programm im Hochlauf

Über Interruptsignale der NC gestarteten Programme werden ebenfalls nicht ausgeführt.

---

### Hinweis

#### Istpositionen der Maschinenachsen von Transformationen

Ist für JOG-Retract eine Transformation aktiv, muss für alle an der Transformation beteiligten Maschinenachsen ein Maschinenbezug vorhanden sein. Siehe Kapitel "Werkzeugrückzug nach POWER ON mit Orientierungstransformation (Seite 663)".

---

### NC/PLC-Nahtstellensignal

Liegen für JOG-Retract relevanten Daten vor, so dass die Funktion ausgeführt werden kann, wird folgendes kanalspezifische NC/PLC-Nahtstellensignal gesetzt:

Mit dem Start der Programmbearbeitung in der Betriebsart AUTO oder MDA wird das Nahtstellensignal wieder zurückgesetzt.

## Anwahl

Die Auswahl der Unterbetriebsart JOG-Retract ist nur möglich, wenn gültige Retract-Daten vorliegen. Die Anzeige erfolgt über das kanalspezifische NC/PLC-Nahtstellensignal:

DB21, ... DBX377.5 == 1 (Retract-Daten verfügbar)

### Automatischer Wechsel in die Unterbetriebsart JOG-Retract

Nach dem Hochlauf wird normalerweise die parametrisierte Default-Betriebsart aktiv:

MD10720 \$MN\_OPERATING\_MODE\_DEFAULT = <Betriebsart>

Sind nach dem Hochlauf Retract-Daten verfügbar, kann nach dem Hochlauf automatisch die Unterbetriebsart JOG-Retract aktiv werden:

MD10721 \$MN\_OPERATING\_MODE\_EXTENDED = 1

### Rückmeldung "JOG-Retract aktiv"

Die Rückmeldung, dass JOG-Retract aktiv ist, erfolgt über das NC/PLC-Nahtstellensignal:

DB21, ... DBX377.4 == 1 (JOG-Retract aktiv)

### Anwahl über PLC-Anwenderprogramm

JOG-Retract kann vom PLC-Anwenderprogramm aus ausgewählt werden:

- Voraussetzungen:
  - DB21, ... DBX377.5 == 1 (Retract-Daten verfügbar)
  - Kanalzustand: Reset
- Auswahl: PI-Dienst "RETRAC" (siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl" > "Bausteinbeschreibungen" > "PI-Dienste" > "PI-Dienst: RETRAC (Seite 1035)")

## Werkzeugrückzug

Der Werkzeugrückzug erfolgt durch manuelles Verfahren der durch die Auswahl von JOG-Retract festgelegten Geometrieachse im WKS. Das manuelle Verfahren kann auch inkrementell erfolgen. Die JOG-Bewegung kann dann mit NC-Stop angehalten und mit NC-Start wieder fortgesetzt werden.

Bei G33 bzw. G331/G332 ist auch eine Spindel am Geometrieachsverbund beteiligt. Die Rückzugsbewegung wird ausgeführt, wenn die Rückzugsachse oder die Spindel manuell verfahren wird, d. h. beim manuellen Verfahren einer Achse des Verbundes werden alle beteiligten Achsen verfahren. Nicht am Geometrieachsverbund beteiligte Achsen bzw. Spindeln können beliebig manuell verfahren werden.

Das Verfahren in Richtung der Werkzeugachse kann sowohl über die Verfahrtasten als auch über Handrad erfolgen.

#### ACHTUNG

Nach Spannungsausfall werden die Achspositionen synchronisiert bzw. restauriert. Nach dem Freifahren des Werkzeugs in der Unterbetriebsart JOG-Retract müssen Achsen, deren Positionen restauriert wurden, referenziert werden (siehe auch MD20700 \$MC\_REFP\_NC\_START\_LOCK und MD34110 \$MA\_REFP\_CYCLE\_NR).

### Werkzeugrückzug bei Gewindeschneiden (G33) oder Gewindebohren (G331, G332)

Bei Gewindebohren (G331, G332) werden die Fahranforderungen für die beteiligten Achsen summiert, d. h. solange für eine beteiligte Achse/Spindel eine Fahranforderung (Verfahrtaste) ansteht, wird auf der Kontur zurückgefahren. Ist für eine beteiligte Achse/Spindel ein Handrad angewählt und wird dieses bewegt, wird dies ignoriert, solange noch ein Verfahren über Verfahrtaste für eine andere beteiligte Achse/Spindel ansteht. Sind für mehrere Achsen bzw. die Spindel Handräder angewählt und werden diese bewegt, werden die Impulse des jeweiligen Handrades in folgender Reihenfolge ausgewertet:

1. Rückzugsachse
2. Spindel
3. Andere Achse

Die Impulse der anderen Handräder werden ignoriert und erst bei Stillstand der vorrangigen Handräder ausgewertet.

Bei G33, G331, G332 werden während eines möglichen Rückzugs folgende Spindel-/Achsanforderungen gesperrt:

- Spindelstart über DB31, ... DBX30
- FC18 für die Spindel
- FC18 für eine am Rückzug beteiligte Achse
- Spindel oder am Rückzug beteiligte Achse zur PLC-kontrollierten Achse umschalten
- Spindel oder am Rückzug beteiligte Achse in einen anderen Kanal oder zur Hauptlaufachse (Kommandoachse, Pendelachse, FC18 / konkurrierende Achse) tauschen

Über MD10735 \$MN\_JOG\_MODE\_MASK, Bit 8 = 0 kann eingestellt werden, dass bei einer Rückzugbewegung für G33, G331, G332 die Rückzugachse nur in positiver Richtung verfahren werden kann. Mit Bit 8 = 1 kann auch in negativer Richtung verfahren werden, allerdings nur bis zum Startpunkt der Kontur, die für die Retract-Bewegung gebildet wurde.

## Abwahl von JOG-Retract

Mit Bedientafel-Reset oder dem kanalspezifischen PI-Dienst \_N\_FINDAB wird die Unterbetriebsart JOG-Retract abgewählt und das System befindet sich in der Betriebsart JOG.

Durch die Abwahl werden die projizierten Reset-Einstellungen aktiv:

- MD20110 \$MC\_RESET\_MODE\_MASK
- MD20150 \$MC\_GCODE\_RESET\_VALUES
- MD20151 \$MC\_GCODE\_RESET\_MODE

## Randbedingungen

- Achskopplungen werden durch die Anwahl der Unterbetriebsart JOG-Retract nicht wieder hergestellt.
- Werkzeugrückzug bei Gewindebohren mit G63 (mit Ausgleichsfutter und drehzahl geregelter Spindel ohne Geber) ist nicht möglich. Das NC/PLC-Nahtstellensignal DB21, ... DBX377.5 (Retract Daten verfügbar) wird in diesem Fall nicht gesetzt.
- OEM-Transformationen für Parallelkinematiken (z. B. für Hexapoden) erlauben manuelles Verfahren nur mit referenzierten Achsen. Restaurierte Achspositionen können in diesem Fall nicht verwendet werden.
- Wurde die Werkzeugorientierung nicht über eine NC-Funktion aktiviert, sondern durch direktes Programmieren der Orientierungsachsen, besteht keine Möglichkeit eines NC-unterstützten Rückzugs in Richtung der Werkzeugachse.
- JOG-Retract kann auch in Systemen mit NCU-Link genutzt werden. Es ist jedoch zu beachten, dass durch Anwahl von JOG-Retract der aktuelle Status von Achscontainern nicht verändert wird und insbesondere alte Achscontainer-Zustände vor Spannungsausfall bzw. Programmabbruch nicht restauriert werden.
- Wird durch JOG-Retract eine Transformation angewählt, müssen für die an der Transformation beteiligten Achsen restaurierte oder synchronisierte Positionen vorhanden sein. Vom Anwender ist dafür zu sorgen, dass Achsen mit inkrementellen Gebern bei Spannungsausfall hinreichend schnell geklemmt werden, um ein Wegdriften von der letzten Sollposition zu verhindern, da sonst die restaurierte Position zu stark von der Istposition abweichen kann. Auch antriebsautarkes Rückziehen darf für diese Achsen nicht aktiviert werden.

## 8.15 Ersetzung von Funktionen durch Unterprogramme

### 8.15.1 Übersicht

#### Funktion

Anwenderspezifische Hilfsfunktionen (z.B.  $M_{101}$ ) lösen keine Systemfunktionen aus. Sie werden lediglich an die NC/PLC-Nahtstelle ausgegeben. Die Funktionalität der Hilfsfunktion ist vom Anwender / Maschinenhersteller im PLC-Anwenderprogramm zu realisieren. In weiteren Verlauf wird beschrieben, wie statt der standardmäßigen Ausgabe an die NC/PLC-Nahtstelle, der Aufruf eines anwenderspezifischen Unterprogramms (Ersetzungsunterprogramm) projiziert wird.

Im Teileprogramm wird dann weiterhin die Funktion  $M_{101}$  programmiert. Bei der Abarbeitung des Teileprogramms erfolgt aber der Aufruf des Ersetzungsunterprogramms. Die Funktion wird somit von der NC durch einen Unterprogrammaufruf ersetzt. Daraus ergeben sich folgende Vorteile:

- Bei Anpassungen an den Fertigungsprozess kann ein bestehendes, getestetes und bewährtes Teileprogramm unverändert weiter verwendet werden. Die Anpassungen werden in anwenderspezifische Unterprogramme ausgelagert.
- Die Realisierung der Funktionalität kann innerhalb des Ersetzungsunterprogramms mit dem vollen Funktionsumfang der NC-Sprache erfolgen.
- Es entfällt der Kommunikations-Overhead zwischen NC und PLC.

#### Ersetzbare Funktionen

Folgende Funktionen können durch Unterprogramme ersetzt werden:

Hilfsfunktionen	
M	Schaltfunktionen
T	Werkzeuganwahl
TCA	Werkzeuganwahl unabhängig vom Status des Werkzeugs
D	Werkzeugkorrektur
DL	additive Werkzeugkorrektur

Spindelbezogene Funktionen bei aktiver Synchronspindelkopplung	
M40	Automatischer Getriebestufenwechsel
M41 - M45	Getriebestufenanwahl 1 ... 5
SPOS	Spindelpositionieren
SPOSA	Spindelpositionieren
M19	Spindelpositionieren

## 8.15.2 Ersetzung von M-, T/TCA- und D/DL-Funktionen

### 8.15.2.1 Ersetzung von M-Funktionen

#### Allgemeine Informationen

Für die Ersetzung von M-Funktionen gelten folgende Bedingungen:

- Pro Satz wird nur eine M-Funktion ersetzt.
- Ein Satz in dem eine M-Funktion ersetzt werden soll, darf folgende Elemente **nicht** enthalten:
  - M98
  - Modalen Unterprogramm-Aufruf
  - Unterprogrammrücksprung
  - Teileprogrammende
- M-Funktionen, die Systemfunktionen auslösen, dürfen nicht durch ein Unterprogramm ersetzt werden (siehe Absatz "Nicht ersetzbare M-Funktionen").

#### Parametrierung

##### M-Funktion und Unterprogramm

Die Parametrierung der M-Funktionen und der Ersetzungsunterprogramme erfolgt in folgenden Maschinendaten:

- MD10715 \$MC\_M\_NO\_FCT\_CYCLE[<Index>] = <M-Funktionsnummer>
- MD10716 \$MC\_M\_NO\_FCT\_CYCLE\_NAME[<Index>] = "<Unterprogrammname>"

Die M-Funktion und das entsprechende Ersetzungsunterprogramm sind durch den gleichen Index verbunden.

Beispiel: M-Funktion  $M_{101}$  wird durch Unterprogramm SUB\_M101 und M-Funktion  $M_{102}$  durch SUB\_M102 ersetzt:

MD10715 \$MC_M_NO_FCT_CYCLE[ 0 ]	= 101
MD10716 \$MC_M_NO_FCT_CYCLE_NAME[ 0 ]	= "SUB_M101"
MD10715 \$MC_M_NO_FCT_CYCLE[ 1 ]	= 102
MD10716 \$MC_M_NO_FCT_CYCLE_NAME[ 1 ]	= "SUB_M102"

### Systemvariable zur Informationsweitergabe

Für eine frei wählbare M-Funktion werden über Systemvariable (siehe Kapitel "Systemvariable (Seite 676)") Informationen bezüglich der ersetzten M-Funktion und weiterer Funktionen (T, TCA, D, DL) zur Auswertung im Ersetzungsunterprogramm zur Verfügung gestellt. Die in den Systemvariablen enthaltenen Daten beziehen sich auf den Satz, in dem die zu ersetzende M-Funktion programmiert wurde.

Die Auswahl der M-Funktion erfolgt mit dem Index des Maschinendatums MD10715 \$MC\_M\_NO\_FCT\_CYCLE[<Index>] in dem die zu ersetzende M-Funktion parametrier ist:

- MD10718 \$MC\_M\_NO\_FCT\_CYCLE\_PAR = <Index>

---

### Hinweis

Bei einer M-Funktionsersetzungen mit Informationsweitergabe über Systemvariable müssen die Adresserweiterung und der Funktionswert der M-Funktion als konstante Werte programmiert werden.

### Zulässige Programmierung:

- M<Funktionswert>
- M=<Funktionswert>
- M[<Adresserweiterung>]=<Funktionswert>

### Unzulässige Programmierung:

- M=<Variable1>
  - M[<Variable2>]=<Variable1>
- 

## Programmierung

Regeln zur Ersetzung von M-Funktionen:

- Das Ersetzungsunterprogramm wird am Satzende aufgerufen
- Innerhalb des Ersetzungsunterprogramms werden keine M-Funktionen ersetzt
- In einem ASUP wird die M-Funktion auch dann ersetzt, wenn das ASUP innerhalb des Ersetzungsunterprogramms gestartet wurde.

## M-Funktionen die nicht ersetzt werden

Folgende M-Funktionen lösen als vordefinierte Hilfsfunktionen Systemfunktionen aus und dürfen nicht durch ein Unterprogramm ersetzt werden:

- M0, ... M5
- M17, M30
- M19
- M40, ... M45
- M98, M99 (Nur bei MD18800 \$MN\_MM\_EXTERN\_LANGUAGE ≠ 0)



Die über Maschinendaten parametrisierten anwenderspezifischen M-Funktionen dürfen ebenfalls nicht durch ein Unterprogramm ersetzt werden, da auch sie Systemfunktionen auslösen.

<b>Maschinendatum</b>	<b>Bedeutung</b>
MD10714 \$MN_M_NO_FCT_EOP	M-Funktion für Spindel aktiv nach RESET
MD10804 \$MN_EXTERN_CHAN_M_NO_SET_INT	M-Funktion für ASUP-Aktivierung (Externmodus)
MD10806 \$MN_EXTERN_CHAN_M_NO_DISABLE_INT	M-Funktion für ASUP-Deaktivierung (Externmodus)
MD10814 \$MN_EXTERN_M_NO_MAC_CYCLE	Makroaufruf über M-Funktion
MD20094 \$MC_SPIND_RIGID_TAPPING_M_NR	M-Funktion für Umschalten in gesteuerten Achsbetrieb
MD20095 \$MC_EXTERN_RIGID_TAPPING_M_NR	M-Funktion für Umschalten in gesteuerten Achsbetrieb (Externmodus)
MD22254 \$MC_AUXFU_ASSOC_M0_VALUE	Zusätzliche M-Funktion für Programm-Halt
MD22256 \$MC_AUXFU_ASSOC_M1_VALUE	Zusätzliche M-Funktion für bedingten Halt
MD26008 \$MC_NIBBLE_PUNCH_CODE	Festlegung der M-Funktionen (für Nibbeln -spezifisch)
MD26012 \$MC_PUNCHNIB_ACTIVATION	Aktivierung der Stanz- und Nibbelfunktionen

**Hinweis**

**Ausnahme**

Die über das Maschinendatum

MD22560 \$MC\_TOOL\_CHANGE\_M\_CODE (Werkzeugwechsel mit M-Funktion)

parametrierte M-Funktion, darf durch ein Unterprogramm ersetzt werden.

### 8.15.2.2 Ersetzung von T/TCA- und D/DL-Funktionen

#### Randbedingungen

Für die Ersetzung der Funktionen T, TCA, D und DL gelten folgende Randbedingungen:

- Pro Satz wird maximal eine Funktionsersetzung wirksam.
- Ein Satz mit Funktionsersetzung darf folgende Elemente **nicht** enthalten:
  - M98
  - Modalen Unterprogramm-Aufruf
  - Unterprogrammrücksprung
  - Teileprogrammende

#### Parametrierung: Ersetzungsunterprogramm

Die Angabe des Ersetzungsunterprogramms erfolgt funktionspezifisch im Maschinendatum:

Funktion	Maschinendatum
T	MD10717 \$MN_T_NO_FCT_CYCLE_NAME
TCA	MD15710 \$MN_TCA_CYCLE_NAME
D/DL	MD11717 \$MN_D_NO_FCT_CYCLE_NAME

#### Hinweis

Es wird empfohlen, für die Ersetzung der T-, TCA- und D/DL-Funktionen dasselbe Unterprogramm zu verwenden.

#### Parametrierung: Verhalten bezüglich D- bzw. DL-Funktion bei gleichzeitiger T-Funktion

Bei gleichzeitiger Programmierung der Funktionen D bzw. DL und T in einem Satz, wird die D- bzw. DL-Nummer entweder als Parameter an das Ersetzungsunterprogramm übergeben oder die D- bzw. DL-Funktion wird vor Aufruf des Ersetzungsunterprogramms ausgeführt. Das Verhalten ist einstellbar über:

MD10719 \$MN\_T\_NO\_FCT\_CYCLE\_MODE (Parametrierung der T-Funktionsersetzung)

Bit	Wert	Bedeutung
0	0	Die D- bzw. DL-Nummer steht im Unterprogramm per Systemvariable zur Verfügung (Grundstellung).
	1	Die D bzw. DL-Nummer wird direkt im Satz verrechnet. <b>Hinweis:</b> Diese Funktion ist nur aktiv, wenn der Werkzeugwechsel mit M-Funktion projiziert wurde: MD22550 \$MC_TOOL_CHANGE_MODE = 1 Anderweitig werden die D- bzw. DL-Werte immer übergeben.

#### Systemvariable zur Informationsweitergabe

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen über Systemvariable (siehe Kapitel "Systemvariable (Seite 676)") zur Verfügung gestellt.

Die in den Systemvariablen enthaltenen Daten beziehen sich auf den Satz, in dem die zu ersetzende Funktion programmiert wurde.

#### Parametrierung: Aufrufzeitpunkt des Ersetzungsunterprogramms

Der Aufrufzeitpunkt des Ersetzungsunterprogramms wird eingestellt über:

MD10719 \$MN\_T\_NO\_FCT\_CYCLE\_MODE, Bit 1 und Bit 2

Bit 2	Bit 1	Aufrufzeitpunkt des Ersetzungsunterprogramms
0	0	Am Satzende Nach der Bearbeitung des Ersetzungsunterprogramms wird die Interpretation mit der Programmzeile fortgesetzt, die auf diejenige folgt, die den Ersetzungsvorgang ausgelöst hat.
0	1	Am Satzanfang Nach der Bearbeitung des Ersetzungsunterprogramms wird die Programmzeile, die zum Aufruf des Ersetzungsunterprogramms geführt hat, interpretiert. Dabei werden die Adressen T, D bzw. DL und die M-Funktion für den Werkzeugwechsel nicht mehr bearbeitet.
1	-	Am Satzanfang und Satzende Das Ersetzungsunterprogramm wird zweimal aufgerufen.

#### Systemvariable für den Aufrufzeitpunkt

Über die Systemvariablen \$P\_SUB\_STAT kann gelesen werden, ob die Ersetzung aktiv ist und wenn ja, wann das Ersetzungsunterprogramm, bezogen auf den Satz, aufgerufen wurde:

Wert	Bedeutung
0	Ersetzung nicht aktiv
1	Ersetzung aktiv, Unterprogrammaufruf am Satzanfang
2	Ersetzung aktiv, Unterprogrammaufruf am Satzende

**Beispiel: Ersetzung der T-Funktion**

Parametrierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 0	Werkzeugwechsel mit T-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Name des Unterprogramms zum Ersetzen der T-Funktion
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 0	Aufrufzeitpunkt: Satzende

Programmierung	Kommentar
N110 D1	; D1
N120 G90 G0 X100 Y100 Z50	; D1 ist aktiv
N130 D2 X110 Z0 T5	; D1 bleibt aktiv
	; Die T-Funktion wird am Satzende durch
	; UP-Aufruf MY_T_CYCLE ersetzt
	; D2 wird MY_T_CYCLE in einer
	; Systemvariable zur Verfügung gestellt

Ein ausführliches Beispiel zur Ersetzung der T-Funktion findet sich in Kapitel: "Beispiele zur M/T-Funktionsersetzung beim Werkzeugwechsel (Seite 678)".

**8.15.2.3 Systemvariable**

**Allgemeine Informationen**

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen (T bzw. TCA, D bzw. DL, M) über Systemvariable zur Verfügung gestellt.


**Ausnahme**

Keine Weitergabe der D- bzw. DL-Nummer, wenn:

- MD10719 \$MN\_T\_NO\_FCT\_CYCLE\_MODE, Bit 0 = 1
- MD22550 \$MC\_TOOL\_CHANGE\_MODE = 1

UND

- D bzw. DL zusammen mit T- oder M-Funktion in einem Satz programmiert

 <b>VORSICHT</b>
<p>Die dem Ersetzungsunterprogramm in den Systemvariablen zur Verfügung gestellten Werte sind noch nicht wirksam. Es liegt in der alleinigen Verantwortung des Anwenders/Maschinenherstellers dies durch geeignete Programmierung im Ersetzungsunterprogramm "nachzuholen".</p>

## Systemvariablen

Systemvariable	Bedeutung
\$C_M_PROG	TRUE, wenn M-Funktion programmiert wurde
\$C_M	Enthält bei \$C_M_PROG == TRUE den Wert der Adresse M Dabei sind zwei Fälle zu unterscheiden: <ul style="list-style-type: none"> <li>Ist für den Werkzeugwechsel mit M-Funktion ein Unterprogramm mit Parameterübergabe projektiert, liefert \$C_M den Wert aus: MD10715 MN_M_NO_FCT_CYCLE</li> <li>Ist nur ein Unterprogramm für die Adressen T und/oder D/DL projektiert und im Programm ist die M-Funktion für den Werkzeugwechsel zusammen mit einer der zu ersetzenden Adressen programmiert, liefert \$C_M den Wert aus: MD22560 \$MN_TOOL_CHANGE_M_MODE</li> </ul>
\$C_AUX_VALUE[0]	Wert der ersetzten M-Funktion
\$C_ME	Enthält bei \$C_M_PROG == TRUE den Wert der Adresserweiterung der M-Funktion
\$C_AUX_EXT[0]	Adresserweiterung der M-Funktion (identisch zu \$C_ME)
\$C_AUX_IS_QUICK[0]	TRUE, wenn M-Funktion mit schneller Ausgabe an PLC programmiert wurde
\$C_T_PROG	TRUE, wenn T-Funktion programmiert wurde
\$C_T	Enthält bei \$C_T_PROG == TRUE den Wert der T-Funktion
\$C_TE	Enthält bei: <ul style="list-style-type: none"> <li>\$C_T_PROG == TRUE</li> <li>\$C_TS_PROG == TRUE</li> </ul> den Wert der Adresserweiterung der T-Funktion
\$C_TS_PROG	TRUE, wenn bei der T- oder TCA-Ersetzung ein Werkzeugbezeichner programmiert wurde.
\$C_TS	Enthält bei \$C_TS_PROG == TRUE den bei T- oder TCA-Ersetzung programmierten Werkzeugbezeichner
\$C_TCA	TRUE, wenn die TCA-Ersetzung aktiv ist
\$C_DUPLO_PROG	TRUE, wenn die Duplo-Nummer bei der TCA-Ersetzung programmiert wurde
\$C_DUPLO	Enthält bei \$C_DUPLO_PROG == TRUE den Wert der programmierten Duplo-Nummer
\$C_THNO_PROG	TRUE, wenn die Toolholder/Spindel-Nummer bei der TCA-Ersetzung programmiert wurde
\$C_THNO	Enthält bei \$C_THNO_PROG == TRUE den Wert der programmierten Toolholder/Spindel-Nummer
\$C_D_PROG	TRUE, wenn D-Funktion programmiert wurde
\$C_D	Enthält bei \$C_D_PROG == TRUE den Wert der D-Funktion
\$C_DL_PROG	TRUE, wenn DL-Funktion programmiert wurde
\$C_DL	Enthält bei \$C_DL_PROG == TRUE den Wert der DL-Funktion
\$P_SUB_STAT	Satzbezogener Aufrufzeitpunkt des Ersetzungsunterprogramms

### 8.15.2.4 Beispiel: Ersetzung einer M-Funktion

#### Beispiel 1

Die Funktion M6 wird durch den Aufruf des Unterprogramms "SUB\_M6" ersetzt.

Die für einen Werkzeugwechsel relevanten Informationen sollen über Systemvariable übergeben werden.

#### Parametrierung

##### Maschinendatum

MD10715 \$MN\_M\_NO\_FCT\_CYCLE[2] = 6

MD10716 \$MN\_M\_NO\_FCT\_CYCLE\_NAME[2] = "SUB\_M6"

MD10718 \$MN\_M\_NO\_FCT\_CYCLE\_PAR = 2

##### Bedeutung

Werkzeugwechsel mit M6

Ersetzungsunterprog. für M6

Info.weitergabe über Systemvariable

#### Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
...	;
N10 T1 D1 M6	; M6 wird durch Unterprogramm "SUB_M6"
	; ersetzt
...	;
N90 M30	

#### Unterprogramm "SUB\_M6"

Programmierung	Kommentar
PROC SUB_M6	
N110 IF \$C_T_PROG==TRUE	; IF Adresse T programmiert
N120 T[\$C_TE]=\$C_T	; T-Anwahl ausführen
N130 ENDIF	; ENDIF
N140 M[\$C_ME]=6	; Werkzeugwechsel ausführen.
N150 IF \$C_D_PROG==TRUE	; IF Adresse D programmiert
N160 D=\$C_D	; D-Anwahl ausführen
N170 ENDIF	; ENDIF
N190 M17	

## Beispiele 2

Das neue Werkzeug wird mit der T-Funktion für den Wechsel vorbereitet. Der Werkzeugwechsel erfolgt erst mit der Funktion M6. Die T-Funktion wird durch den Aufruf des Unterprogramms "MY\_T\_CYCLE " ersetzt. Die D / DL-Nummer wird an das Unterprogramm übergeben.

### Parametrierung

Parametrierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 1	Werkzeugwechsel vorbereiten mit T-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Ersetzungsunterprogramm
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 0	Übergabe der D/DL-Nummer

### Hauptprogramm

Programmcode	Kommentar
N210 D1	;
N220 G90 G0 X100 Y100 Z50	; D1 ist aktiv.
N230 D2 X110 Z0 T5	; D1 bleibt aktiv, programmiertes D2 wird ; dem Unterprogramm als Variable übergeben
N240 M6	; Werkzeugwechsel ausführen

## Beispiele 3

Das neue Werkzeug wird mit der T-Funktion für den Wechsel vorbereitet. Der Werkzeugwechsel erfolgt erst mit der Funktion M6. Die T-Funktion wird durch den Aufruf des Unterprogramms "MY\_T\_CYCLE " ersetzt. Die D / DL-Nummer wird **nicht** an das Unterprogramm übergeben.

### Parametrierung

Parametrierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 1	Werkzeugwechsel vorbereiten mit T-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Ersetzungsunterprogramm
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 1	Keine Übergabe der D/DL-Nummer

**Hauptprogramm**

Programmcode	Kommentar
N310 D1	
N320 G90 G0 X100 Y100 Z50	; D1 ist aktiv.
N330 D2 X110 Z0 T5	; D2 wird aktiv und nicht als Variable an ; das Ersetzungsunterprogramm übergeben.
N340 M6	; Werkzeugwechsel ausführen.

**Beispiele 4**

Die Funktionen T und M6 werden durch das Unterprogramm "MY\_T\_CYCLE" ersetzt.

Die Parameter werden bei der Ersetzung von M6 an das Unterprogramm übergeben.

Wird M6 zusammen mit D bzw. DL im Satz programmiert, wird die D- bzw. DL-Nummer auch dann als Parameter an das Unterprogramm übergeben, wenn keine Übergabe der D/DL-Nummer parametrier ist:

MD10719 \$MN\_T\_NO\_FCT\_CYCLE\_MODE = 1

**Parametrierung**

Projektierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 1	Werkzeugwechsel mit M-Funktion
MD22560 \$MC_TOOL_CHANGE_M_CODE = 6	M-Code für Werkzeugwechsel
MD10715 \$MC_M_NO_FCT_CYCLE[3] = 6	Zu ersetzende M-Funktion
MD10716 \$MC_M_NO_FCT_CYCLE_NAME[3] = "MY_T_CYCLE"	Ersetzungsunterprogramm für M-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Ersetzungsunterprogramm für T-Funktion
MD10718 \$MN_M_NO_FCT_CYCLE_PAR = 3	Parameterübergabe an das Ersetzungsunterprog. für M6
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 1	Keine Übergabe der D/DL-Nummer

**Hauptprogramm**

Programmcode	Kommentar
N410 D1	
N420 G90 G0 X100 Y100 Z50	; D1 ist aktiv.
N330 D2 X110 Z0 T5 M6	; D1 bleibt aktiv, D2 und T5 werden dem M6- Ersetzungsunterprogramm als Variablen übergeben.



### 8.15.2.5 Beispiel: Ersetzung einer T- und D-Funktion

Die Funktionen T und D werden durch den Aufruf des Unterprogramms "D\_T\_SUB\_PROG" ersetzt. Für das Beispiel soll weiter gelten:

- Der Werkzeugwechsel erfolgt mit der Adresse T.
- Der Unterprogrammaufruf erfolgt am Satzanfang.
- Die Werkzeugverwaltung ist nicht aktiv.
- Die Achse B ist eine Teilungsachse mit Hirth-Verzahnung.

### Parametrierung

#### Maschinendatum

MD11717 \$MN\_D\_NO\_FCT\_CYCLE\_NAME = "D\_T\_SUB\_PROG"

MD10717 \$MN\_T\_NO\_FCT\_CYCLE\_NAME = "D\_T\_SUB\_PROG"

MD10719 \$MN\_T\_NO\_FCT\_CYCLE\_MODE = 'H2'

MD22550 \$MC\_TOOL\_CHANGE\_MODE = 0

#### Bedeutung

Ersetzungs-  
unterprogramm  
für D-Funktion

Ersetzungs-  
unterprogramm  
für M-Funktion

Aufruf am Satzanfang

Werkzeugwechsel mit  
T-Funktion

### Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
...	;
N10 G01 F1000 X10 T1=5 D1	; T- und D-Funktion durch Aufruf von
	; "D_T_SUB_PROG" am Satzanfang ersetzt
...	;
N90 M30	

### Unterprogramm "D\_T\_SUB\_PROG"

Programmierung	Kommentar
N1000 PROC D_T_SUB_PROG DISPLOF SBLOF	
N4100 IF \$C_T_PROG==TRUE	; IF Adresse T programmiert
N4120 POS[B]=CAC(\$C_T)	; Teilungsposition anfahren
N4130 T[\$C_TE]=\$C_T	; Werkzeug anwählen (T-Anwahl)
N4140 ENDIF	; ENDIF
N4300 IF \$C_D_PROG==TRUE	; IF Adresse D programmiert
N4320 D=\$C_D	; Korrektur anwählen (D-Anwahl)
N4330 ENDIF	; ENDIF
N4400 IF \$C_DL_PROG==TRUE	; IF Adresse DL programmiert
N4420 D=\$C_DL	; Einsatzkorrektur anwählen
N4430 ENDIF	; ENDIF
N9999 RET	

### 8.15.2.6 Verhalten im Konfliktfall

#### Konfliktfall

Ein Konfliktfall liegt vor, wenn in einem Satz mehrere Funktionen programmiert sind und die Funktionen mit unterschiedlichen Unterprogrammen ersetzt werden sollen:

- Adresse D und DL ersetzen mit Unterprogramm:  
MD11717 \$MN\_FCT\_CYCLE\_NAME = "D\_SUB\_PROG"
- Adresse T ersetzen mit Unterprogramm:  
MD10717 \$MN\_FCT\_CYCLE\_NAME = "T\_SUB\_PROG"
- M-Funktion M6 ersetzen mit Unterprogramm:  
MD10715 \$MN\_M\_NO\_FCT\_CYCLE[0] = 6  
MD10716 \$MN\_M\_NO\_FCT\_CYCLE\_NAME[0] = "M6\_SUB\_PROG"  
MD10718 \$MN\_M\_NO\_FCT\_CYCLE\_PAR = 0  
MD22550 \$MC\_TOOL\_CHANGE\_MODE = 1  
MD22560 \$MC\_TOOL\_CHANGE\_M\_CODE = 6

## Auflösung

Ein Konfliktfall wird entsprechend der nachfolgenden Tabelle aufgelöst:

In einer Programmzeile sind programmiert:			Aufgerufenes Unterprogramm:
D und/oder DL	T oder TCA	M6	
–	–	x	<b>M6_SUB_PROG</b>
–	x	–	<b>T_SUB_PROG</b>
–	x	x	<b>M6_SUB_PROG</b>
x	–	–	<b>D_SUB_PROG</b>
x	–	x	<b>M6_SUB_PROG</b>
x	x	–	<b>T_SUB_PROG</b>
x	x	x	<b>M6_SUB_PROG</b>

## 8.15.3 Ersetzung von Spindel-Funktionen

### 8.15.3.1 Allgemeines

#### Funktion

Bei einer aktiven Kopplung können für Leitspindeln folgende Spindelfunktionen ersetzt werden:

- M40: Automatischer Getriebestufenwechsel
- M41 ... M45: Programmierter Getriebestufenwechsel
- SPOS, SPOSA und M19: Spindelpositionieren

#### Randbedingungen

- Für die Ersetzung einer Spindelfunktion müssen folgende Bedingungen erfüllt sein:
  - Die programmierte Spindel muss Leitspindel einer aktiven Kopplung sein.
  - Leit- und Folgespindel müssen sich im gleichen Kanal befinden. Dies wird nur erkannt, wenn sich die Leitspindel in dem Kanal befindet, in dem die Kopplung geschlossen wurde. Wird die Leitspindel in einen anderen Kanal getauscht, führt ein Getriebestufenwechsel bzw. ein Positionieren dieser Spindel nicht zum Aufruf des Ersetzungsunterprogramms.
  - Ein programmierter Getriebestufenwechsel muss einen realen Getriebestufenwechsel bewirken. Dazu müssen sich programmierte und aktive Getriebestufe unterscheiden.
- In einem Satz kann nur eine Spindelfunktion ersetzt werden. Mehrfachersetzungen führen zum Abbruch der Programmbearbeitung. Die Spindelfunktionen, die ersetzt werden sollen, müssen dann auf mehrere Sätze aufgeteilt werden.

**Parametrierung**

**Spindelfunktion**

Die Spindelfunktionen, die durch das Unterprogramm ersetzt werden sollen, werden ausgewählt im Maschinendatum:

MD30465 \$MA\_AXIS\_LANG\_SUB\_MASK

Bit	Bedeutung	
0	Getriebestufenwechsel automatisch (M40) und direkt (M41-M45)	
	Wert	Bedeutung
	0	Keine Ersetzung
	1	Ersetzung durch das in MD15700 und MD15702 eingestellte Unterprogramm
1	Spindelpositionieren mit SPOS / SPOSA / M19	
	Wert	Bedeutung
	0	Keine Ersetzung
	1	Ersetzung durch das in MD15700 und MD15702 eingestellte Unterprogramm

**Unterprogramm: Name**

Der Name des Ersetzungsunterprogramms wird eingetragen im Maschinendatum:

MD15700 \$MN\_LANG\_SUB\_NAME = "<Unterprogrammname>"

**Unterprogramm: Pfad**

Der Pfad des Ersetzungsunterprogramms wird eingestellt im Maschinendatum:

MD15702 \$MN\_LANG\_SUB\_PATH = <Wert>

Wert	Bedeutung
0	Hersteller-Zyklenverzeichnis: /_N_CMA_DIR
1	Anwender-Zyklenverzeichnis: /_N_CUS_DIR
2	Siemens-Zyklenverzeichnis: /_N_CST_DIR

**Systemvariable: Aufrufzeitpunkt des Ersetzungsunterprogramms**

Der Aufrufzeitpunkt des Ersetzungsunterprogramm kann über die Systemvariable \$P\_SUB\_STAT gelesen werden:

Wert	Bedeutung
0	Ersetzung nicht aktiv
1	Ersetzung aktiv, Unterprogrammaufruf am Satzanfang
2	Ersetzung aktiv, Unterprogrammaufruf am Satzende

### Satzbearbeitung

Wird das Ersetzungsunterprogramm am Satzanfang aufgerufen, wird nach der Bearbeitung des Ersetzungsunterprogramms der Satz, der zum Aufruf geführt hat, abgearbeitet. Die ersetzten Befehle werden nicht mehr bearbeitet.

Wird das Ersetzungsunterprogramm am Satzende aufgerufen, wird zuerst der Satz, der zum Aufruf des Ersetzungsunterprogramms führt, ohne die zu ersetzenden Befehle abgearbeitet. Anschließend wird das Ersetzungsunterprogramm aufgerufen.

## 8.15.3.2 Ersetzung von M40 - M45 (Getriebestufenwechsel)

### Funktion

Die Befehle zum Getriebestufenwechsel (M40, M41 ... M45) einer Leitspindel werden bei aktiver Kopplung durch den Aufruf eines anwenderspezifischen Unterprogramms ersetzt.

### Parametrierung

#### Aktivierung

- MD30465 \$MA\_AXIS\_LANG\_SUB\_MASK, Bit 0 = 1

#### Aufrufzeitpunkt des Unterprogramms

- M40

Der Aufrufzeitpunkt kann nicht eingestellt werden. Das Ersetzungsunterprogramm wird immer am Satzanfang aufgerufen.

- M41 ... M45

Der Aufrufzeitpunkt ist abhängig vom projektierten Ausgabeverhalten der Hilfsfunktion an die PLC (siehe unten MD22080):

- Ausgabe **vor** oder **während** der Bewegung: Unterprogrammaufruf am **Satzanfang**.
- Ausgabe **nach** der Bewegung: Unterprogrammaufruf am **Satzende**

MD22080 \$MC\_AUXFU\_PREDEF\_SPEC[12 ... 16] (Ausgabeverhalten für M41 ... M45)

Bit	Wert	Bedeutung
5	1	Ausgabe der Hilfsfunktion <b>vor</b> der Bewegung
6	1	Ausgabe der Hilfsfunktion <b>während</b> der Bewegung
7	1	Ausgabe der Hilfsfunktion <b>nach</b> der Bewegung

### Systemvariable zu Informationsweitergabe

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen über Systemvariable (siehe Kapitel "Systemvariable (Seite 687)") zur Verfügung gestellt. Die Daten beziehen sich ausschließlich auf den Satz, in dem die zu ersetzende Funktion programmiert wurde.

### 8.15.3.3 Ersetzung von SPOS, SPOSA, M19 (Spindelpositionieren)

#### Funktion

Die Positionierbefehle (*SPOS*, *SPOSA* oder *M19*) einer Leitspindel werden bei aktiver Kopplung durch den Aufruf eines anwenderspezifischen Unterprogramms (Ersetzungsunterprogramm) ersetzt.

#### Anwendungsbeispiel

Bei der Parallelbearbeitung von Werkstücken an einer Doppelspindel-Maschine sind die Spindeln über einen Kopplungsfaktor ungleich 1 gekoppelt. Zum Werkzeugwechsel müssen sie auf dieselbe Position positioniert werden. Das Ersetzungsunterprogramm schaltet dazu die Kopplung aus, positioniert die Spindeln separat auf die Werkzeugwechselposition und schaltet anschließend die Kopplung wieder ein.

#### Parametrierung

##### Aktivierung

- MD30465 \$MA\_AXIS\_LANG\_SUB\_MASK, Bit 1 = 1

##### Aufrufzeitpunkt des Ersetzungsunterprogramms

- *SPOS*, *SPOSA*

Der Aufrufzeitpunkt kann nicht eingestellt werden. Das Ersetzungsunterprogramm wird immer am Satzanfang aufgerufen.

- *M19*

Der Aufrufzeitpunkt ist abhängig vom projektierten Ausgabeverhalten der Hilfsfunktion an die PLC (siehe unten MD22080):

- Ausgabe **vor** oder **während** der Bewegung: Unterprogrammaufruf am **Satzanfang**.
- Ausgabe **nach** der Bewegung: Unterprogrammaufruf am **Satzende**

MD22080 \$MC_AUXFU_PREDEF_SPEC[9]		
Bit	Wert	Bedeutung
5	1	Ausgabe der Hilfsfunktion <b>vor</b> der Bewegung
6	1	Ausgabe der Hilfsfunktion <b>während</b> der Bewegung
7	1	Ausgabe der Hilfsfunktion <b>nach</b> der Bewegung

#### Systemvariable zur Informationsweitergabe

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen über Systemvariable (siehe Kapitel "Systemvariable (Seite 687)") zur Verfügung gestellt. Die Daten beziehen sich ausschließlich auf den Satz, in dem die zu ersetzende Funktion programmiert wurde.

### 8.15.3.4 Systemvariable

Systemvariable	Bedeutung	
\$P_SUB_AXFCT	TRUE, wenn M40, M41 ... M45 Ersetzung aktiv ist	
\$P_SUB_GEAR	Programmierte bzw. errechnete Getriebestufe Außerhalb des Ersetzungsunterprogramms: Getriebestufe der Masterspindel	
\$P_SUB_AUTOGEAR	TRUE, wenn im Satz der den Ersetzungsvorgang ausgelöst hat, M40 aktiv war. Außerhalb des Ersetzungsunterprogramms: aktuelle Einstellung im Interpreter	
\$P_SUB_LA	Enthält den Achsbezeichner der Leitspindel der aktiven Kopplung, die den Ersetzungsvorgang ausgelöst hat. <b>Hinweis</b> Wird die Variable außerhalb des Ersetzungsunterprogramms verwendet, wird die Programmbearbeitung mit Alarm abgebrochen.	
\$P_SUB_CA	Enthält den Achsbezeichner der Folgespindel der aktiven Kopplung, die den Ersetzungsvorgang ausgelöst hat. <b>Hinweis</b> Wird die Variable außerhalb des Ersetzungsunterprogramms aufgerufen, wird die Programmbearbeitung mit Alarm abgebrochen.	
\$P_SUB_AXFCT	Enthält die aktiven Ersetzungsarten entsprechend MD30465 \$MA_AXIS_LANG_SUB_MASK	
\$P_SUB_SPOS	TRUE, wenn die SPOS-Ersetzung aktiv ist	
\$P_SUB_SPOSA	TRUE, wenn die SPOSA-Ersetzung aktiv ist	
\$P_SUB_M19	TRUE, wenn die M19-Ersetzung aktiv ist	
\$P_SUB_SPOSIT	Enthält die programmierte Spindelposition <b>Hinweis</b> Wird die Variable außerhalb des Ersetzungsunterprogramms aufgerufen, wird die Programmbearbeitung mit Alarm abgebrochen.	
\$P_SUB_SPOSMODE	Enthält den Positionsanfahrmodus für die prog. Spindelposition:	
	<b>Wert</b>	<b>Bedeutung</b>
	0	keine Änderung des Positionsanfahrmodus
	1	AC
	2	IC
	3	DC
	4	ACP
	5	ACN
	6	OC
	7	PC
<b>Hinweis:</b> Wird die Variable außerhalb des Ersetzungsunterprogramms aufgerufen, wird die Programmbearbeitung mit Alarm abgebrochen.		
\$P_SUB_STAT	Satzbezogener Aufrufzeitpunkt des Ersetzungsunterprogramms	

### 8.15.3.5 Beispiel: Getriebestufenwechsel

Im Unterprogramm werden alle Befehle zum Getriebestufenwechsel M40, M41 ... M45 ersetzt.

#### Parametrierung

Maschinendatum	Bedeutung
MD15700 \$MN_LANG_SUB_NAME = "LANG_SUB"	Unterprogramm
MD15702 \$MN_LANG_SUB_PATH = 0	Herstellerverzeichnis
MD22080 \$MC_AUXFU_PREDEF_SPEC[12] = 'H21'	M41: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[13] = 'H21'	M42: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[13] = 'H21'	M43: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[15] = 'H21'	M44: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[16] = 'H21'	M45: Ausgabe vor der Bewegung
MD30465 \$MA_AXIS_LANG_SUB_MASK[AX5] = 'H0001'	Getriebewechselbefehle ersetzen

#### Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
N110 COUPON(S2,S1)	; Synchronspindelkopplung einschalten
N120 G01 F100 X100 S5000 M3 M43	; Unterprog.Aufruf aufgrund M43
N130 M40	; Automatischer Getriebestufenwechsel einschalten
N140 M3 S1000	; Unterprog.Aufruf aufgrund S1000 ; und dadurch ausgelöstem autom. ; Getriebestufenwechsel
N9999 M30	



### Ersetzungsunterprogramm "LANG\_SUB", Variante 1

Optimiert auf Einfachheit und Geschwindigkeit durch direkte Adressierung der Spindeln (S1: Leitspindel, S2: Folgespindel).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N1100 IF(\$P_SUB_AXFCT ==1)	; Ersetzung wegen Getriebestufenwechsel
N1140 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N1150 COUPOF(S2,S1)	; Synchronspindelkopplung ausschalten
N1160 ;Getriebestufenwechsel getrennt für Leit- und Folgespindel	
N1170 M1=\$P_SUB_GEAR M2=\$P_SUB_GEAR	
N1180 DELAYFSTON	; Ende Stopp-Delay-Bereich
N1190 COUPON(S2,S1)	; Synchronspindelkopplung einschalten
N1200 ENDIF	
...	
N9999 RET	

### Ersetzungsunterprogramm "LANG\_SUB", Variante 2

Flexibilität durch indirekte Adressierung über Systemvariable (Leitspindel: \$P\_SUB\_LA, Folgespindel: \$P\_SUB\_CA).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N1010 DEF AXIS _LA	; Merker für Leitachse/-spindel
N1020 DEF AXIS _CA	; Merker für Folgeachse/-spindel
N1030 DEF INT _GEAR	; Merker für Getriebestufe
N1100 IF(\$P_SUB_AXFCT==1)	; Ersetzung wegen Getriebestufenwechsel
N1110 _GEAR=\$P_SUB_GEAR	; zu aktivierende Getriebestufe
N1120 _LA=\$P_SUB_LA	; Achsbezeichner der Leitspindel
N1130 _CA=\$P_SUB_CA	; Achsbezeichner der Folgespindel
N1140 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N1150 COUPOF(_CA,_LA)	; Synchronspindelkopplung ausschalten
N1160 ;Getriebestufenwechsel für Leit- und Folgespindel	
N1170 M[AXTOSPI(_LA)]=_GEAR M[AXTOSPI(_CA)]=_GEAR	
N1180 DELAYFSTOF	; Ende Stopp-Delay-Bereich
N1190 COUPON(_CA,_LA)	; Synchronspindelkopplung einschalten
N1200 ENDIF	
...	
N9999 RET	

### 8.15.3.6 Beispiel: Spindelpositionieren

Im Unterprogramm ist nur die Ersetzung der Befehle SPOS und SPOSA explizit ausgeführt. Weitere Ersetzungen sind sinngemäß zu ergänzen.

#### Parametrierung

##### Maschinendatum

MD30465 \$MA\_AXIS\_LANG\_SUB\_MASK[AX5] = 'H0002'

MD22080 \$MC\_AUXFU\_PREDEF\_SPEC[9] = 'H0021'

##### Bedeutung

Positionierbefehle ersetzen

Ausgabe von M19 an die PLC vor der Bewegung

##### Settingdatum

SD43240 \$SA\_M19\_SPOS[AX5] = 260

SD43250 \$SA\_M19\_SPOSMODE[AX5] = 4

##### Bedeutung

Spindelposition bei M19 = 260

Positionsanfahrmodus bei M19: "Anfahren in positiver Richtung (ACP)"

#### Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
...	
N210 COUPON(S2,S1)	; Synchronspindelkopplung aktivieren
N220 SPOS[1]=100	; Leitspindel mit SPOS positionieren
...	
N310 G01 F1000 X100 M19	; Leitspindel mit M19 positionieren

## Ersetzungsunterprogramm "LANG\_SUB", Variante 1

Optimiert auf Einfachheit und Geschwindigkeit durch direkte Adressierung der Spindeln (S1: Leitspindel, S2: Folgespindel).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N2100 IF(\$P_SUB_AXFCT==2)	
N2110 ;Ersetzung von SPOS/SPOSA/M19 bei aktiver Synchronspindelkopplung	
N2185 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N2190 COUPOF(S2,S1)	; Synchronspindelkopplung ausschalten
N2200	; Leit- und Folgespindel positionieren
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220 ;SPOS und SPOSA werden auf SPOS abgebildet	
N2230 CASE \$P_SUB_SPOSMODE OF \	
0 GOTOF LABEL1_DC \	
1 GOTOF LABEL1_IC \	
2 GOTOF LABEL1_AC \	
3 GOTOF LABEL1_DC \	
4 GOTOF LABEL1_ACP \	
5 GOTOF LABEL1_ACN \	
DEFAULT GOTOF LABEL_ERR	
LABEL1_DC: SPOS[1]=DC(\$P_SUB_SPOSIT) SPOS[2]=DC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_IC: DELAYFSTOF	
SPOS[1]=IC(\$P_SUB_SPOSIT) SPOS[2]=IC(\$P_SUB_SPOSIT)	
DELAYFSTON	
GOTOF LABEL1_CONT	
LABEL1_AC: SPOS[1]=AC(\$P_SUB_SPOSIT) SPOS[2]=AC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	

8.15 Ersetzung von Funktionen durch Unterprogramme

Programmierung	Kommentar
LABEL1_ACP: SPOS[1]=ACP(\$P_SUB_SPOSIT) SPOS[2]=ACP(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACN: SPOS[1]=ACN(\$P_SUB_SPOSIT) SPOS[2]=ACN(\$P_SUB_SPOSIT)	
LABEL1_CONT:	
N2250 ELSE	; Spindel mit M19 positionieren
N2270 M1=19 M2=19	; Leit- und Folgespindel
N2280 ENDIF	; Ende Ersetzung SPOS, SPOSA
N2285 DELAYFSTOF	; Ende Stopp-Delay-Bereich
N2290 COUPON(S2,S1)	; Synchronspindelkopplung aktivieren
N2410 ELSE	
N2420 ;ab hier Bearbeitung weiterer Ersetzungen	
...	
N3300 ENDIF	; Ende Ersetzungen
...	
N9999 RET	; Normales Programmende
LABEL_ERR: SETAL(61000)	; Fehler aufgetreten

Ersetzungsunterprogramm "LANG\_SUB", Variante 2

Flexibilität durch indirekte Adressierung über Systemvariable (Leitspindel: \$P\_SUB\_LA, Folgespindel: \$P\_SUB\_CA).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N1010 DEF AXIS _LA	; Leitachse/-spindel
N1020 DEF AXIS _CA	; Folgeachse/-spindel
N1030 DEF INT _LSPI	; Leitspindelnummer (programmierte ; Spindel)
N1040 DEF INT _CSPI	; Folgespindelnummer
...	
N2100 IF(\$P_SUB_AXFCT==2)	
N2110 ;Ersetzung von SPOS/SPOSA/M19 bei aktiver Synchronspindelkopplung	
N2120 _LA=\$P_SUB_LA	; Achsbezeichner der Leitspindel
N2130 _CA=\$P_SUB_CA	; Achsbezeichner der Folgespindel
N2140 _LSPI=AXTOSPI(_LA)	; Nummer der Leitspindel
N2180 _CSPI=AXTOSPI(_LA)	; Nummer der Folgespindel
N2185 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N2190 COUPOF(_CA,_LA)	; Synchronspindelkopplung deaktivieren
N2200	; Leit- und Folgespindel positionieren:
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220 ;SPOS und SPOSA werden auf SPOS abgebildet	
N2230 CASE \$P_SUB_SPOSMODE OF	
0 GOTOF LABEL1_DC \	
1 GOTOF LABEL1_IC \	

Programmierung	Kommentar
2 GOTOF LABEL1_AC \	
3 GOTOF LABEL1_DC \	
4 GOTOF LABEL1_ACP \	
5 GOTOF LABEL1_ACN \	
DEFAULT GOTOF LABEL_ERR	
LABEL1_DC: SPOS[_LSPI]=DC(\$P_SUB_SPOSIT) SPOS[_CSPI]=DC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_IC: DELAYFSTOF	
SPOS[_LSPI]=IC(\$P_SUB_SPOSIT) SPOS[_CSPI]=IC(\$P_SUB_SPOSIT)	
DELAYFSTON	
GOTOF LABEL1_CONT	
LABEL1_AC: SPOS[_LSPI]=AC(\$P_SUB_SPOSIT) SPOS[_CSPI]=AC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACP: SPOS[_LSPI]=ACP(\$P_SUB_SPOSIT) POS[_CSPI]=ACP(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACN: SPOS[_LSPI]=ACN(\$P_SUB_SPOSIT) POS[_CSPI]=ACN(\$P_SUB_SPOSIT)	
LABEL1_CONT:	
N2250 ELSE	; Spindeln mit M19 positionieren
N2270 M[_LSPI]=19 M[_CSPI]=19	
N2280 ENDIF	
N2285 DELAYFSTOF	; Ende Stopp-Delay-Bereich
N2290 COUPON(_CA,_LA)	; Synchronspindelkopplung aktivieren
N2410 ELSE	
N2420 ;ab hier Bearbeitung weiterer Ersetzungen	
...	
N3300 ENDIF	
...	
N9999 RET	; Normales Programmende
LABEL_ERR: SETAL(61000)	; Fehler aufgetreten

### 8.15.4 Eigenschaften der Unterprogramme

#### Allgemeine Regeln

- Das bei der Ersetzung aufgerufene Unterprogramm kann den Befehl `PROC` und die Attribute `SBLOF` und `DISPLOF` enthalten.
- Die Ersetzung wird auch im ISO-Sprachmode ausgeführt. Die Ersetzungsunterprogramme werden aber ausschließlich im Standard-Sprachmode (Siemens) abgearbeitet. Dabei erfolgt eine implizite Umschaltung in den Standard-Sprachmode. Mit Rücksprung aus dem Ersetzungsunterprogramm wird wieder in den ursprünglichen Sprachmode zurückgeschaltet.
- Die Weitergabe von Informationen an das Ersetzungsunterprogramm erfolgt ausschließlich über Systemvariablen. Übergabeparameter sind nicht möglich.
- Das Verhalten bei Einzelsatz und Attribut `SBLOF` ist abhängig von der Einstellung in: MD10702 `IGNORE_SINGLEBLOCK_MASK`, Bit 14 (Einzelsatzstopp verhindern)

Wert	Bedeutung
0	<p>Das Ersetzungsunterprogramm verhält sich wie ein "normales" Unterprogramm:</p> <ul style="list-style-type: none"> <li>• Rücksprung mit <code>M17</code>: Stopp am Ende des Unterprogramms</li> </ul> <p><b>Hinweis</b></p> <p>Die Ausgabe der M-Funktion an die PLC ist abhängig von:  MD20800 <code>\$MC_SPF_END_TO_VDI</code>, Bit 0 (Unterprogrammende an PLC)</p> <ul style="list-style-type: none"> <li>- Bit 0 = 0: keine Ausgabe</li> <li>- Bit 0 = 1: M17 wird an die PLC ausgegeben.</li> </ul> <ul style="list-style-type: none"> <li>• Rücksprung mit <code>RET</code>: Kein Stopp am Ende des Ersetzungsunterprogramms</li> </ul>
1	<p>Im Satz, in dem das Ersetzungsunterprogramm aufgerufen wird, wird <b>nur einmal angehalten</b>. Unabhängig davon ob:</p> <ul style="list-style-type: none"> <li>• Das Unterprogramm am Satzanfang und/oder am Satzende aufgerufen wird</li> <li>• Im Unterprogramm weitere Unterprogramme aufgerufen werden</li> <li>• Das Unterprogramm mit <code>M17</code> oder <code>RET</code> verlassen wird</li> </ul> <p>Der Einzelsatzstopp erfolgt für die Ersetzung von M-Funktionen am Ende des Ersetzungsunterprogramms.</p> <p>Für die Ersetzung von T- und D/DL-Funktionen ist der Zeitpunkt des Einzelsatzstops abhängig vom Aufrufzeitpunkt des Unterprogramms:</p> <ul style="list-style-type: none"> <li>• Aufruf am Satzanfang: Einzelsatzstopp am Ende des Satzes</li> <li>• Aufruf am Satzende: Einzelsatzstopp am Ende des Ersetzungsunterprogramms</li> </ul>

- Bei Ersetzungsunterprogrammen mit dem Attribut `DISPLOF` wird in der Satzanzeige als aktueller Satz die Programmzeile angezeigt, die zum Aufruf des Unterprogramms geführt hat.
- Im Ersetzungsunterprogramm können mit den Befehlen `DELAYFSTON` und `DELAYFSTOF` Bereiche oder das gesamte Ersetzungsunterprogramm gegen Unterbrechungen, wie NC-Stop, Einleseperrre usw., geschützt werden.
- Ersetzungen erfolgen nicht rekursiv. D. h. die Funktion, die zum Aufruf des Ersetzungsunterprogramms geführt hat, wird bei nochmaliger Programmierung im Ersetzungsunterprogramm nicht mehr ersetzt.

### Hilfsfunktionsausgabe an PLC

Bei Ersetzungen von Hilfsfunktionen bewirkt der Aufruf des Ersetzungsunterprogramms noch keine Ausgabe der Hilfsfunktion an die PLC. Die Ausgabe der Hilfsfunktion erfolgt erst wenn die Hilfsfunktion im Ersetzungsunterprogramm erneut programmiert wird.

### Verhalten bei Satzsuchlauf

Das Ersetzungsunterprogramm wird auch in den Suchlaufmodi "Satzsuchlauf mit Berechnung" und "Satzsuchlauf mit Berechnung im Modus Programmtest" (SERUPRO) aufgerufen. Eventuelle Besonderheiten müssen im Ersetzungsunterprogramm unter Verwendung der Systemvariable: `$P_SEARCH`, `$AC_SERUPRO` realisiert werden.

Bezüglich des Aufsammelns von Aktionen beim "Suchlauf mit Berechnung" verhalten sich Ersetzungsunterprogramme wie "normale" Unterprogramme.

### 8.15.5 Randbedingungen

- Funktionsersetzungen sind unzulässig in:
  - Synchronaktionen
  - Technologiezyklen
- Einem Satz, der am Anfang zu ersetzende Funktionen enthält, dürfen keine satzweisen Synchronaktionen vorangestellt werden. Siehe unten Absatz "Beispiel zu: Satzweise Synchronaktionen".
- Im Ersetzungsunterprogramm dürfen nur die für die jeweiligen Ersetzungen notwendigen Aktionen durchgeführt werden.
- In einem Satz, in dem das Ersetzungsunterprogramm am Satzende aufgerufen wird, ist folgendes zu beachten:
  - Es darf kein modaler Unterprogrammaufruf aktiv sein
  - Es darf kein Unterprogrammrückprung programmiert sein
  - Es darf kein Programmende programmiert sein

<b>ACHTUNG</b>
Die Steuerung überwacht nicht, ob die zu ersetzende Funktion im Ersetzungsunterprogramm realisiert wurde.

#### Beispiel zu: Satzweise Synchronaktionen

MD30465 \$MA\_AXIS\_LANG\_SUB\_MASK, Bit 0 = 1 (Getriebestufenwechsel)

```
Programmcode
...
N1000 WHENEVER $AA_IM[X2] <= $AA_IM[X1] + 0.5 DO $AA_OVR[X1]=0
N1010 G1 X100 M43
...
```

Führt in Satz N1010 die Funktion M43 zum Aufruf eines Ersetzungsunterprogramms, wird die Bearbeitung mit abgebrochen und ein Alarm angezeigt.



## 8.16 Programmlaufzeit / Werkstückzähler

Zur Unterstützung des Werkzeugmaschinenbedieners werden Informationen zur Programmlaufzeit und Werkstückzahl bereitgestellt.

Diese Informationen können als Systemvariablen im NC- und/oder PLC-Programm bearbeitet werden. Gleichzeitig stehen diese Informationen für die Anzeige auf der Bedienoberfläche zur Verfügung.

### 8.16.1 Programmlaufzeit

#### Funktion

Die Funktion "Programmlaufzeit" stellt NC-interne Timer zur Überwachung technologischer Prozesse zur Verfügung, die über NC- und Kanal-spezifische Systemvariablen im Teileprogramm und in Synchronaktionen gelesen werden können.

#### Zeit seit dem letzten Steuerungshochlauf

Die Timer für die Messung der Zeit seit dem letzten Steuerungshochlauf sind immer aktiv und können über NC-spezifische Systemvariablen gelesen werden:

Systemvariable	Bedeutung
\$AN_SETUP_TIME	Zeit seit dem letzten Steuerungshochlauf mit Standardwerten ("Kaltstart") in Minuten Wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.
\$AN_POWERON_TIME	Zeit seit dem letzten Normalhochlauf der Steuerung ("Warmstart") in Minuten Wird bei jedem Normalhochlauf der Steuerung automatisch auf "0" zurückgesetzt.

#### Programmlaufzeiten

Die Timer zur Messung der Programmlaufzeiten sind:

- Nur in der Betriebsart AUTOMATIK verfügbar
- Über Kanal-spezifische Systemvariablen lesbar

Einige der Timer sind immer aktiv, andere sind durch MD-Parametrierung aktivierbar / deaktivierbar.

Für die Timer, die immer aktiv sind, stehen folgende Systemvariablen zur Verfügung:

Systemvariable	Bedeutung
\$AC_ACT_PROG_NET_TIME	<p>Aktuelle Netto-Laufzeit des aktuellen Programms in Sekunden</p> <p>Netto-Laufzeit bedeutet, dass die Zeit, in der das Programm gestoppt war, abgezogen ist.</p> <p>Wird in der Betriebsart AUTOMATIK aus dem Kanalzustand RESET heraus ein Teileprogramm neu gestartet, wird \$AC_ACT_PROG_NET_TIME automatisch auf "0" zurückgesetzt.</p> <p>Weitere Eigenschaften:</p> <ul style="list-style-type: none"> <li>• Die Reset-Taste setzt \$AC_ACT_PROG_NET_TIME nicht auf "0" zurück, sondern hält den Timer nur an.</li> <li>• Beim Start eines ASUPs wird \$AC_ACT_PROG_NET_TIME auf "0" gesetzt und zählt die Laufzeit des ASUPs mit. Am Ende eines ASUPs verhält es sich wie bei der RESET-Taste: der Timer wird nur angehalten, aber nicht auf "0" gesetzt.</li> <li>• Beim Start eines ereignisgesteuerten Programms (PROG_EVENT) wird \$AC_ACT_PROG_NET_TIME <b>nicht</b> zurückgesetzt.</li> </ul> <p>Die Programmlaufzeit wird nur weitergezählt, wenn es sich um ein Start-, M30- oder Suchlauf-PROG_EVENT handelt.</p> <ul style="list-style-type: none"> <li>• Das Verhalten von \$AC_ACT_PROG_NET_TIME bei GOTOS und Override = 0% ist über MD27850 parametrierbar (siehe Abschnitt "Parametrierung")</li> </ul> <p><b>Hinweis:</b> Mit \$AC_PROG_NET_TIME_TRIGGER kann \$AC_ACT_PROG_NET_TIME weiter manipuliert werden.</p>
\$AC_OLD_PROG_NET_TIME	<p>Netto-Laufzeit des gerade korrekt beendeten Programms in Sekunden</p> <p>"Korrekt beendet" bedeutet, dass das Programm nicht mit RESET abgebrochen, sondern regulär mit M30 beendet wurde.</p> <p>Wird ein neues Programm gestartet, bleibt \$AC_OLD_PROG_NET_TIME unangetastet, bis M30 erneut erreicht wird.</p> <p>Weitere Eigenschaften:</p> <ul style="list-style-type: none"> <li>• \$AC_OLD_PROG_NET_TIME wird auf "0" gesetzt, wenn das aktuell angewählte Programm editiert wird.</li> <li>• Am Ende eines ASUPs oder eines ereignisgesteuerten Programms (PROG_EVENT) wird \$AC_OLD_PROG_NET_TIME nicht verändert.</li> </ul> <p><b>Hinweis:</b> Der implizite Kopiervorgang von \$AC_ACT_PROG_NET_TIME nach \$AC_OLD_PROG_NET_TIME findet nur statt, wenn \$AC_PROG_NET_TIME_TRIGGER nicht beschrieben wird.</p>

Systemvariable	Bedeutung										
\$AC_OLD_PROG_NET_TIME_COUNT	<p>Änderungen auf \$AC_OLD_PROG_NET_TIME</p> <p>Nach POWER ON steht \$AC_OLD_PROG_NET_TIME_COUNT auf "0".</p> <p>\$AC_OLD_PROG_NET_TIME_COUNT wird immer dann erhöht, wenn die Steuerung \$AC_OLD_PROG_NET_TIME neu geschrieben hat.</p> <p>Wenn der Anwender das laufende Programm mit RESET abbricht, bleiben \$AC_OLD_PROG_NET_TIME und \$AC_OLD_PROG_NET_TIME_COUNT unverändert.</p> <p>Mit \$AC_OLD_PROG_NET_TIME_COUNT lässt sich daher feststellen, ob \$AC_OLD_PROG_NET_TIME geschrieben wurde.</p> <p>Beispiel:</p> <p>Wenn zwei hintereinander laufende Programme die identische Laufzeit haben und korrekt beendet wurden, dann kann der Anwender dies über den veränderten Wert in \$AC_OLD_PROG_NET_TIME_COUNT erkennen.</p>										
\$AC_PROG_NET_TIME_TRIGGER	<p>Trigger zur Laufzeitmessung</p> <p>Dient zur selektiven Messung von Programmabschnitten, d. h. durch Beschreiben von \$AC_PROG_NET_TIME_TRIGGER im NC-Programm kann die Zeitmessung ein- und wieder ausgeschaltet werden:</p> <ol style="list-style-type: none"> <li>1. \$AC_PROG_NET_TIME_TRIGGER = 2 startet die Messung und setzt dabei \$AC_ACT_PROG_NET_TIME auf "0".</li> <li>2. \$AC_PROG_NET_TIME_TRIGGER = 1 beendet die Messung und kopiert den Wert aus \$AC_ACT_PROG_NET_TIME in \$AC_OLD_PROG_NET_TIME.</li> </ol> <p>Um alle Trigger-Möglichkeiten auszuschöpfen, sind bestimmte Werte für \$AC_PROG_NET_TIME_TRIGGER mit besonderen Funktionen belegt:</p> <table border="1"> <tbody> <tr> <td>0</td> <td>Neutraler Zustand Der Trigger ist nicht aktiv.</td> </tr> <tr> <td>1</td> <td>Beenden Beendet die Messung und kopiert den Wert aus \$AC_ACT_PROG_NET_TIME in \$AC_OLD_PROG_NET_TIME. \$AC_ACT_PROG_NET_TIME wird auf "0" gesetzt und läuft danach weiter.</td> </tr> <tr> <td>2</td> <td>Start Startet die Messung und setzt dabei \$AC_ACT_PROG_NET_TIME auf "0". \$AC_OLD_PROG_NET_TIME wird nicht verändert.</td> </tr> <tr> <td>3</td> <td>Stopp Stoppt die Messung. Verändert \$AC_OLD_PROG_NET_TIME nicht und hält \$AC_ACT_PROG_NET_TIME bis zum Fortsetzen konstant.</td> </tr> <tr> <td>4</td> <td>Fortsetzen Fortsetzen der Messung, d. h. eine vorher gestoppte Messung wird wieder aufgenommen. \$AC_ACT_PROG_NET_TIME läuft weiter. \$AC_OLD_PROG_NET_TIME wird nicht verändert.</td> </tr> </tbody> </table>	0	Neutraler Zustand Der Trigger ist nicht aktiv.	1	Beenden Beendet die Messung und kopiert den Wert aus \$AC_ACT_PROG_NET_TIME in \$AC_OLD_PROG_NET_TIME. \$AC_ACT_PROG_NET_TIME wird auf "0" gesetzt und läuft danach weiter.	2	Start Startet die Messung und setzt dabei \$AC_ACT_PROG_NET_TIME auf "0". \$AC_OLD_PROG_NET_TIME wird nicht verändert.	3	Stopp Stoppt die Messung. Verändert \$AC_OLD_PROG_NET_TIME nicht und hält \$AC_ACT_PROG_NET_TIME bis zum Fortsetzen konstant.	4	Fortsetzen Fortsetzen der Messung, d. h. eine vorher gestoppte Messung wird wieder aufgenommen. \$AC_ACT_PROG_NET_TIME läuft weiter. \$AC_OLD_PROG_NET_TIME wird nicht verändert.
0	Neutraler Zustand Der Trigger ist nicht aktiv.										
1	Beenden Beendet die Messung und kopiert den Wert aus \$AC_ACT_PROG_NET_TIME in \$AC_OLD_PROG_NET_TIME. \$AC_ACT_PROG_NET_TIME wird auf "0" gesetzt und läuft danach weiter.										
2	Start Startet die Messung und setzt dabei \$AC_ACT_PROG_NET_TIME auf "0". \$AC_OLD_PROG_NET_TIME wird nicht verändert.										
3	Stopp Stoppt die Messung. Verändert \$AC_OLD_PROG_NET_TIME nicht und hält \$AC_ACT_PROG_NET_TIME bis zum Fortsetzen konstant.										
4	Fortsetzen Fortsetzen der Messung, d. h. eine vorher gestoppte Messung wird wieder aufgenommen. \$AC_ACT_PROG_NET_TIME läuft weiter. \$AC_OLD_PROG_NET_TIME wird nicht verändert.										
Durch POWER ON werden alle Systemvariablen auf "0" zurückgesetzt!											

---

**Hinweis**

**Restzeit für ein Werkstück**

Wenn nacheinander gleiche Werkstücke produziert werden, kann aus den Timerwerten:

- Bearbeitungszeit für das zuletzt produzierte Werkstück (siehe \$AC\_OLD\_PROG\_NET\_TIME)

und

- Aktuelle Bearbeitungszeit (siehe \$AC\_ACT\_PROG\_NET\_TIME)

die verbleibende Restzeit für ein Werkstück ermittelt werden.

Die Restzeit wird zusätzlich zur aktuellen Bearbeitungszeit auf der Bedienoberfläche angezeigt.

---

**ACHTUNG**

**Verwendung von STOPRE**

Die Systemvariablen \$AC\_OLD\_PROG\_NET\_TIME und \$AC\_OLD\_PROG\_NET\_TIME\_COUNT erzeugen keinen impliziten Vorlaufstopp. Bei der Verwendung im Teileprogramm ist das unkritisch, wenn der Wert der Systemvariablen aus dem vorangegangenen Programmlauf stammt. Wenn aber der Trigger zur Laufzeitmessung (\$AC\_PROG\_NET\_TIME\_TRIGGER) hochfrequent geschrieben wird und sich dadurch \$AC\_OLD\_PROG\_NET\_TIME sehr oft ändert, dann sollte im Teileprogramm ein explizites STOPRE verwendet werden.

Für die Timer, die durch MD-Parametrierung aktiviert / deaktiviert werden, stehen folgende Systemvariablen zur Verfügung:

Systemvariable	Bedeutung
\$AC_OPERATING_TIME	<p>Gesamtlaufzeit von NC-Programmen in der Betriebsart Automatik in Sekunden</p> <p>Aufsummiert werden in der Betriebsart Automatik die Laufzeiten aller Programme zwischen NC-Start und Programmende / NC-Reset.</p> <p>Bei NC-Stop und Override = 0% wird standardmäßig nicht gezählt. Das Weiterzählen bei Override 0% kann über MD27860 aktiviert werden.</p> <p>Der Wert wird mit jedem Steuerungshochlauf automatisch auf "0" zurückgesetzt.</p>
\$AC_CYCLE_TIME	<p>Laufzeit des angewählten NC-Programms in Sekunden</p> <p>Im angewählten NC-Programm wird die Laufzeit zwischen NC-Start und Programmende / NC-Reset gemessen.</p> <p>Bei NC-Stop und Override = 0% wird standardmäßig nicht gezählt. Das Weiterzählen bei Override 0% kann über MD27860 aktiviert werden.</p> <p>Der Wert wird mit dem Start eines neuen NC-Programms automatisch auf "0" zurückgesetzt. Über MD27860 ist einstellbar, ob diese Löschung auch bei einem Sprung auf den Programmanfang mit GOTOS oder bei Start von ASUPs und PROG_EVENTS erfolgen soll.</p>
\$AC_CUTTING_TIME	<p>Bearbeitungszeit in Sekunden</p> <p>Gemessen wird die Laufzeit der Bahnachsen (mindestens eine ist aktiv) ohne aktiven Eilgang in allen NC-Programmen zwischen NC-Start und Programmende / NC-Reset. Über MD27860 ist einstellbar, ob nur mit aktivem Werkzeug oder werkzeugunabhängig gemessen werden soll.</p> <p>Die Messung wird zusätzlich bei aktiver Verweilzeit unterbrochen.</p> <p>Der Wert wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.</p>

### Aktivierung / Deaktivierung

Das Ein-/Ausschalten der aktivierbaren Timer erfolgt über das Maschinendatum:

MD27860 \$MC\_PROCESSTIMER\_MODE, Bit 0-2

Bit	Wert	Bedeutung
0	0	Timer für \$AC_OPERATING_TIME nicht aktiv.
	1	Timer für \$AC_OPERATING_TIME aktiv.
1	0	Timer für \$AC_CYCLE_TIME nicht aktiv.
	1	Timer für \$AC_CYCLE_TIME aktiv.
2	0	Timer für \$AC_CUTTING_TIME nicht aktiv.
	1	Timer für \$AC_CUTTING_TIME aktiv.

### Parametrierung

#### Verhalten der immer aktiven Timer

Das Verhalten der immer aktiven Timer bei GOTOS und Override = 0% wird eingestellt mit dem Maschinendatum:

MD27850 \$MC\_PROG\_NET\_TIMER\_MODE

Bit	Wert	Bedeutung
0	0	\$AC_ACT_PROG_NET_TIME wird bei einem Sprung mit GOTOS auf den Programmanfang nicht auf "0" zurückgesetzt (Grundeinstellung).
	1	\$AC_ACT_PROG_NET_TIME wird bei einem Sprung mit GOTOS auf den Programmanfang auf "0" zurückgesetzt, der Wert vorher in \$AC_OLD_PROG_NET_TIME gespeichert und der Programmzähler \$AC_OLD_PROG_NET_TIME_COUNT erhöht.
1	0	\$AC_ACT_PROG_NET_TIME wird bei Override = 0% nicht erhöht. D. h. die Programmlaufzeit wird ohne die Zeit gemessen, für die der Override auf "0" eingestellt wurde (Grundeinstellung).
	1	\$AC_ACT_PROG_NET_TIME wird auch bei Override = 0% erhöht. D. h. die Programmlaufzeit wird mit der Zeit gemessen, für die der Override auf "0" eingestellt wurde.

**Verhalten der aktivierbaren Timer**

Das Verhalten der aktivierbaren Timer bei bestimmten Funktionen (z. B. Probelaufvorschub, Programmtest) wird eingestellt mit dem Maschinendatum:

MD27860 \$MC\_PROCESSTIMER\_MODE

Bit	Wert	Bedeutung
4	0	Keine Messung bei aktivem Probelaufvorschub.
	1	Messung auch bei aktivem Probelaufvorschub.
5	0	Keine Messung bei Programm-Test.
	1	Messung auch bei Programm-Test.
6	Nur bei Bit 1 = 1 (Timer für \$AC_CYCLE_TIME ist aktiv)	
	0	\$AC_CYCLE_TIME wird auch bei Start durch ASUP und PROG_EVENTS auf "0" zurückgesetzt.
	1	\$AC_CYCLE_TIME wird bei Start durch ASUP und PROG_EVENTS nicht auf "0" zurückgesetzt.
7	Nur bei Bit 2 = 1 (Timer für \$AC_CUTTING_TIME ist aktiv)	
	0	Timer für \$AC_CUTTING_TIME zählt nur bei aktivem Werkzeug.
	1	Timer für \$AC_CUTTING_TIME zählt werkzeugunabhängig.
8	Nur bei Bit 1 = 1 (Timer für \$AC_CYCLE_TIME ist aktiv)	
	0	\$AC_CYCLE_TIME wird bei einem Sprung mit GOTOS auf den Programmanfang nicht auf "0" zurückgesetzt.
	1	\$AC_CYCLE_TIME wird bei einem Sprung mit GOTOS auf den Programmanfang auf "0" zurückgesetzt.
9	Nur bei Bit 0, 1 = 1 (Timer für \$AC_OPERATING_TIME und \$AC_CYCLE_TIME sind aktiv)	
	0	Kein Weiterzählen der Programmlaufzeit bei Override = 0%
	1	Weiterzählen der Programmlaufzeit bei Override = 0%

**Randbedingungen**

- **Satzsuchlauf**

Bei Satzsuchlauf werden keine Programmlaufzeiten ermittelt.

- **REPOS**

Die Zeitdauer eines REPOS-Vorgangs wird der aktuellen Bearbeitungszeit (\$AC\_ACT\_PROG\_NET\_TIME) angerechnet.

## Beispiele

### Beispiel 1: Parametrierung der Laufzeitmessung über MD27860

- Aktivierung der Laufzeitmessung für das aktive NC-Programm, dabei kein Messen bei aktivem Probelaufvorschub und Programmtest:  
MD27860 \$MC\_PROCESSTIMER\_MODE = 'H2'
- Aktivierung der Messung für die Werkzeug-Eingriffszeit, dabei auch Messen bei aktivem Probelaufvorschub und Programmtest:  
MD27860 \$MC\_PROCESSTIMER\_MODE = 'H34'
- Aktivierung der Messung für die Gesamtlaufzeit und die Bearbeitungszeit mit aktivem Werkzeug, dabei auch Messen bei Programmtest:  
MD27860 \$MC\_PROCESSTIMER\_MODE = 'H25'
- Aktivierung der Messung für die Gesamtlaufzeit und die Bearbeitungszeit (werkzeugunabhängig), dabei auch Messen bei Programmtest:  
MD27860 \$MC\_PROCESSTIMER\_MODE = 'Ha5'
- Aktivierung der Messung für die Bearbeitungszeit mit aktivem Werkzeug, dabei auch Messen bei Override = 0% , aber nicht bei aktivem Probelauf-Vorschub:  
MD27860 \$MC\_PROCESSTIMER\_MODE = 'H22'

### Beispiel 2: Zeitdauer von "mySubProgrammA" messen

Programmcode
--------------

---

```
...  
N50 DO $AC_PROG_NET_TIME_TRIGGER=2  
N60 FOR ii= 0 TO 300  
N70 mySubProgrammA  
N80 DO $AC_PROG_NET_TIME_TRIGGER=1  
N95 ENDFOR  
N97 mySubProgrammB  
N98 M30
```



Nachdem das Programm die Zeile N80 verarbeitet hat, steht in \$AC\_OLD\_PROG\_NET\_TIME die Nettolaufzeit von "mySubProgrammA".

Der Wert von \$AC\_OLD\_PROG\_NET\_TIME:

- Bleibt über M30 hinaus erhalten
- Wird nach jedem Schleifendurchlauf aktualisiert

### Beispiel 3: Zeitdauer von "mySubProgrammA" und "mySubProgrammC" messen

```
Programcode
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

## 8.16.2 Werkstückzähler

### Funktion

Mit der Funktion "Werkstückzähler" stehen in Form von kanalspezifische Systemvariablen diverse Zähler mit einem Wertebereich von 0 bis 999.999.999 zur Verfügung. Auf die Systemvariablen kann lesend und schreibend zugegriffen werden.

Über die nachfolgend beschriebenen kanalspezifischen Maschinendaten kann auf die Zähler-Aktivierung, den Zeitpunkt der Nullung und den Zählalgorithmus Einfluss genommen werden.

**Systemvariablen zur Werkstückzählung**

Systemvariable	Bedeutung
\$AC_REQUIRED_PARTS	Anzahl der zu fertigenden Werkstücke (Soll-Werkstückzahl) In diesem Zähler kann die Anzahl der Werkstücke definiert werden, bei dessen Erreichen die Ist-Werkstückzahl (\$AC_ACTUAL_PARTS) auf "0" zurückgesetzt wird. Über MD27880 kann die Generierung des Anzeige-Alarms: "Kanal %1 Werkstueck-Soll = %2 erreicht" und des kanalspezifischen NC/PLC-Nahtstellensignals: DB21, ... DBX317.1 (Werkstück-Soll erreicht) aktiviert werden.
\$AC_TOTAL_PARTS	Anzahl der insgesamt gefertigten Werkstücke (Ist-Werkstückzahl Gesamt) Dieser Zähler gibt die Anzahl aller ab Startzeitpunkt gefertigten Werkstücke an. Der Wert wird nur bei einem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.
\$AC_ACTUAL_PARTS	Anzahl der gefertigten Werkstücke (Ist-Werkstückzahl) In diesem Zähler wird die Anzahl aller ab Startzeitpunkt gefertigten Werkstücke registriert. Bei einem Erreichen der Soll-Werkstückzahl (\$AC_REQUIRED_PARTS) wird der Zähler automatisch auf "0" zurückgesetzt (\$AC_REQUIRED_PARTS > 0 vorausgesetzt).
\$AC_SPECIAL_PARTS	Anzahl der vom Anwender gezählten Werkstücke Dieser Zähler erlaubt dem Anwender eine Werkstückzählung nach eigener Definition. Definiert werden kann eine Alarmausgabe bei Erreichen der Soll-Werkstückzahl (\$AC_REQUIRED_PARTS). Eine Nullung des Zählers muss der Anwender selbst vornehmen.

**Hinweis**

Alle Werkstückzähler werden bei einem Steuerungshochlauf mit Standardwerten, auf den Wert "0" gesetzt und können, unabhängig von ihrer Aktivierung, gelesen und geschrieben werden.

## Aktivierung

Die Werkstückzähler werden aktiviert mit dem Maschinendatum:

MD27880 \$MC\_PART\_COUNTER (Aktivierung der Werkstückzähler)

Bit	Wert	Bedeutung
0	1	\$AC_REQUIRED_PARTS ist aktiv.
1	0	Alarm- / Signalausgabe bei: \$AC_ACTUAL_PARTS = \$AC_REQUIRED_PARTS
	1	Alarm- / Signalausgabe bei: \$AC_SPECIAL_PARTS = \$AC_REQUIRED_PARTS
4	1	\$AC_TOTAL_PARTS ist aktiv.
5	0	\$AC_TOTAL_PARTS wird durch M02 / M30 um den Wert "1" erhöht.
	1	\$AC_TOTAL_PARTS wird durch den mit MD27882[0] definierten M-Befehl um den Wert "1" erhöht.
6	0	\$AC_TOTAL_PARTS ist auch bei Programm-Test / Satzsuchlauf aktiv.
7	1	\$AC_TOTAL_PARTS wird bei einem Rücksprung mit GOTOS um den Wert "1" erhöht.
8	1	\$AC_ACTUAL_PARTS ist aktiv.
9	0	\$AC_ACTUAL_PARTS wird durch M02 / M30 um den Wert "1" erhöht.
	1	\$AC_ACTUAL_PARTS wird durch den mit MD27882[1] definierten M-Befehl um den Wert "1" erhöht.
10	0	\$AC_ACTUAL_PARTS ist auch bei Programm-Test / Satzsuchlauf aktiv.
11	1	\$AC_ACTUAL_PARTS wird bei einem Rücksprung mit GOTOS um den Wert "1" erhöht.
12	1	\$AC_SPECIAL_PARTS ist aktiv.
13	0	\$AC_SPECIAL_PARTS wird durch M02 / M30 um den Wert "1" erhöht.
	1	\$AC_SPECIAL_PARTS wird durch den mit MD27882[2] definierten M-Befehl um den Wert "1" erhöht.
14	0	\$AC_SPECIAL_PARTS ist auch bei Programm-Test / Satzsuchlauf aktiv.
15	1	\$AC_SPECIAL_PARTS wird bei einem Rücksprung mit GOTOS um den Wert "1" erhöht.

## Werkstückzählung mit anwenderdefiniertem M-Befehl

Ist das entsprechende Bit in MD27880 gesetzt, wird der Zählimpuls statt über das Programmende  $M2/M30$  über einen über folgendes Maschinendatum parametrierbaren M-Befehl ausgelöst:

MD27882 \$MC\_PART\_COUNTER\_MCODE[<n>] (Werkstückzählung mit anwenderdefiniertem M-Befehl)

<n>	Bedeutung
0	MD27882[0] definiert den M-Befehl, bei dem \$AC_TOTAL_PARTS hochgezählt wird.
1	MD27882[1] definiert den M-Befehl, bei dem \$AC_ACTUAL_PARTS hochgezählt wird.
2	MD27882[2] definiert den M-Befehl, bei dem \$AC_SPECIAL_PARTS hochgezählt wird.

Bei Aufruf eines anwenderdefinierten M-Befehls wird der betreffende Werkstückzähler um den Wert "1" erhöht.

## Randbedingungen

- **Betriebsartenwechsel / NC-RESET**  
Durch Betriebsartenwechsel oder NC-RESET werden die Werkstückzähler nicht beeinflusst.
- **\$AC\_REQUIRED\_PARTS ≤ 0**  
Bei  $\$AC\_REQUIRED\_PARTS \leq 0$  und MD27880  $\$MC\_PART\_COUNTER.Bit\ 0 == 1$  wird für alle aktiven Zähler die Zählung und der mit MD27880 eingestellte Identitätsvergleich **nicht** durchgeführt.

## Beispiele

- **Aktivierung des Werkstückzählers \$AC\_REQUIRED\_PARTS:**  
MD27880  $\$MC\_PART\_COUNTER = 'H3'$ 
  - $\$AC\_REQUIRED\_PARTS$  ist aktiv.
  - Anzeige-Alarm bei:  $\$AC\_REQUIRED\_PARTS == \$AC\_SPECIAL\_PARTS$
- **Aktivierung des Werkstückzählers \$AC\_TOTAL\_PARTS:**  
MD27880  $\$MC\_PART\_COUNTER = 'H10'$   
MD27882  $\$MC\_PART\_COUNTER\_MCODE[0] = 80$ 
  - $\$AC\_TOTAL\_PARTS$  ist aktiv, mit jedem  $M02$  wird der Zähler um den Wert "1" erhöht.
  - $\$MC\_PART\_COUNTER\_MCODE[0]$  hat keine Bedeutung.
- **Aktivierung des Werkstückzählers \$AC\_ACTUAL\_PARTS:**  
MD27880  $\$MC\_PART\_COUNTER = 'H300'$   
MD27882  $\$MC\_PART\_COUNTER\_MCODE[1] = 17$ 
  - $\$AC\_TOTAL\_PARTS$  ist aktiv, mit jedem  $M17$  wird der Zähler um den Wert "1" erhöht.
- **Aktivierung des Werkstückzählers \$AC\_SPECIAL\_PARTS:**  
MD27880  $\$MC\_PART\_COUNTER = 'H3000'$   
MD27882  $\$MC\_PART\_COUNTER\_MCODE[2] = 77$ 
  - $\$AC\_SPECIAL\_PARTS$  ist aktiv.
  - Mit jedem  $M77$  erfolgt:  $\$AC\_SPECIAL\_PARTS + 1$
- **Abschalten des Werkstückzählers \$AC\_ACTUAL\_PARTS:**  
MD27880  $\$MC\_PART\_COUNTER = 'H200'$   
MD27882  $\$MC\_PART\_COUNTER\_MCODE[1] = 50$ 
  - $\$AC\_ACTUAL\_PARTS$  ist nicht aktiv.

- **Aktivierung aller Zähler:**

MD27880 \$MC\_PART\_COUNTER = 'H3313'

MD27882 \$MC\_PART\_COUNTER\_MCODE[0] = 80

MD27882 \$MC\_PART\_COUNTER\_MCODE[1] = 17

MD27882 \$MC\_PART\_COUNTER\_MCODE[2] = 77

- \$AC\_REQUIRED\_PARTS ist aktiv.
- Anzeige-Alarm bei: \$AC\_REQUIRED\_PARTS == \$AC\_SPECIAL\_PARTS
- \$AC\_TOTAL\_PARTS ist aktiv, mit jedem M02 wird der Zähler um den Wert "1" erhöht.
- \$MC\_PART\_COUNTER\_MCODE[0] hat keine Bedeutung.
- \$AC\_ACTUAL\_PARTS ist aktiv, mit jedem M17 wird der Zähler um den Wert "1" erhöht.
- \$AC\_SPECIAL\_PARTS ist aktiv, mit jedem M77 wird der Zähler um den Wert "1" erhöht.

- **Keine Bearbeitung des Werkstückzählers \$AC\_ACTUAL\_PARTS bei Programm-Test / Satzsuchlauf:**

MD27880 \$MC\_PART\_COUNTER = 'H700'

MD27882 \$MC\_PART\_COUNTER\_MCODE[1] = 75

- \$AC\_ACTUAL\_PARTS ist aktiv, mit jedem M75 wird der Zähler um den Wert "1" erhöht, außer bei Programm-Test und Satzsuchlauf.

- **Aufhebung der im MD27880 \$MC\_PART\_COUNTER getroffenen Zählmodi bei Bit 0 = 1:**

MD27882 \$MC\_PART\_COUNTER\_MCODE[0] = 41

MD27882 \$MC\_PART\_COUNTER\_MCODE[1] = 42

MD27882 \$MC\_PART\_COUNTER\_MCODE[2] = 43

Programmcode	Kommentar
...	
N100 \$AC_REQUIRED_PARTS=-10	; Wert < 0 ⇒ Zählung einstellen.
N200 M41 M43	; Keine Zählung.
N300 M42	
...	
N500 \$AC_REQUIRED_PARTS=52	; Wert > 0: Zählung entsprechend MD27880 aktiv.
N501 M43	; Zählung.
N502 M42 M41	; Zählung.
...	

## 8.17 Datenlisten

### 8.17.1 Maschinendaten

#### 8.17.1.1 Allgemeine Maschinendaten

##### Anzeige-Maschinendaten

Nummer	Bezeichner: \$MM_	Beschreibung
<b>SINUMERIK Operate</b>		
9421	MA_AXES_SHOW_GEO_FIRST	Geo-Achsen des Kanals zu erst anzeigen
9422	MA_PRESET_MODE	PRESET / Basisverschiebung in JOG
9423	MA_MAX_SKP_LEVEL	Maximale Anzahl von Ausblendeebenen

##### NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10010	ASSIGN_CHAN_TO_MODE_GROUP	Kanal gültig in BAG
10280	PROG_FUNCTION_MASK	Vergleichsbefehle ">" und "<"
10700	PREPROCESSING_LEVEL	Programmvorverarbeitungsstufe
10702	IGNORE_SINGLEBLOCK_MASK	Einzelsatzstopp verhindern
10707	PROG_TEST_MASK	Programmtest Modi
10708	SERUPRO_MASK	Satzsuchlauf Modi
10710	PROG_SD_RESET_SAVE_TAB	Zu aktualisierende Settingdaten
10711	NC_LANGUAGE_CONFIGURATION	Art und Weise, wie mit Sprachbefehlen verfahren wird, deren zugehörige Option bzw. Funktion nicht aktiviert ist.
10713	M_NO_FCT_STOPRE	M-Funktion mit Vorlaufstopp
10715	M_NO_FCT_CYCLE	Durch UP zu ersetzende M-Funktion
10716	M_NO_FCT_CYCLE_NAME	UP-Name für M-Funktions-Ersetzung
10717	T_NO_FCT_CYCLE_NAME	Name des Werkzeugwechselzyklus für T-Funktionsersetzung
10718	M_NO_FCT_CYCLE_PAR	M-Funktionsersetzung mit Parametern
10719	T_NO_FCT_CYCLE_MODE	Parametrierung der T-Funktionsersetzung
11450	SEARCH_RUN_MODE	Suchlauf Parametrierung
11470	REPOS_MODE_MASK	Repositioniereigenschaften
11600	BAG_MASK	BAG-Verhalten bezüglich ASUP
11602	ASUP_START_MASK	Stoppgründe für ASUP ignorieren

Nummer	Bezeichner: \$MN_	Beschreibung
11604	ASUP_START_PRIO_LEVEL	Prioritäten, ab denen \$MN_ASUP_START_MASK wirksam ist
11610	ASUP_EDITABLE	Aktivierung eines anwenderspezifischen ASUP Programmes
11612	ASUP_EDIT_PROTECTION_LEVEL	Schutzstufe des anwenderspezifischen ASUP Programms
11620	PROG_EVENT_NAME	Programmname für PROG_EVENT
11717	D_NO_FCT_CYCLE_NAME	Unterprogrammname für D-Funktionsersetzung
15700	LANG_SUB_NAME	Name für Substitutionsunterprogramm
15702	LANG_SUB_PATH	Aufrufpfad für Substitutionsunterprogramm
17200	GMMC_INFO_NO_UNIT	globale HMI Info (ohne physikalische Einheit)
17201	GMMC_INFO_NO_UNIT_STATUS	globale HMI Statusinfo (ohne physik. Einheit)
18360	MM_EXT_PROG_BUFFER_SIZE	FIFO-Buffer Größe für Abarbeiten von Extern (DRAM)
18362	MM_EXT_PROG_NUM	Anzahl der gleichzeitig von 'Extern abarbeitbaren Programmebenen

### 8.17.1.2 Kanal-spezifische Maschinendaten

#### Grundmaschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20000	CHAN_NAME	Kanalname
20050	AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrieachse zu Kanalachse
20060	AXCONF_GEOAX_NAME_TAB	Geometrieachsname im Kanal
20070	AXCONF_MACHAX_USED	Maschinenachsnummer gültig im Kanal
20080	AXCONF_CHANAX_NAME_TAB	Kanalachsname im Kanal [Kanalachsnr.]: 0...7
20090	SPIND_DEF_MASTER_SPIND	Löschstellung der Masterspindel im Kanal
20100	DIAMETER_AX_DEF	Geometrieachse mit Planachsfunktion
20106	PROG_EVENT_IGN_SINGLEBLOCK	Prog-Events ignorieren den Einzelsatz
20107	PROG_EVENT_IGN_INHIBIT	Prog-Events ignorieren die Einlesesperre
20108	PROG_EVENT_MASK	Ereignisgesteuerte Programmaufrufe
20109	PROG_EVENT_MASK_PROPERTIES	Eigenschaften Prog-Events
20114	MODESWITCH_MASK	Einstellung für Repos
20116	IGNORE_INHIBIT_ASUP	Anwender-ASUPs trotz Einlesesperre komplett abarbeiten
20117	IGNORE_SINGLEBLOCK_ASUP	Anwender-ASUPs trotz Einzelsatzbearbeitung komplett abarbeiten
20160	CUBIC_SPLINE_BLOCKS	Anzahl der Sätze beim C-Spline
20170	COMPRESS_BLOCK_PATH_LIMIT	Maximale Verfahränge eines NC-Satzes bei Kompression

8.17 Datenlisten

Nummer	Bezeichner: \$MC_	Beschreibung
20191	IGN_PROG_STATE_ASUP	Ausführung der Interruptroutine auf der Bedienoberfläche nicht anzeigen
20192	PROG_EVENT_IGN_PROG_STATE	Ausführung des Prog-Events auf der Bedienoberfläche nicht anzeigen
20193	PROG_EVENT_IGN_STOP	Prog-Events ignorieren die Stop-Taste
20210	CUTCOM_CORNER_LIMIT	Maximalwinkel für Ausgleichssätze bei WRK
20220	CUTCOM_MAX_DISC	Maximaler Wert für DISC
20230	CUTCOM_CURVE_INSERT_LIMIT	Maximalwinkel für Schnittpunktberechnung bei WRK
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	Sätze für vorausschauende Konturberechnung bei WRK
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	Maximale Satzanzahl ohne Verfahrbewegung bei WRK
20270	CUTTING_EDGE_DEFAULT	Grundstellung der Werkzeugschneide ohne Programmierung
20400	LOOKAH_USE_VELO_NEXT_BLOCK	Lookahead auf programmierte Folgesatzgeschwindigkeit
20430	LOOKAH_NUM_OVR_POINTS	Anzahl der Override-Eckwerte bei Lookahead
20440	LOOKAH_OVR_POINTS	Korrekturschalter-Eckwerte bei Lookahead
20500	CONST_VELO_MIN_TIME	Minimale Zeit mit konstanter Geschwindigkeit
20600	MAX_PATH_JERK	Bahnbezogener Maximalruck
20610	ADD_MOVE_ACCEL_RESERVE	Beschleunigungsreserve für überlagerte Bewegungen
20700	REFP_NC_START_LOCK	NC-Startsperre ohne Referenzpunkt
20750	ALLOW_GO_IN_G96	G0-Logik bei G96, G961
20800	SPF_END_TO_VDI	Unterprogrammende an PLC
21000	CIRCLE_ERROR_CONST	Kreisendpunktüberwachung Konstante
21010	CIRCLE_ERROR_FACTOR	Kreisendpunktüberwachung Faktor
21100	ORIENTATION_IS_EULER	Winkeldefinition bei Orientierungsprogrammierung
21110	X_AXIS_IN_OLD_X_Z_PLANE	Koordinatensystem bei automatischer Framedefinition
21200	LIFTFAST_DIST	Verfahrstrecke bei Schnellabheben von der Kontur
21210	SETINT_ASSIGN_FASTIN	HW-Zuordnung des ext. NCK-Eingangsbytes für NC-Programm-Interrupt
21202	LIFTFAST_WITH_MIRROR	Schnellabheben mit Spiegeln



**Satzsuchlauf**

<b>Nummer</b>	<b>Bezeichner: \$MC_</b>	<b>Beschreibung</b>
20128	COLLECT_TOOL_CHANGE	Werkzeugwechselbefehle an PLC nach Satzsuchlauf
22600	SERUPRO_SPEED_MODE	Geschwindigkeit bei Suchlauf-Typ 5
22601	SERUPRO_SPEED_FACTOR	Geschwindigkeitsfaktor bei Suchlauftyp 5
22621	ENABLE_START_MODE_MASK_PRT	Schaltet MD22620: START_MODE_MASK_PRT bei Suchlauf SERUPRO frei
22622	DISABLE_PLC_START	Teileprogrammstart via PLC erlauben
22680	AUTO_IPTR_LOCK	Unterbrechungszeiger sperren

**Reset-Verhalten**

<b>Nummer</b>	<b>Bezeichner: \$MC_</b>	<b>Beschreibung</b>
20110	RESET_MODE_MASK	Grundstellung nach RESET/Teileprogrammende
20112	START_MODE_MASK	Grundstellung bei NC-Start nach Teileprogrammstart
20118	GEOAX_CHANGE_RESET	Automatischen Geometrieachswechsel erlauben
20120	TOOL_RESET_VALUE	Werkzeug-Längenkorrektur im Hochlauf (RESET/Teileprogrammende)
20121	TOOL_PRESEL_RESET_VALUE	Vorgewähltes Werkzeug bei RESET
20130	CUTTING_EDGE_RESET_VALUE	Werkzeugschneiden-Längenkorrektur im Hochlauf (RESET/Teileprogrammende)
20140	TRAFO_RESET_VALUE	Transformationsdatensatz im Hochlauf (RESET/Teileprogrammende)
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen
20152	GCODE_RESET_MODE	Resetverhalten der G-Gruppen
20156	MAXNUM_GCODES_EXT	Resetverhalten der externen G-Gruppen
22620	START_MODE_MASK_PRT	Grundstellung bei speziellem NC-Start nach Hochlauf und bei RESET

## Hilfsfunktionseinstellungen

Nummer	Bezeichner: \$MC_	Beschreibung
22000	AUXFU_ASSIGN_GROUP	Hilfsfunktionsgruppe
22010	AUXFU_ASSIGN_TYPE	Hilfsfunktionsart
22020	AUXFU_ASSIGN_EXTENSION	Hilfsfunktionserweiterung
22030	AUXFU_ASSIGN_VALUE	Hilfsfunktionswert
22200	AUXFU_M_SYNC_TYPE	Ausgabezeitpunkt der M-Funktionen
22210	AUXFU_S_SYNC_TYPE	Ausgabezeitpunkt der S-Funktionen
22220	AUXFU_T_SYNC_TYPE	Ausgabezeitpunkt der T-Funktionen
22230	AUXFU_H_SYNC_TYPE	Ausgabezeitpunkt der H-Funktionen
22240	AUXFU_F_SYNC_TYPE	Ausgabezeitpunkt der F-Funktionen
22250	AUXFU_D_SYNC_TYPE	Ausgabezeitpunkt der D-Funktionen
22400	S_VALUES_ACTIVE_AFTER_RESET	S-Funktion über RESET hinaus wirksam
22410	F_VALUES_ACTIVE_AFTER_RESET	F-Funktion über RESET hinaus wirksam
22510	GCODE_GROUPS_TO_PLC	G-Codes, die bei Satzwechsel/Reset an die Nahtstelle NCK/PLC ausgegeben werden
22550	TOOL_CHANGE_MODE	Neue Werkzeugkorrektur bei M-Funktion
22560	TOOL_CHANGE_M_CODE	M-Funktion für Werkzeugwechsel

## Speichereinstellungen

Nummer	Bezeichner: \$MC_	Beschreibung
27900	REORG_LOG_LIMIT	Prozentsatz des IPO-Puffers für Freigabe des Logfiles
28000	MM_REORG_LOG_FILE_MEM	Speichergröße für REORG (DRAM)
28010	MM_NUM_REORG_LUD_MODULES	Anzahl der Bausteine für lokale Anwendervariablen bei REORG
28020	MM_NUM_LUD_NAMES_TOTAL	Anzahl der lokalen Anwendervariablen (DRAM)
28040	MM_LUD_VALUES_MEM	Speichergröße für lokale Anwendervariablen (DRAM)
28050	MM_NUM_R_PARAM	Anzahl der kanalspezifischen R-Parameter (SRAM)
28060	MM_IPO_BUFFER_SIZE	Anzahl der NC-Sätze im IPO-Puffer (DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	Anzahl der Sätze für die Satzaufbereitung. (DRAM)
28080	MM_NUM_USER_FRAMES	Anzahl der einstellbaren Frames (SRAM)
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl Satzelemente für Compile-Zyklen (DRAM)
28100	MM_NUM_CC_BLOCK_USER_MEM	Größe des Satzspeichers für Compile-Zyklen (DRAM)
28400	MM_ABSBLOCK	Basissätze mit Absolutwerten aktivieren
28402	MM_ABSBLOCK_BUFFER_CONF	Größe des Upload-Buffers dimensionieren
28500	MM_PREP_TASK_STACK_SIZE	Stackgröße der Präparationstask (DRAM)

## Programmlaufzeit und Werkstückzähler

Nummer	Bezeichner: \$MC_	Beschreibung
27860	PROCESSTIMER_MODE	Aktivierung der Laufzeit-Messung
27880	PART_COUNTER	Aktivierung der Werkstück-Zähler
27882	PART_COUNTER_MCODE[ ]	Werkstückzählung über einen M-Befehl

### 8.17.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30465	AXIS_LANG_SUB_MASK	Substituierung von NC-Sprachbefehlen
30550	AXCONF_ASSIGN_MASTER_CHAN	Löschstellung des Kanals für Achswechsel
30600	FIX_POINT_POS	Festwertpositionen der Achsen bei G75
33100	COMPRESS_POS_TOL	Maximale Abweichung bei Kompression

## 8.17.2 Settingdaten

### 8.17.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42000	THREAD_START_ANGLE	Startwinkel bei Gewinde
42010	THREAD_RAMP_DISP	Beschleunigungsverhalten der Achse beim Gewindeschneiden
42100	DRY_RUN_FEED	Probelaufvorschub
42200	SINGLEBLOCK2_STOPRE	Debugmode für SBL2 aktivieren
42444	TARGET_BLOCK_INCR_PROG	Aufsetzmodus nach Satzsuchlauf mit Berechnung
42700	EXT_PROG_PATH	Programmpfades für externen Unterprogrammaufruf EXTCALL
42750	ABSBLOCK_ENABLE	Basis-Satzanzeige freigeben
42990	MAX_BLOCKS_IN_IPOBUFFER	Maximale Anzahl der Sätze im Interpolationspuffer

### 8.17.3 Signale

#### 8.17.3.1 Signale an NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Not-Halt	DB10.DBX56.1	DB2600.DBX0.1

#### 8.17.3.2 Signale an BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Betriebsart AUTOMATIK	DB11.DBX0.0	DB3000.DBX0.0
Betriebsart MDA	DB11.DBX0.1	DB3000.DBX0.1
Betriebsart JOG	DB11.DBX0.2	DB3000.DBX0.2
Betriebsart-Wechselsperre	DB11.DBX0.4	DB3000.DBX0.4
BAG-Stopp	DB11.DBX0.5	-
BAG-Stopp Achsen plus Spindel	DB11.DBX0.6	-
BAG-RESET	DB11.DBX0.7	DB3000.DBX0.7
Maschinenfunktion Teach In	DB11.DBX1.0	DB3000.DBX1.0
Maschinenfunktion REPOS	DB11.DBX1.1	-
Maschinenfunktion REF	DB11.DBX1.2	DB3000.DBX1.2

#### 8.17.3.3 Signale von BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
angewählte Betriebsart AUTOMATIK	DB11.DBX4.0	-
angewählte Betriebsart MDA	DB11.DBX4.1	-
angewählte Betriebsart JOG	DB11.DBX4.2	-
angewählte Maschinenfunktion Teach In	DB11.DBX5.0	-
angewählte Maschinenfunktion REPOS	DB11.DBX5.1	-
angewählte Maschinenfunktion REF	DB11.DBX5.2	-
aktive Betriebsart AUTOMATIK	DB11.DBX6.0	DB3100.DBX0.0
aktive Betriebsart MDA	DB11.DBX6.1	DB3100.DBX0.1
aktive Betriebsart JOG	DB11.DBX6.2	DB3100.DBX0.2

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
BAG betriebsbereit	DB11.DBX6.3	DB3100.DBX0.3
BAG reseted	DB11.DBX6.4	-
NCK Intern JOG aktiv	DB11.DBX6.5	-
alle Kanäle im Reset-Zustand	DB11.DBX6.7	-
aktive Maschinenfunktion Teach In	DB11.DBX7.0	DB3100.DBX1.0
aktive Maschinenfunktion REPOS	DB11.DBX7.1	-
aktive Maschinenfunktion REF	DB11.DBX7.2	DB3100.DBX1.2

#### 8.17.3.4 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
DRF aktivieren	DB21, ... .DBX0.3	DB3200.DBX0.3
Einzelsatz aktivieren	DB21, ... .DBX0.4	DB3200.DBX0.4
M01 aktivieren	DB21, ... .DBX0.5	DB3200.DBX0.5
Probelaufvorschub aktivieren	DB21, ... .DBX0.6	DB3200.DBX0.6
PLC-Aktion beendet	DB21, ... .DBX1.6	-
Programmtest aktivieren	DB21, ... .DBX1.7	DB3200.DBX1.7
Satz ausblenden Ebenen: /0 bis /7	DB21, ... .DBX2.0-7	DB3200.DBX2.0-7
Einlesesperre	DB21, ... .DBX6.1	DB3200.DBX6.1
Programmebenen-Abbruch	DB21, ... .DBX6.4	DB3200.DBX6.4
NC-Startsperre	DB21, ... .DBX7.0	DB3200.DBX7.0
NC-Start	DB21, ... .DBX7.1	DB3200.DBX7.1
NC-Stop an Satzgrenze	DB21, ... .DBX7.2	DB3200.DBX7.2
NC-Stop	DB21, ... .DBX7.3	DB3200.DBX7.3
NC-Stop Achsen plus Spindeln	DB21, ... .DBX7.4	DB3200.DBX7.4
Reset	DB21, ... .DBX7.7	-
REPOSPATHMODE	DB21, ... .DBX31.0-2	-
REPOSMODEEDGE	DB21, ... .DBX31.4	-

#### 8.17.3.5 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
DRF angewählt	DB21, ... .DBX24.3	DB1700.DBX0.3
NCK assoziiertes M01 auswählen	DB21, ... .DBX24.4	-
M01 angewählt	DB21, ... .DBX24.5	DB1700.DBX0.5
Probelaufvorschub angewählt	DB21, ... .DBX24.6	DB1700.DBX0.6
REPOSPATHMODE 0 - 2	DB21, ... .DBX25.0-2	-
Vorschubkorrektur für Eilgang angewählt	DB21, ... .DBX25.3	DB1700.DBX1.3
REPOS MODE EDGE	DB21, ... .DBX25.4	-

## 8.17 Datenlisten

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Programmtest angewählt	DB21, ... .DBX25.7	DB1700.DBX1.7
Satzausblenden angewählt: /0 - /7	DB21, ... .DBX26.0-7	DB1700.DBX2.0-7
Satzausblenden angewählt: /8	DB21, ... .DBX27.0	DB1700.DBX3.0
Satzausblenden angewählt: /9	DB21, ... .DBX27.1	DB1700.DBX3.1
REPOSPATHMODE: 0 - 2	DB21, ... .DBX31.0-2	-
REPOS MODE EDGE	DB21, ... .DBX31.4	-
Satzausblenden aktiv /8	DB21, ... .DBX31.6	DB3200.DBX15.6
Satzausblenden aktiv /9	DB21, ... .DBX31.7	DB3200.DBX15.7
Abarbeiten von extern aktiv	DB21, ... .DBX32.0	DB3300.DBX0.0
Aktionssatz aktiv	DB21, ... .DBX32.3	DB3300.DBX0.3
Anfahrersatz aktiv	DB21, ... .DBX32.4	DB3300.DBX0.4
M0/M1 aktiv	DB21, ... .DBX32.5	DB3300.DBX0.5
letzter Aktionssatz aktiv	DB21, ... .DBX32.6	DB3300.DBX0.6
Satzsuchlauf aktiv	DB21, ... .DBX33.4	DB3300.DBX1.4
M02/M30 aktiv	DB21, ... .DBX33.5	DB3300.DBX1.5
Transformation aktiv	DB21, ... .DBX33.6	DB3300.DBX1.6
Programmtest aktiv	DB21, ... .DBX33.7	DB3300.DBX1.7
Programm Zustand: läuft	DB21, ... .DBX35.0	DB3300.DBX3.0
Programm Zustand: warten	DB21, ... .DBX35.1	DB3300.DBX3.1
Programm Zustand: angehalten	DB21, ... .DBX35.2	DB3300.DBX3.2
Programm Zustand: unterbrochen	DB21, ... .DBX35.3	DB3300.DBX3.3
Programm Zustand: abgebrochen	DB21, ... .DBX35.4	DB3300.DBX3.4
Kanal Zustand: aktiv	DB21, ... .DBX35.5	DB3300.DBX3.5
Kanal Zustand: unterbrochen	DB21, ... .DBX35.6	DB3300.DBX3.6
Kanal Zustand: Reset	DB21, ... .DBX35.7	DB3300.DBX3.7
Interrupt-Behandlung aktiv	DB21, ... .DBX36.4	-
Kanal betriebsbereit	DB21, ... .DBX36.5	-
Einlesefreigabe wird ignoriert	DB21, ... .DBX37.6	-
Stoppen am Blockende bei SBL wird unterdrückt	DB21, ... .DBX37.7	-
Nr. der aktiven G-Funktion der G-Funktionsgruppe 1 – n (8 Bit-Int)	DB21, ... .DBB208-271	DB3500.DBB0-63
Werkstück-Soll erreicht	DB21, ... .DBX317.1	DB3300.DBX4001.1
ASUP ist angehalten	DB21, ... .DBX318.0	DB3300.DBX4002.0
Satzsuchlauf via Programmtest ist aktiv	DB21, ... .DBX318.1	-
REPOS MODE EDGEACKN	DB21, ... .DBX319.0	-
Repos Path Mode Quitt: 0 - 2	DB21, ... .DBX319.1-3	-
Repos DEFERENTIAL Chan	DB21, ... .DBX319.5	-
Anzeige des auslösenden Ereignisses beim ereignisgesteuerten Programmaufruf	DB21, ... .DBX376.0-7	DB3300.DBB4004
ASUP ist aktiv	DB21, ... .DBX378.0	DB3300.DBB4006.0
ASUP mit unterdrückter Anzeigenaktualisierung ist aktiv	DB21, ... .DBX378.1	DB3300.DBB4006.1

**8.17.3.6 Signale an Achse/Spindel**

<b>Signalname</b>	<b>SINUMERIK 840D sl</b>	<b>SINUMERIK 828D</b>
REPOSDELAY	DB31, ... .DBX10.0	-

**8.17.3.7 Signale von Achse/Spindel**

<b>Signalname</b>	<b>SINUMERIK 840D sl</b>	<b>SINUMERIK 828D</b>
REPOS Verschiebung	DB31, ... .DBX70.0	-
REPOS Verschiebung gültig	DB31, ... .DBX70.1	-
REPOS Delay Quitt	DB31, ... .DBX70.2	-
REPOSDELAY	DB31, ... .DBX72.0	-
Bahnachse	DB31, ... .DBX76.4	DB390x.DBX1002.4





## K2: Achsen, Koordinatensysteme, Frames

### 9.1 Kurzbeschreibung

#### 9.1.1 Achsen

##### Maschinenachsen

Maschinenachsen sind die real an der (Werkzeug-)Maschine vorhandenen Achsen.

##### Kanalachsen

Jede Geometrieachse und jede Zusatzachse wird einem Kanal und somit einer Kanalachse zugewiesen. Geometrieachsen und Zusatzachsen werden immer in "ihrem" Kanal verfahren.

##### Geometrieachsen

Die drei Geometrieachsen bilden immer ein fiktives rechtwinkliges Koordinatensystem, das Basiskoordinatensystem (BKS).

Durch Verwendung von FRAMES (Verschiebung, Drehung, Skalierung, Spiegelung) können Geometrieachsen des Werkstückkoordinatensystems (WKS) auf das BKS abgebildet werden.

##### Zusatzachsen

Bei Zusatzachsen ist, im Gegensatz zu Geometrieachsen, kein geometrischer Zusammenhang zwischen den Achsen definiert.

##### Bahnachsen

Bahnachsen zeichnen sich dadurch aus, dass sie gemeinsam interpoliert werden (alle Bahnachsen eines Kanals haben gemeinsam einen Bahninterpolator).

Alle Bahnachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

##### Positionierachsen

Positionierachsen zeichnen sich dadurch aus, dass sie getrennt interpoliert werden (jede Positionierachse hat einen eigenen Achsinterpolator). Jede Positionierachse hat einen eigenen Vorschub und eine eigene Beschleunigungskennlinie.

## Synchronachsen

Synchronachsen werden gemeinsam mit Bahnachsen interpoliert (alle Bahnachsen und Synchronachsen eines Kanals haben einen gemeinsamen Bahninterpolator).

Alle Bahnachsen und alle Synchronachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

## Achskonfiguration

Die Zuordnung zwischen den Geometrieachsen, Zusatzachsen, Kanalachsen und Maschinenachsen, sowie die Festlegung der Namen der einzelnen Achstypen wird über folgende Maschinendaten getroffen:

MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB (Zuordnung Geometrieachse zu Kanalachse)

MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB (Geometrieachsname im Kanal)

MD20070 \$MC\_AXCONF\_MACHAX\_USED (Maschinenachsnummer gültig im Kanal)

MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB (Kanalachsname im Kanal)

MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB (Maschinenachsname)

MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX (Zuordnung Spindel zu Maschinenachse)

## Umschaltbare Geometrieachsen

Mit der Funktion "Umschaltbare Geometrieachsen" kann aus dem Teileprogramm heraus der Verbund der Geometrieachsen aus anderen Kanalachsen zusammengesetzt werden.

Achsen, die in einem Kanal zunächst als synchrone Zusatzachsen projektiert sind, können durch einen Programmbefehl eine beliebige Geometrieachse ersetzen.

## Link-Achse

Link-Achsen sind Achsen, die an einer anderen NCU physikalisch angeschlossen sind und deren Lageregelung unterliegen. Link-Achsen können dynamisch Kanälen einer **anderen** NCU zugeordnet werden. Link-Achsen sind aus Sicht einer bestimmten NCU nicht-lokale Achsen.

Der dynamischen Änderung der Zuordnung zu einer NCU dient das Konzept der **Achscontainer**. Achstausch mit `GET` und `RELEASE` aus dem Teileprogramm ist für Link-Achsen über NCU-Grenzen hinweg nicht verfügbar.

Die Link-Achsen sind beschrieben in:

### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3)

## Achscontainer

Ein Achscontainer ist eine Ringpuffer-Datenstruktur, in der die Zuordnung von lokalen Achsen und/oder Link-Achsen zu Kanälen erfolgt. Die Einträge im Ringpuffer sind **zyklisch verschiebbar**.

Die Link-Achsen Konfiguration lässt im logischen Maschinenachs-Abbild neben dem direkten Verweis auf lokale Achsen oder Link-Achsen den Verweis auf Achscontainer zu.

Ein solcher Verweis besteht aus:

- Achscontainer-Nummer
- Slot (Ringpuffer-Platz innerhalb des entsprechenden Achscontainers)

Als Eintrag in einem Ringpuffer-Platz steht:

- Eine lokale Achse  
oder
- Eine Link-Achse

Die Funktion "Achscontainer" ist beschrieben in:

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3)

## 9.1.2 Koordinatensysteme

### MKS

Das Maschinenkoordinatensystem (MKS) zeichnet sich durch folgende Eigenschaften aus:

- Es wird durch die Maschinenachsen gebildet.
- Die Maschinenachsen können rechtwinkelig kartesisch oder beliebig zueinander angeordnet sein.
- Die Maschinenachsen haben einstellbare Namen.
- Die Maschinenachsen können Linearachsen oder Rundachsen sein.

### BKS

Das Basiskoordinatensystem (BKS) zeichnet sich durch folgende Eigenschaften aus:

- Die Geometrieachsen bilden ein rechtwinkeliges kartesisches Koordinatensystem.
- Das BKS geht durch kinematische Transformation aus dem MKS hervor.

### BNS

Das Basis-Nullpunktsystem (BNS) ist das Basis-Koordinatensystem mit Basis-Verschiebung.

### ENS

Das Einstellbare Nullpunktsystem (ENS) ist das Werkstückkoordinatensystem mit programmierbarem Frame aus Sicht vom WKS. Der Werkstücknullpunkt wird durch die einstellbaren Frames G54 ... G599 festgelegt.

## WKS

Das Werkstückkoordinatensystem (WKS) zeichnet sich durch folgende Eigenschaften aus:

- Im WKS werden alle Koordinaten der Achsen programmiert (Teileprogramm).
- Es wird durch Geometrieachsen und Zusatzachsen gebildet.
- Geometrieachsen bilden immer ein rechtwinkeliges kartesisches Koordinatensystem
- Zusatzachsen bilden ein Koordinatensystem ohne geometrischen Zusammenhang zwischen den Zusatzachsen.
- Geometrieachsen und Zusatzachsen haben einstellbare Namen.
- Durch FRAMES kann das WKS verschoben, gedreht, skaliert oder gespiegelt werden (TRANS, ROT, SCALE, MIRROR).

Auch Mehrfachverschiebungen, Mehrfachdrehungen, etc. sind möglich.

## Externe Nullpunktverschiebung

Die externe Nullpunktverschiebung hat folgende Eigenschaften:

- Zu einem von PLC bestimmten Zeitpunkt wird eine zuvor festgelegte zusätzliche Nullpunktverschiebung zwischen Basis- und Werkstück-Koordinatensystem aktiviert.
- Die Beträge der Verschiebungen können pro beteiligte Achse gesetzt werden durch:
  - PLC
  - Bedientafel
  - Teileprogramm
- Aktivierte Verschiebungen wirken sich ab dem Abarbeiten des ersten Bewegungssatzes dieser Achsen nach der Aktivierung aus. Die Verschiebungen werden dem programmierten Weg überlagert (keine Interpolation).

Die Geschwindigkeit, mit der die externe Nullpunktverschiebung herausgefahren wird, beträgt:

Programmierter  $F$ -Wert + 1/2 JOG-Geschwindigkeit

In  $G0$ -Sätzen wird die externe Nullpunktverschiebung am Satzende gefahren.

- Die aktivierten Verschiebungen werden bei `RESET` und Programmende beibehalten.
- Nach Power On ist die zuletzt aktive Verschiebung weiterhin in der Steuerung gespeichert, muss aber über die PLC wieder aktiviert werden.

### 9.1.3 Frames

#### Frame

Ein Frame stellt eine geschlossene Rechenvorschrift dar, die kartesische Koordinatensysteme ineinander überführt.

#### Frame-Komponenten

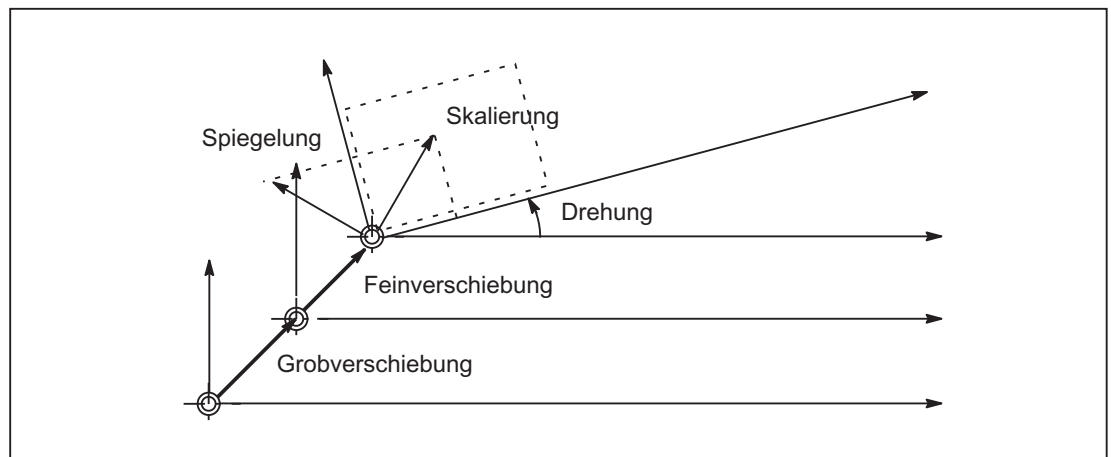


Bild 9-1 Frame-Komponenten

Ein Frame setzt sich aus folgenden Komponenten zusammen:

Frame-Komponenten		Programmierbar mit:
Verschiebung	Grobverschiebung	TRANS ATRANS (additiver Translationsanteil) CTRANS (Nullpunktverschiebung für mehrere Achsen) G58 (axiale Nullpunktverschiebung)
	Feinverschiebung	CFINE G59 (axiale Nullpunktverschiebung)
Drehung		ROT / ROTs AROT / AROTS CROTS
Skalierung		SCALE ASCALE
Spiegelung		MIRROR AMIRROR

#### Besonderheiten für Achsen

Die Grob- und Fein-Verschiebung, Skalierung und Spiegelung können für Geometrieachsen und Zusatzachsen programmiert werden. Für Geometrieachsen kann zusätzlich eine Drehung programmiert werden.

## Grob- und Feinverschiebung

Der Translationsanteil von Frames besteht aus:

- Grobverschiebung mit `TRANS`, `ATRANS` und `CTRANS`

Die Grobverschiebung wird normalerweise vom Maschineneinrichter vorgegeben.

Mit `TRANS` werden die programmierbaren Verschiebungen für alle Geometrieachsen und Zusatzachsen vorgegeben.

- Feinverschiebung mit `CFINE`

Kann innerhalb bestimmter Eingabegrenzen vom Maschinenbediener vorgegeben werden.

## G58, G59 (nur 840D sl)

Mit `G58` und `G59` kann bei SINUMERIK 840D sl die Grob- bzw. Feinverschiebung des programmierbaren Frames axial ersetzt werden. Diese Funktionen sind nur einsetzbar, wenn die Feinverschiebung projiziert ist.

- Grobverschiebung mit `G58`

`G58` verändert nur den absoluten Translationsanteil (Grobverschiebung) für die angegebene Achse, die Summe der additiv-programmierten Translationen (Feinverschiebung) bleibt erhalten.

- Feinverschiebung mit `G59`

`G59` dient dem axialen Überschreiben der additiv-programmierten Translationen für die angegebenen Achsen, die mit `ATRANS` programmiert wurden.

## Frame-Drehungen

Orientierungen im Raum werden über Frame-Drehungen wie folgt festgelegt:

- Drehung mit `ROT` legt die Einzeldrehungen für alle Geometrieachsen fest.
- Raumwinkel mit `ROTS`, `AROTS`, `CROTS` legen die Orientierung einer Ebene im Raum fest.
- Framedrehung mit `TOFRAME` definiert einen Frame, dessen Z-Achse in Werkzeugrichtung zeigt.

## Skalierung

Mit `SCALE` werden die programmierbaren Skalierungen (Maßstabfaktor) für alle Geometrieachsen und Zusatzachsen programmiert.

Soll eine neue Skalierung auf einer anderen Skalierung, Drehung, Verschiebung oder Spiegelung aufbauen, muss `ASCALE` programmiert werden.

## Spiegeln

Mit dem folgenden Maschinendatum kann eingestellt werden, um welche Achse gespiegelt wird:

MD10610 MIRROR\_REF\_AX (Bezugsachse für das Spiegeln)

Wert	Bedeutung
0	Es wird um die programmierte Achse gespiegelt.
1, 2 oder 3	Je nach Eingabewert wird das Spiegeln auf ein Spiegeln einer bestimmten Bezugsachse und Drehung von zwei anderen Geometrieachsen abgebildet.

## Frame-Kettung

Frame-Komponenten oder gesamte Frames lassen sich über den Verkettungsoperator ":" zu einem Gesamt-Frame zusammenfassen. So setzt sich z. B. der aktuelle Frame `$P_ACTFRAME` aus der Verkettung von Gesamt-Basisframe, einstellbarem Frame, den System-Frames und dem programmierbaren Frame zusammen:

```
$P_ACTFRAME = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :
               $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
               $P_ACTBFRAME : $P_IFRAME : $P_TOOLFRAME :
               $P_WPFRAME : $P_TRAFRAME : $P_PFRAME
               $P_ISO4FRAME : $P_CYCFRAME
```

## Frames bei G91

Kettenmaß-Programmierung mit `G91` ist so definiert, dass bei Anwahl einer Nullpunktverschiebung der Korrekturwert additiv zum inkrementell programmierten Wert gefahren wird.

Das Verhalten ist abhängig von der Einstellung im Settingdatum:

SD42440 `$SC_FRAME_OFFSET_INCR_PROG` (Nullpunktverschiebungen in Frames)

Wert	Bedeutung
1	Nullpunktverschiebung wird bei <code>FRAME</code> und inkrementeller Programmierung einer Achse herausgefahren (= Standardeinstellung).
0	Nur der programmierte Weg wird gefahren.

### Unterdrückung von Frames

Die aktuellen Frames können mit folgenden Anweisungen unterdrückt werden:

Befehl	Bedeutung
G53	Unterdrückung der aktuelle Nullpunktverschiebung (satzweise)
G153	Unterdrückung der aktuellen Frame inklusive Basis-Frame
SUPA	Unterdrückung der aktuellen Nullpunktverschiebung, einschließlich programmierter Verschiebungen, System-Frames, Handradverschiebungen (DRF), externer Nullpunktverschiebung und überlagerter Bewegung

### NCU-globale Basis-Frames

Über NCU-globale Basis-Frames können aus einem Kanal heraus Frames für andere Kanäle **vorbelegt** werden.

Eigenschaften der NCU-globalen Basis-Frames:

- Können von allen Kanälen aus geschrieben bzw. gelesen werden
- Sind nur in den Kanälen aktivierbar
- Verschiebungen, Skalierungen und Spiegelungen für Kanal- und Maschinenachsen

Alle globalen und kanalspezifischen Basis-Frames werden verkettet und ergeben so das Gesamt-Basis-Frame. Standardmäßig ist mindestens ein Basis-Frame pro Kanal vorhanden.

### Einstellbare Frames

Einstellbare Frames können als NCU-globale oder kanalspezifische Frames definiert werden.

### Konsistenz

Es liegt in der alleinigen Verantwortung des Anwenders beim Schreiben, Lesen und Aktivieren von Frames z. B. mittels Kanalkoordinierung, ein konsistentes Verhalten innerhalb der Kanäle zu erzielen. Eine kanalübergreifende Aktivierung von Frames ist nicht möglich.



## 9.2 Achsen

### 9.2.1 Übersicht

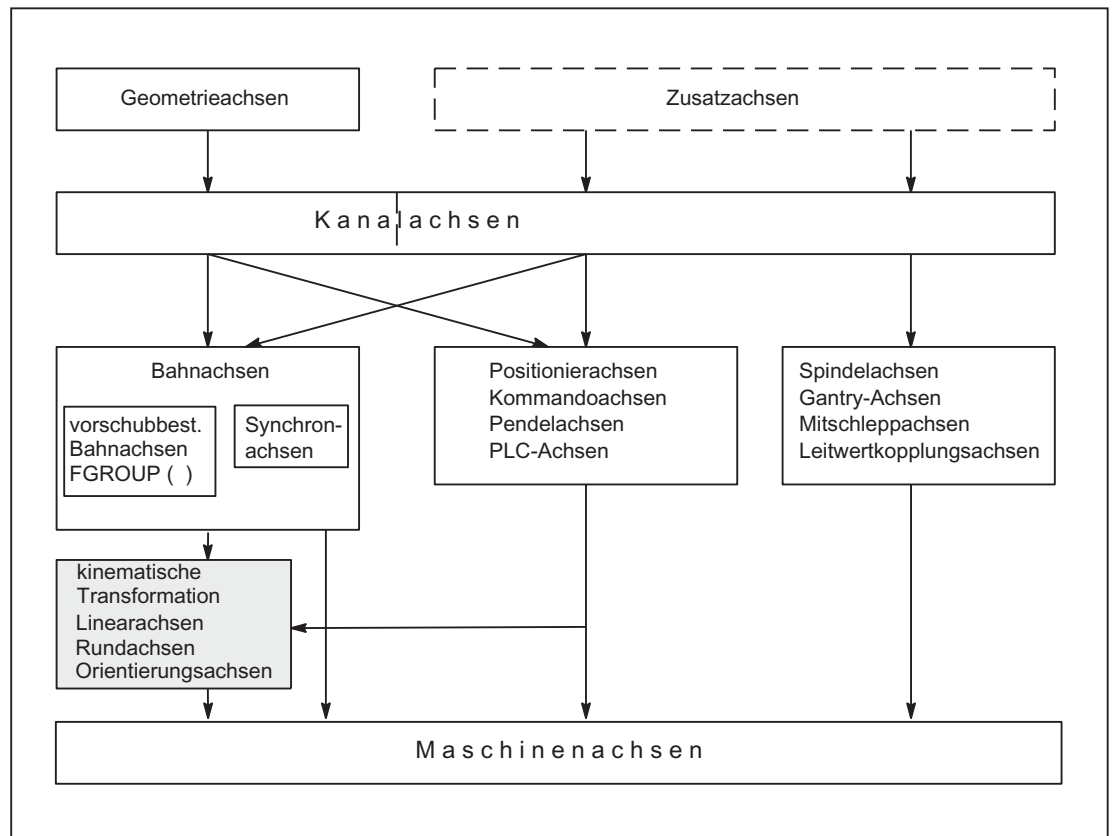


Bild 9-2 Zusammenhang zwischen Geometrie-, Zusatz- und Maschinenachsen

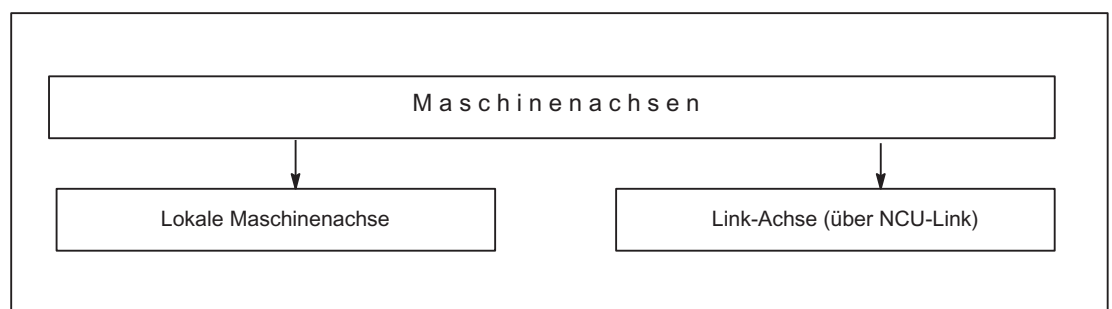


Bild 9-3 Lokale und externe Maschinenachsen (Link-Achsen)

## 9.2.2 Maschinenachsen

### Bedeutung

Maschinenachsen sind die real an der (Werkzeug-)Maschine vorhandenen Achsen.

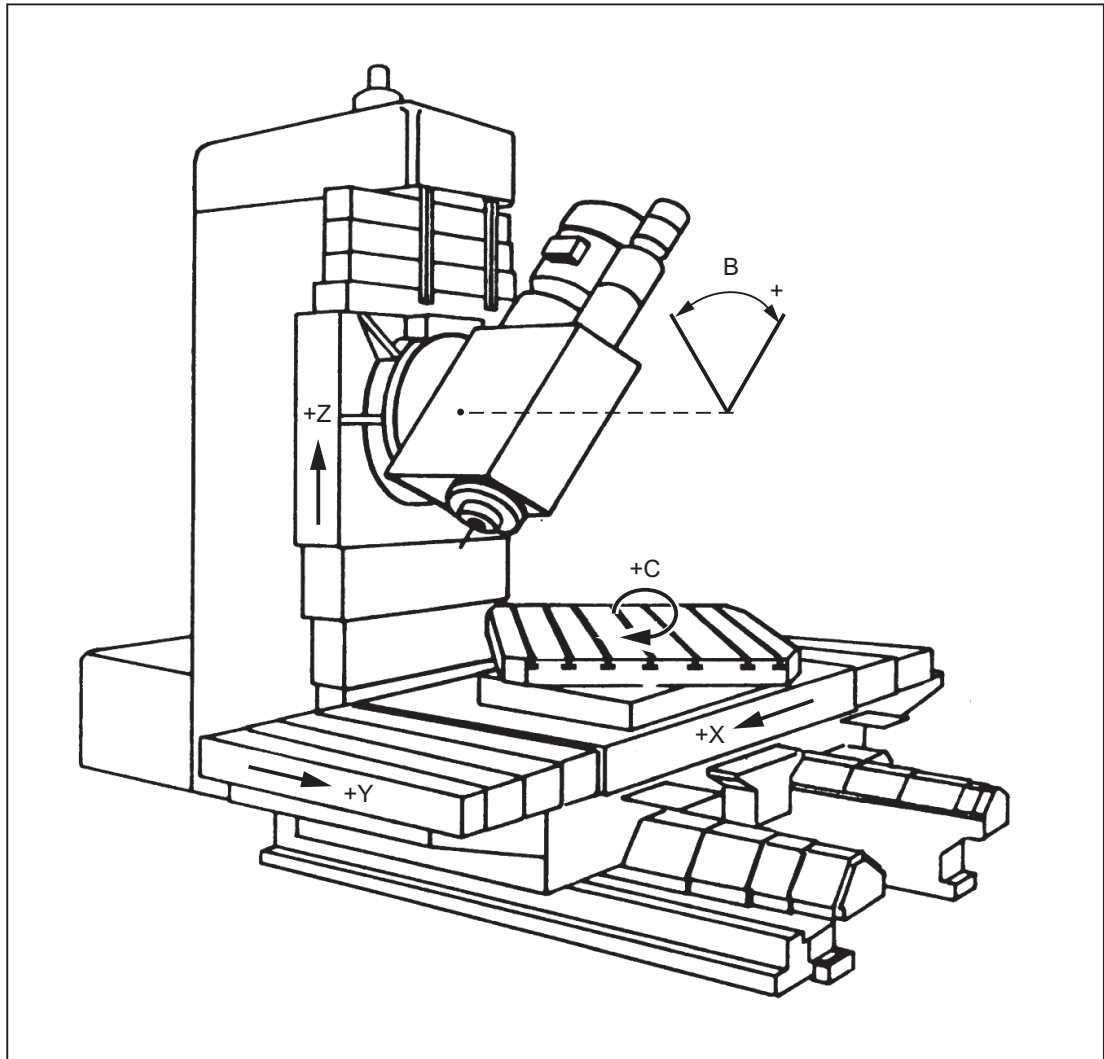


Bild 9-4 Maschinenachsen X, Y, Z, B, S einer kartesischen Maschine

## Anwendung

Maschinenachsen können sein:

- Geometrieachsen X, Y, Z
- Orientierungsachsen A, B, C
- Laderachsen
- Werkzeugrevolver
- Achsen für Werkzeugmagazin
- Achsen für Werkzeugwechsler
- Pinole
- Achsen für Palettenwechsler
- etc.

### 9.2.3 Kanalachsen

#### Bedeutung

Jede Geometrieachse und jede Zusatzachse wird einem Kanal zugewiesen.  
Geometrieachsen und Zusatzachsen werden immer in "ihrem" Kanal verfahren.

### 9.2.4 Geometrieachsen

#### Bedeutung

Die drei Geometrieachsen bilden immer ein fiktives rechtwinkliges Koordinatensystem.

Durch Verwendung von FRAMES (Verschiebung, Drehung, Skalierung, Spiegelung) können Geometrieachsen des Werkstückkoordinatensystems (WKS) auf das BKS abgebildet werden.

#### Anwendung

Geometrieachsen werden für Programmierung der Werkstückgeometrie (der Kontur) verwendet.

Die Ebenenanwahl  $G_{17}$ ,  $G_{18}$  und  $G_{19}$  (DIN 66217) bezieht sich immer auf die drei Geometrieachsen. Deshalb ist es vorteilhaft, die drei Geometrieachsen X, Y und Z zu nennen.

### 9.2.5 Umschaltbare Geometrieachsen

#### Bedeutung

Mit der Funktion "Umschaltbare Geometrieachsen" kann aus dem Teileprogramm heraus der Verbund der Geometrieachsen aus anderen Kanalachsen zusammengesetzt werden.

Achsen, die in einem Kanal zunächst als synchrone Zusatzachsen projiziert sind, können durch einen Programmbefehl eine beliebige Geometrieachse ersetzen.

#### Beispiel

An einer Maschine mit zwei Z-Achsen Z1, Z2 kann vom Teileprogramm gesteuert Z1 oder Z2 die Geometrieachse bilden.

#### Aktivierung

Die Umschaltung erfolgt durch folgenden Programmbefehl:

```
GEOAX([n, Kanalachsname]...)
```

n=0:	um eine Achse aus dem Geometrieachsverbund herauszunehmen
n=1, 2, 3:	Index der Geometrieachse
GEOAX( ):	stellt die über Maschinendaten festgelegte Grundeinstellung der Zuordnung von Kanal- zu Geometrieachsen her
Kanalachsname:	Name der Kanalachse, die Geometrieachse werden soll

Eine Kanalachse, die zur Geometrieachse gemacht wurde, kann nur unter ihrem Geometrieachsnamen adressiert werden. Die Geometrieachsamen selbst bleiben unverändert.

Die Geometrieachsen können einzeln oder zu mehreren in einem Befehl umgeschaltet werden.

## Randbedingungen

Grundsätzlich kann jede Kanalachse, die Geometrieachse ist, durch eine andere Kanalachse ersetzt werden.

Dabei gelten folgende Einschränkungen:

- Rundachsen dürfen nicht zu Geometrieachsen werden.
- Eine Geometrieachse, die den gleichen Namen wie eine Kanalachse hat, kann nicht durch eine andere Kanalachse ersetzt werden (Alarmmeldung). Eine solche Achse kann auch nicht aus dem Geometrieachsverbund herausgenommen werden. Sie kann ihren Platz innerhalb des Geometrieachsverbundes nicht wechseln.
- Bei den an der Umschaltung beteiligten Achspaaren müssen jeweils beide Achsen sachsbezogen synchronisiert sein.
- Beim Umschalten dürfen folgende Funktionen nicht aktiv sein:
  - Transformation
  - Spline-Interpolation
  - Werkzeugradiuskorrektur
  - Werkzeugfeinkorrektur
- Eine mögliche DRF-Verschiebung und externe Nullpunktverschiebung bleiben wirksam. Sie wirken jeweils auf Kanalachsen. Die Kanalachsuzuordnung wird durch das Umschalten der Geometrieachsen nicht beeinflusst.

## Umschalten Geometrieachsen

Es werden alle Frames, Schutzbereiche und Arbeitsfeldbegrenzungen gelöscht. Sie müssen gegebenenfalls nach dem Umschaltvorgang neu programmiert werden.

Damit ist das Verhalten beim Umschalten der Geometrieachsen identisch zu dem beim Wechsel (Ein-, Aus- oder Umschalten) einer kinematischen Transformation.

## Werkzeuglängenkorrektur

Eine aktive Werkzeuglängenkorrektur bleibt wirksam und wirkt nach dem Umschalten auf die neuen Geometrieachsen.

Als noch nicht herausgefahren gelten die Werkzeuglängenkorrekturen für folgende Geometrieachsen:

- Alle Geometrieachsen, die neu in den Geometrieachsverbund aufgenommen wurden.
- Alle Geometrieachsen, die ihren Platz innerhalb des Geometrieachsverbundes gewechselt haben.

Geometrieachsen, die bei einer Umschaltung ihre Position innerhalb des Geometrieachsverbundes beibehalten, behalten auch ihren Status bezüglich der Werkzeuglängenkorrektur bei.

**RESET**

Das Reset-Verhalten der geänderten Geometrieachsordnung wird festgelegt mit den Maschinendaten:

MD20110 \$MC\_RESET\_MODE\_MASK (Festlegung der Steuerungs-Grundstellung nach Reset / TP-Ende)

MD20118 \$MC\_GEOAX\_CHANGE\_RESET (Automatischen Geometrieachswechsel erlauben)

**MD20110 \$MC\_RESET\_MODE\_MASK**

Bit	Wert	Bedeutung
12	0	Bei gesetztem Maschinendatum MD20118 \$MC_GEOAX_CHANGE_RESET (Automatischen Geometrieachswechsel erlauben) wird eine geänderte Geometrieachsordnung bei Reset bzw. Teileprogrammende gelöscht. Die in den Maschinendaten festgelegte Grundeinstellung für die Geometrieachsordnung wird aktiv.
	1	Eine geänderte Geometrieachsordnung bleibt über Reset / Teileprogrammende hinaus aktiv.

**MD20118 \$MC\_GEOAX\_CHANGE\_RESET**

Wert	Bedeutung
0	Die aktuelle Konfiguration der Geometrieachsen bleibt bei Reset und bei Programmstart unverändert. Mit dieser Einstellung ist das Verhalten identisch zu älteren Softwareständen ohne Geoachstausch.
1	Die Konfiguration der Geometrieachsen wird bei Reset bzw. Teileprogrammende in Abhängigkeit vom Maschinendatum MD20110 \$MC_RESET_MODE_MASK und bei Teileprogrammstart in Abhängigkeit vom Maschinendatum MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START) unverändert beibehalten oder in den durch das Maschinendatum MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB (Zuordnung Geometrieachse zu Kanalachse) definierten Grundzustand gebracht.

**Programmstart**

Analog zum Verhalten bei Reset richtet sich das Verhalten bei Programmstart nach der Einstellung im Maschinendatum:

MD20112 \$MC\_START\_MODE\_MASK (Festlegung der Steuerungsgrundstellung bei NC-Start)

Bit	Wert	Bedeutung
12	0	Eine geänderte Geometrieachsordnung bleibt bei Teileprogrammstart aktiv.
	1	Eine geänderte Geometrieachsordnung wird bei Teileprogrammstart gelöscht.

## Referenzpunkt anfahren

Beim Umschalten in die Betriebsart "Referenzpunktanfahren" wird automatisch die durch die Maschinendaten definierte Geometrieachskonfiguration eingestellt.

## M-Code

Ein Umschalten der Geometrieachsen mit `GEOAX ( )` kann der PLC über die Ausgabe eines M-Code mitgeteilt werden:

`MD22532 $MC_GEOAX_CHANGE_M_CODE` (M-Code bei Werkzeugträgerwechsel)

---

### Hinweis

Hat dieses Maschinendatum einen der Werte 0 bis 6, 17, 30, wird kein M-Code ausgegeben.

---

## Transformationswechsel

Folgende Zusammenhänge sind bei kinematischer Transformation und Geometrieachsumschaltung zu beachten:

- Ein Umschalten der Geometrieachszuordnung ist bei aktiver Transformation nicht möglich.
- Das Aktivieren einer Transformation löscht die programmierte Geometrieachskonfiguration und ersetzt sie durch die in den Maschinendaten der aktivierten Transformation abgelegte Geometrieachszuordnung.
- Nach Ausschalten der Transformation ist die über MD festgelegte Grundeinstellung für die Geometrieachskonfiguration wirksam.

Besteht die Notwendigkeit, im Zusammenhang mit Transformationen die Geometrieachszuordnung zu ändern, ist dies durch das Projektieren einer weiteren Transformation zu erreichen. Dazu ist die Gesamtzahl der gleichzeitig verfügbaren Transformationen im Kanal gleich 8.

Von den folgenden Transformationsgruppen können maximal zwei Transformationen pro Kanal gleichzeitig zur Verfügung stehen:

- Orientierungstransformationen  
(3-Achs-, 4-Achs-, 5-Achs- und Nutator-Transformation)
- `TRAANG` (Schräge Achse)
- `TRANSMIT`
- `TRACYL`

### Literatur:

Funktionshandbuch Sonderfunktionen; 3 bis 5-Achs-Transformation (F2)

Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformation (M1)

**Beispiel**

Im folgenden Beispiel wird angenommen, dass es 6 Kanalachsen mit den Kanalachsennamen XX, YY, ZZ, U, V, W und drei Geometrieachsen mit den Namen X, Y, Z gibt. Über Maschinendaten ist die Grundeinstellung so festgelegt, dass die Geometrieachsen auf die ersten drei Kanalachsen, d. h. auf XX, YY und ZZ abgebildet werden.

Programmcode	Kommentar
GEOAX( )	; Die Geometrieachsuzuordnung, die über das Maschinendatum: MD AXCONF_GEOAX_ASSIGN_TAB festgelegt ist, wird wirksam, d. h. XX, YY und ZZ werden Geometrieachsen.
G0 X0 Y0 Z0 U0 V0 W0	; Alle Achsen im Eilgang auf Position 0 verfahren.
GEOAX(1, U, 2, V, 3, W)	; Die Kanalachse U wird zur ersten, V zur zweiten und W zur dritten Geometrieachse.
GEOAX(1, XX, 3, ZZ)	; Die Kanalachse XX wird zur ersten, ZZ zur dritten Geometrieachse. Die zweite Geometrieachse bleibt unverändert.
G17 G2 X20 I10 F1000	; Halbkreis in der X-Y-Ebene. Es fahren die Kanalachsen XX und V.
GEOAX(2, W)	; Die Kanalachse W wird zur zweiten Geometrieachse. Die erste und dritte Geometrieachse bleiben unverändert.
G17 G2 X20 I10 F1000	; Vollkreis in der X-Y-Ebene. Es fahren die Kanalachsen XX und W.
GEOAX( )	; Die Geometrieachsuzuordnung, die über das Maschinendatum: MD AXCONF_GEOAX_ASSIGN_TAB festgelegt ist, wird wirksam, d. h. XX, YY und ZZ werden Geometrieachsen.
GEOAX(1, U, 2, V, 3, W)	; U, V und W werden erste, zweite und dritte Geometrieachse
G1 X10 Y10 Z10 XX=25	; Die Kanalachsen U, V, W fahren jeweils auf die Position 10, XX fährt auf Position 25.
GEOAX(0, V)	; V wird wieder aus dem Geometrieachsverbund herausgenommen. U und W bleiben Geometrieachsen. Die zweite Geometrieachse ist nicht mehr belegt.
GEOAX(1, U, 2, V, 3, W)	; U, V und W werden erste, zweite und dritte Geometrieachse, d. h. U und W bleiben unverändert.
GEOAX(3, V)	; V wird zur dritten Geometrieachse. Dadurch wird W, das bisher die dritte Geometrieachse war, aus dem Geometrieachsverbund herausgenommen. Die zweite Geometrieachse ist nicht mehr belegt.



## 9.2.6 Zusatzachsen

### Bedeutung

Bei Zusatzachsen ist, im Gegensatz zu Geometrieachsen, kein geometrischer Zusammenhang zwischen den Achsen definiert.

---

### Hinweis

Geometrieachsen haben einen exakt definierten Zusammenhang in Form eines rechtwinkligen Koordinatensystems.

---

Zusatzachsen sind Bestandteile des Basiskoordinatensystems (BKS). Durch Verwendung von FRAMES (Verschiebung, Skalierung, Spiegelung) können Zusatzachsen des Werkstückkoordinatensystems (WKS) auf das BKS abgebildet werden.

### Anwendung

Typische Zusatzachsen sind:

- Rundachsen
- Werkzeugmagazinachsen
- Werkzeugrevolverachsen
- Laderachsen

## 9.2.7 Bahnachsen

### Bedeutung

Bahnachsen zeichnen sich dadurch aus, dass sie gemeinsam interpoliert werden (alle Bahnachsen eines Kanals haben gemeinsam einen Bahninterpolator).

Alle Bahnachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

Der unter der Adresse  $F$  programmierte Vorschub (Bahnvorschub) gilt für alle im Satz programmierten Bahnachsen, mit folgenden Ausnahmen:

- Es wurde eine Achse programmiert, die mit der Anweisung  $F_{GROUP}$  als nicht bahngeschwindigkeitsbestimmend festgelegt wurde.
- Achsen, die mit den Anweisungen  $POS$  oder  $POSA$  programmiert wurden, besitzen einen eigenen Vorschub (Achsinterpolator).

### Anwendung

Bahnachsen dienen zum Bearbeiten des Werkstücks mit der programmierten Kontur.

## 9.2.8 Positionierachsen

### Bedeutung

Positionierachsen zeichnen sich dadurch aus, dass sie getrennt interpoliert werden (jede Positionierachse hat einen eigenen Achsinterpolator). Jede Positionierachse hat einen eigenen Vorschub und eine eigene Beschleunigungskennlinie. Positionierachsen können zusätzlich zu Bahnachsen (auch im gleichen Satz) programmiert werden. Die Interpolation der Bahnachsen (Bahninterpolator) wird durch Positionierachsen nicht beeinflusst. Bahnachsen und die einzelnen Positionierachsen erreichen nicht notwendigerweise gleichzeitig ihre Satzendpunkte.

Durch die Anweisungen `POS` und `POSA` werden Positionierachsen programmiert und das Satzwechselkriterium festgelegt:

- `POS`

Der Satzwechsel erfolgt, wenn Bahnachsen und Positionierachsen ihre Satzendpunkte erreicht haben.

- `POSA`

Der Satzwechsel erfolgt, wenn Bahnachsen ihre Satzendpunkte erreicht haben. Positionierachsen laufen über Satzgrenzen hinweg bis zu ihrem Satzendpunkt weiter.

Konkurrierende Positionierachsen unterscheiden sich von Positionierachsen durch folgende Punkte:

- Sie bekommen ihre Satzendpunkte ausschließlich von der PLC.
- Sie können zu jedem beliebigen Zeitpunkt (nicht an Satzgrenzen) gestartet werden.
- Sie beeinflussen das laufende Teileprogramm in ihrer Bearbeitung nicht.

### Anwendung

Typische Positionierachsen sind:

- Lader für Werkstücktransport
- Werkzeugmagazin / -revolver

### Verweis

Für weitere Informationen siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)" und "S1: Spindeln (Seite 1383)".

#### Literatur:

- Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)
- Funktionshandbuch Sonderfunktionen; Gantry-Achsen (G1)
- Funktionshandbuch Sonderfunktionen; Achskopplungen und ESR (M3)
- Funktionshandbuch Erweiterungsfunktionen; Pendeln (P5)
- Funktionshandbuch Synchronaktionen

## 9.2.9 Hauptlaufachsen

### Bedeutung

Als Hauptlaufachse wird eine Achse bezeichnet, die vom Hauptlauf interpoliert wird.

Diese Interpolation kann folgendermaßen gestartet werden:

- Aus Synchronaktionen  
(als Kommandoachsen auf Grund eines Ereignisses über satzbezogene, modale oder statische Synchronaktionen)
- Vom PLC über spezielle Funktionsbausteine im PLC-Grundprogramm  
(als konkurrierende Positionierachse oder auch PLC-Achse genannt)
- Durch Settingdatum oder aus dem Teileprogramm  
(als asynchrone oder satzsynchrone Pendelachse)

### Beeinflussung

Eine vom Hauptlauf interpolierte Achse reagiert bezüglich:

- NC-Stop
- Alarmbehandlung
- Programmbeeinflussungen
- Programmende
- RESET

---

#### Hinweis

Das Verhalten am Programmende ist unterschiedlich. Die Achsbewegung muss nicht immer am Programmende abgeschlossen sein und kann damit auch über das Programmende erfolgen.

---

### Anwendung

Bestimmte Achsen können im Hauptlauf von dem durch den NC-Programmablauf getriggerten Kanalverhalten entkoppelt und vom PLC aus kontrolliert werden. Auch diese Achsen werden im Hauptlauf interpoliert und verhalten sich unabhängig vom Kanal- und Programmablauf.

Eine vom PLC kontrollierte Achse kann dann autark von der NC beeinflusst werden. Dies betrifft folgende Aktionen:

- Ablauf der Achse abbrechen (entspricht Restweg löschen)
- Achse stoppen bzw. unterbrechen
- Achse weiterfahren (Bewegungsablauf fortsetzen)
- Achse in den Grundzustand zurücksetzen

### 9.2.10 Synchronachsen

#### Bedeutung

Synchronachsen sind Teil der Bahnachsen, die nicht zur Berechnung der Bahngeschwindigkeit herangezogen werden. Sie werden gemeinsam mit Bahnachsen interpoliert (alle Bahnachsen und Synchronachsen eines Kanals haben einen gemeinsamen Bahninterpolator).

Alle Bahnachsen und alle Synchronachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

Der unter der Adresse  $F$  programmierte Vorschub (Bahnvorschub) gilt für alle im Satz programmierten Bahnachsen, jedoch nicht für die Synchronachsen.

Synchronachsen benötigen für ihren programmierten Weg die gleiche Zeit wie die Bahnachsen.

#### Befehl FGROU P

Mit dem Befehl  $FGROU P$  wird festgelegt, ob die Achse eine vorschubbestimmende **Bahnachse** (wird zur Berechnung der Bahngeschwindigkeit herangezogen) oder eine **Synchronachse** (wird zur Berechnung der Bahngeschwindigkeit nicht herangezogen) ist.

#### Beispiel

Programmcode	Kommentar
N05 G00 G94 G90 M3 S1000 X0 Y0 Z0	;
N10 FGROU P(X,Y)	; Achsen X und Y sind Bahnachsen Achse Z ist Synchronachse
N20 G01 X100 Y100 F1000	; progr. Vorschub 1000 mm/min Vorschub der Achse X = 707 mm/min Vorschub der Achse Y = 707 mm/min
N30 FGROU P(X)	; Achse X ist eine Bahnachse Achse Y ist eine Synchronachse
N20 X200 Y150	; progr. Vorschub 1000 mm/min Vorschub der Achse X = 1000 mm/min Vorschub der Achse Y stellt sich auf 500 mm/min ein, da nur der halbe Weg zu fahren ist.

#### Hinweis

Beim Befehl  $FGROU P$  ist der Kanalachsname zu verwenden.

Dies wird definiert durch das Maschinendatum:

MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB (Kanalachsname im Kanal)

## Anwendung

Bei Schraubenlinieninterpolation (Helixinterpolation) kann durch `FGROUP` wahlweise eingestellt werden:

- Ob der programmierte Vorschub auf der Bahn gelten soll  
(alle 3 programmierten Achsen sind Bahnachsen)
- Ob der programmierte Vorschub auf dem Kreis gelten soll  
(2 Achsen sind Bahnachsen und die Zustellachse ist eine Synchronachse)

### 9.2.11 Achskonfiguration

#### Zuordnung von Geometrie-, Zusatz-, Kanal- und Maschinenachsen

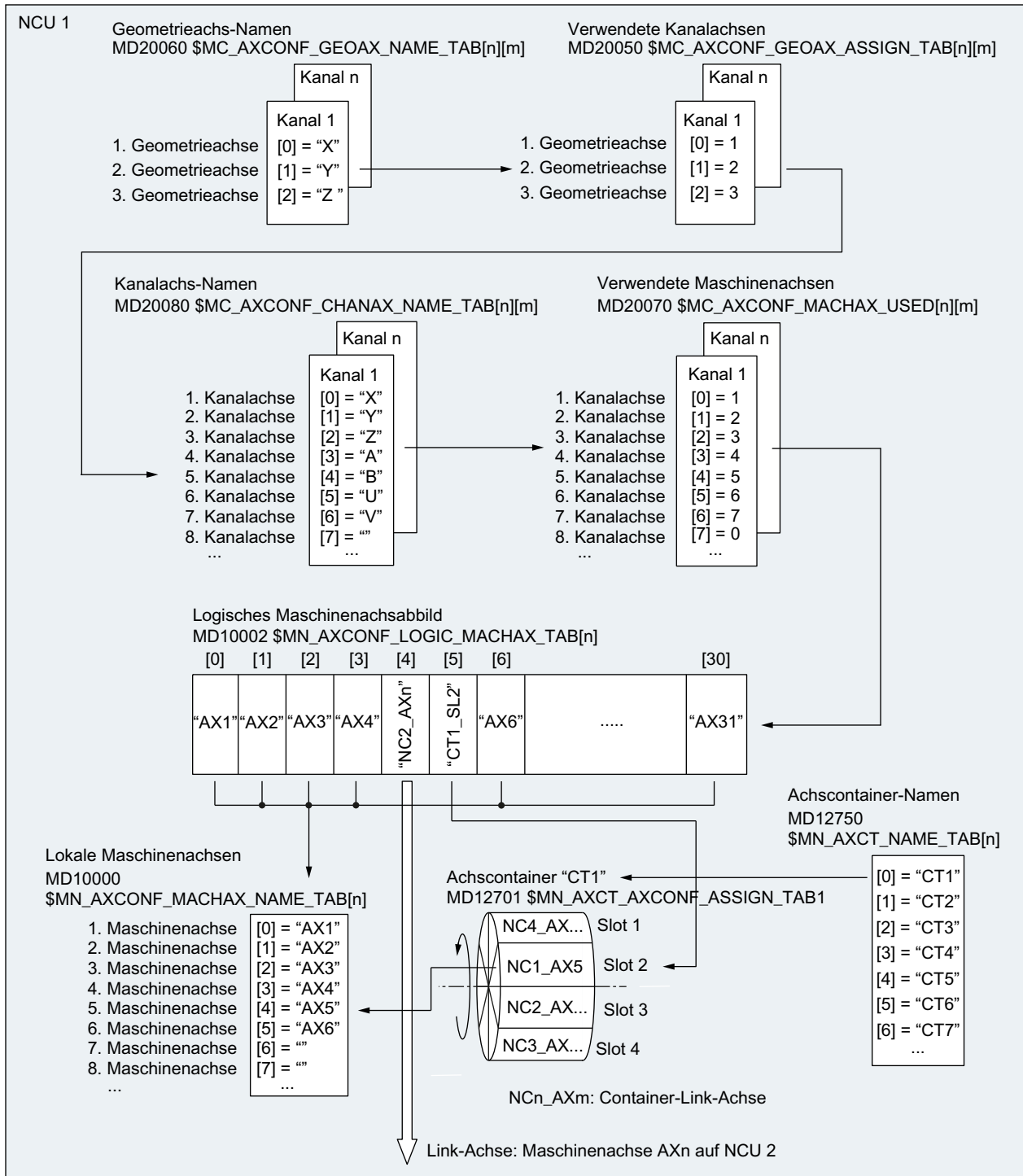


Bild 9-5 Achskonfiguration

## Besonderheiten

- Führende Nullen bei anwenderdefinierten Achsbezeichnern werden ignoriert:  
MD10000 ` \$MN\_AXCONF\_MACHAX\_NAME\_TAB[0] = X01 entspricht X1
- Die Zuordnung der Geometrieachsen zu den Kanalachsen muss lückenlos und in aufsteigender Reihenfolge erfolgen.
- Alle Kanalachsen, die keine Geometrieachsen sind, sind Zusatzachsen.

## Kanalachslücken

Im Normalfall muss jeder im Maschinendatum MD20080  
\$MC\_AXCONF\_CHANAX\_NAME\_TAB definierten Kanalachse eine Maschinenachse zugeordnet werden.

Zur Vereinfachung der Inbetriebnahme von Maschinenbaureihen mit unterschiedlicher Anzahl von Maschinenachsen, dürfen auch Kanalachsen definiert sein, denen keine Maschinenachse zugeordnet ist. In der Reihenfolge der Aufzählung der Kanalachsen können dadurch Lücken entstehen.

Die Zulässigkeit von Kanalachslücken muss explizit freigegeben werden:

MD11640 \$MN\_ENABLE\_CHAN\_AX\_GAP = 1

Ohne Freigabe beendet ein Wert von 0 im Maschinendatum:

MD20070 \$MC\_AXCONF\_MACHAX\_USED

die Zuweisung eventuell vorhandener weiterer Maschinenachsen zu Kanalachsen.

### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme ( B3)

---

### Hinweis

Kanalachsen ohne zugeordnete Maschinenachsen (Kanalachslücken) werden bezüglich Anzahl und Indizierung der Kanalachsen wie normale Kanalachsen mit zugeordneten Maschinenachsen behandelt.

Wird eine Kanalachse ohne zugeordnete Maschinenachse (Kanalachslücke) als Geometrieachse definiert, wird dies ohne Alarm abgewiesen.

---

**Beispiel: Kanalachslücke**

Im folgenden Beispiel ist der Kanalachse B keine Maschinenachse zugeordnet.

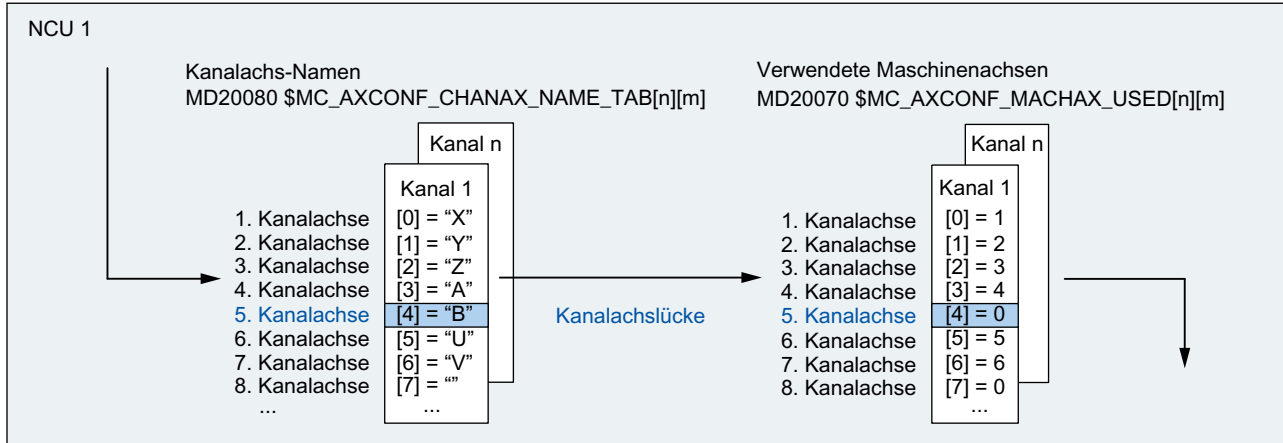


Bild 9-6 Achskonfiguration mit Kanalachslücke (Ausschnitt)

**Besonderheiten: Kanalachslücken**

Bezüglich Kanalachslücken ist weiter zu beachten:

- Kanalachsen ohne zugeordnete Maschinenachsen (Kanalachslücken) werden bezüglich Anzahl und Indizierung der Kanalachsen wie normale Kanalachsen mit zugeordneten Maschinenachsen behandelt.
- Wird eine Kanalachse ohne zugeordnete Maschinenachse (Kanalachslücke) als Geometrieachse definiert, wird dies ohne Alarm abgewiesen.



## 9.2.12 Link-Achsen

### Bedeutung

Eine Link-Achse ist eine Maschinenachse, die nicht auf der NCU liegt von der aus sie verfahren wird. Im Maschinendatum für das logische Maschinenachsabbild der verfahrenenden NCU wird dazu nicht der Bezeichner einer lokalen Maschinenachse eingetragen, sondern der NCU- und der Maschinenachs-Bezeichner der NCU an der sie physikalisch angeschlossen ist.

Beispielhaft soll die Maschinenachse AX1 der NCU2 von NCU1 aus verfahren werden:

- NCU1: MD10002 \$MN\_AXCONF\_LOGIC\_MACHAX\_TAB[n] = NC2\_AX1

### Voraussetzung

Als Voraussetzung für die Verwendung von Link-Achsen müssen die beteiligten NCUs per Link-Kommunikation verbunden sein. Die Funktionen Link-Achsen und Link-Kommunikation sind ausführlich beschrieben in:

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3)





## 9.3 Nullpunkte und Referenzpunkte

### 9.3.1 Bezugspunkte im Arbeitsraum

#### Nullpunkte und Referenzpunkte

Aus den Koordinatenachsen und den konstruktiven Merkmalen der Maschine ergibt sich deren Nullstellung. Den Nullpunkt des Koordinatensystems erhält man durch Festlegung eines zweckmäßigen Bezugspunktes an der Maschine in ihrer Nullstellung.

Die Lage der Koordinatensysteme (MKS, BKS, BNS, ENS, WKS) wird durch Nullpunkte festgelegt.

Nullpunkte		Referenzpunkte	
	<b>M</b> = Maschinennullpunkt		<b>R</b> = Referenzpunkt
	<b>W</b> = Werkstücknullpunkt		<b>T</b> = Werkzeugträgerbezugspunkt

#### Maschinennullpunkt M

Mit dem Maschinennullpunkt M wird das Maschinen-Koordinatensystem MKS festgelegt. Auf den Maschinennullpunkt beziehen sich alle anderen Bezugspunkte.

#### Werkstücknullpunkt W

Der Werkstücknullpunkt W legt das Werkstück-Koordinatensystem in Bezug auf den Maschinennullpunkt M fest. Im Werkstück-Koordinatensystem WKS werden die programmierten Teileprogrammsätze abgefahren.

#### Referenzpunkt R

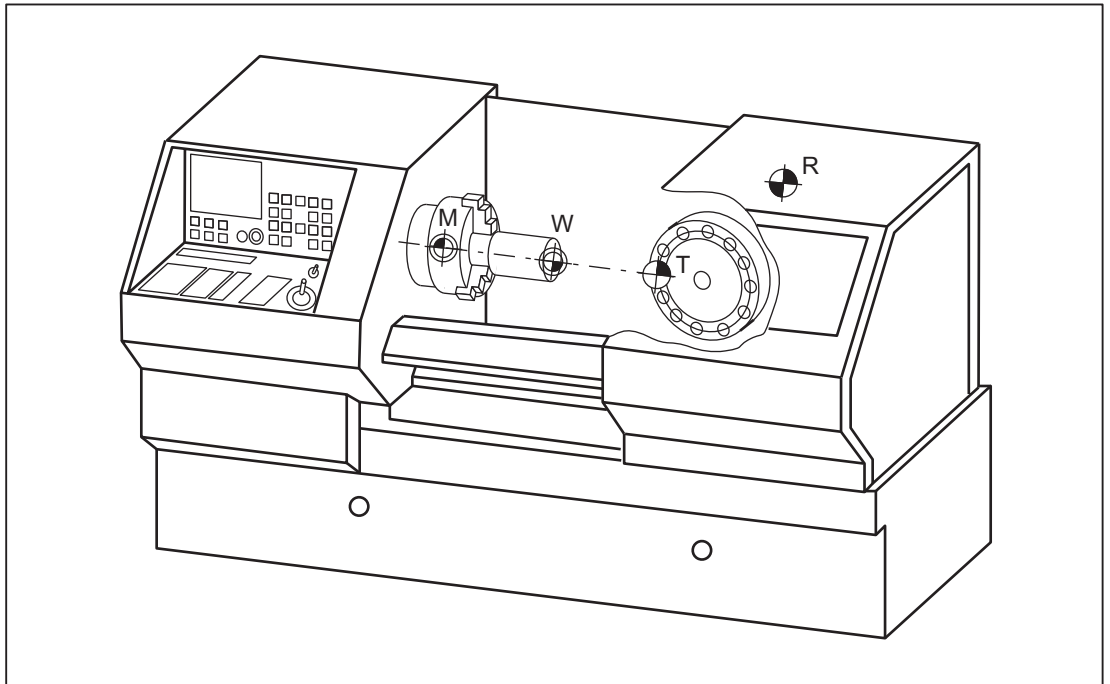
Die Position des Referenzpunktes R wird durch Nockenschalter vorgegeben. Durch ihn wird das Wegmesssystem geeicht.

Bei inkrementellen Messgebern muss der Referenzpunkt nach jedem Einschalten der Steuerung angefahren werden. Erst dann kann die Steuerung mit dem Messsystem arbeiten und alle Positionswerte auf die Koordinatensysteme übertragen.

#### Werkzeugträgerbezugspunkt T

Der Werkzeugträgerbezugspunkt T befindet sich an der Werkzeughalteraufnahme. Durch Eingabe der Werkzeuglängen berechnet die Steuerung den Abstand der Werkzeugspitze (TCP-Tool Center Position) vom Werkzeugträgerbezugspunkt.

Beispiel: Nullpunkte und Referenzpunkte bei einer Drehmaschine



### 9.3.2 Lage der Koordinatensysteme und Referenzpunkte

#### Einschalten der Steuerung

Bei inkrementellen Messgebern muss der Referenzpunkt nach jedem Einschalten der Steuerung angefahren werden, damit die Steuerung alle Positionswerte auf das Koordinatensystem übertragen kann.

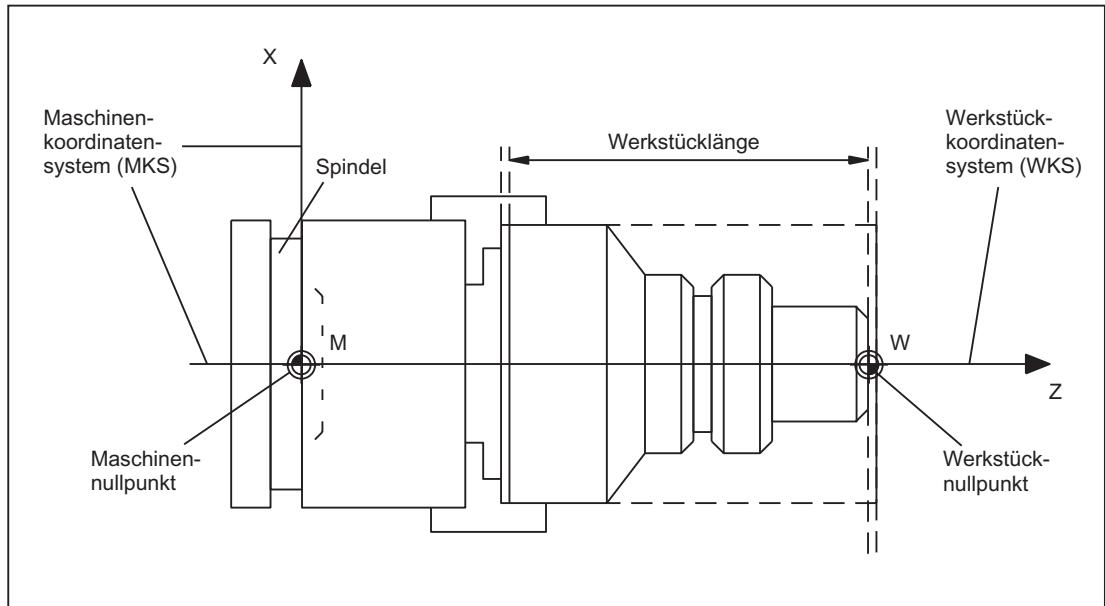


Bild 9-7 Lage der Koordinatensysteme durch Maschinennullpunkt M und Werkstücknullpunkt W

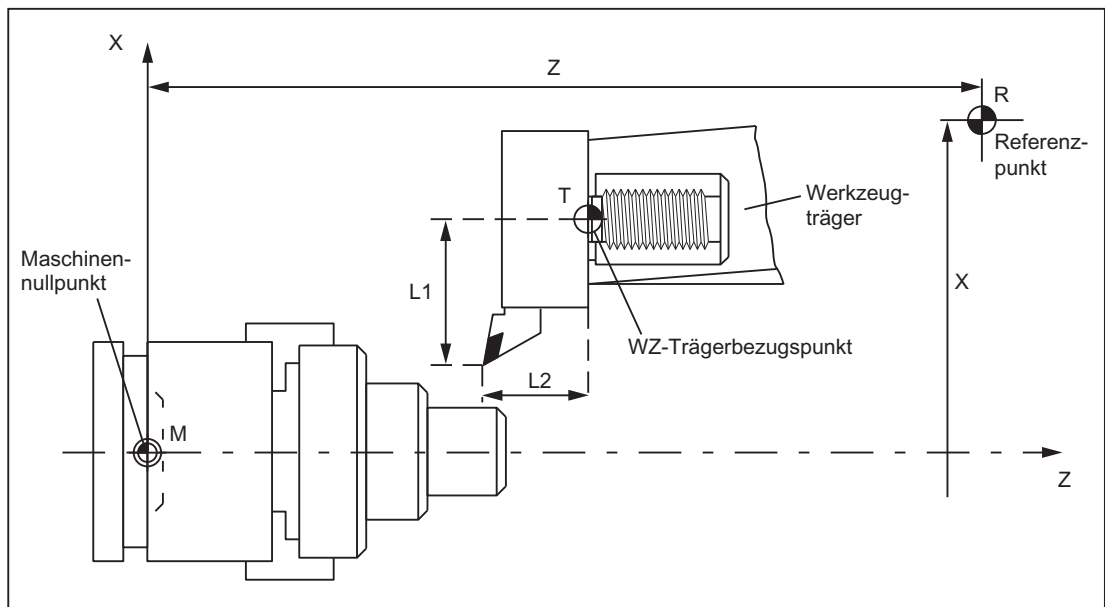


Bild 9-8 Lage des Referenzpunktes zum Maschinennullpunkt

## 9.4 Koordinatensysteme

### 9.4.1 Übersicht

#### Kartesische Koordinatensysteme

Nach DIN 66217 werden für Werkzeugmaschinen rechtwinkelige (kartesische) Koordinatensysteme benutzt. Mit der "Rechten-Hand-Regel" werden die positiven Richtungen der Koordinatenachsen festgelegt. Das Koordinatensystem wird auf das Werkstück bezogen und die Programmierung erfolgt unabhängig davon, ob das Werkzeug oder das Werkstück bewegt wird. Bei der Programmierung wird immer davon ausgegangen, dass sich das Werkzeug relativ zum Koordinatensystem des stillstehend gedachten Werkstückes bewegt.

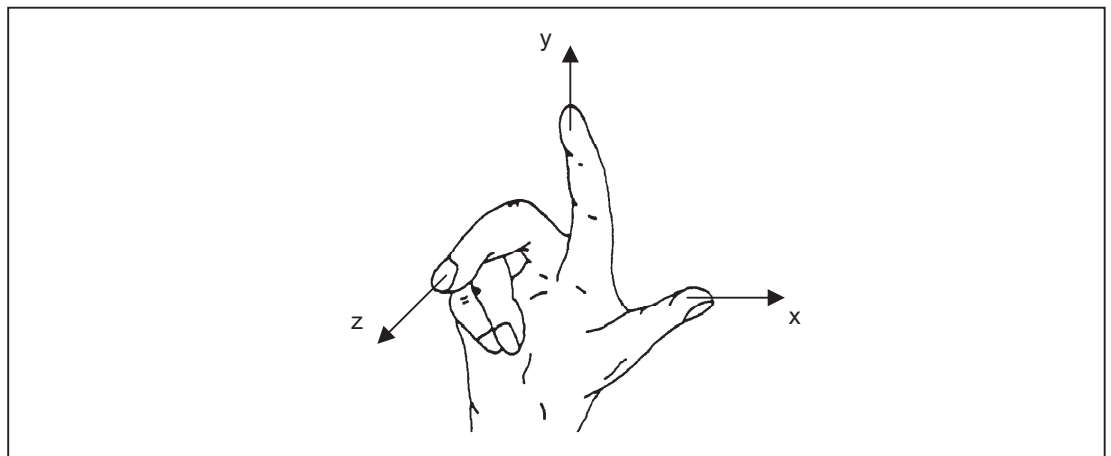


Bild 9-9 Rechte-Hand-Regel

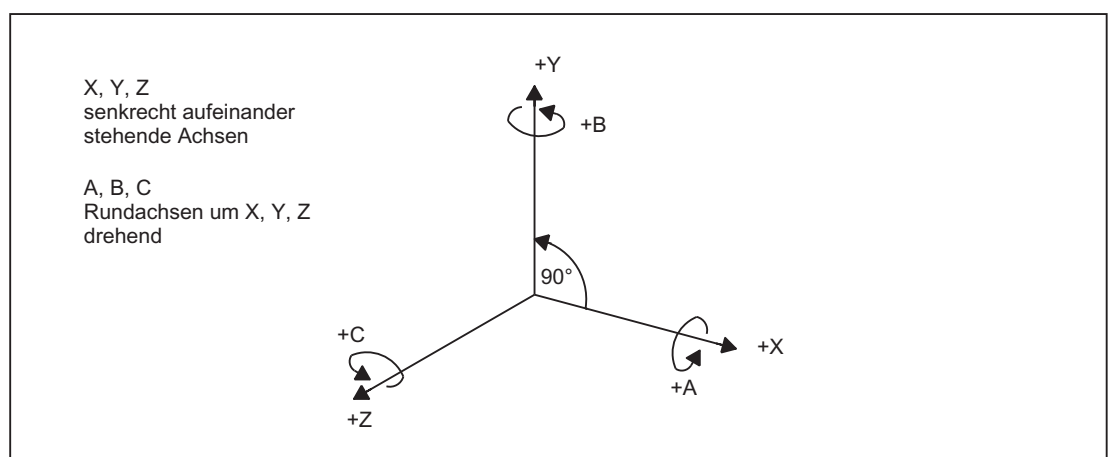


Bild 9-10 Rechtsdrehendes, rechtwinkliges kartesisches Koordinatensystem

Es sind folgende Koordinatensysteme definiert:

MKS	<b>M</b> aschinen- <b>K</b> oordinaten- <b>S</b> ystem
BKS	<b>B</b> asis- <b>K</b> oordinaten- <b>S</b> ystem
BNS	<b>B</b> asis- <b>N</b> ullpunkt- <b>S</b> ystem
ENS	<b>E</b> instellbares <b>N</b> ullpunkt- <b>S</b> ystem
WKS	<b>W</b> erkstück- <b>K</b> oordinaten- <b>S</b> ystem

### Zusammenhänge zwischen den Koordinatensystemen

Die Koordinatensysteme sind durch die kinematische Transformation und die FRAMES bestimmt.

Durch eine kinematische Transformation wird das MKS in das BKS überführt. Ist keine kinematische Transformation aktiv, so entspricht das BKS dem MKS.

Der Basisframe bildet das BKS auf das BNS ab.

Ein aktivierter einstellbarer `FRAME G54...G599` führt das BNS in das ENS über.

Durch den programmierbaren `FRAME` wird das WKS festgelegt, welches die Basis für die Programmierung darstellt.

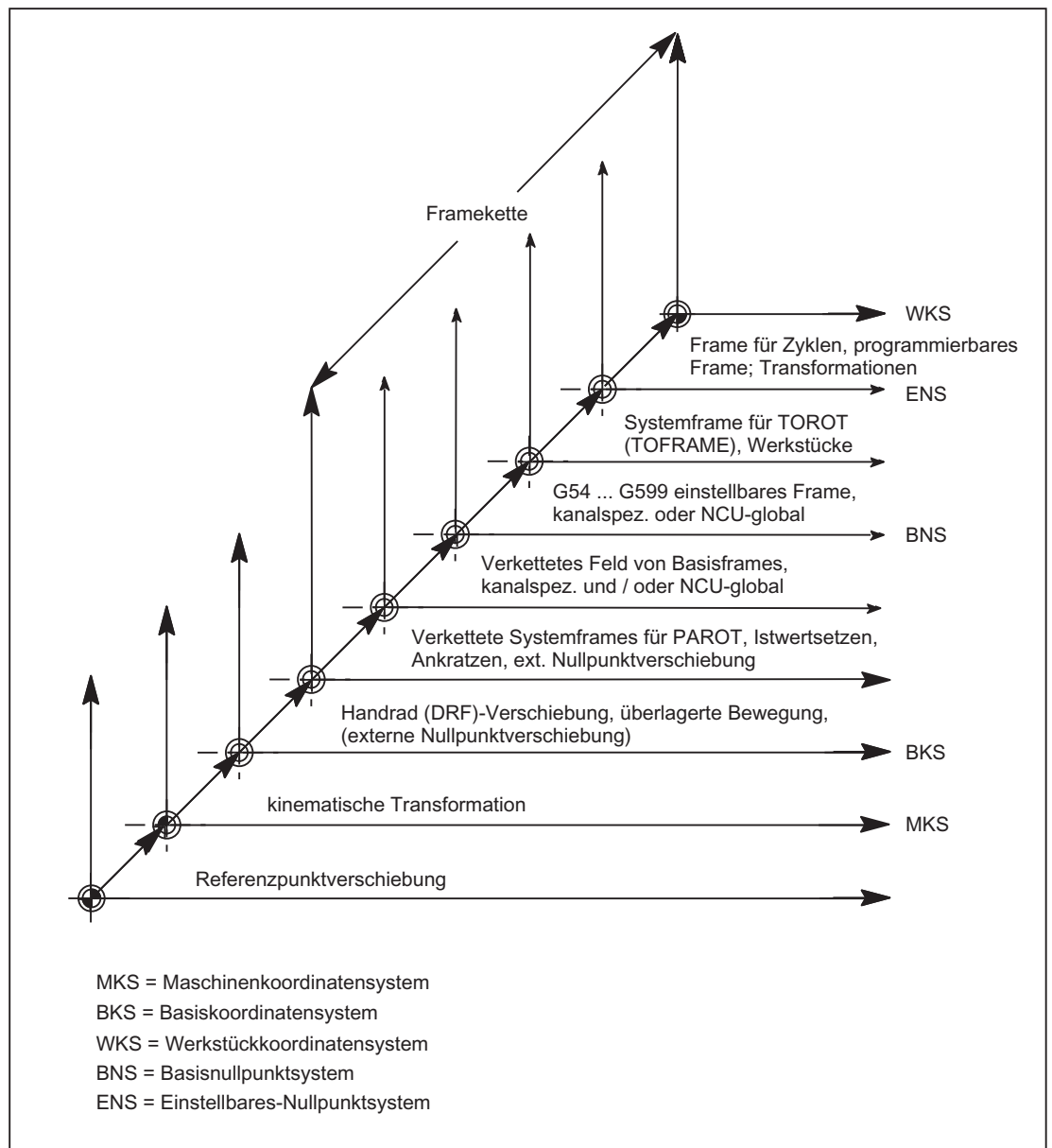


Bild 9-11 Zusammenhänge zwischen den Koordinatensystemen

## 9.4.2 Maschinenkoordinatensystem (MKS)

### Maschinenkoordinatensystem (MKS)

Das Maschinenkoordinatensystem (MKS) wird aus allen physikalisch vorhandenen Maschinenachsen gebildet.

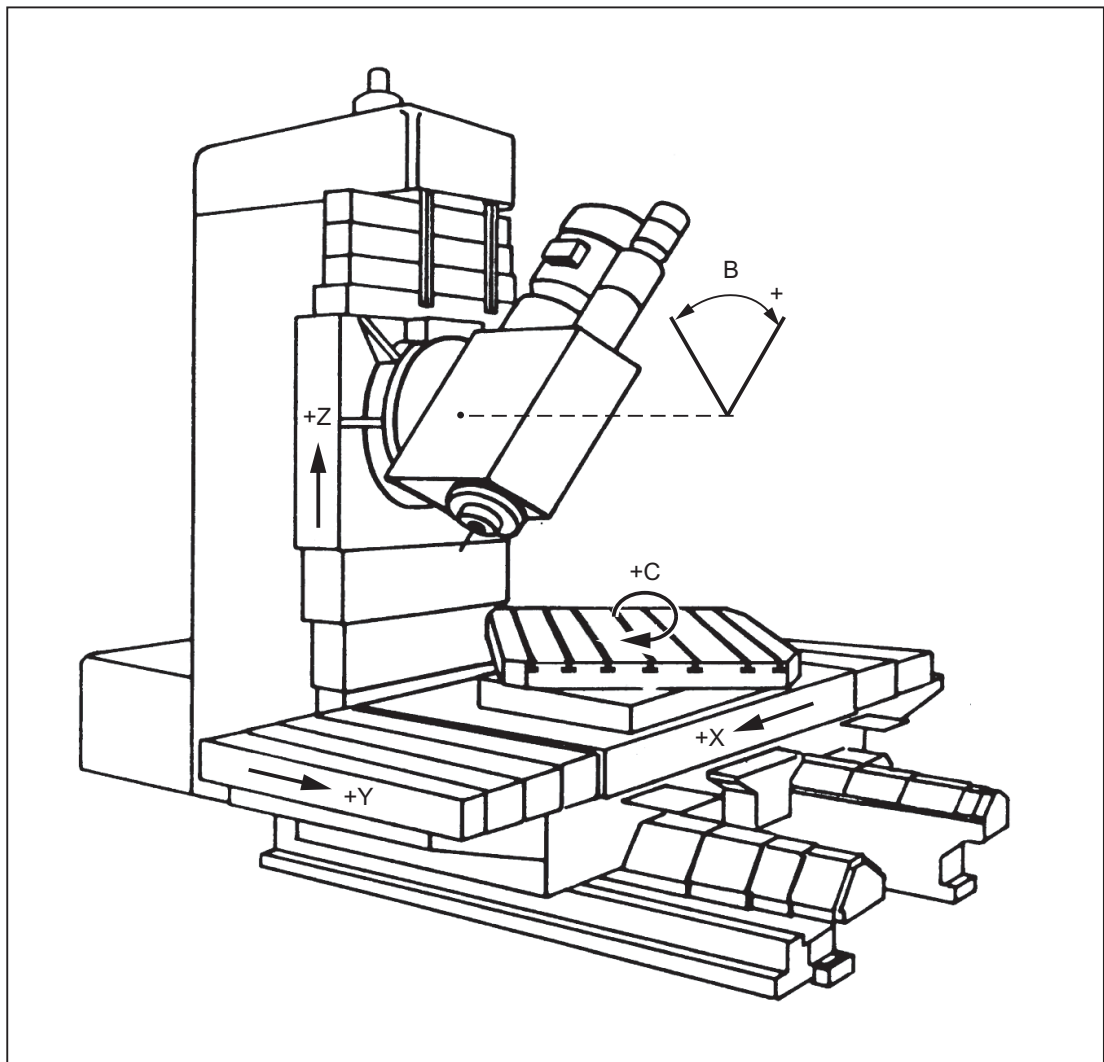


Bild 9-12 MKS mit den Maschinenachsen X, Y, Z, B, C (5-Achs-Fräsmaschine)



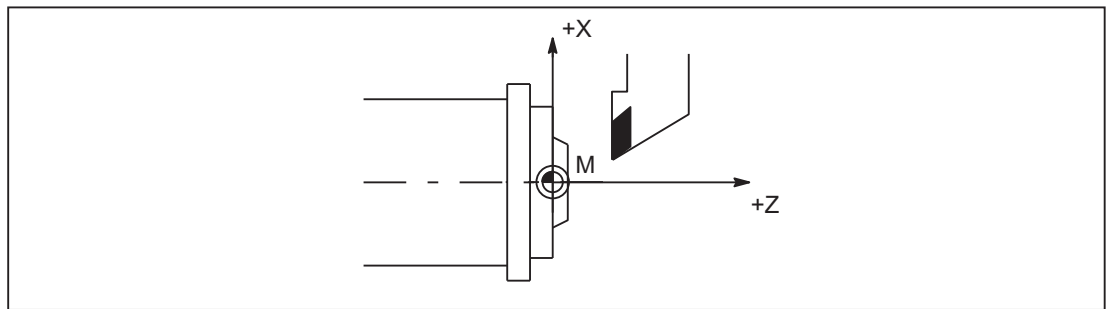


Bild 9-13 MKS mit den Maschinenachsen X, Z (Drehmaschine)

### Axiale Preset-Verschiebung

Über die Funktion "Preset-Verschiebung (PRESETON)" kann der Bezugspunkt der Steuerung im Maschinenkoordinatensystem (Maschinennullpunkt) neu gesetzt werden.

---

#### Hinweis

Es wird empfohlen, die Funktion **nur** für Maschinenachsen ohne Referenzpunktpflicht zu verwenden.

---

Zum Wiederherstellen des ursprünglichen Maschinenkoordinatensystems muss die Maschinenachse z. B. mit G74 (Referenzpunktfahren) erneut referenziert werden.

Bei der Preset-Verschiebung findet keine Bewegung der Maschinenachsen statt.

#### VORSICHT

Nach einer Preset-Verschiebung ist die entsprechende Maschinenachse im Status "nicht referenziert"! Das bedeutet bei Verwendung von Absolutgebern, dass die Geber-Justage verlorengegangen und neu durchgeführt werden muss (z. B. durch Einmessen mit einem Laser-Interferometer). Die Verwendung von PRESETON in Kombination mit Absolutgebern ist deshalb nicht zu empfehlen.

#### Literatur

- Programmieranleitung Grundlagen  
Kapitel: "Ergänzende Befehle" > "Referenzpunktfahren (G74)"
- Programmierhandbuch Arbeitsvorbereitung  
Kapitel: "Koordinatentransformationen (FRAMES)" > "Preset-Verschiebung (PRESETON)"

### 9.4.3 Basiskoordinatensystem (BKS)

#### Basiskoordinatensystem (BKS)

Das Basiskoordinatensystem (BKS) besteht aus drei rechtwinklig angeordneten Achsen (Geometrieachsen), sowie aus weiteren Achsen (Zusatzachsen) ohne geometrischen Zusammenhang.

#### WZ-Maschinen ohne kinematische Transformation

Das BKS und das MKS fallen immer dann zusammen, wenn das BKS ohne kinematische Transformation (z. B. TRANSMIT / Stirnflächen-Transformation, 5-Achstransformation und max. drei Maschinenachsen) auf das MKS abgebildet werden kann.

Bei diesen Maschinen können Maschinenachsen und Geometrieachsen den gleichen Namen haben.

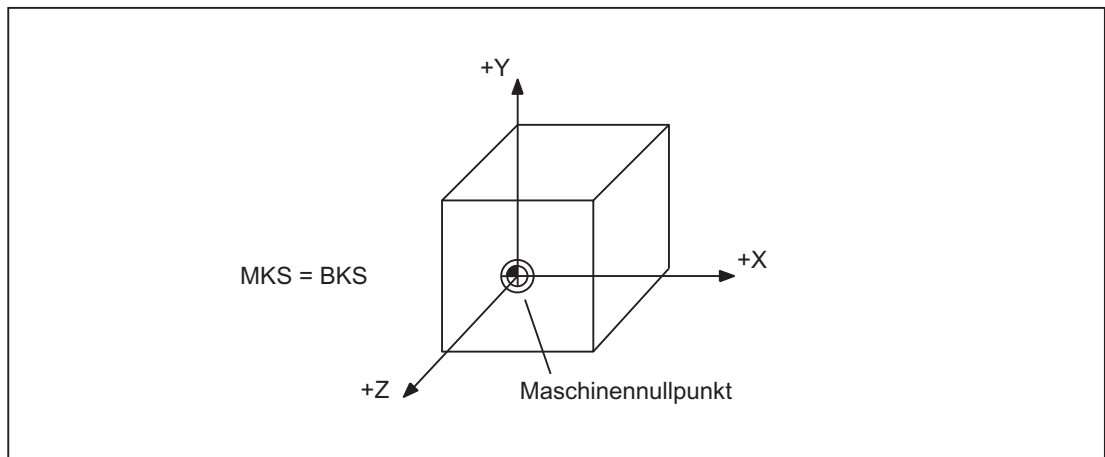


Bild 9-14 MKS=BKS ohne kinematische Transformation

#### WZ-Maschinen mit kinematischer Transformation

Das BKS und das MKS fallen nicht zusammen, wenn das BKS mit kinematischer Transformation (z. B. TRANSMIT / Stirnflächen-Transformation, 5-Achstransformation oder mehr als drei Achsen) auf das MKS abgebildet wird.

Bei diesen Maschinen müssen Maschinenachsen und Geometrieachsen unterschiedliche Namen haben.

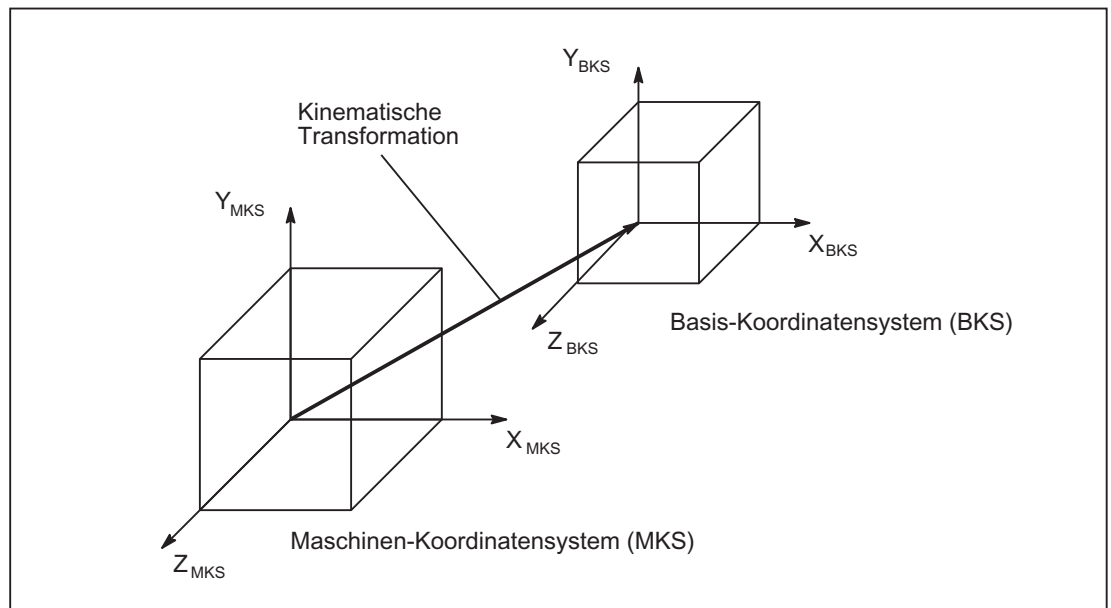


Bild 9-15 Kinematische Transformation zwischen MKS und BKS

## Maschinenkinematik

Das Werkstück wird immer in einem zwei- oder dreidimensionalen rechtwinkligen Koordinatensystem (WKS) programmiert. Zur Fertigung dieser Werkstücke werden aber immer häufiger Werkzeugmaschinen mit Rundachsen oder nicht rechtwinklig angeordneten Linearachsen eingesetzt. Zur Abbildung der im WKS programmierten Koordinaten (rechtwinklig) in reale Maschinenachsbewegungen dient die kinematische Transformation.

### Literatur:

Funktionshandbuch Sonderfunktionen; 3- bis 5-Achs-Transformation (F2)

Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformation (M1)

### 9.4.4 Additive Korrekturen

#### Externe Nullpunktverschiebungen

Die "Externe Nullpunktverschiebung" ist eine axiale Verschiebung. Im Gegensatz zu Frames sind keine Anteile für Rotation, Skalierung und Spiegelung möglich.

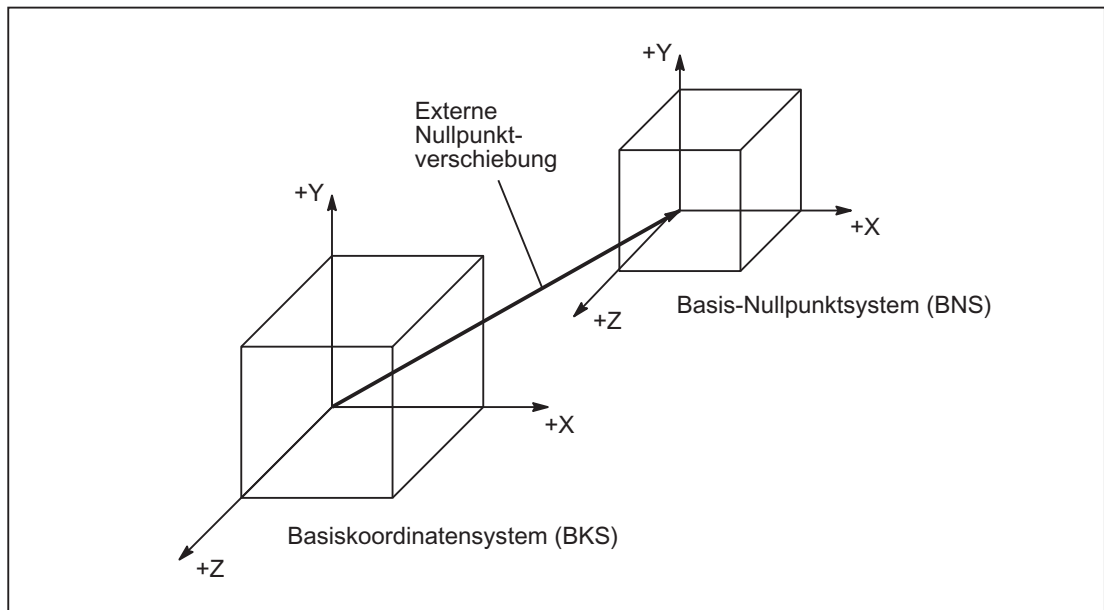


Bild 9-16 Externe Nullpunktverschiebung zwischen BKS und BNS

#### Vorgabe der Verschiebe-Werte

Verschiebe-Werte werden vorgegeben von:

- PLC  
Durch Beschreiben der Systemvariablen.
- Über Bedientafelfront  
Aus dem Menü "Aktuelle Nullpunkt-Verschiebungen".
- NC-Programm  
Zuweisung an Systemvariable \$AA\_ETRANS[Achse].

#### Aktivierung der Verschiebe-Werte

Die 0/1 Flanke des folgenden PLC-Signals aktiviert die zuvor definierten Verschiebe-Werte:

DB31, ... DBX3.0 (Externe Nullpunktverschiebung übernehmen)

Die Auswertung des 0/1 Flankenwechsels erfolgt nur in der Betriebsart "Automatik".

## Wirkung der Aktivierung

Die aktivierte Verschiebung einer Achse wird ab dem Bearbeiten des nächsten Bewegungssatzes für diese Achse nach der Aktivierung wirksam.

**Beispiel für möglichen zeitlichen Ablauf:**

Programmcode	Kommentar
G0 X100	
X150	; während dieser Bewegung wird eine neue "Externe Nullpunktverschiebung" von der PLC aktiviert
X200	; die neue "Externe Nullpunktverschiebung" wird wegen G0-Programmierung am Satzende (X200) herausgefahren, falls keine Geschwindigkeitsreserve vorhanden ist (Override 100%)

Die "Externe Nullpunktverschiebung über Systemframe" wird sofort herausgefahren.

Kanalspezifische Systemframes können über das folgende Maschinendatum projiziert werden:

MD28082 \$MC\_MM\_SYSTEM\_FRAME\_MASK (Systemframes SRAM)

## Programmierung

Setzen einer neuen Verschiebung über die achsspezifischen Systemvariablen:

\$AA\_ETRANS[Achse]=R<sub>i</sub>

Die folgende Anweisung liest den achsspezifischen aktiven Verschiebe-Wert:

R<sub>i</sub>=\$AA\_ETRANS[Achse]

---

### Hinweis

Der gelesene Wert kann sich von einem zuvor gesetzten Wert dann unterscheiden, wenn noch keine Aktivierung auf den gesetzten Wert erfolgt ist.

Der gelesene Wert entspricht einem früher gesetzten Wert, wenn für den zuletzt gesetzten Wert noch keine Aktivierung stattfand. Das Systemframe für die "Externe Nullpunktverschiebung" ist nur vorhanden, wenn es auch projiziert wurde.

---

## DRF-Verschiebung

Die DRF-Verschiebung ermöglicht das Einstellen einer additiven inkrementellen Nullpunktverschiebung für Geometrie- und Zusatzachsen im Basiskoordinatensystem **über Handrad**.

Die DRF-Verschiebung kann gelesen werden über die achsspezifische Systemvariable:

\$AC\_DRF[<Achse>]

### Literatur

Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1), Kapitel: DRF-Verschiebung

## Überlagerte Bewegungen

Die "Überlagerte Bewegung" für die programmierte Achse ist nur aus Synchronaktionen über die Systemvariable \$AA\_OFF[Achse] zugreifbar.

## Hochlauf

Nach Hochlauf (Power On) sind die zuletzt berücksichtigten Verschiebe-Werte der "Externen Nullpunktverschiebung" gespeichert und werden erst mit erneutem Aktivierungssignal wirksam.

Abhängig von folgendem Maschinendatum bleiben Systemframes bei Power On erhalten:

MD24008 \$MC\_CHSFRAME\_POWERON\_MASK (Systemframes nach Power On zurücksetzen)

## RESET / Programmende

Die aktivierten Werte bleiben über RESET und Programmende hinweg aktiv.

Reset-Verhalten der kanalspezifischen Systemframes wie folgt:

Mit der folgenden Maschinendatum-Einstellung ist Systemframe für die "Externe Nullpunktverschiebung" nach RESET aktiv:

MD24006 \$MC\_CHSFRAME\_RESET\_MASK, Bit 1 = 1

Mit der folgenden Maschinendatum-Einstellung wird die "Externe Nullpunktverschiebung" im aktiven Systemframe in der Datenhaltung gelöscht:

MD24006 \$MC\_CHSFRAME\_RESET\_MASK, Bit 1 = 0

Nach RESET sind folgende Frames aktiv:

- Systemframe für:
  - MD24006 \$MC\_CHSFRAME\_RESET\_MASK, Bit 4 = 1 (Werkstückbezugspunkte)
  - MD24006 \$MC\_CHSFRAME\_RESET\_MASK, Bit 5 = 1 (Zyklen)

## Unterdrückung

Die NC-Programm-Anweisung *SUPA* unterdrückt die "Externe Nullpunktverschiebung" für die Dauer der Bearbeitung des Satzes.

Der Befehl *G74* (Referenzpunktfahren) sowie die entsprechende Bedienung in der Betriebsart "Referenzpunktfahren" unterdrücken die "Externe Nullpunktverschiebung" für die Dauer des Referenzpunktfahrens.

Bei *G74*, d. h. Betriebsart "Automatik" oder "MDA", wird die zuvor aktive "Externe Nullpunktverschiebung" mit der nächsten Verfahrbewegung im Satz wieder automatisch aktiv.

Nach einem Betriebsartenwechsel aus der Betriebsart "Referenzpunktfahren" muss für die referierten Achsen das NC/PLC-Nahtstellensignal zur erneuten Aktivierung gesetzt werden.

## 9.4.5 Basis-Nullpunktsystem (BNS)

### Basis-Nullpunktsystem (BNS)

Das Basis-Nullpunktsystem (BNS) ergibt sich aus dem Basis-Koordinatensystem durch die Basisverschiebung.

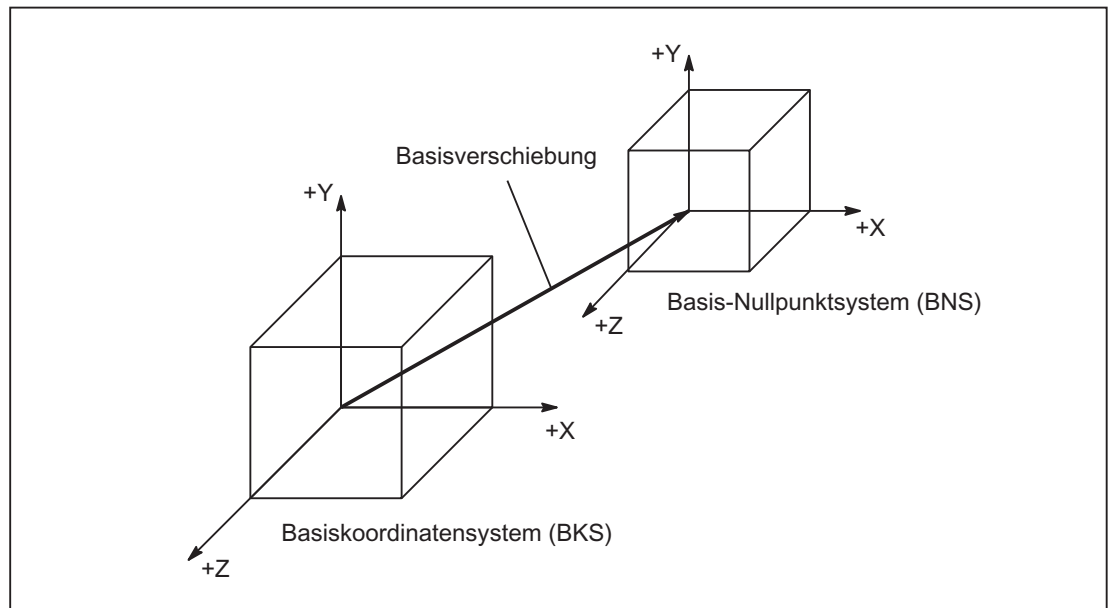


Bild 9-17 Basisverschiebung zwischen BKS und BNS

### Basisverschiebung

Die Basisverschiebung beschreibt die Koordinatentransformation zwischen dem BKS und BNS. Mit ihr kann z. B. der Paletten-Nullpunkt festgelegt werden.

Die Basisverschiebung setzt sich zusammen aus:

- Externe Nullpunktverschiebung
- DRF-Verschiebung
- Überlagerte Bewegung
- Verkettete Systemframes
- Verkettete Basisframes

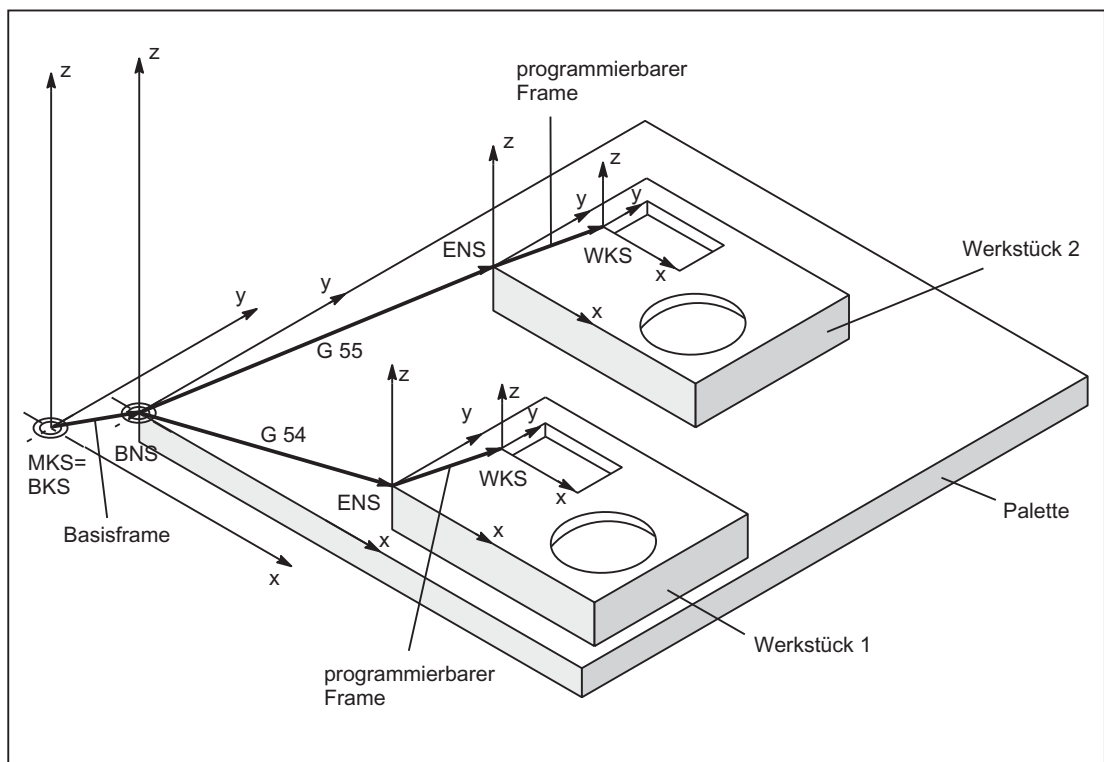


Bild 9-18 Beispiel für Anwendung der Basisverschiebung



Es gilt:

- Der Anwender kann die Basisverschiebung aus dem Teileprogramm, der Bedienung und von der PLC verändern.
- Soll die Basisverschiebung sofort wirksam werden, so kann über PLC mit `FC9` ein ASUP gestartet werden, der den entsprechenden `G`-Code ausführt.

---

#### Hinweis

#### Empfehlung an den Maschinenhersteller

Verwenden Sie für eigene Anwendungen Verschiebungen ab der 3. Basisverschiebung.

Die 1. und 2. Basisverschiebung sind für das "Istwertsetzen" und die "Externe Nullpunktverschiebung" vorgesehen.

---

## 9.4.6 Einstellbares Nullpunktsystem (ENS)

### Einstellbare Nullpunktsystem (ENS)

Das "Einstellbare Nullpunktsystem" (ENS) ist das Werkstückkoordinatensystem WKS mit programmierbarem FRAME (gesehen aus der Perspektive WKS). Der Werkstücknullpunkt wird durch die einstellbaren FRAMES `G54...G599` festgelegt.

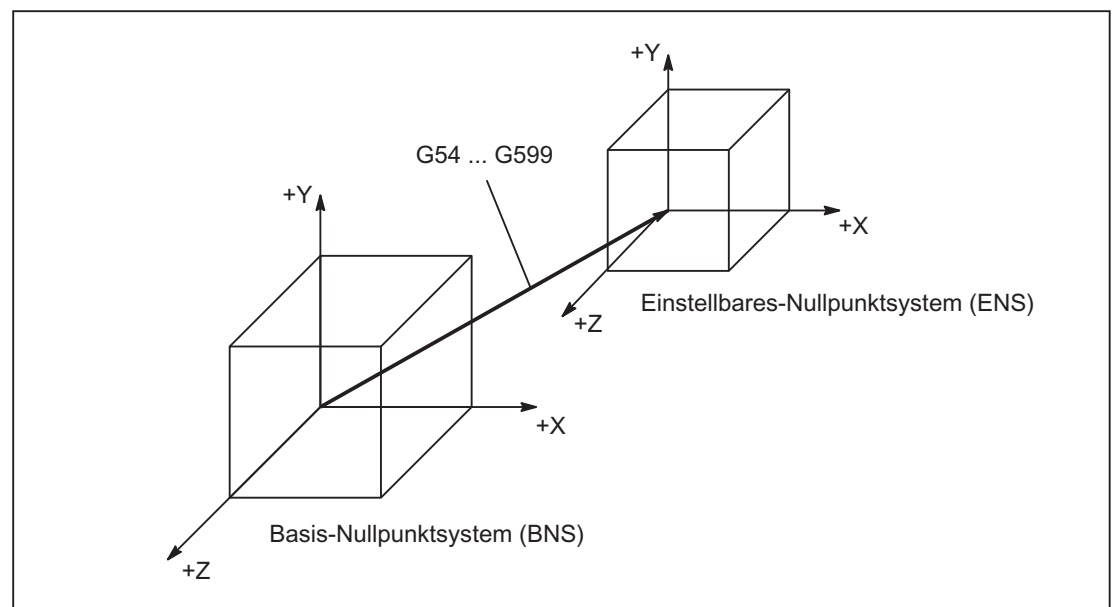


Bild 9-19 Einstellbarer FRAME `G54 ... G599` zwischen BNS und ENS

Vom "Einstellbaren-Nullpunktsystem" aus wirken programmierbare Verschiebungen. Alle programmierbaren Verschiebungen beziehen sich auf das "Einstellbare-Nullpunktsystem".

### WKS-Istwertanzeige im WKS oder ENS

Auf der HMI-Bedienoberfläche können die Istwerte der Achsen im Maschinenkoordinatensystem (MKS) oder im WKS angezeigt werden. Bei Anzeige im WKS können die Istwerte auch bezogen auf das ENS angezeigt werden. Die entsprechende Parametrierung erfolgt über das Maschinendatum:

MD9424 \$MM\_MA\_COORDINATE\_SYSTEM (Koordinatensystem für Istwertanzeige)

Wert	Bedeutung
0	Istwertanzeige bezogen auf das WKS
1	Istwertanzeige bezogen auf das ENS

#### Hinweis

##### Anzeige des aktuellen Koordinatensystems

Auf der HMI-Bedienoberfläche wird auch bei aktiver "Istwertanzeige bezogen auf das ENS" das WKS als Koordinatensystem angezeigt, auf das sich die Istwertanzeige bezieht.

### Beispiel

Istwertanzeige bezogen auf das WKS bzw. ENS

Code (Ausschnitt)	Istwertanzeige: Achse X (WKS)	Istwertanzeige: Achse X (ENS)
N10 X100	100	100
N20 X0	0	0
N30 \$P_PFRAME = CTRANS(X,10)	0	10
N40 X100	100	110

## 9.4.7 Werkstückkoordinatensystem (WKS)

### Werkstückkoordinatensystem WKS

Das Werkstückkoordinatensystem (WKS) ist die Basis für die Programmierung.

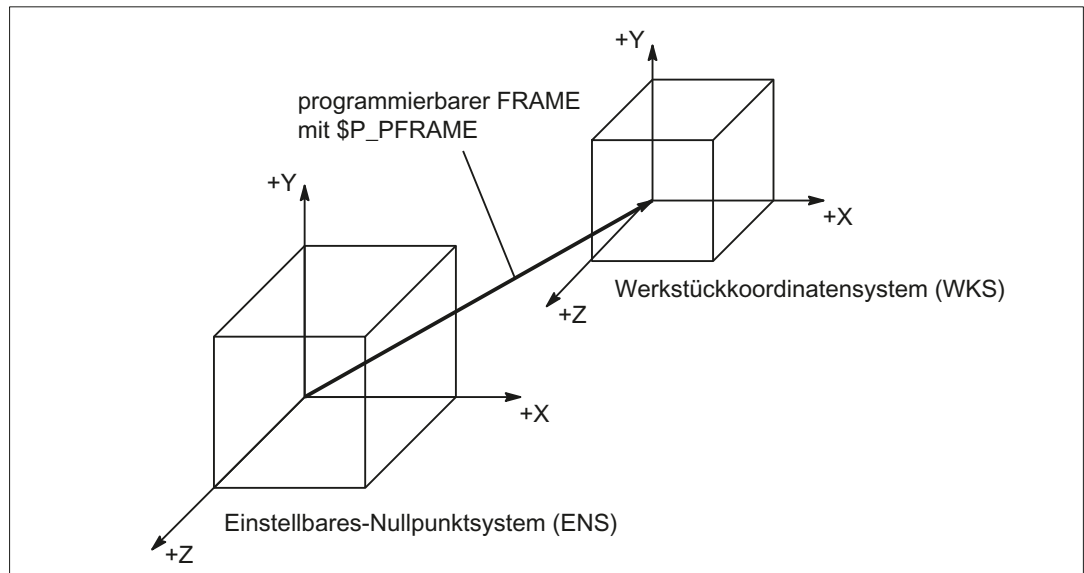


Bild 9-20 Programmierbarer FRAME zwischen ENS und WKS

## 9.5 Frames

### 9.5.1 Frame-Arten

Ein Frame ist eine Datenstruktur die Werte für Verschiebung (**TRANS**), Feinverschiebung (**FINE**), Drehung (**ROT**), Spiegelung (**MIRROR**) und Skalierung (**SCALE**) für Achsen enthält.

Bei Aktivierung des Frames wird anhand der Frame-Werte über eine definierte Rechenvorschrift eine statische Koordinatentransformation für die im Frame enthaltenen Achsen durchgeführt.

#### Axialer Frame

Ein axialer Frame enthält die Frame-Werte einer Achse.

Beispielhafte Datenstruktur eines axialen Frames für Achse X:

Achse	TRANS	FINE	ROT	MIRROR	SCALE
X	10.0	0.1	0.0	0	1

#### Kanalspezifischer Frame

Ein kanalspezifischer Frame enthält die Frame-Werte für alle Kanalachsen (Geometrie-, Zusatz- und Maschinenachsen).

Drehungen (ROT) werden nur bei Geometrieachsen eingerechnet.

Ein kanalspezifischer Frame wirkt nur in dem Kanal in dem der Frame definiert ist.

Beispielhafte Datenstruktur eines kanalspezifischen Frames:

- Geometrieachsen: X, Y, Z
- Zusatzachsen: A
- Maschinenachsen: AX1

Achse	TRANS	FINE	ROT	MIRROR	SCALE
X	10.0	0.1	0.0	0	1
Y	0.0	0.0	0.0	1	1
Z	0.0	0.0	45.0	0	1
A	2.0	0.1	0.0	0	2
AX1	0.0	0.0	0.0	0	0

## Globaler Frame

Ein globaler Frame enthält die Frame-Werte für alle Maschinenachsen.

Ein globaler Frame wirkt in allen Kanälen der NC.

Beispielhafte Datenstruktur eines kanalspezifischen Frames:

- Maschinenachsen: AX1, ... AX5

Achse	TRANS	FINE	ROT	MIRROR	SCALE
AX1	10.0	0.1	-	0	1
AX2	0.0	0.0	-	1	1
AX3	0.0	0.0	-	0	1
AX4	2.0	0.1	-	0	2
AX5	0.0	0.0	-	1	1

## 9.5.2 Frame-Komponenten

### 9.5.2.1 Translation

#### Programmierung

Die Programmierung der Translation bzw. Grobverschiebung kann über folgende Befehle erfolgen:

- Beispiel Datenhaltungsframes  $\$P\_UIFR$ 
  - Gesamtframe:  $\$P\_UIFR[\langle n \rangle] = CTRANS(\langle K1 \rangle, \langle V1 \rangle [, \langle K2 \rangle, \langle V2 \rangle] [, \langle K3 \rangle, \langle V3 \rangle])$   
mit  $K_m$  = Koordinate x, y oder z und  $V_m$  = Verschiebung m
  - Frame-Komponente:  $\$P\_UIFR[\langle n \rangle, \langle K \rangle, TR] = \langle V \rangle$   
mit  $K$  = Koordinate x, y oder z und  $V$  = Verschiebung
- Beispiel Programmierbarer Frame
  - $TRANS \langle K1 \rangle \langle V1 \rangle [ \langle K2 \rangle \langle V2 \rangle ] [ \langle K3 \rangle \langle V3 \rangle ]$   
mit  $K_m$  = Koordinate x, y oder z und  $V_m$  = Verschiebung m

Programmbeispiele:

Programmcode	Bemerkung
$\$P\_UIFR[1] = CTRANS(X,10,Y,10)$	Gesamtframe
$\$P\_UIFR[1,X,TR] = 10$	Frame-Komponente
$TRANS X=10 Y=10$	Programmierbarer Frame

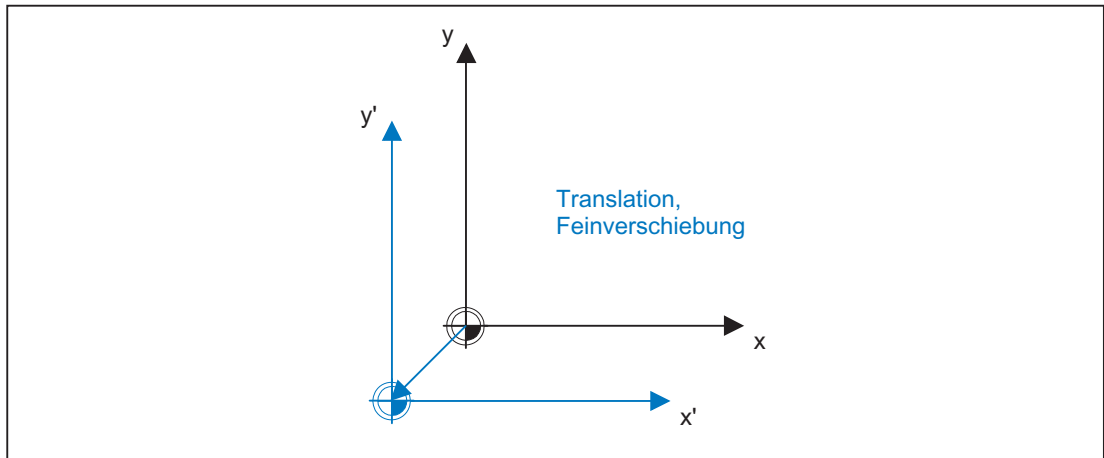


Bild 9-21 Verschiebung in Z-Richtung

### 9.5.2.2 Feinverschiebung

#### Parametrierung

Die Freigabe der Feinverschiebung erfolgt über das Maschinendatum:

MD18600 \$MN\_MM\_FRAME\_FINE\_TRANS = <Wert>

Wert	Bedeutung
0	Feinverschiebung kann nicht eingegeben bzw. nicht programmiert werden.
1	Feinverschiebung für einstellbare Frames, Basisframes und das Programmierbare Frame ist von der Bedienung oder über Programm möglich.

#### Programmierung

Die Programmierung der Translation bzw. Grobverschiebung kann über folgende Befehle erfolgen:

- Beispiel Datenhaltungsframes \$P\_UIFR
  - Gesamtframe: \$P\_UIFR[<n>] = CFINE(<K1>, <V1> [, <K2>, <V2>] [, <K3>, <V3>])  
mit Km = Koordinate x, y oder z und Vm = Verschiebung m
  - Frame-Komponente: \$P\_UIFR[<n>, <K>, FI] = <V>  
mit K = Koordinate x, y oder z und V = Verschiebung
- Beispiel Programmierbarer Frame
  - TRANS <K1> <V1> [, <K2> <V2>] [, <K3> <V3>]  
mit Km = Koordinate x, y oder z und Vm = Verschiebung m

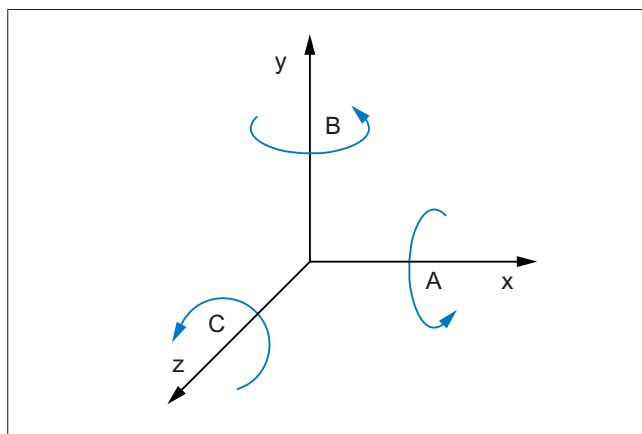
Programmierbeispiele:

Programmcode	Bemerkung
<code>\$P_UIFR[1] = CTRANS(X,10,Y,10)</code>	Gesamtframe
<code>\$P_UIFR[1,X,TR] = 10</code>	Frame-Komponente
<code>TRANS X=10 Y=10</code>	Programmierbarer Frame

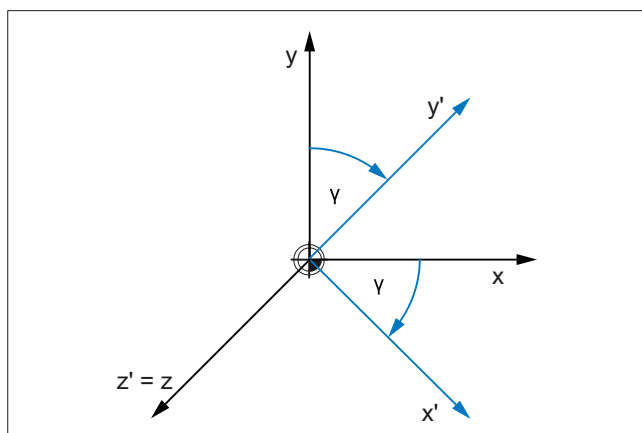
### 9.5.2.3 Drehung: Übersicht (nur Geometrieachsen)

#### Funktion

Die Drehrichtung um die Koordinatenachsen wird durch ein rechtshändiges, rechtwinkliges Koordinatensystem mit den Achsen  $x$ ,  $y$  und  $z$  bestimmt. Der Drehsinn der Drehung ist positiv, wenn die Drehbewegung bei Blick in die positive Richtung der Koordinatenachse im Uhrzeigersinn erfolgt. A, B und C bezeichnen Drehungen, deren Achsen parallel zu den Koordinatenachsen sind.



Das folgende Bild zeigt die neue Lage des Koordinatensystems  $x'$ ,  $y'$  und  $z'$  nach der Drehung um  $z$  mit  $\gamma = -45^\circ$



**Parametrierung der Drehreihenfolge**

Über das folgende Maschinendatum wird eingestellt, um welche Koordinatenachsen und in welcher Reihenfolge die Drehungen ausgeführt werden, wenn mehr als ein Drehwinkel programmiert ist:

MD10600 \$MN\_FRAME\_ANGLE\_INPUT\_MODE = <Wert>

Wert	Bedeutung
1	Euler-Winkel in zy'x"-Konvention (RPY-Winkel)
2	Euler-Winkel in zx'z"-Konvention

**Hinweis**

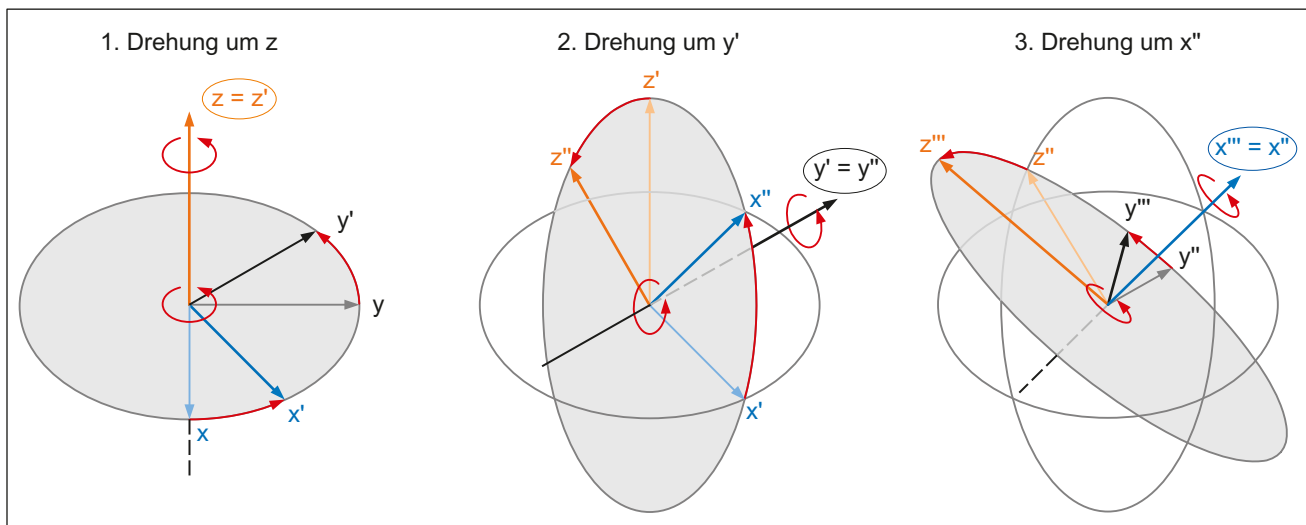
Aus historischen Gründen ist die Möglichkeit der Verwendung von Euler-Winkeln in zx'z"-Konvention vorhanden. Es wird aber dringend empfohlen ausschließlich Euler-Winkel in zy'x"-Konvention (RPY-Winkel) (siehe Kapitel "Drehung mit Euler-Winkeln: ZY'X"-Konvention (RPY-Winkel) (Seite 768)") zu verwenden.

**9.5.2.4 Drehung mit Euler-Winkeln: ZY'X"-Konvention (RPY-Winkel)**

Euler-Winkel in zy'x"-Konvention werden auch als RPY-Winkel bezeichnet. RPY ist abgeleitet aus dem Englischen:

- R: Roll (rollen) → Drehung um x
- P: Pitch (nicken) → Drehung um y'
- Y: Yaw (gieren) → Drehung um z''

Bei RPY-Winkeln erfolgen die Drehungen in der Reihenfolge z, y', x''.





**Zuordnung Drehachse zu Geometrieachse**

Drehachse	Geometrieachse im Kanal
x''	1. Geometrieachse
y'	2. Geometrieachse
z	3. Geometrieachse

**Wertebereich**

Bei RPY-Winkeln können programmierte Werte nur innerhalb folgender Wertebereiche eindeutig zurückgerechnet werden:

$$\begin{array}{rclcl}
 -180 & \leq & \mathbf{x} & \leq & 180 \\
 -90 & < & \mathbf{y} & < & 90 \\
 -180 & \leq & \mathbf{z} & \leq & 180
 \end{array}$$

**Programmierung des Gesamtframes**

Bei der Programmierung des Gesamtframes werden immer alle Drehkomponenten des Frames geschrieben. Nicht programmierte Komponenten werden implizit mit dem Wert 0° belegt.

**Syntax**

```

<Frame> = CROT([<1. GAx>,<Winkel>],[<2. GAx>,<Winkel>],[<3.
GAx>,<Winkel>])
ROT [<1. GAx><Winkel>] [<2. GAx><Winkel>] [<3. GAx><Winkel>]
AROT [<1. GAx><Winkel>] [<2. GAx><Winkel>] [<3. GAx><Winkel>]

```

**Bedeutung**

CROT:	Drehung absolut, Bezugsframe: beliebiger programmierter Frame								
<Frame>:	beliebiger aktiver oder Datenhaltungsframe								
ROT:	Drehung absolut, Bezugsframe: Programmierbarer Frame \$P_PFRAME, Bezugspunkt: Nullpunkt des aktuellen mit G54 ... G57, G505 ... G599 eingestellten Werkstückkoordinatensystems								
AROT:	Drehung additiv, Bezugsframe: Programmierbarer Frame \$P_PFRAME, Bezugspunkt: Nullpunkt des aktuellen mit G54 ... G57, G505 ... G599 eingestellten Werkstückkoordinatensystems								
<n-te GAx>:	Bezeichner der n-ten Geometrieachse um die um den angegebenen Winkel gedreht werden soll. Für nicht programmierte Geometrieachse wird als Drehwinkel implizit der Wert 0° gesetzt. Zuordnung von Geometrieachse zu Drehachse:								
	<table border="0"> <thead> <tr> <th style="text-align: center;">Geometrieachse</th> <th style="text-align: center;">Drehachse</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1. Geometrieachse</td> <td style="text-align: center;">x''</td> </tr> <tr> <td style="text-align: center;">2. Geometrieachse</td> <td style="text-align: center;">y'</td> </tr> <tr> <td style="text-align: center;">3. Geometrieachse</td> <td style="text-align: center;">z</td> </tr> </tbody> </table>	Geometrieachse	Drehachse	1. Geometrieachse	x''	2. Geometrieachse	y'	3. Geometrieachse	z
Geometrieachse	Drehachse								
1. Geometrieachse	x''								
2. Geometrieachse	y'								
3. Geometrieachse	z								
<Winkel>:	Winkelangabe in Grad								

**Programmierung einer Framekomponente**

Bei der Programmierung einer Framekomponente wird nur die programmierte Komponente des Frames geschrieben. Die nicht programmierten Komponenten bleiben unverändert.

**Syntax**

<Frame> [<Index>, <GAx>, RT] = <Winkel>

**Bedeutung**

<Frame>:	beliebiger aktiver oder Datenhaltungsframe
<Index>:	Feldindex des Frames, z.B. \$P_UIFR[0 ... n]
<GAx>:	Bezeichner der Geometrieachse um die um den angegebenen Winkel gedreht werden soll.
RT:	Schlüsselwort für Drehung "RoTation"
<Winkel>:	Winkelangabe in Grad

## Zurücklesen der Drehkomponenten

Im Allgemeinen erhält man beim Zurücklesen der Drehkomponenten eines Frames die gleichen Werte wie die, welche programmiert wurden:

Programmiert	Gespeicherte Drehkomponenten <sup>1)</sup>		
	x, RT	y, RT	z, RT
<Frame>=CROT(X, 45, Y, 30, Z, -20)	45	30	-20
1) die Werte der gespeicherten Drehkomponenten erhält man beim Zurücklesen			

## Werte außerhalb des Wertebereichs

Programmierte Werte außerhalb eines Wertebereichs werden auf die Bereichsgrenzen abgebildet:

Programmiert	Gespeicherte Drehkomponenten <sup>1)</sup>		
	x, RT	y, RT	z, RT
<Frame>=CROT(X, 190, Y, 0, Z, -200)	-170	0	160
1) die Werte der gespeicherten Drehkomponenten erhält man beim Zurücklesen			

## Hinweis

Es wird empfohlen, beim Schreiben der Drehkomponenten des Frames die angegebenen Wertebereiche einzuhalten, um beim Rücklesen der Drehkomponenten wieder die gleichen Werte zu erhalten.

## Kardanische Blockade (engl. Gimbal-Lock)

Kardanische Blockade (engl. Gimbal-Lock) bezeichnet ein geometrisches Problem, bei dem die Drehkomponenten nicht mehr eindeutig aus dem Ortsvektor zurückgerechnet werden können. Die Kardanische Blockade tritt bei RPY-Winkeln bei einer Winkelstellung der Drehkomponente y von 90° auf. In diesem Fall werden von der Steuerung die Drehkomponenten so umgerechnet, dass gilt:

- Drehkomponente x = 0°
- Drehkomponente y = 90°
- Drehkomponente z = resultierender Drehwinkel aus:  $Z_{\text{programmiert}} - X_{\text{programmiert}}$

Programmiert	Gespeicherte Drehkomponenten		
	x, RT	y, RT	z, RT
<Frame>=CROT(X, 30, Y, 90, Z, 40)	0	90	40 - 30 = 10

**Unterschiede beim Schreiben von Gesamtframe und Framekomponenten**

Beim Schreiben der Drehkomponenten eines Frames sind zwei Fälle zu unterscheiden:

1. Schreiben des Gesamtframes: `<Frame>=CROT(X, a, Y, b, Z, c)`

Beim Schreiben des Gesamtframes, erfolgt die Umrechnung sofort zum Zeitpunkt des Schreibens.

2. Schreiben einzelner Drehkomponenten z.B. Drehung um X: `<Frame>[0, X, RT]=a`

Beim Schreiben einzelner Drehkomponenten, erfolgt die Umrechnung abhängig vom Speicherort des Frames:

- Datenhaltungsframes

Bei Datenhaltungsframes erfolgt die Umrechnung zum Zeitpunkt des Aktivierens des Frames aufgrund der bis dahin geschriebenen Drehkomponenten. Bezüglich der Umrechnung der Drehkomponenten verhält sich somit ein Datenhaltungsframe nach dem Schreiben einzelner Drehkomponenten wie beim Schreiben des Gesamtframes.

- Aktive Frames

Bei aktiven Frames erfolgt die Umrechnung sofort zum Zeitpunkt des Schreibens der Drehkomponente.

**Beispiele: Schreiben des Gesamtframes**


Programmiert	Gespeicherte Drehkomponenten		
	x, RT	y, RT	z, RT
<code>&lt;Frame&gt;=CROT(X, 0, Y, 90, Z, 90)</code>	0	90	90
<code>&lt;Frame&gt;=CROT(X, 90, Y, 90)</code>	0	90	-90 <sup>1)</sup>
<code>&lt;Frame&gt;=CROT(X, 90, Y, 90, Z, 90)</code>	0	90	0 <sup>1)</sup>
1) unterschiedliche Werte gegenüber dem Schreiben einzelner Drehkomponenten eines aktiven Frames			

**Beispiele: Schreiben einzelner Drehkomponenten eines Datenhaltungsframes**

Programmiert	Gespeicherte Drehkomponenten		
	x, RT	y, RT	z, RT
<code>&lt;Datenhaltungsframe&gt; [0, X, RT]=0</code> <code>&lt;Datenhaltungsframe&gt; [0, Y, RT]=90</code> <code>&lt;Datenhaltungsframe&gt; [0, Z, RT]=90</code>	0	90	90
<code>&lt;Datenhaltungsframe&gt; [0, X, RT]=90</code> <code>&lt;Datenhaltungsframe&gt; [0, Y, RT]=90</code> <code>&lt;Datenhaltungsframe&gt; [0, Z, RT]=0</code>	0	90	-90 <sup>1)</sup>
<code>&lt;Datenhaltungsframe&gt; [0, X, RT]=90</code> <code>&lt;Datenhaltungsframe&gt; [0, Y, RT]=90</code> <code>&lt;Datenhaltungsframe&gt; [0, Z, RT]=90</code>	0	90	0 <sup>1)</sup>
1) unterschiedliche Werte gegenüber dem Schreiben einzelner Drehkomponenten eines aktiven Frames			

**Beispiele: Schreiben einzelner Drehkomponenten eines aktiven Frames**

Programmiert	Gespeicherte Drehkomponenten		
	x, RT	y, RT	z, RT
<pre>&lt;aktiver Frame&gt; [0, X, RT] = 0 &lt;aktiver Frame&gt; [0, Y, RT] = 90 &lt;aktiver Frame&gt; [0, Z, RT] = 90</pre>	0	90	90
<pre>&lt;aktiver Frame&gt; [0, X, RT] = 90 &lt;aktiver Frame&gt; [0, Y, RT] = 90 &lt;aktiver Frame&gt; [0, Z, RT] = 0</pre>	0	90	0 <sup>1)</sup>
<pre>&lt;aktiver Frame&gt; [0, X, RT] = 90 &lt;aktiver Frame&gt; [0, Y, RT] = 90 &lt;aktiver Frame&gt; [0, Z, RT] = 90</pre>	0	90	90 <sup>1)</sup>
1) unterschiedliche Werte gegenüber dem Schreiben des <b>Gesamtframes</b> bzw. dem Schreiben <b>einzelner Drehkomponenten</b> eines <b>Datenhaltungsframes</b>			

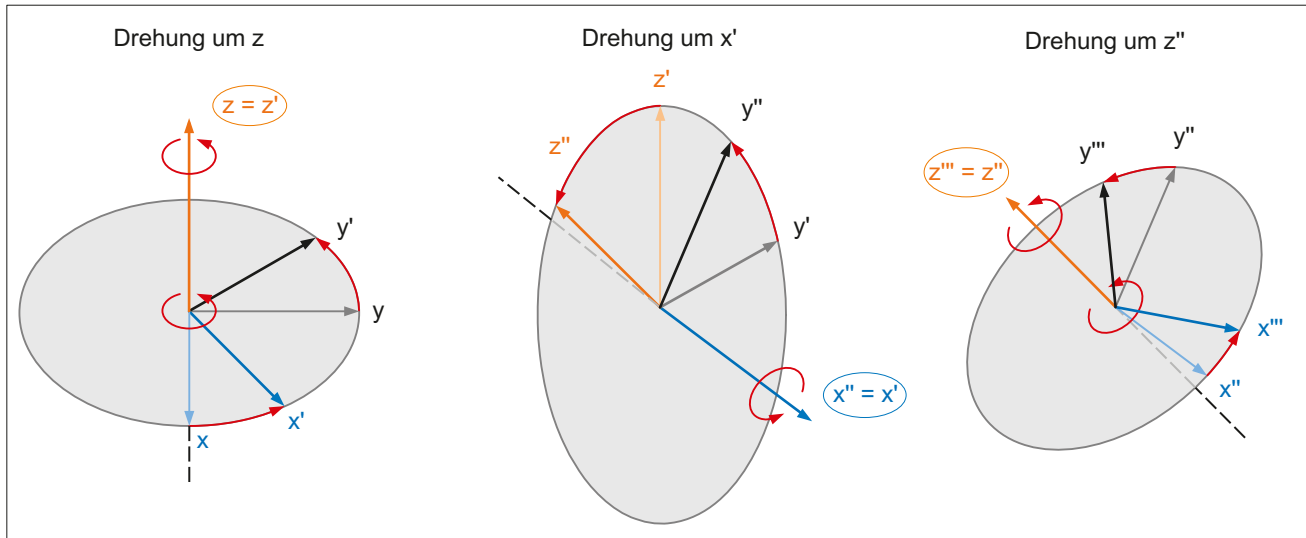
 VORSICHT
Aufgrund des unterschiedlichen Umrechnungszeitpunktes können nach dem Schreiben des <b>Gesamtframes</b> bzw. dem Schreiben <b>einzelner Drehkomponenten</b> eines <b>Datenhaltungsframes</b> und dem Schreiben <b>einzelner Drehkomponenten</b> eines <b>aktiven Frames</b> für die Drehkomponente z unterschiedliche Werte zurückgelesen werden.

**9.5.2.5 Drehung mit Euler-Winkeln: ZX'Z''-Konvention**

Mit Euler-Winkeln erfolgen die Drehungen in der Reihenfolge z, x', z''.

**Hinweis****Verwendungsempfehlung**

Aus historischen Gründen ist die Möglichkeit der Verwendung von Euler-Winkeln in zx'z''-Konvention vorhanden. Es wird aber dringend empfohlen ausschließlich Euler-Winkel in zy'x''-Konvention (RPY-Winkel) (siehe Kapitel "Drehung mit Euler-Winkeln: ZY'X''-Konvention (RPY-Winkel) (Seite 768)") zu verwenden.



**Zuordnung Drehachse zu Geometrieachse**

Drehachse	Geometrieachse im Kanal
$z''$	3. Geometrieachse
$x'$	1. Geometrieachse
$z$	3. Geometrieachse

**Wertebereich**

Angaben von Euler-Winkeln können nur innerhalb folgender Wertebereiche eindeutig rückgerechnet werden:

$$\begin{aligned}
 0 &\leq x < 180 \\
 -180 &\leq y \leq 180 \\
 -180 &\leq z \leq 180
 \end{aligned}$$

Angaben außerhalb der angegebenen Wertebereiche werden modulo der Bereichsgrenzen gerechnet.

**Hinweis**

Es wird empfohlen, beim Schreiben der Drehkomponenten des Frames die angegebenen Wertebereiche einzuhalten, um beim Zurücklesen der Drehkomponenten wieder die gleichen Werte zu erhalten.

### 9.5.2.6 Drehung in beliebiger Ebene

#### CRPL - Constant Rotation Plane

Mit der vordefinierten Funktion "**Constant Rotation Plane**" kann für einen Frame eine Drehung in einer beliebigen Ebene ( $G_{17}$ ,  $G_{18}$ ,  $G_{19}$ ), ohne Angabe des Bezeichners einer Geometrieachse, programmiert werden. Somit können auch Drehungen in der dritten Ebene programmiert werden, wenn aufgrund der spezifischen Maschinenkonstellation nur zwei Geometrieachsen im Kanal vorhanden sind.

#### Syntax

CRPL(<Drehachse>, <Drehwinkel>)

#### Bedeutung

CRPL:	Drehung in beliebiger Ebene		
<Drehachse>:	Achse um die gedreht wird		
	Typ:	INT	
	<b>Wert</b>	<b>Bedeutung</b>	
	0	Drehung in der aktiven Ebene	
	1	Drehung um Z	
	2	Drehung um Y	
	3	Drehung um X	
<Drehwinkel>:	Winkel in Grad um den gedreht wird		
	Typ:	REAL	
	Es wird dringend empfohlen die angegebenen Winkelbereiche einzuhalten. Werden die Grenzen nicht eingehalten, ist keine eindeutige Rückrechnung möglich. Winkelangaben außerhalb der Grenzen werden nicht abgelehnt.		
	RPY-Winkel:	X	$-180 \leq \langle \text{Drehwinkel} \rangle \leq 180$
		Y	$-90 \leq \langle \text{Drehwinkel} \rangle \leq 90$
		Z	$-180 \leq \langle \text{Drehwinkel} \rangle \leq 180$
	ZX'Z"-Konvention:	X	$-180 \leq \langle \text{Drehwinkel} \rangle \leq 180$
		Y	$0 \leq \langle \text{Drehwinkel} \rangle \leq 180$
		Z	$-180 \leq \langle \text{Drehwinkel} \rangle \leq 180$

### Verkettung mit Frames

CRPL() kann mit Frames und Frame-Funktionen wie CTRANS(), CROT(), CMIRROR(), CSCALE(), CFINE() etc. verkettet werden.

Beispiele:

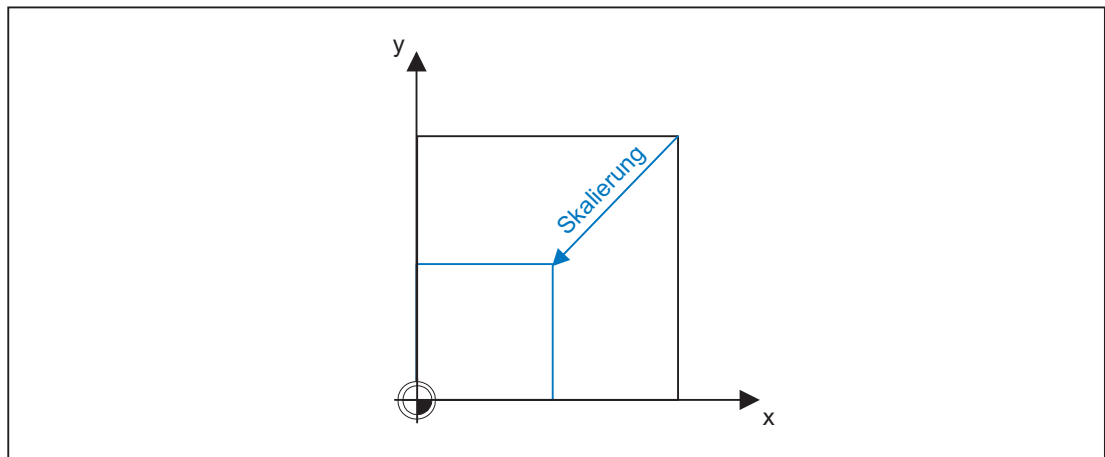
```
$P_PFRAME = $P_PFRAME : CRPL(0,30.0)
```

```
$P_PFRAME = CTRANS(X,10) : CRPL(1,30.0)
```

```
$P_PFRAME = CROT(X,10) : CRPL(2,30.0)
```

```
$P_PFRAME = CRPL(3,30.0) : CMIRROR(Y)
```

### 9.5.2.7 Skalierung



### Programmierung

Die Programmierung einer Skalierung erfolgt über folgende Programmbefehle:

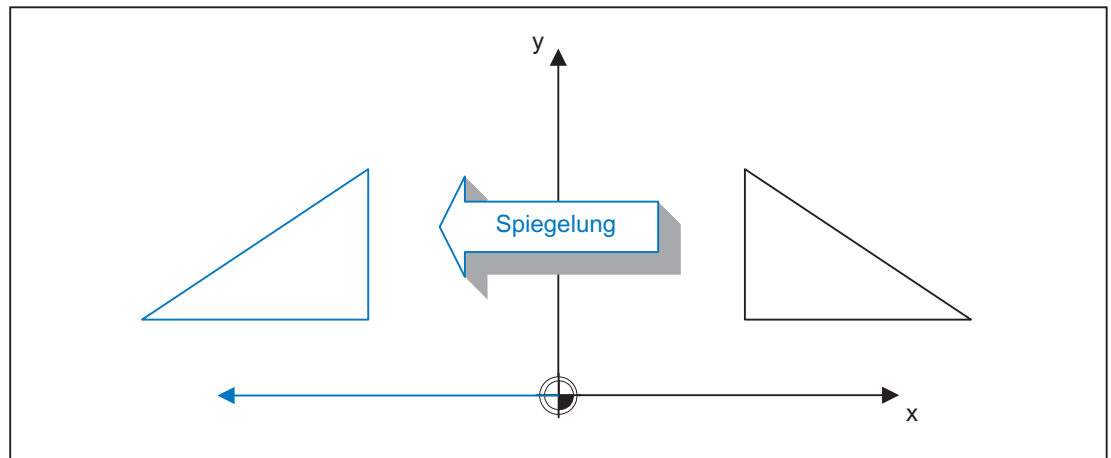
```
$P_UIFR[1] = CSCALE(x,1,y,1)
```

```
SCALE x = 1y = 1
```

```
$P_UIFR[1,x,sc] = 1
```



### 9.5.2.8 Spiegelung



#### Programmierung

Die Programmierung einer Spiegelung erfolgt über folgende Programmbefehle:

```
$P_UIFR[1] = CMIRROR(x,1,y,1)
```

```
MIRROR x = 1 y = 1
```

```
$P_UIFR[1,x,mi] = 1
```

### 9.5.2.9 Verkettungsoperator

Framekomponenten oder gesamte Frames lassen sich über den Verkettungsoperator ( : ) zu einem Gesamtframe zusammenfassen.

### 9.5.2.10 Programmierbare Achsbezeichner

In den Frame-Befehlen können Geo-, Kanal- und Maschinen-Achsbezeichner verwendet werden. Die programmierte Achse muss bei den kanalspezifischen Frames im Kanal bekannt sein.

#### SPI

Bei der Programmierung von Frame-Anweisungen kann an Stelle eines Achsbezeichners auch die Achsfunktion `SPI(<spindelnummer>)` verwendet werden.

`SPI(<spindelnummer>)` stellt dabei den Bezug der Spindel zu der Kanalachse.

→ siehe MD35000 `$MA_SPIND_ASSIGN_TO_MACHAX[ ]` (Zuordnung Spindel zur Maschinenachse)

Folgende Frame-Anweisungen können mit `SPI(spino)` programmiert werden:

`CTRANS()`

`CFINE()`

`CMIRROR()`

`CSCALE()`

Eine Spindel kann immer nur einer Rundachse zugewiesen werden. Deshalb kann die Funktion `CROT(..)` nicht mit `SPI()` programmiert werden, da für `CROT()` nur Geometrie-Achsen erlaubt sind.

Bei der Rückübersetzung von Frames wird immer der Kanalachsbezeichner bzw. der Maschinenachsbezeichner der zur Spindel gehörenden Achse ausgegeben, auch wenn im Teileprogramm Achsbezeichner mit `SPI(..)` programmiert worden sind.

Ist die Spindel z. B. der Kanalachse "A" zugeordnet, dann wird aus der Programmierung:

```
N10 $P_UIFR[1] = CTRANS(SPI(1),33.33,X,1):CSCALE(SPI(1),33.33):CMIRROR(SPI(1))
```

bei der Rückübersetzung:

```
$P_UIFR[1]=CTRANS(X,1,A,33.33):CSCALE(A,33.33):CMIRROR(A)
```

Wird in einer Frame-Anweisung eine Spindel und die zugeordnete Achse programmiert, wird der Alarm 16420 "Achse % mehrfach programmiert" ausgegeben.

**Beispiel:**

```
$P_UIFR[1] = CTRANS(SPI(1),33.33,X,1,A,44)
```

(Die Spindel ist der Achse A zugeordnet.)

**Programmierbeispiele**

```
$P_PFRAME[SPI(1),TR]=22.22
```

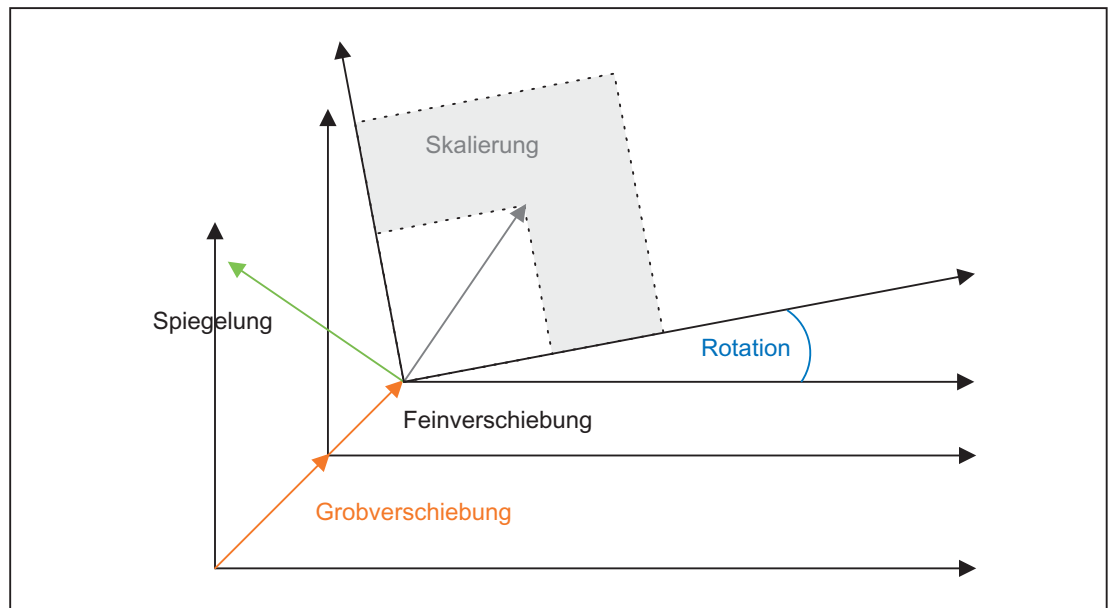
```
$P_PFRAME=CTRANS(X,Achswert,Y,Achswert,SPI(1),Achswert)
```

```
$P_PFRAME=CSCALE(X,Maßstab,Y,Maßstab,SPI(2),Maßstab)
```

```
$P_PFRAME=CMIRROR(S1,Y,Z)
```

```
$P_UBFR=CTRANS(A,10):CFINE(SPI(1),0.1)
```

### 9.5.2.11 Koordinatentransformation



Die Koordinatentransformation für Geometrieachsen ergibt sich anhand folgender Formeln:

WKS → BKS	$\vec{v} = R * \underline{S} * \underline{M} * \vec{v}' + \vec{t}$
BKS → WKS	$\vec{v}' = \text{inv}(\underline{M}) * \text{inv}(\underline{S}) * \text{inv}(\underline{R}) + (\vec{v} - \vec{t})$

V: Positionsvektor im BKS

V': Positionsvektor im WKS

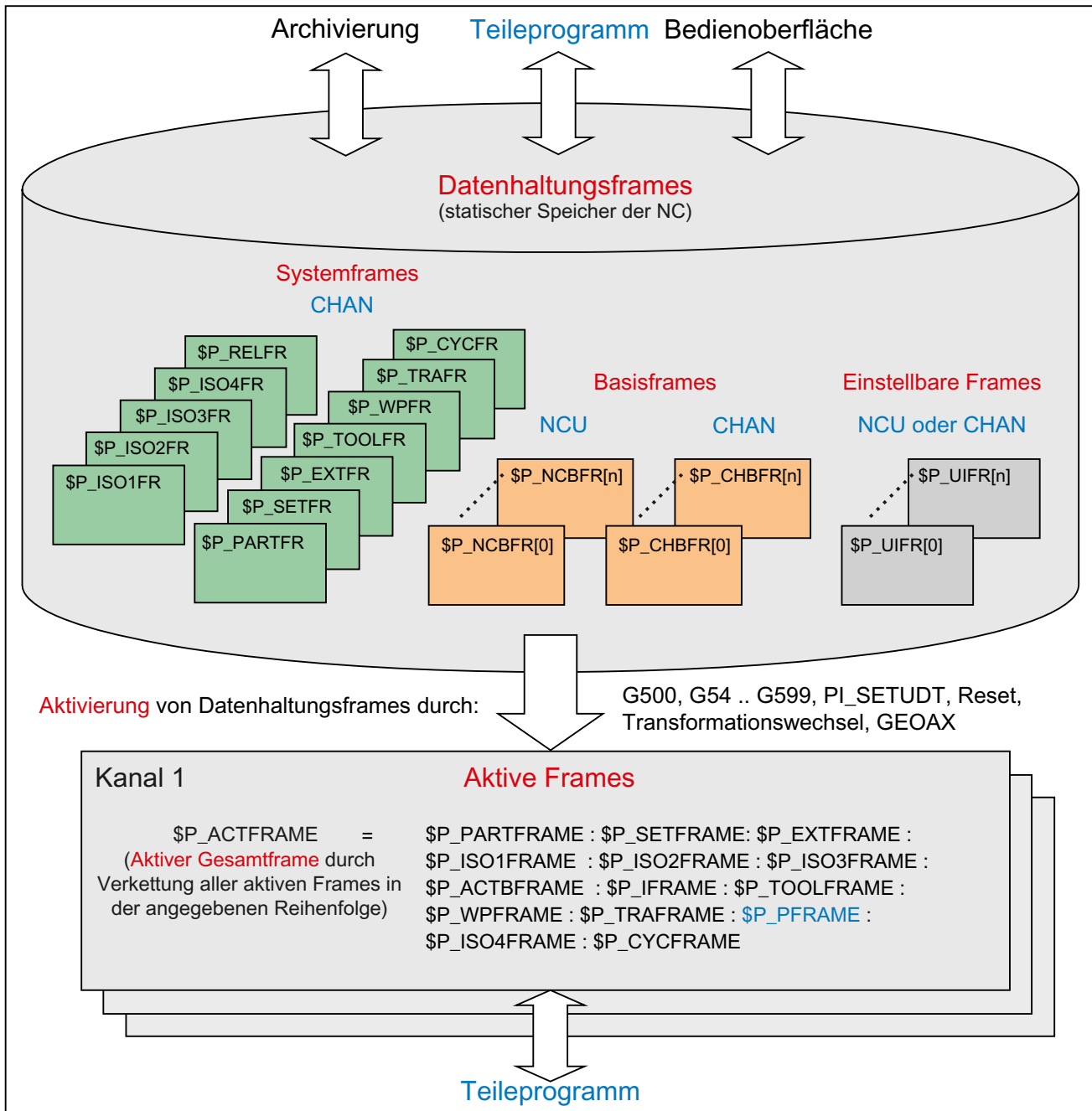
## 9.5.3 Frames in der Datenhaltung und aktive Frames

### 9.5.3.1 Übersicht

Es gibt folgende Frame-Typen:

- Systemframes (siehe Bild)
- Basisframes ( $\$P\_NCBFR[n]$ ,  $\$P\_CHBFR[n]$ )
- Einstellbare Frames ( $\$P\_UIFR[n]$ )
- Programmierbares Frame ( $\$P\_PFRAME[n]$ )

Bei allen Frames, außer dem programmierbaren Frame, existiert ein Frame in der Datenhaltung (Datenhaltungsframe) und ein aktiver Frame. Beim programmierbaren Frame existiert nur ein aktiver Frame.



### Schreiben von Frames

Aus dem Teileprogramm heraus können Datenhaltungsframes und aktive Frames geschrieben werden. Über die Bedienoberfläche können nur Datenhaltungsframes geschrieben werden.

### Archivieren von Frames

Es können nur Datenhaltungsframes archiviert werden.

### 9.5.3.2 Aktivierung von Datenhaltungsframes

Datenhaltungsframes werden zu aktiven Frames durch folgende Aktionen:

- Teileprogrammbefehle zur Aktivierung/Deaktivierung von Verschiebungen: G54...G599, G500
- RESET und MD20110 \$MC\_RESET\_MODE\_MASK, Bit14 = 1
- Transformationswechsel
- Änderung der Geometrieachszuordnung GEOAX
- Über SINUMERIK Operate mit PI-Dienst "\_N\_SETUDT"

#### Aktivierung über SINUMERIK Operate

Die Aktivierung eines Datenhaltungsframes mit PI-Dienst "\_N\_SETUDT" wirkt im Kanal erst nach einem Fortsetzstart für das angewählte Teileprogramm. Die Aktivierung wirkt im Reset-Zustand sofort, wenn folgendes Maschinendatum gesetzt ist:

MD9440 \$MM\_ACTIVATE\_SEL\_USER\_DATA (Aktive Verschiebung sofort wirksam setzen)

#### Aktivierung von Systemframes

Die Aktivierung von Systemframes erfolgt durch:

- Programmierung der entsprechenden Systemfunktion im Teileprogramm
- Bedienhandlung an SINUMERIK Operate

---

#### Hinweis

##### Modifizieren von Systemframes der Datenhaltung

Systemframes der Datenhaltung können zwar prinzipiell durch den Zyklen-Programmierer modifiziert und über eine G500, G54...G599-Anweisung aktiviert werden. Diese Möglichkeit sollte aber nur unter Vorbehalt verwendet werden.

---

### Aktivierung von Datenhaltungsframes

Das Verhalten bei Aktivierung von Datenhaltungsframes kann über folgendes Maschinendatum eingestellt werden:

MD24050 \$MC\_FRAME\_SAA\_MODE (Speichern und aktivieren von Datenhaltungsframes)

Bit	Wert	Bedeutung
0	0	Datenhaltungsframes werden <b>nur</b> durch Programmierung der Bitmasken \$P_CHBFRMASK, \$P_NCBFRMASK und \$P_CHSFRMASK aktiv. G500...G599 aktiviert nur das entsprechende einstellbare Frame. Das Reset-Verhalten ist unabhängig davon.
	1	Datenhaltungsframes werden durch Systemfunktionen wie TOROT, PAROT, externe Nullpunktverschiebung, Transformationen nicht implizit beschrieben.

#### Systemvariable \$P\_CHSFRMASK

Die Systemframes der Datenhaltung können über die Systemvariable **\$P\_CHSFRMASK** aktiviert werden. Der Wert der Variablen wird bitcodiert angegeben, entsprechend dem Maschinendatum:

MD28082 \$MC\_MM\_SYSTEM\_FRAME\_MASK (Systemframes der Datenhaltung)

Durch Setzen eines Bits der Systemvariable **\$P\_CHSFRMASK** auf den Wert 1, wird der entsprechende Systemframe der Datenhaltung im Kanal aktiv. Bei einem Wert 0 bleibt der im Kanal aktuell aktive Systemframe weiterhin aktiv.

#### Aktivierung von Systemframes nach RESET

Nach RESET werden im Kanal die Systemframes aktiviert, deren Bits im folgenden Maschinendatum gesetzt sind:

MD24006 \$MC\_CHSFRAME\_RESET\_MASK (Aktive Systemframes nach Reset)

#### Aktivierung der Systemframes für TCARR, PAROT und TOROT, TOFRAME

Die Systemframes für TCARR, PAROT und TOROT, TOFRAME werden entsprechend der Einstellung im folgenden Maschinendatum aktiviert:

MD20150 \$MC\_GCODE\_RESET\_VALUES (Löschstellung der G-Gruppen)

Beim Umschalten von Geometrieachsen durch Transformations-An/Abwahl oder Befehl GEOAX, wird das aktuelle Gesamtframe \$P\_ACTFRAME entweder gelöscht oder anhand der neuen Geometrieachs-Konstellation neu berechnet und aktiviert. Die Systemframes werden, ebenso wie alle anderen Frames, bezüglich der Geometrieachsen neu aufbereitet.

### 9.5.3.3 NCU-Globale Frames

Alle einstellbaren Frames  $G_{54} \dots G_{599}$  und alle Basisframes können NCU-global oder kanalspezifisch projiziert werden. Bei den Basisframes ist auch die Kombination möglich. Die globalen Frames wirken auf alle Kanäle einer NCU. Alle Kanäle können darauf lesend und schreibend zugreifen. Globale Frames haben nur axiale Frameanteile, also Translationen, Skalierungen und Spiegelungen einzelner Achsen. Jeder Kanal kann für beliebige Maschinenachsen globale Frames lesen oder modifizieren.

Globale Frames zeichnen sich dadurch aus, dass sie in allen Kanälen einer NCU eingerechnet werden. Da die Zuordnung von Maschinenachsen zu Kanalachsen und speziell zu den Geometrieachsen in allen Kanälen unterschiedlich sein kann, existiert kein geometrischer Zusammenhang. Globale Frames beschreiben Verschiebungen, Skalierungen und Spiegelungen von Maschinenachsen. Auf globale Frames lassen sich also keine Drehungen anwenden.

Alle einstellbaren Frames können mit dem folgenden Maschinendatum zu globalen Frames umprojiziert werden:

MD18601 \$MN\_MM\_NUM\_GLOBAL\_USER\_FRAMES (Anzahl der globalen, vordefinierten Anwenderframes(SRAM))

Hat dieses Maschinendatum einen Wert größer als Null, so gibt es keine kanalspezifischen einstellbaren Frames.

Das folgende Maschinendatum ist dann irrelevant und wird nicht ausgewertet:

MD28080 \$MC\_MM\_NUM\_USER\_FRAMES (Anzahl der einstellbaren Frames (SRAM))

Die Anzahl von globalen Basisframes wird über das folgende Maschinendatum parametrier:

MD18602 \$MN\_MM\_NUM\_GLOBAL\_BASE\_FRAMES (Anzahl der globalen Basisframes(SRAM))

Gleichzeitig können über das folgende Maschinendatum kanalspezifische Basisframes vorhanden sein:

MD28081 \$MC\_MM\_NUM\_BASE\_FRAMES (Anzahl Basisframes (SRAM))

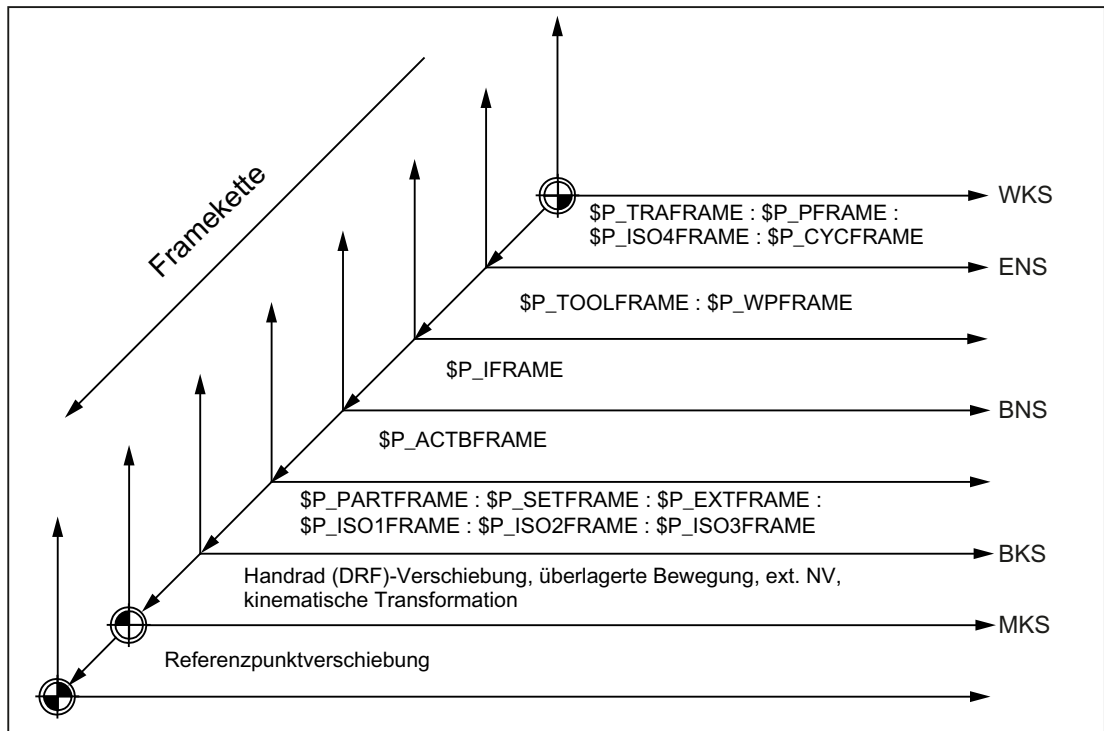
Die globalen Frames können von allen Kanälen einer NCU geschrieben und gelesen werden. Beim Schreiben von globalen Frames ist vom Anwender für eine Kanalkoordinierung Sorge zu tragen. Dies kann z. B. durch Wait-Marken realisiert werden.

## 9.5.4 Framekette und Koordinatensysteme

### 9.5.4.1 Übersicht

Im folgenden Bild ist die Framekette für das aktuelle Gesamtframe abgebildet. Die Framekette liegt zwischen dem Basiskoordinatensystem (BKS) und dem Werkstückkoordinatensystem (WKS).

Das Einstellbare Nullpunkt-System (ENS) entspricht dem durch das Programmierbare Frame transformierte WKS. Das Basis-Nullpunkt-System (BNS) beinhaltet noch das aktuelle Einstellbare Frame. Das Systemframe für die externe Nullpunktverschiebung ist nur vorhanden, wenn es projiziert wurde, ansonsten wird die externe Nullpunktverschiebung als überlagerte Bewegung der Achse heraus gefahren.



WKS: **W**erkstück **K**oordinaten **S**ystem

ENS: **E**instellbares **N**ullpunkt **S**ystem

BNS: **B**asis **N**ullpunkt **S**ystem

BKS: **B**asis **K**oordinaten **S**ystem

MKS: **M**aschinen **K**oordinaten **S**ystem

Das aktuelle Gesamtframe  $\$P\_ACTFRAME$  ergibt sich aus der Verkettung aller aktiven Frames der Framekette:

$$\begin{aligned} \$P\_ACTFRAME = & \$P\_PARTFRAME : \$P\_SETFRAME : \$P\_EXTFRAME : \\ & \$P\_ISO1FRAME : \$P\_ISO2FRAME : \$P\_ISO3FRAME : \\ & \$P\_ACTBFRAME : \$P\_IFRAME : \$P\_TOOLFRAME : \\ & \$P\_WPFRAME : \$P\_TRAFRAME : \$P\_PFRAME : \\ & \$P\_ISO4FRAME : \$P\_CYCFRAME \end{aligned}$$



### 9.5.4.2 Relative Koordinatensysteme

Relative Koordinatensysteme dienen nur zur Anzeige von aktuellen Achs-Sollpositionen, die relativ zu einem vorgegebenen Referenzpunkt im aktiven angezeigten Koordinatensystem liegen. Bezüglich den relativen Koordinatensystemen kann nicht programmiert werden. Es können nur Achspositionen in diesen Systemen über Systemvariablen gelesen werden.

Die neuen Anzeige-Koordinatensysteme liegen relativ zum WKS- und ENS-Koordinatensystem und ergeben sich durch Transformation der WKS- oder ENS-Achspositionen mit dem aktiven Systemframe  $\$P\_RELFRAME$ . Die relativen Koordinatensysteme können nicht nur linear verschoben sein, sondern auch noch gedreht, gespiegelt, gestaucht bzw. gedehnt werden.

Die Positionsanzeige für Achssollwerte erfolgt im WKS oder im ENS. Die Projektierung erfolgt über HMI-Maschinendaten. Es ist immer nur ein Anzeige-Koordinatensystem im Kanal aktiv. Deshalb wird auch nur ein relatives Frame zur Verfügung gestellt, welches die beiden relativen Koordinatensysteme im gleichen Verhältnis erzeugt. Die HMI zeigt die relativen Koordinaten entsprechend der Projektierung an.

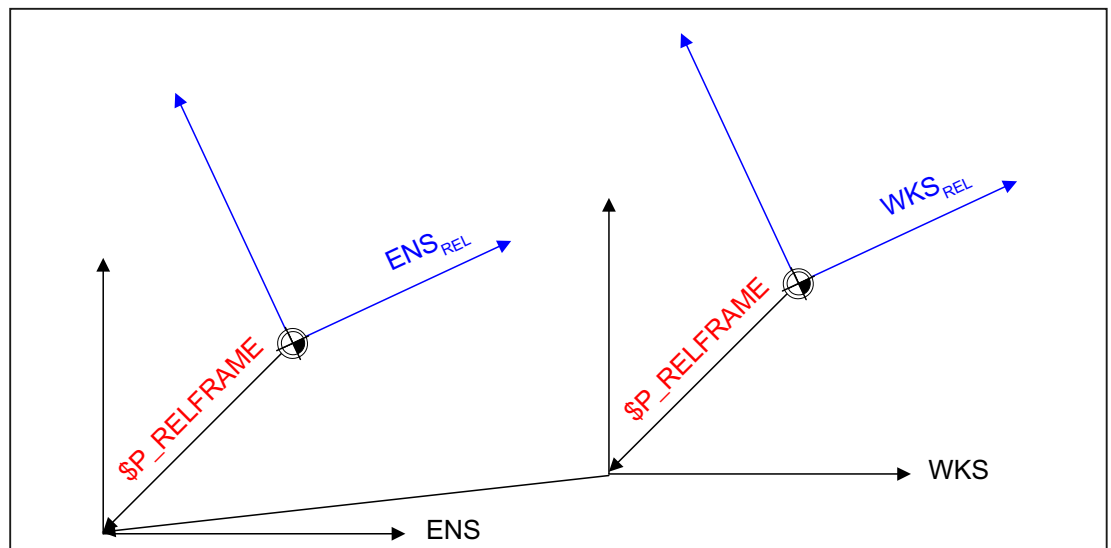


Bild 9-22 Relative Koordinatensysteme

Das Datenhaltungsframe  $\$P\_RELFR$  kann im Teileprogramm und über BTSS beschrieben werden. Es können alle Framekomponenten verändert werden.

Das aktive Systemframe  $\$P\_RELFRAME$  kann im Teileprogramm beschrieben und über BTSS gelesen werden.

Die Projektierung des Systemframes \$P\_RELFR erfolgt über folgende Maschinendaten:

Maschinendatum	Bit	Bedeutung
MD28082 \$MC_MM_SYSTEM_FRAME_MASK	11	Anlegen von \$P_RELFR; damit werden relative Koordinatensysteme existent.
MD28083 \$MC_MM_SYSTEM_DATAFRAME_MASK	11	Datenhaltungsframe \$P_RELFR
MD24006 \$MC_CHSFRAME_RESET_MASK.	11	\$P_RELFR wird bei Reset aktiv
MD24007 \$MC_CHSFRAME_RESET_CLEAR_MASK	11	\$P_RELFR wird bei Reset gelöscht
MD24008 \$MC_CHSFRAME_POWERON_MASK	11	\$P_RELFR wird bei PowerOn gelöscht

Die Achspositionen im relativen Koordinatensystem  $WKS_{Rel}$  können über die Variable \$AA\_PCS\_REL[ax] gelesen werden. Die Variable ist im Teileprogramm, BTSS und über Synchronaktionen lesbar.

Die Achspositionen im relativen Koordinatensystem  $ENS_{Rel}$  können über die Variable \$AA\_ACS\_REL[ax] gelesen werden. Die Variable ist im Teileprogramm, BTSS und über Synchronaktionen lesbar.

Das Setzen eines relativen Bezugspunktes über die Bedienoberfläche erfolgt über die allgemeine Kommandoschnittstelle für die Werkstück- und Werkzeugvermessung. Das Systemframe \$P\_RELFR für relative Koordinatensysteme wird folgendermaßen berechnet und aktiviert:

- \$AC\_MEAS\_TYPE = 14
- PI-Dienste \_N\_SETUDT(6, 7)

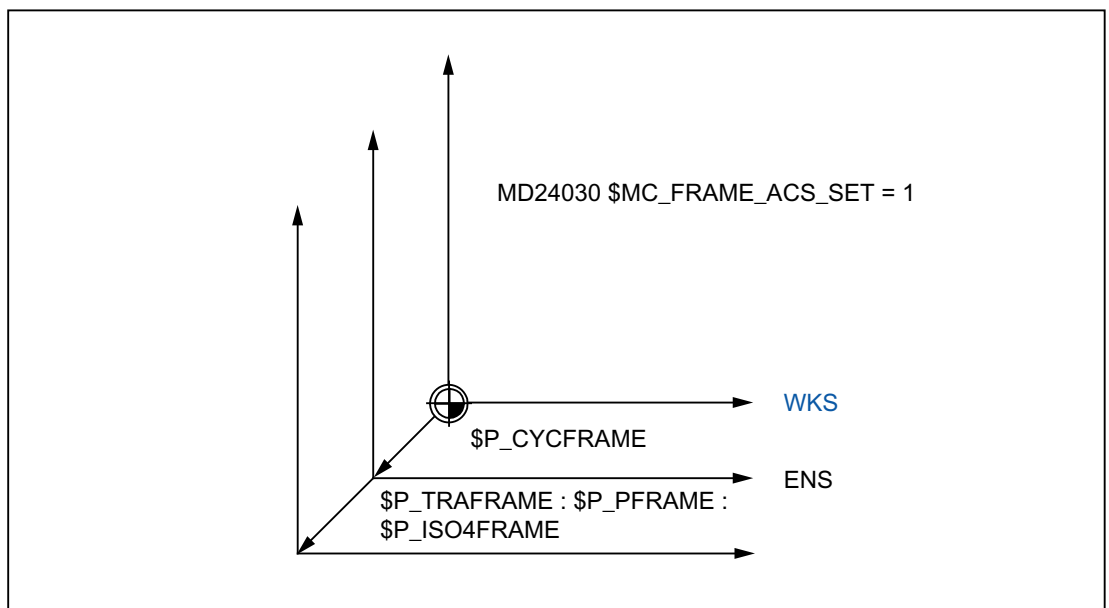
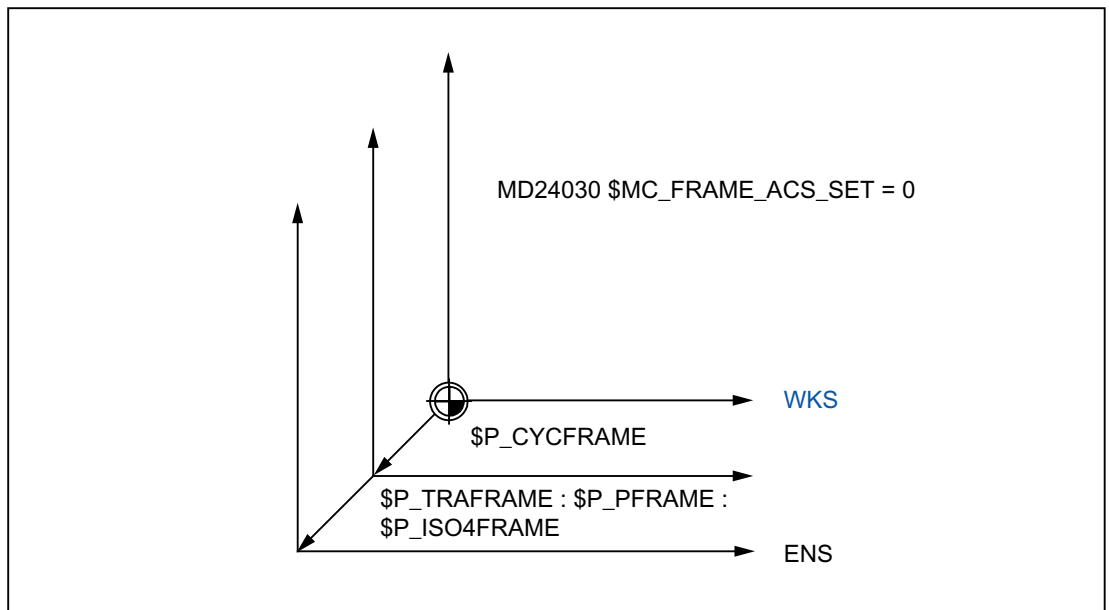
Ein Beispiel zum Setzen von relativen Achspositionen findet sich in:

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Messen (M5),  
 Kapitel "Messen von Geo- und Zusatzachsen (Mess Type 14, 15)"

### 9.5.4.3 Projektierbares ENS

Das ENS-Koordinatensystem dient zur Istwertanzeige und für Verfahrbewegungen während eines unterbrochenen Zyklus. Zyklen nutzen für die Realisierung ihrer Funktionalität Frames in der Framekette. Sie tragen Translationen oder Drehungen entweder in das programmierbare Frame oder in das Zyklen-Systemframe ein. Dadurch verändern die Zyklen das WKS. Ein Anwender, der einen Zyklus durch Stopp unterbricht, möchte jedoch nicht in dem "Zyklen-Koordinatensystem" verfahren, sondern in dem programmierten WKS. Deshalb wird für die Anzeige das ENS genutzt. Aus Kompatibilitätsgründen wird das ENS projektierbar gemacht.



Mit dem folgenden Maschinendatum kann eingestellt werden, ob das ENS mit oder ohne dem programmierbaren Frame, dem Transformationsframe und \$P\_ISO4FRAME ist:

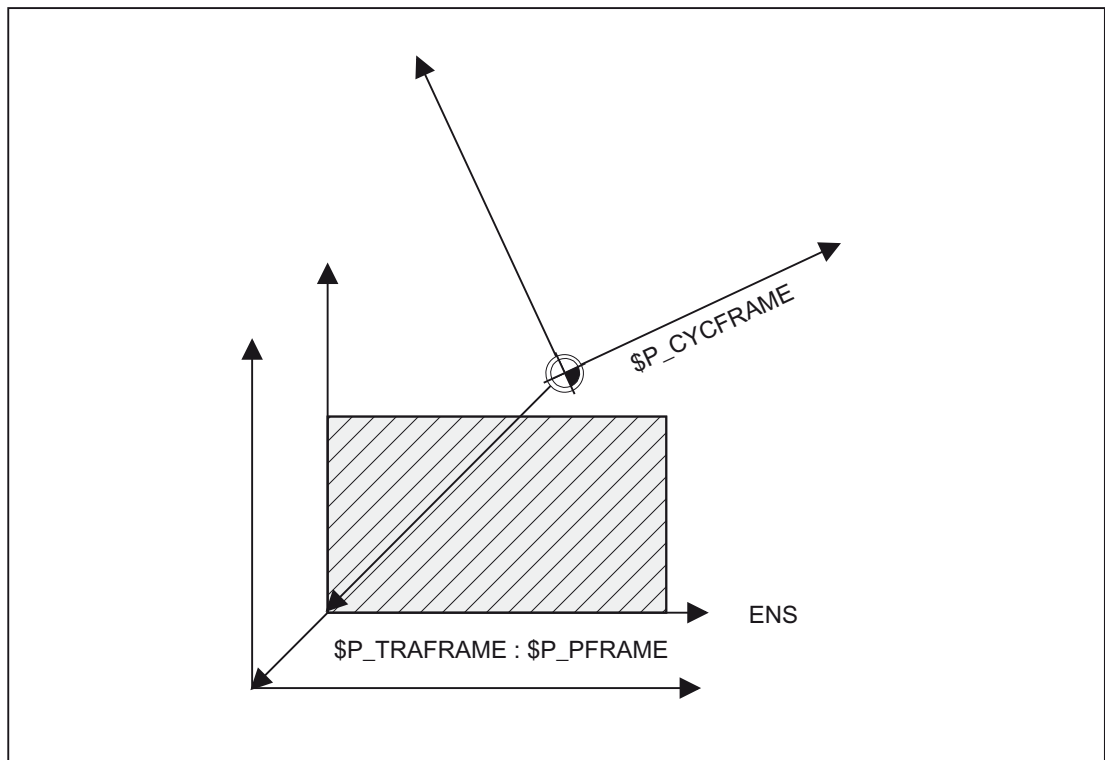
MD24030 \$MC\_FRAME\_ACS\_SET (Einstellung des ENS Koordinatensystems)

Standardmäßig ist der Wert 1 eingestellt.

Die Umprojektierung des ENS hat Rückwirkung auf alle ENS-Istwert-Anzeigen und auf die Systemvariablen \$AA\_IEN[achse]. Das Verfahren von Geometrieachsen in JOG im ENS ist ebenso von der Projektierung abhängig.

#### 9.5.4.4 Handverfahren im ENS-Koordinatensystem

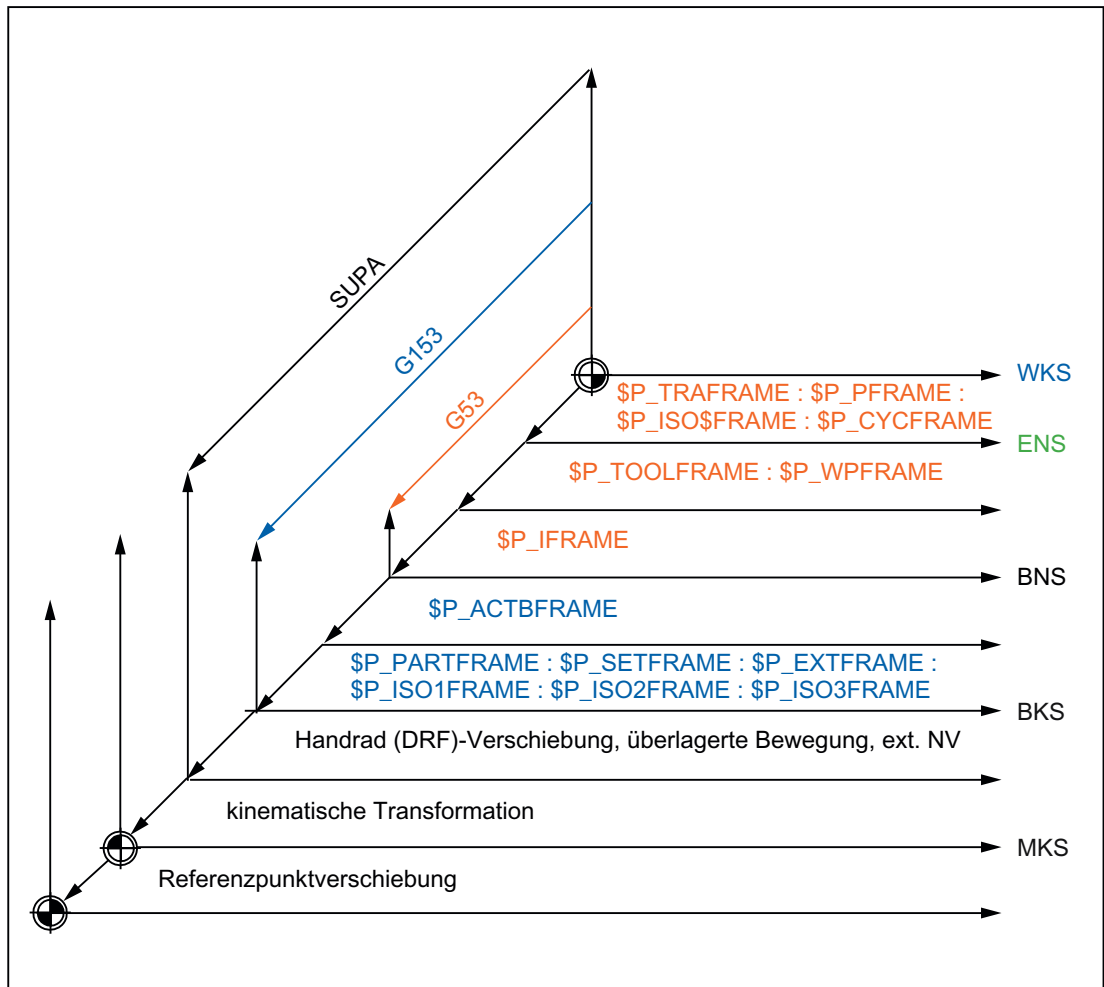
Die Geometrieachsen werden bisher beim Handverfahren in der Betriebsart JOG im WKS verfahren. Zusätzlich dazu gibt es die Möglichkeit, das Handverfahren im ENS-Koordinatensystem durchzuführen. Dazu gibt es die Variable \$AC\_JOG\_COORD, mit der zwischen Handverfahren im WKS und ENS umgeschaltet werden kann. Der Anwender hat jetzt die Möglichkeit auszuwählen, ob er im ENS oder im WKS verfahren möchte.



## 9.5.4.5 Unterdrückung von Frames

## Programmierung

Befehl	Bedeutung
G53	Satzweises Unterdrücken folgender Frames: <ul style="list-style-type: none"> <li>• Systemframe für Zyklen</li> <li>• Programmierbares Frame</li> <li>• Systemframe für Transformationen, Werkstücke , TOROT und TOFRAME</li> <li>• Aktives einstellbares Frame</li> </ul>
G153	Satzweises Unterdrücken folgender Frames: <ul style="list-style-type: none"> <li>• Systemframe für Zyklen</li> <li>• Programmierbares Frame</li> <li>• Systemframe für TOROT und TOFRAME, Werkstücke</li> <li>• Aktives einstellbares Frame</li> <li>• Alle kanalspez. und NCU-globalen Basisframes</li> <li>• Systemframes für PAROT, Istwertsetzen, Ankratzen, ext. NPV</li> </ul>
SUPA	Impliziter Vorlaufstopp und satzweises Unterdrücken von Frames analog G153 und zusätzlich: <ul style="list-style-type: none"> <li>• Handradverschiebungen (DRF)</li> <li>• [ext. Nullpunktverschiebung]</li> <li>• Überlagerte Bewegung</li> </ul>
G500	Modales Aktivieren des G500-Frames. Das G500-Frame sollte ein Nullframe sein.
DRFOF	Ausschalten (Löschen) der Handradverschiebungen (DRF)



**Parametrierung**

Die Frame-Unterdrückungen SUPA, G153 und G53 führen dazu, dass das WKS, ENS und evtl. das BNS bei aktiver Frame-Unterdrückung springt. Über das folgende Maschinendatum kann diese Eigenschaft für die Positionsanzeige und für die vordefinierten Positionsvariablen geändert werden:

MD24020 \$MC\_FRAME\_SUPPRESS\_MODE (Positionen bei Frameunterdrückung)

Bit	Bedeutung
0	Positionen für die Anzeige (BTSS) ist ohne Frame-Unterdrückung.
1	Positionsvariablen sind ohne Frame-Unterdrückung.

Bei gesetztem Bit wird die Position für die Anzeige bzw. für die Variablen ohne Frame-Unterdrückung berechnet, so dass keine Positionssprünge mehr auftreten.

## 9.5.5 Frames der Framekette

### 9.5.5.1 Übersicht

Man unterscheidet bis zu vier Varianten von Frames:

- Einstellbare Frames ( $G_{500}, G_{54}..G_{599}$ )
- Basisframes
- Programmierbares Frame
- Systemframes

### 9.5.5.2 Einstellbare Frames $\$P\_UIFR[n]$

Die Anzahl von NCU-globalen einstellbaren Frames wird über das folgende Maschinendatum eingestellt:

MD18601  $\$MN\_MM\_NUM\_GLOBAL\_USER\_FRAMES$  (Anzahl der globalen, vordefinierten Anwenderframes (SRAM))

Die Anzahl kann zwischen 0 und 100 variieren. Hat das MD einen Wert größer Null, dann gibt es nur NCU-globale einstellbare Frames, ansonsten gibt das folgende Maschinendatum die Anzahl der kanalspezifischen einstellbaren Frames an:

MD28080  $\$MC\_MM\_NUM\_USER\_FRAMES$  (Anzahl der einstellbaren Frames(SRAM))

Über die Systemvariable  $\$P\_UIFR[n]$  können die Frame-Feldelemente gelesen und geschrieben werden. Beim Schreiben eines Feldelementes wird das Frame nicht gleichzeitig aktiviert, sondern die Aktivierung erfolgt erst mit der Ausführung einer  $G_{500}, G_{54}..G_{599}$ -Anweisung. Bei NCU-globalen Frames wird das geänderte Frame in jedem Kanal der NCU, der eine  $G_{500}, G_{54}..G_{599}$ -Anweisung ausführt, aktiv. Die Variable dient vorwiegend als Speicher für Schreibvorgänge von HMI oder PLC. Diese Frame-Variablen werden über die Datensicherung gesichert.

### Aktuelles einstellbares Frame $\$P\_IFRAME$

Über die vordefinierte Frame-Variable  $\$P\_IFRAME$  kann das aktuelle einstellbare Frame, welches im Kanal gültig ist, im Teileprogramm gelesen und geschrieben werden. Das geschriebene einstellbare Frame wird sofort eingerechnet. Bei NCU-globalen einstellbaren Frames wirkt das geänderte Frame nur in dem Kanal, in dem das Frame programmiert wurde. Soll das Frame für alle Kanäle einer NCU geändert werden, so muss gleichzeitig  $\$P\_UIFR[n]$  und  $\$P\_IFRAME$  beschrieben werden. Die anderen Kanäle müssen dann noch das entsprechende Frame mit z. B.  $G_{54}$  aktivieren.

### Programmierung von einstellbaren Frames

Einstellbare Frames können über das Teileprogramm und über BTSS von der Bedienung und von der PLC geschrieben und gelesen werden. Über BTSS können jedoch nur die Datenhaltungsframes geschrieben werden. Der Index des aktiven einstellbaren Frames kann über die Systemvariable  $\$P\_UIFRNUM$  ermittelt werden.

### 9.5.5.3 Kanal Basisframes \$P\_CHBFR[n]

Über das Maschinendatum kann die Anzahl der Basisframes im Kanal projiziert werden:

MD28081 \$MC\_MM\_NUM\_BASE\_FRAMES (Anzahl Basisframes (SRAM))

Die Minimalkonfiguration ist so ausgelegt, dass es mindestens ein Basisframe pro Kanal gibt. Maximal sind 16 Basisframes pro Kanal möglich. Zusätzlich zu den 16 Basisframes im Kanal kann es noch 16 NCU-globale Basisframes geben.

Über die Systemvariable \$P\_CHBFR[n] können die Basisframe-Feldelemente gelesen und geschrieben werden. Beim Schreiben eines Basisframe-Feldelementes wird das verkettete Gesamt-Basisframe nicht aktiviert, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54..G599-Anweisung. Die Variable dient vorwiegend als Speicher für Schreibvorgänge auf das Basisframe von HMI oder PLC. Diese Frame-Variablen werden über die Datensicherung gesichert.

#### Aktuelle Kanal Basisframes \$P\_CHBFRAME[n]

Über die Systemvariable \$P\_CHBFRAME[n] können die aktuellen Kanal-Basisframe-Feldelemente gelesen und geschrieben werden. Das resultierende Gesamt-Basisframe wird durch den Schreibvorgang im Kanal eingerechnet. Beim Schreiben eines Basisframes wird das Gesamt-Basisframe neu berechnet.

#### Basisframe im Kanal \$P\_UBFR

Die Systemvariable bleibt aus Kompatibilitätsgründen erhalten, obwohl sie redundant zu der Variablen \$P\_CHBFR[0] ist.

Ein Schreiben auf die vordefinierte Variable \$P\_UBFR aktiviert das Basisframe mit dem Feldindex 0 nicht gleichzeitig, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54, ..G599-Anweisung. Bei NCU-globalen Frames wird das geänderte Frame in jedem Kanal der NCU, der eine G500, G54..G599-Anweisung ausführt, aktiv. Die Variable dient vorwiegend als Speicher für Schreibvorgänge auf das Basisframe von HMI oder PLC. Die Variable kann auch im Programm geschrieben und gelesen werden.

\$P\_UBFR ist identisch mit \$P\_CHBFR[0]. Standardmäßig gibt es immer ein Basisframe im Kanal, so dass die Systemvariable kompatibel zu älteren Ständen ist. Gibt es kein kanalspezifisches Basisframe, so wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.



### Aktuelles 1. Basisframe im Kanal \$P\_BFRAME

Die Systemvariable bleibt aus Kompatibilitätsgründen erhalten, obwohl sie redundant zu der Variablen \$P\_CHBFRAME[0] ist.

Über die vordefinierte Framevariable \$P\_BFRAME kann das aktuelle Basisframe mit dem Feldindex 0, welches im Kanal gültig ist, im Teileprogramm gelesen und geschrieben werden. Das geschriebene Basisframe wird sofort eingerechnet. Bei NCU-globalen einstellbaren Frames wirkt das geänderte Frame nur in dem Kanal, in dem das Frame programmiert wurde. Soll das Frame für alle Kanäle einer NCU geändert werden, so muss gleichzeitig \$P\_UBFR und \$P\_BFRAME beschrieben werden. Die anderen Kanäle müssen dann noch das entsprechende Frame mit z.B. G54 aktivieren.

\$P\_BFRAME ist identisch mit \$P\_CHBFRAME[0]. Die Systemvariable hat standardmäßig immer einen gültigen Wert. Gibt es kein kanalspezifisches Basisframe, so wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.

### Programmierung von Basisframes

Basisframes können über das Teileprogramm und über BTSS von der Bedienung und von der PLC geschrieben und gelesen werden. Über BTSS können jedoch nur die Datenhaltungsframes geschrieben werden.

#### 9.5.5.4 NCU-globale Basisframes \$P\_NCBFR[n]

Über das folgende Maschinendatum kann die Anzahl der globalen Basisframes projiziert werden:

MD18602 \$MN\_MM\_NUM\_GLOBAL\_BASE\_FRAMES (Anzahl der globalen Basisframes(SRAM))

Es gibt maximal 16 globale Basisframes. Alle Basisframes werden als Feld angelegt.

Über die Systemvariable \$P\_NCBFR[n] können die Basisframe-Feldelemente gelesen und geschrieben werden. Beim Schreiben eines Basisframe-Feldelementes wird das verkettete Gesamt-Basisframe nicht aktiviert, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54..G599-Anweisung. Soll das geänderte Frame in jedem Kanal der NCU aktiv werden, so muss jeder Kanal eine G500, G54..G599-Anweisung ausführen. Die Variable dient vorwiegend als Speicher für Schreibvorgänge auf das Basisframe von HMI oder PLC. Diese Frame-Variablen werden über die Datensicherung gesichert.

### Aktuelle NCU-globale Basisframes \$P\_NCBFRAME[n]

Über die Systemvariable \$P\_NCBFRAME[n] können die aktuellen globalen Basisframe-Feldelemente gelesen und geschrieben werden. Das resultierende Gesamt-Basisframes wird durch den Schreibvorgang im Kanal eingerechnet. Das geänderte Frame wird nur in dem Kanal, in dem das Frame programmiert wurde, aktiv. Soll das Frame für alle Kanäle einer NCU geändert werden, so muss gleichzeitig \$P\_NCBFR[n] und \$P\_NCBFRAME[n] beschrieben werden. Die anderen Kanäle müssen dann noch das Frame mit z.B. G54 aktivieren. Beim Schreiben eines Basisframes wird das Gesamt-Basisframe neu berechnet.

### Programmierung globaler Frames

Die Programmierung der globalen Frames erfolgt analog, wie bei den kanalspezifischen Frames. D.h. ein globales Basisframe wird über \$P\_NCBFR[n], und ein globales einstellbares Frame wird über \$P\_UIFR[n] programmiert.

Als Achsbezeichner bei den Frame-Programmbefehlen können Geometrie-Achs-, Kanalachs- und Maschinenachs-Bezeichner verwendet werden. Die Programmierung mit Kanal-Achs-Bezeichnern wird mit dem Alarm 18314 "Frame: Typkonflikt" abgelehnt, wenn es zu der Kanalachse keine Maschinenachse auf dieser NCU gibt. Kanalspezifische Frames können mit Geometrie-Achs-, Kanalachs- und Maschinenachs-Bezeichner programmiert werden. Die Programmierung mit Maschinen-Achs-Bezeichnern wird mit dem Alarm 18314 "Frame Typkonflikt" abgelehnt, wenn es zu der Maschinenachse keine korrespondierende Kanalachse auf dieser NCU gibt. Wird eine Maschinenachse oder eine Kanalachse, die gleichzeitig auch Geometrieachse ist, mit Framekomponenten beaufschlagt, so wird gleichzeitig auch die entsprechende Geometrie-Achs-Komponente verändert.

Beispiel:

```
$P_NCBFR[0] = CTRANS( ax1, 10 )  
$P_NCBFR[0] = CTRANS(x, 10)  
$P_NCBFR[0, ax1, FI ] = 0.1  
$P_NCBFR[0, x, FI] = 0.1
```

Auf globale Frames lassen sich keine Rotationen anwenden. Die Programmierung einer Rotation wird mit dem Alarm: "18310 Kanal %1 Satz %2 Frame: Rotation unzulässig", abgelehnt.

Die programmierte Verkettung von globalen Frames und kanalspezifischen Frames ist nicht möglich und wird mit dem Alarm 18314 "Frame: Typkonflikt" abgelehnt. Die interne Verkettung von allen globalen Frames und allen kanalspezifischen Frames zum Gesamtframe erfolgt im Kanal nur mit allen im Kanal bekannten Kanalachsen. Die Zuweisung eines Frames mit Rotationsanteilen an ein globales Frame wird mit dem Alarm "Frame: Rotation unzulässig" abgelehnt.

Beispiel:

```
$P_NCBFR[0] = CTRANS( x, 10 ):CROT( y, 45 ; fehlerhafte Zuweisung bei globalem
)                                     Basisframe
```

Die folgenden Frames sind kanalspezifisch:

`$P_UBFR`, `$P_BFRAME`, `$P_CHBFR[n]`,

`$P_CHBFRAME[n]`, `$P_NCBFRAME[n]`,

`$P_ACTBFRAME` und `$P_ACTFRAME`

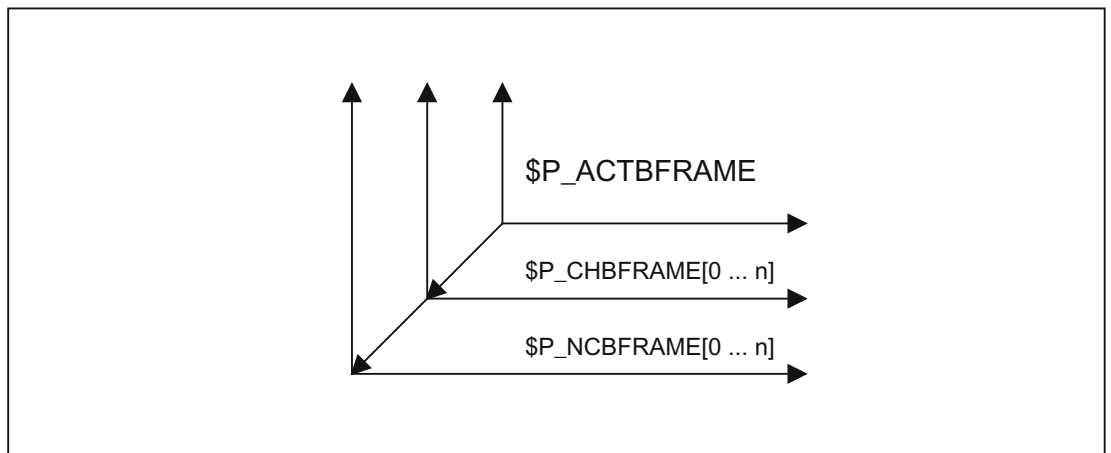
Diese Frames können Rotationsanteile enthalten.

Diese Frames wirken nur für den eingestellten Kanal.

Im SW-Stand 5.1 wird die Programmierung einer Kanalachse, die eine Link-Achse ist, mit dem Alarm "14092 Kanal %1 Satz %2 Achse %3 ist falscher Achstyp" abgelehnt. Eine Achse kann also nur programmiert werden, wenn sie physikalisch auf der NCU vorhanden ist.

#### 9.5.5.5 Gesamt-Basisframe `$P_ACTBFRAME`

Die Variable ermittelt das verkettete Gesamt-Basisframe. Die Variable ist nur lesbar.



`$P_ACTBFRAME=`

`$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n]`

### Programmierbarkeit des Gesamt-Basisframe

Über die Systemvariablen `$P_CHBFRMASK` und `$P_NCBFRMASK` kann der Anwender auswählen, welche Basisframes er in die Berechnung des "Gesamt"-Basisframes mit einbeziehen möchte. Die Variablen können nur im Programm programmiert werden und über BTSS gelesen werden. Der Wert der Variablen wird als Bitmaske interpretiert und gibt an, welches Basisframe-Feldelement von `$P_ACTBFRAME` mit in die Berechnung einfließt. Mit `$P_CHBFRMASK` kann vorgegeben werden, welche kanalspezifische Basisframes und mit `$P_NCBFRMASK`, welche NCU-globalen Basisframes eingerechnet werden. Mit der Programmierung der Variablen wird das Gesamt-Basisframe und das Gesamt-Frame neu berechnet. Nach `RESET` und in der Grundeinstellung ist der Wert von `$P_CHBFRMASK` gleich `$MC_CHBFRAME_RESET_MASK` und der Wert von `$P_NCBFRMASK` gleich `$MN_NCBFRAME_RESET_MASK`.

```
$P_NCBFRMASK = 'H81'      ; $P_NCBFRAME[0] : $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11'      ; $P_CHBFRAME[0] : $P_CHBFRAME[4]
```

#### 9.5.5.6 Programmierbarer Frame `$P_PFRAME`

Den programmierbaren Frame gibt es nur als aktiven Frame. Dieser Frame ist für den Programmierer reserviert.

Das programmierbare Frame bleibt bei `RESET` erhalten, wenn:

MD24010 `$MC_PFRAME_RESET_MODE` (Reset-Modus für programmierbaren Frame) = 1

Diese Funktionalität ist vor allem wichtig, wenn man nach `RESET` noch aus einer schrägen Bohrung herausfahren möchte.

### MIRROR

Spiegelungen einer Geometrieachse wurden bisher (bis SW-P4) anhand des Maschinendatums:

MD10610 `$MN_MIRROR_REF_AX` (Bezugsachse für das Spiegeln) auf eine festgelegte Bezugsachse bezogen.

Diese Festlegung ist aus Anwendersicht schwer nachvollziehbar. Bei Spiegelung der z-Achse wurde in der Anzeige übermittelt, dass die x-Achse gespiegelt ist und die y-Achse um 180° gedreht wurde. Bei Spiegelungen von zwei Achsen wird das noch komplexer und es ist nicht mehr leicht nachvollziehbar, welche Achsen gespiegelt wurden und welche nicht.

Ab SW-P5 gibt es die zusätzliche Möglichkeit das Spiegeln einer Achse eindeutig darzustellen. Ein Spiegeln wird dann nicht auf ein Spiegeln einer Bezugsachse und Verdrehungen von anderen Achsen abgebildet.

Projektiert werden kann diese Einstellung durch:

MD10610 `$MN_MIRROR_REF_AX = 0`

Die Programmierung des programmierbaren Frames mit `MIRROR` und `AMIRROR` wird erweitert. Bisher wurde der angegebene Wert der Koordinatenachse, wie z.B. bei `MIRROR X0` der Wert 0, nicht ausgewertet, sondern `AMIRROR` hatte eine Toggle-Funktion, d. h. `MIRROR X0` schaltet die Spiegelung ein und ein weiteres `AMIRROR X0` schaltet sie wieder aus. `MIRROR` wirkt immer absolut und `AMIRROR` additiv.

Mit der Maschinendatum-Einstellung:

MD10612 \$MN\_MIRROR\_TOGGLE = 0 ("Mirror Toggle")

kann festgelegt werden, dass die programmierten Werte ausgewertet werden.

Bei einem Wert von 0, wie bei `AMIRROR X0`, wird die Spiegelung der Achse ausgeschaltet, und bei Werte ungleich 0 wird die Achse gespiegelt, wenn sie noch nicht gespiegelt ist.

Das komponentenweise Lesen oder Schreiben von Spiegelungen ist unabhängig vom Maschinendatum:

MD10612 \$MN\_MIRROR\_TOGGLE

Ein Wert = 0 bedeutet, dass danach die Achse nicht gespiegelt ist und ein Wert = 1 heißt, dass die Achse danach immer gespiegelt wird, egal, ob die Achse schon gespiegelt war oder nicht.

```
$P_NCBFR[0,x,mi]=1           ; x-Achse wird immer gespiegelt.
$P_NCBFR[0,x,mi]=0           ; x-Achse spiegeln aus.
```

### Axiales Ersetzen G58, G59 (nur 840D sl)

Der Translationsanteil des programmierbaren Frames wird in einen absoluten Anteil und in einen Anteil für die Summe aller additiv-programmierten Translationen aufgespalten. Der absolute Anteil kann über `TRANS`, `CTRANS` oder durch Schreiben der Translationskomponente geändert werden, wobei der additive Anteil zu Null gesetzt wird. `G58` verändert nur den absoluten Translationsanteil für die angegebene Achse, die Summe der additiv-programmierten Translationen bleibt erhalten.

G58 X... Y... Z... A... ...

`G59` dient zum axialen Überschreiben der additiv-programmierten Translationen für die angegebenen Achsen, die mit `ATRANS` programmiert wurden.

G59 X... Y... Z... A... ...

Beispiel:

```
TRANS X10 Y10 Z10
ATRANS X5 Y5           ; Gesamttranslationen X15 Y15 Z10
G58 X20                ; Gesamttranslationen X25 Y15 Z10
G59 X10 Y10            ; Gesamttranslationen X30 Y20 Z10
```

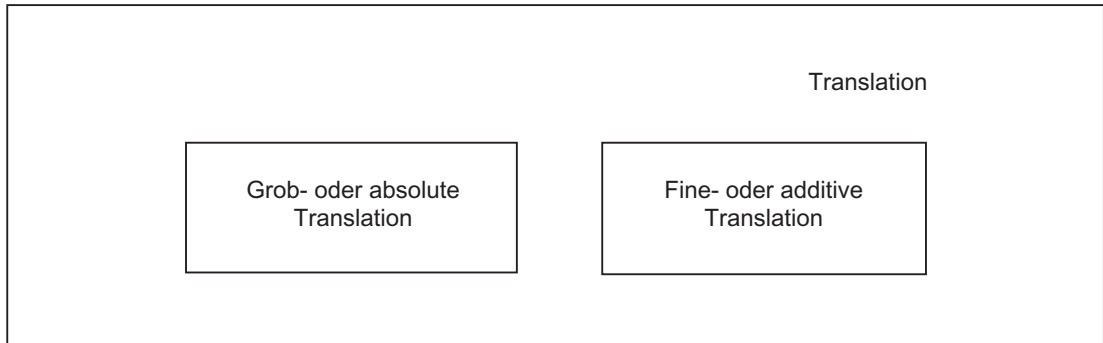
`G58` und `G59` ist nur einsetzbar, wenn:

MD24000 \$MC\_FRAME\_ADD\_COMPONENTS (Framekomponenten für `G58` / `G59`) = TRUE

Ansonsten wird der Alarm "18311 Kanal %1 Satz %2 Frame: Anweisung unzulässig" ausgegeben.

Die Funktion ist auch nur in Verbindung mit einer projizierten Feinverschiebung für das programmierbare Frame einsetzbar. Wird `G58` oder `G59` ohne projizierte Feinverschiebung verwendet, so wird der Alarm "18312 Kanal %1 Satz %2 Frame: Feinverschiebung nicht projiziert" ausgegeben.

Der absolute Anteil der Translation wird im Grobverschiebungsanteil und der additive Translationsanteil wird im Feinverschiebungsanteil gespeichert. Dazu wird das programmierbare Frame bzgl. der Feinverschiebung erweitert.



Beim Sichern des programmierbaren Frames in eine lokale Framevariable (LUD oder GUD) und beim Rückschreiben wird der Fine-Anteil mit übertragen.

Nachfolgende Tabelle beschreibt die Wirkung von diversen Programmbefehlen auf die absolute und die additive Translation.

	Grob- bzw. absolute Translation	Fine- bzw. additive Translation
TRANS X10	10	0
ATRANS X10	unverändert	alt_fine + 10
CTRANS (X, 10)	10	0
CTRANS ()	0	0
CFINE (X, 10)	0	10
\$P_PFRAME [X, TR] = 10	10	unverändert
\$P_PFRAME [X, FI] = 10	unverändert	10
G58 X10	10	unverändert
G59 X10	unverändert	10

### 9.5.5.7 Kanalspezifische Systemframes

#### Kanalspezifische Systemframes

Systemframes werden nur von Systemfunktionen, wie Istwertsetzen, Ankratzen, externe Nullpunktverschiebung und Schrägbearbeitung, beschrieben. Pro Kanal gibt es bis zu sieben Systemframes.

Die gültigen Systemframes im Kanal können festgelegt werden über das Maschinendatum:

MD28082 \$MC\_MM\_SYSTEM\_FRAME\_MASK (Systemframes (SRAM))

Aus Speicherplatzgründen sollten nur die Systemframes projektiert werden, die für die Systemfunktionen notwendig sind. Jedes Systemframe belegt pro Kanal ca. 1 kByte SRAM und ca. 6 kByte DRAM. Voreingestellt ist das Systemframe für Istwertsetzen und Ankratzen und das Systemframe für die Zyklen. Die Projektierung von kanalspezifischen Systemframes erfolgt bitcodiert entsprechend der nachfolgenden Tabelle:

Bit	Vorbesetzung	Systemframe
0	1	Istwertsetzen und Ankratzen
1	0	Externe Nullpunktverschiebung über Systemframe
2	0	TCARR und PAROT bei orientierbarem Werkzeugträger
3	0	TOROT und TOFRAME
4	0	Frame für Werkstückbezugspunkte
5	1	Frame für Zyklen
6	0	Frame für An- und Abwahl von Transformationen
7	0	\$P_ISO1FRAME: Frame für G51.1 Spiegeln (ISO)
8	0	\$P_ISO2FRAME: Frame für G68 2DROT (ISO)
9	0	\$P_ISO3FRAME: Frame für G68 3DROT (ISO)
10	0	\$P_ISO4FRAME: Frame für G51 Scale (ISO)
11	0	\$P_RELFR: Frame für relative Koordinatensysteme

Beispiel:

\$MC\_MM\_SYSTEM\_FRAME\_MASK = 'B001101' heißt:

Es gibt drei Systemframes, eines für Istwertsetzen, eines für PAROT und eines für TOROT und TOFRAME.

Über die Systemframe-Maske wird festgelegt, ob es ein Systemframe für die entsprechende Funktion gibt. Bei nicht projektiertem Frame wird die jeweilige Funktion unter Umständen mit einem Alarm abgelehnt.

### Systemframes in der Datenhaltung

Die Systemframes werden im statischen NC-Speicher gesichert und sind deshalb archivierbar und wieder ladbar. Die Systemframes in der Datenhaltung können über folgende Variablen im Programm geschrieben und gelesen werden:

Systemvariable	Bedeutung
\$P_SETFR	Systemframe für Istwertsetzen und Ankratzen ( <b>Set-Frame</b> )
\$P_EXTFR	Systemframe für Externe Nullpunktverschiebung ( <b>Ext-Frame</b> )
\$P_PARTFR	Systemframe für TCARR und PAROT ( <b>Part-Frame</b> )
\$P_TOOLFR	Systemframe für TOROT und TOFRAME ( <b>Tool-Frame</b> )
\$P_WPFR	Systemframe für Werkstückbezugspunkte ( <b>Work-Piece-Frame</b> )
\$P_CYCFR	Systemframe für Zyklen ( <b>Cycle-Frame</b> )
\$P_TRAFR	Systemframe für Transformationen ( <b>Transformation-Frame</b> )
\$P_ISO1FR	Systemframe für G51.1 Spiegeln (ISO)
\$P_ISO2FR	Systemframe für G68 2DROT (ISO)
\$P_ISO3FR	Systemframe für G68 3DROT (ISO)
\$P_ISO4FR	Systemframe für G51 Scale (ISO)
\$P_RELFR	Systemframe für relative Koordinatensysteme

Das Schreiben auf diese Frames sollte ausschließlich über Systemfunktionen erfolgen. Für Zyklen-Programmierer wurde die Möglichkeit geschaffen, die Frames über o. g. Variablen zu beschreiben. Das Schreiben auf ein nicht projektiertes Systemframe wird mit dem Alarm "Kanal %1 Satz %2 Name %3 nicht definiert o. Option nicht vorhanden" abgelehnt.

Die Systemframes in der Datenhaltung werden entweder durch die Systemfunktion (TOROT, PAROT, ..) direkt oder durch eine G500, G54...G599-Anweisung aktiviert.

### Aktive Systemframes

Die aktiven Systemframes sind die im Hauptlauf aktiven Frames. Es gibt zu jedem Systemframe in der Datenhaltung ein entsprechendes aktuelles Systemframe. Erst mit der Aktivierung des Datenhaltungsframes werden die Werte vorlaufbezogen eingerechnet.



Es gibt folgende aktuelle Systemframes:

- **\$P\_SETFRAME**  
Über die Variable `$P_SETFRAME` kann man im Teileprogramm das aktuelle Systemframe für das Istwertsetzen und Ankratzen lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_EXTFRAME**  
Über die Variable `$P_EXTFRAME` kann man im Teileprogramm das aktuelle Systemframe für die externe Nullpunktverschiebung lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_PARTFRAME**  
Über die Variable `$P_PARTFRAME` kann man im Teileprogramm das aktuelle Systemframe für `TCARR` und `PAROT` bei orientierbarem Werkzeugträger lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_TOOLFRAME**  
Über die Variable `$P_TOOLFRAME` kann man im Teileprogramm das aktuelle Systemframe für `TOROT` und `TOFRAME` lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_WPFRAME**  
Über die Variable `$P_WPFRAME` kann man im Teileprogramm das aktuelle Systemframe für das Setzen von Werkstückbezugspunkten lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_CYCFRAME**  
Über die Variable `$P_CYCFRAME` kann man im Teileprogramm das aktuelle Systemframe für Zyklen lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_TRAFRAME**  
Über die Variable `$P_TRAFRAME` kann man im Teileprogramm das aktuelle Systemframe für Transformationen lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_ISO1FRAME, \$P\_ISO2FRAME, \$P\_ISO3FRAME, \$P\_ISO4FRAME**  
Über die Variablen kann man im Teileprogramm die aktuellen Systemframes für spezielle ISO-Sprachbefehle lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_RELFRAME**  
Über die Variable `$P_RELFRAME` kann man im Teileprogramm das aktuelle Systemframe für relative Koordinatensysteme lesen und schreiben. Ist das Systemframe über MD28082 nicht projiziert, so liefert die Variable ein Nullframe zurück.
- **\$P\_ACSFRAME**  
Der aktuelle resultierende Frame, der das ENS-(ACS) Koordinatensystem festlegt, kann über die Variable `$P_ACSFRAME` gelesen und geschrieben werden.

Für MD24030 \$MC\_FRAME\_ACS\_SET = 0 berechnet sich das Frame wie folgt:

\$P\_ACSFRAME = \$P\_PARTFRAME : \$P\_SETFRAME : \$P\_EXTFRAME :  
\$P\_ISO1FRAME : \$P\_ISO2FRAME : \$P\_ISO3FRAME :  
\$P\_ACTBFRAME : \$P\_IFRAME :  
\$P\_TOOLFRAME : \$P\_WPFRAME

Für MD24030 \$MC\_FRAME\_ACS\_SET = 1 berechnet sich das Frame wie folgt:

\$P\_ACSFRAME = \$P\_PARTFRAME : \$P\_SETFRAME : \$P\_EXTFRAME :  
\$P\_ISO1FRAME : \$P\_ISO2FRAME : \$P\_ISO3FRAME :  
\$P\_ACTBFRAME : \$P\_IFRAME :  
\$P\_TOOLFRAME : \$P\_WPFRAME : \$P\_TRAFRAME :  
\$P\_PFRAME : \$P\_ISO4FRAME

- \$P\_ACTFRAME

Der aktuelle resultierende Gesamtframe \$P\_ACTFRAME ergibt sich nun als Verkettung aller Systemframes, Basisframes, dem aktuellen einstellbaren Frame und dem programmierbaren Frame. Der aktuelle Frame wird immer dann aktualisiert, wenn sich ein Frameanteil ändert.

Das aktuelle Gesamtframe ergibt sich nach folgender Formel:

\$P\_ACTFRAME = \$P\_PARTFRAME : \$P\_SETFRAME : \$P\_EXTFRAME :  
\$P\_ISO1FRAME : \$P\_ISO2FRAME : \$P\_ISO3FRAME :  
\$P\_ACTBFRAME : \$P\_IFRAME :  
\$P\_TOOLFRAME : \$P\_WPFRAME : \$P\_TRAFRAME :  
\$P\_PFRAME : \$P\_ISO4FRAME : \$P\_CYCFRAME

## 9.5.6 Implizite Frame-Änderungen

### 9.5.6.1 Frames beim Umschalten von Geometrieachsen

Die Geometrieachskonstellation kann sich im Kanal beim Ein- und Ausschalten einer Transformation und beim `GEOAX()`-Befehl (R3) ändern.

Über das Maschinendatum:

`MD10602 $MN_FRAME_GEOAX_CHANGE_MODE`

kann für alle Kanäle des Systems projiziert werden, ob das aktuelle Gesamtframe anhand der neuen Geometrieachsen neu berechnet wird, oder ob das Gesamtframe gelöscht wird.

Über das Maschinendatum sind vier Modi einstellbar:

- `MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 0`

Bei Umschaltungen von Geometrieachsen, wie bei An- und Abwahl von Transformationen und `GEOAX()` wird das aktuelle Gesamtframe gelöscht. Erst nach Aktivieren eines neuen Frames wird dann die geänderte Geometrieachskonstellation berücksichtigt.

- `MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1`

Das aktuelle Gesamtframe wird beim Umschalten von Geometrieachsen neu berechnet, wobei die Translationen, Skalierungen und Spiegelungen der neuen Geometrieachsen wirksam werden. Die Drehungen der Geometrieachsen, die vor dem Umschalten programmiert wurden, bleiben auch für die neuen Geometrieachsen wirksam. Für `TRANSMIT`, `TRACYL` und `TRAANG` siehe Kapitel "Frames bei An- und Abwahl von Transformationen (Seite 807)".

- `MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 2`

Das aktuelle Gesamtframe wird beim Umschalten von Geometrieachsen neu berechnet, wobei die Translationen, Skalierungen und Spiegelungen der neuen Geometrieachsen wirksam werden. Sind vor der Umschaltung in den aktuellen Basisframes, dem aktuellen einstellbaren Frame oder im programmierbaren Frame, Drehungen aktiv, so wird die Umschaltung mit Alarm "Frame: Umschaltung der Geometrieachsen unzulässig" abgebrochen.

Für `TRANSMIT`, `TRACYL` und `TRAANG` siehe Kapitel "Frames bei An- und Abwahl von Transformationen (Seite 807)".

- `MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 3`

Das aktuelle Gesamtframe wird bei An- und Abwahl von Transformationen gelöscht. Bei `GEOAX()` wird das Gesamtframe neu berechnet, wobei die Translationen, Skalierungen und Spiegelungen der neuen Geometrieachsen wirksam werden. Die Drehungen der Geometrieachsen, die vor dem Umschalten programmiert wurden, bleiben auch für die neuen Geometrieachsen wirksam.

Die Werkstückgeometrie wird von einem Koordinatensystem beschrieben, das von den Geometrieachsen aufgespannt wird. Jeder Geometrieachse ist eine Kanalachse und jeder Kanalachse eine Maschinenachse zugeordnet. Für jedes Frame (Systemframe, Basisframes, einstellbare Frames, prog. Frame) gibt es für jede Maschinenachse ein axiales Frame. Wird einer Geometrieachse eine neue Maschinenachse zugeordnet, so bringt die Maschinenachse ihre axialen Frameanteile, wie Translationen (grob und fein), Skalierung und Spiegelung, des entsprechenden Frames mit. Die neue Geometrie im Kanal wird dann von den neuen Konturframes gebildet, die sich aus den bis zu drei neuen Geometrieachsen bilden.

Die aktuell gültigen Frames werden bei der Umschaltung von Geoachsen neu berechnet und ein resultierendes Gesamtframe daraus gewonnen. Die Datenhaltungsframes werden erst nach Aktivierung berücksichtigt.

Beispiel:

Die Kanalachse a soll durch einen `GeoAx`-Tausch zur Geometrieachse x werden. Durch den Tausch soll das Programmierbare Frame einen Trans-Anteil von 10 in der x-Achse haben. Das aktuelle einstellbare Frame soll erhalten bleiben.

MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 1

Programmcode	Kommentar
\$P_UIFR[1] = CROT(x,10,y,20,z,30)	; Frame bleibt nach GeoAx-Tausch erhalten.
G54	; Einstellbares Frame wird aktiv.
TRANS a10	; Achsiale Verschiebung von a wird mit getauscht.
GEOAX(1, a)	; a wird zur x-Achse; \$P_ACTFRAME=CROT(x,10, y,20,z,30):CTRANS(x10).

Bei Transformationswechsel können gleichzeitig mehrere Kanalachsen zu Geometrieachsen werden.

Beispiel:

Durch eine 5-Achs-Orientierungstranformation werden die Kanalachsen 4, 5 und 6 zu Geometrieachsen der Transformation. Die Geometrieachsen vor der Transformation werden also alle ersetzt. Beim Einschalten der Transformation ändern sich alle aktuellen Frames. Zur Berechnung des neuen WKS-Systems werden die achsialen Frameanteile der Kanalachsen, die zu Geometrieachsen werden, berücksichtigt. Programmierte Drehungen vor der Transformation werden beibehalten. Nach dem Ausschalten der Transformation wird das alte WKS wieder hergestellt. Der häufigste Anwendungsfall wird wohl der sein, dass sich die Geometrieachsen vor und nach der Transformation nicht ändern und die Frames so beibehalten werden sollen, wie sie vor der Transformation waren.

**Maschinendaten:**

\$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 1

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[0] = "CAX"

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[1] = "CAY"

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[2] = "CAZ"

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[3] = "A"

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[4] = "B"

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[5] = "C"

\$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[0] = 1

\$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[1] = 2

\$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[2] = 3

\$MC\_AXCONF\_GEOAX\_NAME\_TAB[0] = "X"

\$MC\_AXCONF\_GEOAX\_NAME\_TAB[1] = "Y"

\$MC\_AXCONF\_GEOAX\_NAME\_TAB[2] = "Z"

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[0]=4

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[1]=5

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[2]=6

\$MC\_TRAFO\_AXES\_IN\_1[0]=4

\$MC\_TRAFO\_AXES\_IN\_1[1]=5

\$MC\_TRAFO\_AXES\_IN\_1[2]=6

\$MC\_TRAFO\_AXES\_IN\_1[3]=1

\$MC\_TRAFO\_AXES\_IN\_1[4]=2

**Programm:**

**Programmcode**

```

$P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
$P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
$P_IFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(z,45)
$P_PFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(x,10,y,20,z,30)
    
```

**Programmcode**

**Kommentar**

Programmcode	Kommentar
TRAORI	; Trafo setzt GeoAx(4,5,6) ; \$P_NCBFRAME[0] = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3) ; \$P_ACTBFRAME =ctrans(x,8,y,10,z,12,cax,2,cay,4,caz,6) ; \$P_PFRAME = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3):crot(x,10,y,20,z,30) ; \$P_IFRAME = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3):crot(z,45)
TRAFOOF	; Ausschalten der Transformation setzt GeoAx(1,2,3) ; \$P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6) ; \$P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6) ; \$P_IFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(z,45) ; \$P_PFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(x,10,y,20,z,30)

### 9.5.6.2 Frames bei An- und Abwahl von Transformationen

Diese Funktion gibt es ab NCK 51.00.00. Es werden die Transformationen TRANSMIT, TRACYL und TRAANG unterstützt.

Bei An- und Abwahl von Transformationen ändert sich in der Regel die Geometrieachszuordnung zu den Kanalachsen. Transformationen, bei denen Rundachsen zu Linearachsen und Linearachsen zu Rundachsen werden, ist keine eindeutige Zuordnung von axialen Frameanteilen zu geometrischen Konturframeanteilen möglich. Bei diesen nichtlinearen Transformationen muss eine Sonderbehandlung zur Aufbereitung des Konturframes erfolgen.

Es wird der Modus, der durch MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 1 und 2 eingestellt wird, so erweitert, dass auch die o. g. Transformationen berücksichtigt werden.

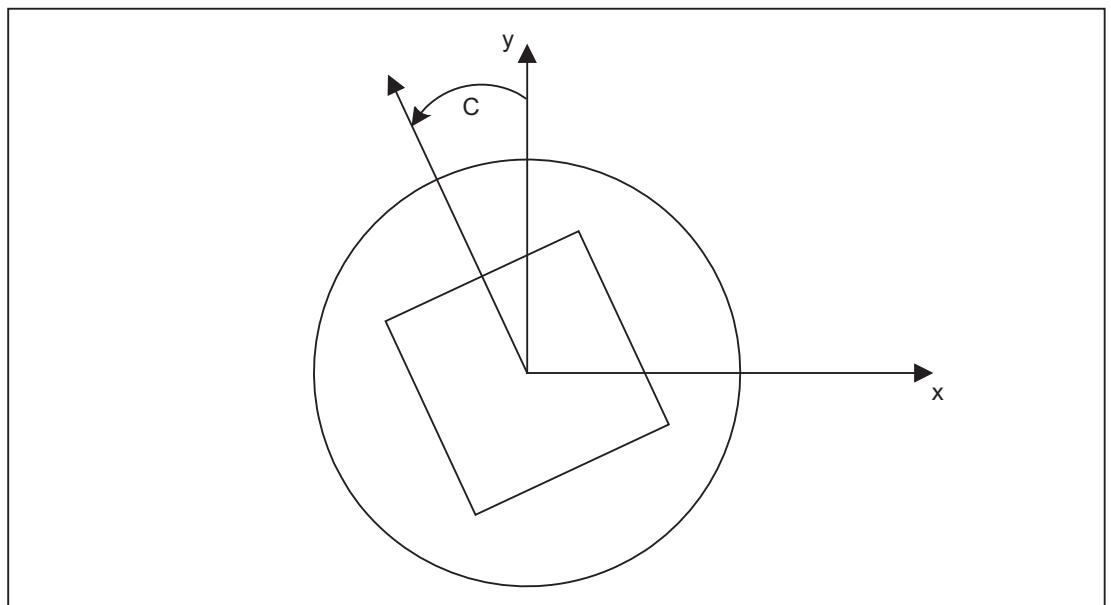
Bei Transformationsanwahl wird das Konturframe anhand der axialen Frames aufgebaut. Die virtuelle Geometrieachse bei den Transformationen TRANSMIT, TRACYL und TRAANG unterliegt dabei einer Sonderbehandlung.

#### ACHTUNG

##### Transformationen mit virtuellen Achsen

Bei der Anwahl von TRANSMIT oder TRACYL werden Verschiebungen, Skalierungen und Spiegelungen der realen Y-Achse nicht in die virtuelle Y-Achse übernommen. Verschiebungen, Skalierungen und Spiegelungen der virtuellen Y-Achse werden bei TRAFOOF gelöscht.

### TRANSMIT



#### **Transmit-Erweiterungen:**

Das axiale Gesamtframe der TRANSMIT-Rundachse, d. h. die Translation, die Feinverschiebung, die Spiegelung und die Skalierung kann über folgende Maschinendaten in der Transformation berücksichtigt werden:

MD24905 \$MC\_TRANSMIT\_ROT\_AX\_FRAME\_1 = 1

MD24955 \$MC\_TRANSMIT\_ROT\_AX\_FRAME\_2 = 1

Eine Verschiebung der Rundachse kann z. B. durch eine Kompensation der Schräglage eines Werkstückes in einem Frame innerhalb der Framekette eingetragen werden. Diese Verschiebung soll in der Regel auch in der Transformation als Offset der Rundachse berücksichtigt werden. Eine Verschiebung der c-Achse, wie im obigen Bild, führt dann zu entsprechenden x- und y-Werten.

MD24905 \$MC\_TRANSMIT\_ROT\_AX\_FRAME\_1 = 2

MD24955 \$MC\_TRANSMIT\_ROT\_AX\_FRAME\_2 = 2

Mit dieser Einstellung wird die axiale Verschiebung der Rundachse bis zum ENS in der Transformation berücksichtigt. Die in den ENS-Frames enthaltenen axialen Verschiebungen der Rundachse werden in das Transformationsframe als Drehung eingetragen. Diese Einstellung ist nur wirksam, wenn das Transformationsframe projiziert ist.

#### **Frame-Erweiterungen:**

Nachfolgend beschriebene Erweiterungen gelten nur für folgende Maschinendaten-Einstellungen:

MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 1

MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 2

Mit Anwahl der Transformation TRANSMIT entsteht gekoppelt über die Rundachse eine virtuelle Geometrieachse, die keinen Bezug zu einem axialen Frame hat, sondern nur im Konturframe berücksichtigt wird. Der geometrische Wert ergibt sich aus der Drehung einer Rundachse. Alle anderen Geometrieachsen übernehmen bei Trafo-Anwahl ihre axialen Anteile.

#### **Translationen:**

Die Translationen der virtuellen Achse werden bei TRANSMIT-Anwahl gelöscht. Die Translationen der Rundachse können in der Transformation berücksichtigt werden.

#### **Drehungen:**

Die Drehungen vor der Transformation werden übernommen.

#### **Spiegelungen:**

Die Spiegelung der virtuellen Achse wird gelöscht. Die Spiegelung Rundachse kann in der Transformation berücksichtigt werden.

#### **Skalierungen:**

Die Skalierung der virtuellen Achse wird gelöscht. Die Skalierung der Rundachse kann in der Transformation berücksichtigt werden.



**Beispiel:****Maschinendaten für TRANSMIT**

```

; FRAME-Projektierungen

$MC_MM_SYSTEM_FRAME_MASK='H41'           ; TRAFRAME, SETFRAME
$MC_CHSFRAME_RESET_MASK='H41'           ; Frames sind nach Reset aktiv.
$MC_CHSFRAME_POWERON_MASK='H41'        ; Frames werden bei Power On gelöscht.

$MN_FRAME_GEOAX_CHANGE_MODE=1          ; Frames werden nach GeoAx-Umschaltung
                                         umgerechnet.
$MC_RESET_MODE_MASK='H4041'            ; Basisframe wird nach RESET nicht
                                         abgewählt.
; $MC_RESET_MODE_MASK='H41'            ; Basisframe wird nach RESET abgewählt.

; $MC_GCODE_RESET_VALUES[7]=2          ; G54 ist Voreinstellung.
$MC_GCODE_RESET_VALUES[7]=1           ; G500 ist Voreinstellung.

$MN_MM_NUM_GLOBAL_USER_FRAMES=0
$MN_MM_NUM_GLOBAL_BASE_FRAMES=3

$MC_MM_NUM_USER_FRAMES=10              ; von 5 bis 100
$MC_MM_NUM_BASE_FRAMES=3               ; von 0 bis 8

$MN_NCBFRAME_RESET_MASK='HFF'
$MC_CHBFRAME_RESET_MASK='HFF'

$MN_MIRROR_REF_AX=0                    ; Keine Normierung bei der Spiegelung.
$MN_MIRROR_TOGGLE=0

$MN_MM_FRAME_FINE_TRANS=1              ; Feinverschiebung
$MC_FRAME_ADD_COMPONENTS=TRUE           ; G58, G59 ist möglich.

```

```
; TRANSMIT ist 1. Trafo

$MC_TRAFO_TYPE_1=256

$MC_TRAFO_AXES_IN_1[0]=1
$MC_TRAFO_AXES_IN_1[1]=6
$MC_TRAFO_AXES_IN_1[2]=3
$MC_TRAFO_AXES_IN_1[3]=0
$MC_TRAFO_AXES_IN_1[4]=0

$MA_ROT_IS_MODULO[AX6]=TRUE;

$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=6
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3

$MC_TRANSMIT_BASE_TOOL_1[0]=0.0
$MC_TRANSMIT_BASE_TOOL_1[1]=0.0
$MC_TRANSMIT_BASE_TOOL_1[2]=0.0

$MC_TRANSMIT_ROT_AX_OFFSET_1=0.0
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1=TRUE

$MC_TRANSMIT_ROT_AX_FRAME_1=1
```

```

; TRANSMIT ist 2. Trafo

$MC_TRAFO_TYPE_2=256

$MC_TRAFO_AXES_IN_2[0]=1
$MC_TRAFO_AXES_IN_2[1]=6
$MC_TRAFO_AXES_IN_2[2]=2
$MC_TRAFO_AXES_IN_2[3]=0
$MC_TRAFO_AXES_IN_2[4]=0

$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0]=1
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1]=6
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2]=2

$MC_TRANSMIT_BASE_TOOL_2[0]=4.0
$MC_TRANSMIT_BASE_TOOL_2[1]=0.0
$MC_TRANSMIT_BASE_TOOL_2[2]=0.0

$MC_TRANSMIT_ROT_AX_OFFSET_2=19.0
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_2=TRUE

$MC_TRANSMIT_ROT_AX_FRAME_2=1

```

**Teileprogramm:**

```

; Frameeinstellungen
N820 $P_UIFR[1] = ctrans(x,1,y,2,z,3,c,4)
N830 $P_UIFR[1] = $P_UIFR[1] : crot(x,10,y,20,z,30)
N840 $P_UIFR[1] = $P_UIFR[1] : cmirror(x,c)
N850
N860 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,c,15)
N870

; Werkzeuganwahl, Aufspannkompensation, Ebenenanwahl
N890 T2 D1 G54 G17 G90 F5000 G64 SOFT
N900

```

```
; Anfahren der Ausgangsstellung
N920 G0 X20 Z10
N930
N940 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,C,15)
N950 setal(61000)
N960 endif
N970 if $P_BFRAME <> $P_CHBFR[0]
N980 setal(61000)
N990 endif
N1000 if $P_IFRAME <>
CTRANS(X,1,Y,2,Z,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1010 setal(61000)
N1020 endif
N1030 if $P_IFRAME <> $P_UIFR[1]
N1040 setal(61000)
N1050 endif
N1060 if $P_ACTFRAME <>
CTRANS(X,11,Y,22,Z,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1070 setal(61000)
N1080 endif
N1090
N1100 TRANSMIT(2)
N1110
N1120 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,20,CAZ,30,C,15)
N1130 setal(61000)
N1140 endif
N1180 if $P_IFRAME <>
CTRANS(X,1,Y,0,Z,2,CAZ,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1190 setal(61000)
N1200 endif
N1240 if $P_ACTFRAME <>
CTRANS(X,11,Y,0,Z,22,CAZ,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1250 setal(61001)
N1260 endif
N1270
N1280
N1290 $P_UIFR[1,x,tr] = 11
N1300 $P_UIFR[1,y,tr] = 14
N1310
N1320 g54
N1330
```

```
| ; Frame einstellen  
N1350 ROT RPL=-45  
N1360 ATRANS X-2 Y10  
N1370
```

```
| ; Vierkant schrappen  
N1390 G1 X10 Y-10 G41 OFFN=1; Aufmass 1mm  
N1400 X-10  
N1410 Y10  
N1420 X10  
N1430 Y-10  
N1440
```

```
| ; Werkzeugwechsel  
N1460 G0 Z20 G40 OFFN=0  
N1470 T3 D1 X15 Y-15  
N1480 Z10 G41  
N1490
```

```
| ; Vierkant schlichten  
N1510 G1 X10 Y-10  
N1520 X-10  
N1530 Y10  
N1540 X10  
N1550 Y-10  
N1560
```

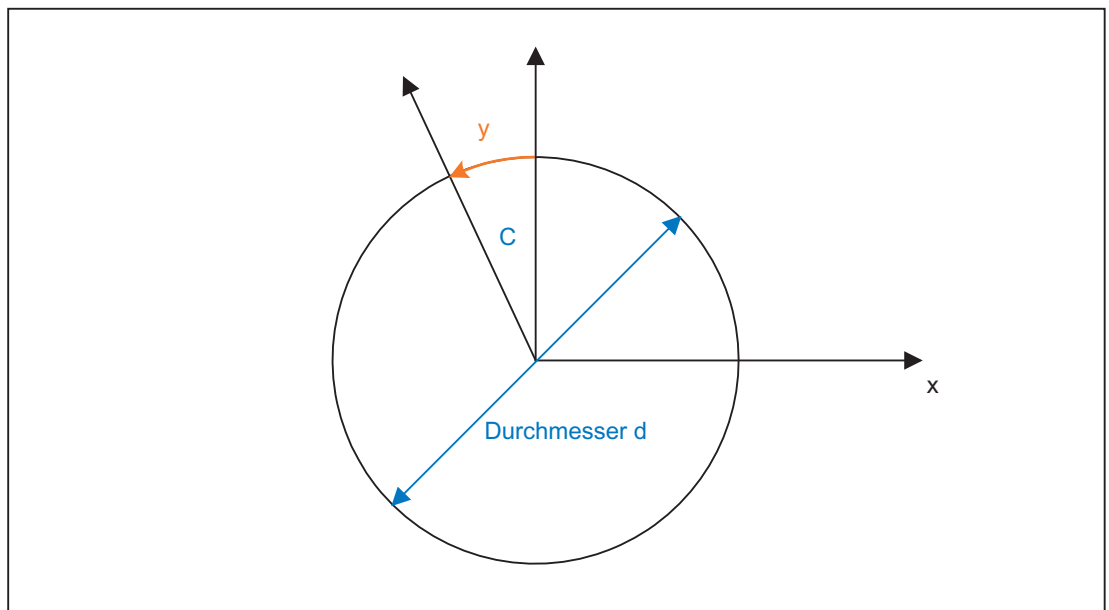
```
; Frame abwählen
N2950 m30 N1580 Z20 G40
N1590 TRANS
N1600
N1610 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,20,CAZ,30,C,15)
N1620 setal(61000)
N1630 endif
N1640 if $P_BFRAME <> $P_CHBFR[0]
N1650 setal(61000)
N1660 endif
N1670 if $P_IFRAME <>
TRANS(X,11,Y,0,Z,2,CAZ,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1680 setal(61000)
N1690 endif
N1730 if $P_ACTFRAME <>
TRANS(X,21,Y,0,Z,22,CAZ,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1740 setal(61001)
N1750 endif
N1760
N1770 TRAF00F
N1780
N1790 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,C,15)
N1800 setal(61000)
N1810 endif
N1820 if $P_BFRAME <> $P_CHBFR[0]
N1830 setal(61000)
N1840 endif
N1850 if $P_IFRAME <>
TRANS(X,11,Y,2,Z,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1860 setal(61000)
N1870 endif
N1880 if $P_IFRAME <> $P_UIFR[1]
N1890 setal(61000)
N1900 endif
N1910 if $P_ACTFRAME <>
TRANS(X,21,Y,22,Z,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1920 setal(61002)
N1930 endif
N1940
N2010 $P_UIFR[1] = ctrans()
N2011 $P_CHBFR[0] = ctrans()
N2020 $P_UIFR[1] = ctrans(x,1,y,2,z,3,c,0)
N2021 G54
N2021 G0 X20 Y0 Z10 C0
N2030 TRANSMIT(1)
```

```

N2040 TRANS x10 y20 z30
N2041 ATRANS y200
N2050 G0 X20 Y0 Z10
N2051 if $P_IFRAME <> CTRANS (X,1,Y,0,Z,3,CAY,2)
N2052 setal(61000)
N2053 endif
N2054 if $P_ACTFRAME <> CTRANS (X,11,Y,20,Z,33,CAY,2):CFINE(Y,200)
N2055 setal(61002)
N2056 endif
N2060 TRAFOOF
N2061 if $P_IFRAME <> $P_UIFR[1]
N2062 setal(61000)
N2063 endif
N2064 if $P_ACTFRAME <> CTRANS (X,11,Y,2,Z,33):CFINE(Y,0)
N2065 setal(61002)
N2066 endif

```

## TRACYL



### TRACYL-Erweiterungen:

Das axiale Gesamtframe der TRACYL-Rundachse, d. h. die Translation, die Feinverschiebung, die Spiegelung und die Skalierung, kann über folgende Maschinendaten in der Transformation berücksichtigt werden:

MD24805 \$MC\_TRACYL\_ROT\_AX\_FRAME\_1 = 1

MD24855 \$MC\_TRACYL\_ROT\_AX\_FRAME\_2 = 1

Eine Verschiebung der Rundachse kann z. B. durch eine Kompensation der Schräglage eines Werkstückes in einem Frame innerhalb der Framekette eingetragen werden. Diese Verschiebung soll in der Regel auch in der Transformation als Offset der Rundachse oder als y-Verschiebung berücksichtigt werden. Eine Verschiebung der c-Achse (wie im obigen Bild) führt dann zu entsprechenden x- und y-Werten.

MD24805 \$MC\_TRACYL\_ROT\_AX\_FRAME\_1 = 2

MD24855 \$MC\_TRACYL\_ROT\_AX\_FRAME\_2 = 2

Mit dieser Einstellung wird die axiale Verschiebung der Rundachse bis zum ENS in der Transformation berücksichtigt. Die in den ENS-Frames enthaltenen axialen Verschiebungen der Rundachse werden in das Transformationsframe als Verschiebung auf der Manteloberfläche eingetragen. Diese Einstellung ist nur wirksam, wenn das Transformationsframe projiziert ist.

**Frame-Erweiterungen:**

Nachfolgend beschriebene Erweiterungen gelten nur für folgende Maschinendaten-Einstellungen:

MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 1

MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 2

Mit Anwahl der Transformation TRACYL entsteht gekoppelt über die Rundachse eine virtuelle Geometrieachse auf der Manteloberfläche, die keinen Bezug zu einem axialen Frame hat, sondern nur im Konturframe berücksichtigt wird. Alle Komponenten der virtuellen Geometrieachse werden abgelöscht. Alle anderen Geometrieachsen übernehmen bei Trafo-Anwahl ihre axialen Anteile.

**Translationen:**

Die Translationen der virtuellen Achse werden bei TRACYL-Anwahl gelöscht. Die Translationen der Rundachse können in der Transformation berücksichtigt werden.

**Drehungen:**

Die Drehungen vor der Transformation werden übernommen.

**Spiegelungen:**

Die Spiegelung der virtuellen Achse wird gelöscht. Die Spiegelung Rundachse kann in der Transformation berücksichtigt werden.

**Skalierungen:**

Die Skalierung der virtuellen Achse wird gelöscht. Die Skalierung der Rundachse kann in der Transformation berücksichtigt werden.



**Beispiel:****Maschinendaten für TRACYL:**

```

; FRAME-Projektierungen

$MC_MM_SYSTEM_FRAME_MASK = 'H41'           ; TRAFRAME, SETFRAME
$MC_CHSFRAME_RESET_MASK = 'H41'           ; Frames sind nach Reset aktiv.
$MC_CHSFRAME_POWERON_MASK = 'H41'        ; Frames werden bei Power On gelöscht.

$MN_FRAME_GEOAX_CHANGE_MODE = 1           ; Frames werden nach GeoAx-Umschaltung
                                           umgerechnet.

$MC_RESET_MODE_MASK = 'H4041'             ; Basisframe wird nach Reset nicht
                                           abgewählt.
; $MC_RESET_MODE_MASK = 'H41'             ; Basisframe wird nach Reset abgewählt.

; $MC_GCODE_RESET_VALUES[7] = 2           ; G54 ist Voreinstellung.
$MC_GCODE_RESET_VALUES[7] = 1           ; G500 ist Voreinstellung.

$MN_MM_NUM_GLOBAL_USER_FRAMES = 0
$MN_MM_NUM_GLOBAL_BASE_FRAMES = 3

$MC_MM_NUM_USER_FRAMES = 10               ; von 5 bis 100
$MC_MM_NUM_BASE_FRAMES = 3               ; von 0 bis 8

$MN_NCBFRAME_RESET_MASK = 'HFF'
$MC_CHBFRAME_RESET_MASK = 'HFF'

$MN_MIRROR_REF_AX = 0                     ; Keine Normierung bei der Spiegelung.
$MN_MIRROR_TOGGLE = 0
$MN_MM_FRAME_FINE_TRANS = 1               ; Feinverschiebung
$MC_FRAME_ADD_COMPONENTS = TRUE           ; G58, G59 ist möglich

```

```
; TRACYL mit Nutwandkorrektur ist 3. Trafo

$MC_TRAFO_TYPE_3 = 513; TRACYL

$MC_TRAFO_AXES_IN_3[0] = 1
$MC_TRAFO_AXES_IN_3[1] = 5
$MC_TRAFO_AXES_IN_3[2] = 3
$MC_TRAFO_AXES_IN_3[3] = 2

$MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] = 1
$MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] = 5
$MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] = 3

$MC_TRACYL_BASE_TOOL_1[0] = 0.0
$MC_TRACYL_BASE_TOOL_1[1] = 0.0
$MC_TRACYL_BASE_TOOL_1[2] = 0.0

$MC_TRACYL_ROT_AX_OFFSET_1 = 0.0
$MC_TRACYL_ROT_SIGN_IS_PLUS_1 = TRUE

$MC_TRACYL_ROT_AX_FRAME_1 = 1
```

**Teileprogramm:**

```
;Einfacher Verfahrttest mit Nutwandkorrektur
N450 G603
N460
```

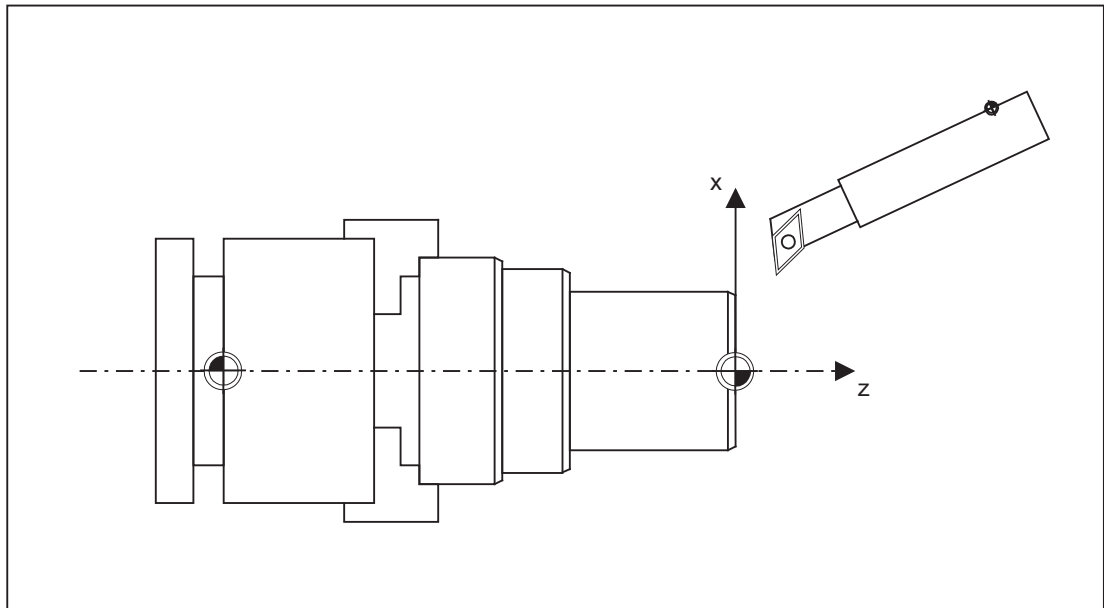
```
; Frameeinstellungen
N500 $P_UIFR[1] = ctrans(x,1,y,2,z,3,b,4)
N510 $P_UIFR[1] = $P_UIFR[1] : crot(x,10,y,20,z,30)
N520 $P_UIFR[1] = $P_UIFR[1] : cmirror(x,b)
N530
N540 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,b,15)
N550
N560 G54
N570
```

```
; Bahnsteuerbetrieb mit angewähltem Überschleifen
N590 G0 x0 y0 z-10 b0 G90 F50000 T1 D1 G19 G641 ADIS=1 ADISPOS=5
N600
N610 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,15)
N620 setal(61000)
N630 endif
N640 if $P_BFRAME <> $P_CHBFR[0]
N650 setal(61000)
N660 endif
N670 if $P_IFRAME <>
TRANS(X,1,Y,2,Z,3,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N680 setal(61000)
N690 endif
N700 if $P_IFRAME <> $P_UIFR[1]
N710 setal(61000)
N720 endif
N730 if $P_ACTFRAME <>
TRANS(X,11,Y,22,Z,33,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N740 setal(61000)
N750 endif
N760
```

```
; Transformation ein
N780 TRACYL(40.)
N790
N800 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N810 setal(61000)
N820 endif
N830 if $P_CHBFR[0] <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N840 setal(61000)
N850 endif
N860 if $P_IFRAME <>
TRANS(X,1,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N870 setal(61000)
N880 endif
N890 if $P_UIFR[1] <>
TRANS(X,1,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N900 setal(61000)
N910 endif
N920 if $P_ACTFRAME <>
TRANS(X,11,Y,0,Z,33,CAY,22,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N930 setal(61001)
N940 endif
N950
N960 $P_UIFR[1,x,tr] = 11
N970 $P_UIFR[1,y,tr] = 14
N980
N990 g54
N1000
N1010 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N1020 setal(61000)
N1030 endif
N1040 if $P_BFRAME <> $P_CHBFR[0]
N1050 setal(61000)
N1060 endif
N1070 if $P_IFRAME <>
TRANS(X,11,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1080 setal(61000)
N1090 endif
N1100 if $P_IFRAME <> $P_UIFR[1]
N1110 setal(61000)
N1120 endif
N1130 if $P_ACTFRAME <>
TRANS(X,21,Y,0,Z,33,CAY,22,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1140 setal(61001)
N1150 endif
N1160
```

```
; Transformation aus
N1180 TRAFOOF
N1190
N1200 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,15)
N1210 setal(61000)
N1220 endif
N1230 if $P_BFRAME <> $P_CHBFR[0]
N1240 setal(61000)
N1250 endif
N1260 if $P_IFRAME <>
TRANS(X,11,Y,2,Z,3,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1270 setal(61000)
N1280 endif
N1290 if $P_IFRAME <> $P_UIFR[1]
N1300 setal(61000)
N1310 endif
N1320 if $P_ACTFRAME <>
TRANS(X,21,Y,22,Z,33,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1330 setal(61002)
N1340 endif
N1350
N1360 G00 x0 y0 z0 G90
N1370
N1380 m30
```

## TRAANG



### Frame-Erweiterungen:

Nachfolgend beschriebene Erweiterungen gelten nur für folgende Maschinendaten-Einstellungen:

MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 1

MD10602 \$MN\_FRAME\_GEOAX\_CHANGE\_MODE = 2

### Translationen:

Die Translationen der virtuellen Achse werden bei TRAANG-Anwahl beibehalten.

### Drehungen:

Die Drehungen vor der Transformation werden übernommen.

### Spiegelungen:

Die Spiegelung der virtuellen Achse wird übernommen.

### Skalierungen:

Die Skalierung der virtuellen Achse wird übernommen.

**Beispiel:****Maschinendaten für TRAANG:**

```

; FRAME - Projektierungen

$MC_MM_SYSTEM_FRAME_MASK = 'H1'           ; SETFRAME
$MC_CHSFRAME_RESET_MASK = 'H41'          ; Frames sind nach RESET aktiv.
$MC_CHSFRAME_POWERON_MASK = 'H41'        ; Frames werden bei "Power On" gelöscht.

$MN_FRAME_GEOAX_CHANGE_MODE = 1          ; Frames werden nach GeoAx-Umschaltung
                                           umgerechnet.

$MC_RESET_MODE_MASK = 'H4041'            ; Basisframe wird nach RESET nicht
                                           abgewählt.
; $MC_RESET_MODE_MASK = 'H41'            ; Basisframe wird nach RESET abgewählt.

; $MC_GCODE_RESET_VALUES[7] = 2          ; G54 ist Voreinstellung.
$MC_GCODE_RESET_VALUES[7] = 1            ; G500 ist Voreinstellung.

$MN_MM_NUM_GLOBAL_USER_FRAMES = 0
$MN_MM_NUM_GLOBAL_BASE_FRAMES = 3

$MC_MM_NUM_USER_FRAMES = 10              ; von 5 bis 100
$MC_MM_NUM_BASE_FRAMES = 3              ; von 0 bis 8

$MN_NCBFRAME_RESET_MASK = 'HFF'
$MC_CHBFRAME_RESET_MASK = 'HFF'

$MN_MIRROR_REF_AX = 0                    ; Keine Normierung bei der Spiegelung.
$MN_MIRROR_TOGGLE = 0
$MN_MM_FRAME_FINE_TRANS = 1              ; Feinverschiebung
$MC_FRAME_ADD_COMPONENTS = TRUE          ; G58, G59 ist möglich.

```

```
; TRAANG ist 1. Trafo

$MC_TRAFO_TYPE_1 = 1024

$MC_TRAFO_AXES_IN_1[0] = 4           ; schräge Achse
$MC_TRAFO_AXES_IN_1[1] = 3           ; Achse parallel zu z
$MC_TRAFO_AXES_IN_1[2] = 2
$MC_TRAFO_AXES_IN_1[3] = 0
$MC_TRAFO_AXES_IN_1[4] = 0

$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0] = 4
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1] = 2
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2] = 3

$MC_TRAANG_ANGLE_1 = 85.
$MC_TRAANG_PARALLEL_VELO_RES_1 = 0.
$MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.

$MC_TRAANG_BASE_TOOL_1[0] = 0.0
$MC_TRAANG_BASE_TOOL_1[1] = 0.0
$MC_TRAANG_BASE_TOOL_1[2] = 0.0
```

```
; TRAANG ist 2. Trafo

$MC_TRAFO_TYPE_2 = 1024

$MC_TRAFO_AXES_IN_2[0] = 4
$MC_TRAFO_AXES_IN_2[1] = 3
$MC_TRAFO_AXES_IN_2[2] = 0
$MC_TRAFO_AXES_IN_2[3] = 0
$MC_TRAFO_AXES_IN_2[4] = 0

$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] = 4
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] = 0
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] = 3

$MC_TRAANG_ANGLE_2 = -85.
$MC_TRAANG_PARALLEL_VELO_RES_2 = 0.2
$MC_TRAANG_PARALLEL_ACCEL_RES_2 = 0.2

$MC_TRAANG_BASE_TOOL_2[0] = 0.0
$MC_TRAANG_BASE_TOOL_2[1] = 0.0
$MC_TRAANG_BASE_TOOL_2[2] = 0.0
```



**Teileprogramm:**

```

; Frameeinstellungen
N820 $P_UIFR[1] = ctrans(x,1,y,2,z,3,b,4,c,5)
N830 $P_UIFR[1] = $P_UIFR[1] : crot(x,10,y,20,z,30)
N840 $P_UIFR[1] = $P_UIFR[1] : cmirror(x,c)
N850
N860 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,b,40,c,15)
N870

; Werkzeuganwahl, Aufspannkompensation, Ebenenanwahl
N890 T2 D1 G54 G17 G90 F5000 G64 SOFT
N900

; Anfahren der Ausgangsstellung
N920 G0 X20 Z10
N930
N940 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,40,C,15)
N950 setal(61000)
N960 endif
N970 if $P_BFRAME <> $P_CHBFR[0]
N980 setal(61000)
N990 endif
N1000 if $P_IFRAME <>
TRANS(X,1,Y,2,Z,3,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1010 setal(61000)
N1020 endif
N1030 if $P_IFRAME <> $P_UIFR[1]
N1040 setal(61000)
N1050 endif
N1060 if $P_ACTFRAME <>
TRANS(X,11,Y,22,Z,33,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1070 setal(61000)
N1080 endif
N1090
N1100 TRAANG(,1)
N1110
N1120 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,CAX,10,B,40,C,15)
N1130 setal(61000)
N1140 endif
N1150 if $P_BFRAME <> $P_CHBFR[0]
N1160 setal(61000)

```

```
N1170 endif
N1180 if $P_IFRAME <>
CTRANS(X,1,Y,2,Z,3,CAX,1,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1190 setal(61000)
N1200 endif
N1210 if $P_IFRAME <> $P_UIFR[1]
N1220 setal(61000)
N1230 endif
N1240 if $P_ACTFRAME <>
TRANS(X,11,Y,22,Z,33,CAX,11,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1250 setal(61001)
N1260 endif
N1270
N1280
N1290 $P_UIFR[1,x,tr] = 11
N1300 $P_UIFR[1,y,tr] = 14
N1310
N1320 g54
N1330

; Frame einstellen
N1350 ROT RPL=-45
N1360 ATRANS X-2 Y10
N1370
```

```
| ; Vierkant schrappen  
N1390 G1 X10 Y-10 G41 OFFN=1; Aufmass 1mm  
N1400 X-10  
N1410 Y10  
N1420 X10  
N1430 Y-10  
N1440
```

```
| ; Werkzeugwechsel  
N1460 G0 Z20 G40 OFFN=0  
N1470 T3 D1 X15 Y-15  
N1480 Z10 G41  
N1490
```

```
| ; Vierkant schlichten  
N1510 G1 X10 Y-10  
N1520 X-10  
N1530 Y10  
N1540 X10  
N1550 Y-10  
N1560
```

```
; Frame abwaehlen
N1580 Z20 G40
N1590 TRANS
N1600
N1610 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,CAX,10,B,40,C,15)
N1620 setal(61000)
N1630 endif
N1640 if $P_BFRAME <> $P_CHBFR[0]
N1650 setal(61000)
N1660 endif
N1670 if $P_IFRAME <>
TRANS(X,11,Y,14,Z,3,CAX,1,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1680 setal(61000)
N1690 endif
N1700 if $P_IFRAME <> $P_UIFR[1]
N1710 setal(61000)
N1720 endif
N1730 if $P_ACTFRAME <>
TRANS(X,21,Y,34,Z,33,CAX,11,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1740 setal(61001)
N1750 endif
N1760
N1770 TRAFOOF
N1780
N1790 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,40,C,15)
N1800 setal(61000)
N1810 endif
N1820 if $P_BFRAME <> $P_CHBFR[0]
N1830 setal(61000)
N1840 endif
N1850 if $P_IFRAME <>
TRANS(X,1,Y,14,Z,3,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1860 setal(61000)
N1870 endif
N1880 if $P_IFRAME <> $P_UIFR[1]
N1890 setal(61000)
N1900 endif
N1910 if $P_ACTFRAME <>
TRANS(X,11,Y,34,Z,33,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1920 setal(61002)
N1930 endif
N1940
N1950 m30
```

### 9.5.6.3 Adaptionen von aktiven Frames

Die Geometrie-Achskonstellation kann sich während der Programmbearbeitung oder bei `RESET` ändern. Die Anzahl der vorhandenen Geometrieachsen können dabei von null bis drei variieren. Bei nicht-vorhandenen Geometrieachsen können Komponenten in den aktiven Frames (z.B. Drehungen) dazu führen, dass die aktiven Frames für diese Achskonstellation ungültig werden. Dies wird durch folgenden Alarm angezeigt:

Kanal %1 Satz %2 Rotation fuer nicht vorhandene Geometrieachse programmiert.

Dieser Alarm bleibt solange anstehen, bis die Frames entsprechend geändert wurden.

Mit folgendem Maschinendatum kann man die automatische Anpassung von aktiven Frames einschalten und somit den Alarm 16440 verhindern:

```
MD24040 $MC_FRAME_ADAPT_MODE
```

Bitmaske zur Anpassung der aktiven Frames bzgl. der Achskonstellation.

Es gilt:

Bit 0: Drehungen in aktiven Frames, die Koordinatenachsen verdrehen, fuer die es keine Geometrieachsen gibt, werden aus den aktiven Frames geloescht.

Bit 1: Scherungswinkel in aktiven Frames werden orthogonalisiert.

Bit 2: Skalierungen aller Geometrieachsen in den aktiven Frames werden auf den Wert 1 gesetzt.

Mit der Maschinendatum-Einstellung:

```
MD24040 $MC_FRAME_ADAPT_MODE = 1
```

werden alle Drehungen in den aktiven Frames gelöscht, die zu Koordinaten-Achsbewegungen für nicht vorhandene Geometrieachsen führen könnten.

Die Datenhaltungsframes werden dabei nicht verändert. Bei der Aktivierung von Datenhaltungsframes, werden dann auch nur die möglichen Drehungen übernommen.

Beispiel:

Es existiert keine y-Achse:

```
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
```

```
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0
```

```
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3
```

```
$P_UIFR[1] = crot(x,45,y,45,z,45)
```

```
N390 G54 G0 X10 z10 f10000
```

```
if $P_IFRAME <> crot(y,45)
```

```
; es wird nur die Drehung um y übernommen
```

```
setal(61000)
```

```
endif
```

### 9.5.6.4 Mapped Frames

#### Übersicht

Die Funktion "Mapped Frames" unterstützt die kanalübergreifende konsistente Änderung axialer Frames innerhalb kanalspezifischer oder globaler Datenhaltungsframes. In axialen Maschinendaten wird dazu festgelegt zwischen welchen Achsen das Mapping erfolgen soll.

Ist das Frame-Mapping z.B. für die Maschinenachsen AX1 und AX4 aktiv und es wird in einem kanalspezifischen Datenhaltungsframe (z.B. Basisframe \$P\_CHBFR[x]) der axiale Frame der Achse AX1 verändert (Translation, Fine-Translation, Skalierung, Spiegelung), werden diese Frame-Daten für AX1 und AX4 auf alle kanalspezifischen Datenhaltungsframes (z.B. Basisframe \$P\_CHBFR[x]) in allen Kanälen in denen sie als Kanalachsen parametrisiert sind, übertragen.

Kein Frame-Mapping erfolgt bei Änderung der axialen Frame-Daten für die Drehung.

#### Voraussetzungen

Für das Frame-Mapping müssen folgende Voraussetzungen erfüllt sein:

- Die Datenhaltungsframes die für das Mapping verwendet werden, müssen projiziert sein:  
MD28083 \$MC\_MM\_SYSTEM\_DATAFRAME\_MASK (Systemframes)
- **Kanalspezifische** Datenhaltungsframes müssen für das Mapping explizit freigegeben werden:  
MD10616 \$MN\_MAPPED\_FRAME\_MASK (Freigabe Frame-Mapping)

#### Hinweis

Bei **globalen** Datenhaltungsframes wird das Mapping immer durchgeführt. Eine Freigabe ist nicht erforderlich.

#### Parametrierung

Die Parametrierung der Mapping-Beziehungen erfolgt in den achsspezifischen Maschinendaten:

MD32075 \$MA\_MAPPED\_FRAME[<AXn>] = "AXm"

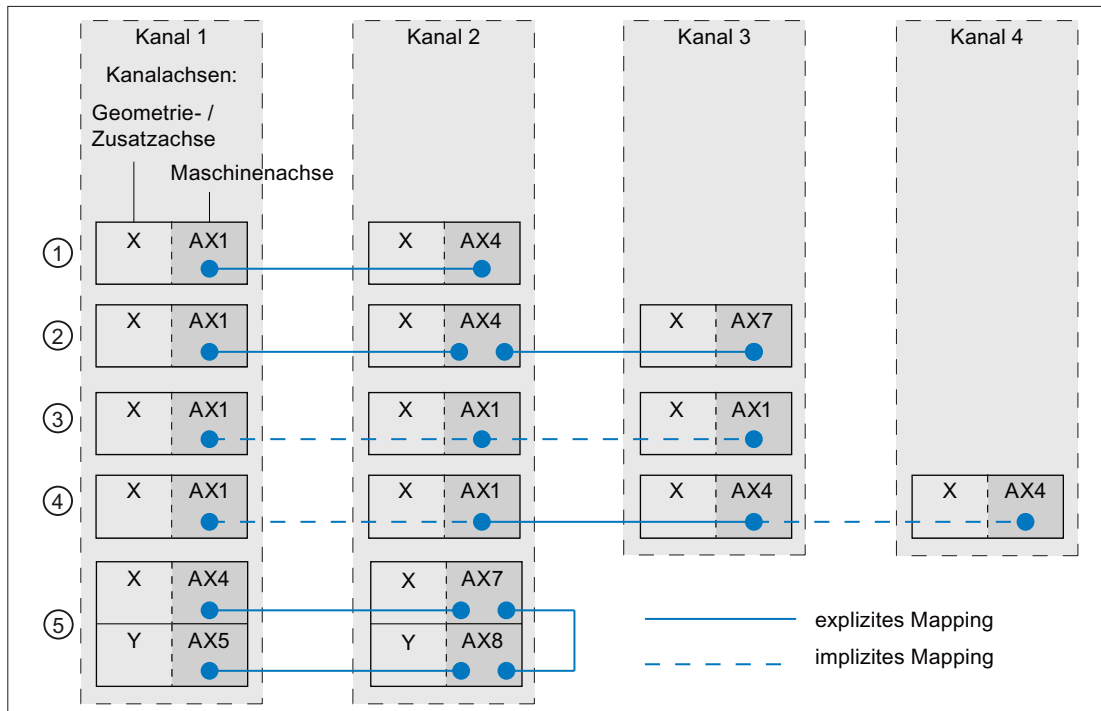
AXn, AXm: Maschinenachsbezeichner mit n, m = 1, 2, ... max. Anzahl Maschinenachsen

## Mapping-Regeln

Für das Frame-Mapping gelten folgende Regeln:

- Das Mapping ist bidirektional.  
Ein axialer Frame kann für die Achse AXn oder AXm geschrieben werden. Die Frame-Daten werden immer für die jeweils andere Achse übernommen.
- Es werden immer alle parametrisierten Mapping-Beziehungen ausgewertet.  
Beim Schreiben eines axialen Frames einer Achsen AXn werden alle Mapping-Beziehungen ausgewertet und die Frame-Daten für alle direkt und indirekt betroffenen Achsen übernommen.
- Das Mapping ist kanalglobal.  
Beim Schreiben eines axialen Frames der Achse AXn oder AXm für einen kanalspezifischen Frame, werden die Frame-Daten für alle Kanäle übernommen in denen AXn oder AXm als Kanalachsen parametrisiert sind.
- Beim Schreiben eines axialen Frames mittels Geometrie- oder Zusatzachsbezeichner, erfolgt die Auswertung der Mapping-Beziehungen über die der Geometrie- oder Zusatzachse aktuell zugeordneten Maschinenachse.
- Das Mapping ist Frame-spezifisch.  
Beim Schreiben eines axialen Frames erfolgt das Mapping der Frame-Daten nur innerhalb des gleichen kanalspezifischen oder globalen Datenhaltungsframes.

<b>ACHTUNG</b>
<b>Datenkonsistenz</b>
Es liegt ausschließlich in der Verantwortung des Anwenders/Maschinenhersteller, z.B. durch Kanalsynchronisation, dass nach dem Schreiben eines Frames in allen Kanälen konsistente Frame-Daten vorliegen.



**Beschreibung**

- ① Einfache Mapping-Beziehung:  
AX1(K1) ↔ AX4(K2)
- ② Verkettete Mapping-Beziehungen:  
AX1(K1) ↔ AX4(K2) ↔ AX7(K3)
- ③ Mapping-Beziehung auf sich selbst, mit AX1 als Kanalachse von Kanal 1, 2 und 3:  
AX1(K1+K2+K3)
- ④ Mapping-Beziehung zwischen zwei Achsen, die Kanalachsen in jeweils zwei Kanälen sind:  
AX1(K1+K2) ↔ AX4(K3+K4)
- ⑤ Verkettete Mapping-Beziehungen bei der mehrere Kanalachsen im gleichen Kanal geschrieben werden:  
AX4(K1) ↔ AX7(K2) ↔ AX8(K2) ↔ AX5(K1)

**Parametrierung: \$MA\_**

- MAPPED\_FRAME[<AX1>] = "AX4"
- MAPPED\_FRAME[<AX1>] = "AX4"  
MAPPED\_FRAME[<AX4>] = "AX7"
- MAPPED\_FRAME[<AX1>] = "AX1"
- MAPPED\_FRAME[<AX1>] = "AX4"
- MAPPED\_FRAME[<AX4>] = "AX7"  
MAPPED\_FRAME[<AX7>] = "AX8"  
MAPPED\_FRAME[<AX8>] = "AX5"

Bild 9-23 Mapping Beispiele



## Aktivieren der Datenhaltungsframes

Die Datenhaltungsframes können im Teileprogramm und über die Bedienoberfläche von SINUMERIK Operate geschrieben werden. Bei der Aktivierung der direkt und über Frame-Mapping geschriebenen Datenhaltungsframes in den Kanälen ist folgendes zu beachten:

- Schreiben im Teileprogramm  
Die Datenhaltungsframes müssen in jedem Kanal explizit (G500, G54 ... G599) aktiviert werden
- Schreiben über Bedienoberfläche  
Datenhaltungsframes werden über die Bedienoberfläche z.B. durch die Eingabe neuer Nullpunktverschiebungen geschrieben. Ein veränderter Datenhaltungsframe wird in allen betroffenen Kanälen sofort aktiv, wenn sich keiner dieser Kanäle im Zustand "Kanal aktiv" befindet. Ist einer der betroffenen Kanäle im Zustand "Kanal aktiv", wird der Datenhaltungsframe in keinem Kanal aktiv. Die Aktivierung muss dann in jedem Kanal explizit im Teileprogramm (G500, G54 ... G599) programmiert werden. Oder er wird mit dem nächsten Wechsel des Kanalzustandes nach "Kanal-Reset" aktiv.

## Beispiel

An einer Steuerung sind folgende Kanäle und Kanalachsen parametriert:

- Kanal 1
  - Z: Geometrieachse
  - AX1: Maschinenachse der Geometrieachse Z
- Kanal 2
  - Z: Geometrieachse
  - AX4: Maschinenachse der Geometrieachse Z

Der Nullpunkt der Z-Achse soll in beiden Kanäle immer gleich sein:

- Mapping-Beziehung: \$MA\_MAPPED\_FRAME[AX1] = "AX4"

**Programmierung im Teileprogramm**

```

Kanal 1
...
N100 WAIT (10,1,2)
N110 $P_UIFR[1] = CTRANS(Z, 10)
N120 WAIT (20,1,2)
N130 G54
N140 IF ($P_IFRAME[0, Z, TR] <> 10)
N150   SETAL(61000)
N160 ENDIF
...

Kanal 2
...
N200 WAIT (10,1,2)
N220 WAIT (20,1,2)
N230 G54
N230 IF ($P_IFRAME[0, Z, TR] <> 10)
N250   SETAL(61000)
N260 ENDIF
...
    
```

**Beschreibung:**

- N100 / N200 Kanalsynchronisation für konsistentes Schreiben und Mapping der Frame-Daten
- N110 Schreiben des einstellbaren Datenhaltungs-Frames \$P\_UIFR[1]:  
Verschieben des Nullpunktes der Z-Achse auf 10 mm  
Mapping der axialen Frame-Daten:  
Kanal1: Z  $\triangleq$  AX1  $\leftrightarrow$  Kanal2: Z  $\triangleq$  AX4
- N120 / N220 Kanalsynchronisation für konsistentes Aktivieren der neuen Frame-Daten
- N130 / N230 Aktivieren der neuen Frame-Daten
- N140 / N240 Überprüfung des Nullpunktes der Z-Achse auf: 10 mm

**9.5.7 Vordefinierte Frame-Funktionen**

**9.5.7.1 Inverses Frame**

Zur Abrundung der Frame-Arithmetik im Teileprogramm wird eine Funktion zur Verfügung gestellt, die aus einem Frame den inversen Frame berechnet. Die Frame-Verkettung eines Frames mit seinem inversen Frame ergibt immer einen Nullframe.

```
FRAME INVFRAME( FRAME )
```

Die Frameinvertierung ist ein Hilfsmittel für die Koordinatentransformationen. Die Berechnung von Messframes erfolgt meist im WKS. Möchte man dieses berechnete Frame in ein anderes Koordinatensystem transformieren, d.h. das berechnete Frame soll in einen beliebigen Frame innerhalb der Framekette eingetragen werden, so ist dies mit folgenden Berechnungen möglich:

Das neue Gesamtframe ergibt sich als Verkettung des alten Gesamtframes mit dem berechneten Frame.

```
$P_ACTFRAME = $P_ACTFRAME : $AC_MEAS_FRAME
```

Das neue Frame in der Framekette ergibt sich danach:

Zielframe ist \$P\_SETFRAME:

$$\begin{aligned} \$P\_SETFRAME &= \$P\_ACTFRAME : \$AC\_MEAS\_FRAME : INVFRAME(\$P\_ACTFRAME) : \\ & \$P\_SETFRAME \end{aligned}$$

Zielframe ist n-tes Kanalbasisframe \$P\_CHBFRAME[n]:

$$\begin{aligned} n = 0: & \text{TMP} = \$P\_PARTFRAME : \$P\_SETFRAME : \$P\_EXTFRAME : \$P\_NCBFRAME[0..k] \\ n \neq 0: & \text{TMP} = \$P\_PARTFRAME : \$P\_SETFRAME : \$P\_EXTFRAME : \$P\_NCBFRAME[0..k] \\ & : \$P\_CHBFRAME[0..n-1] \end{aligned}$$

$$k = \$MN\_MM\_NUM\_GLOBAL\_BASE\_FRAMES$$

$$\begin{aligned} \$P\_CHBFRAME[n] &= INVFRAME(\text{TMP}) : \$P\_ACTFRAME : \$AC\_MEAS\_FRAME : \\ & INVFRAME(\$P\_ACTFRAME) : \text{TMP} : \$P\_CHBFRAME[n] \end{aligned}$$

Zielframe ist \$P\_IFRAME:

$$\begin{aligned} \text{TMP} &= \$P\_PARTFRAME : \$P\_SETFRAME : \$P\_EXTFRAME : \$P\_BFRAME \\ \$P\_IFRAME &= INVFRAME(\text{TMP}) : \$P\_ACTFRAME : \$AC\_MEAS\_FRAME : \\ & INVFRAME(\$P\_ACTFRAME) : \text{TMP} : \$P\_IFRAME \end{aligned}$$

Anwendungsbeispiel:

Ein Frame, das z.B. über eine Messfunktion ermittelt wurde, soll im aktuellen SETFRAME so eingetragen werden, dass das neue Gesamtframe sich als Verkettung des alten Gesamtframes mit dem Mess-Frame ergibt. Das SETFRAME wird mit Hilfe von Frame-Invertierungen entsprechend umgerechnet.

```
DEF INT RETVAL
```

```
DEF FRAME TMP
```

```
$TC_DP1[1,1]=120 ; Typ
```

```
$TC_DP2[1,1]=20.;0
```

```
$TC_DP3[1,1]= 10. ; (z) Laengenkorrekturvektor
```

```
$TC_DP4[1,1]= 0. ; (y)
```

```
$TC_DP5[1,1]= 0. ; (x)
```

```
$TC_DP6[1,1]= 2. ; Radius
```

```
T1 D1
```

```
g0 x0 y0 z0 f10000
```

```
G54
```

```
$P_CHBFRAME[0] = crot(z,45)
```

```
$P_IFRAME[x,tr] = -sin(45)
$P_IFRAME[y,tr] = -sin(45)

$P_PFRAME[z,rt] = -45

; Ecke mit 4 Messpunkten vermessen
$AC_MEAS_VALID = 0

; 1. Messpunkt anfahren
g1 x-1 y-3

; 1. Messpunkt abspeichern
$AC_MEAS_LATCH[0] = 1

; 2. Messpunkt anfahren
g1 x5 y-3

; 2. Messpunkt abspeichern
$AC_MEAS_LATCH[1] = 1

; 3. Messpunkt anfahren
g1 x-4 y4

; 3. Messpunkt abspeichern
$AC_MEAS_LATCH[2] = 1

; 4. Messpunkt anfahren
g1 x-4 y1

; 4. Messpunkt abspeichern
$AC_MEAS_LATCH[3] = 1
```

```
; Sollposition der Ecke setzen
$AA_MEAS_SETPOINT[x] = 0
$AA_MEAS_SETPOINT[y] = 0
$AA_MEAS_SETPOINT[z] = 0

; Sollschnittwinkel vorgeben
$AC_MEAS_CORNER_SETANGLE = 90
$AC_MEAS_WP_SETANGLE = 30

; Ebene fuer die Messung ist G17
$AC_MEAS_ACT_PLANE = 0

; Werkzeug auswaehlen
$AC_MEAS_T_NUMBER = 1
$AC_MEAS_D_NUMBER = 1

; Messtype auf Ecke 1 setzen
$AC_MEAS_TYPE = 4

; Messvorgang starten
RETVAl = MEASURE()

if RETVAL <> 0
  setal(61000 + RETVAL)
endif

if $AC_MEAS_WP_ANGLE <> 30
  setal(61000 + $AC_MEAS_WP_ANGLE)
endif
```

### 9.5 Frames

```
if $AC_MEAS_CORNER_ANGLE <> 90
setal(61000 + $AC_MEAS_CORNER_ANGLE)
endif

; Gemessenes Frame so transformieren und nach $P_SETFRAME schreiben,
; dass ein Gesamtframe entsteht, welches aus dem alten Gesamtframe
; verkettet mit dem Messframe ergibt.

$P_SETFRAME = $P_ACTFRAME : $AC_MEAS_FRAME : INVFRAME($P_ACTFRAME) :
$P_SETFRAME

; Systemframe in der Datenhaltung beschreiben
$P_SETFR = $P_SETFRAME

; Fahre die Ecke an
g1 x0 y0

; Um 30 Grad gedrehtes Rechteck abfahren
g1 x10
y10
x0
y0

m30
```

#### 9.5.7.2 Additives Frame in der Framekette

Durch Messungen am Werkstück oder durch Berechnungen im Teileprogramm und Zyklen ergibt sich meist ein Frame, das additiv zum aktuellen Gesamtframe wirken sollen. Das WKS und damit der Nullpunkt der Programmierung soll also verschoben und evtl. gedreht werden. Dieses gemessene Frame liegt als temporäres Frame vor und ist noch nicht aktiv in der Framekette enthalten. Zur Einrechnung und evtl. Aktivierung dieses Frames dient folgende Funktion:

```
INT ADDFRAME (FRAME, STRING)
```

## Programmierung

Parameter 1:      Typ: `FRAME`      Additives gemessenes oder berechnetes Frame

Parameter 2:      Typ: `STRING`      Strings für aktuelle Frames:  
`"$P_CYCFRAME"`, `"$P_ISO4FRAME"`,  
`"$P_PFRAME"`, `"$P_WPFRAME"`,  
`"$P_TOOLFRAME"`, `"$P_IFRAME"`,  
`"$P_CHBFRAME[0..16]"`, `"$P_NCBFRAME[0..16]"`,  
`"$P_ISO1FRAME"`, `"$P_ISO2FRAME"`,  
`"$P_ISO3FRAME"`,  
`"$P_EXTFRAME"`, `"$P_SETFRAME"`  
`"$P_PARTFRAME"`

Strings für Datenhaltungs-Frames:  
`"$P_CYCFR"`, `"$P_ISO4FR"`, `"$P_TRAFR"`,  
`"$P_WPFR"`,  
`"$P_TOOLFR"`, `"$P_UIFR[0..99]"`,  
`"$P_CHBFR[0..16]"`, `"$P_NCBFR[0..16]"`,  
`"$P_ISO1FR"`, `"$P_ISO2FR"`, `"$P_ISO3FR"`,  
`"$P_EXTFR"`, `"$P_SETFR"`, `"$P_PARTFR"`

Funktionswert:	Typ: <code>INT</code>	Wert:	Bedeutung:
		0	OK
		1	Zielangabe (String) ist falsch
		2	Zielframe ist nicht projiziert
		3	Drehung im Frame ist nicht erlaubt

Die Funktion `ADDFRAME()` berechnet den Zielframe, der durch den `STRING` spezifiziert ist. Der Zielframe wird so berechnet, dass sich das neue Gesamtframe als Verkettung des alten Gesamtframes mit dem übergebenen Frame ergibt, z. B.:

```
ERG = ADDFRAME(TMPFRAME, "$P_SETFRAME")
```

Das neue Gesamtframe ergibt sich als:

$$\$P\_ACTFRAME_{\text{neu}} = \$P\_ACTFRAME_{\text{alt}} : \text{TMPFRAME}$$

Wurde als Zielframe ein aktuelles Frame spezifiziert, so wird das neue Gesamtframe im Vorlauf aktiv. Ist das Zielframe ein Datenhaltungs-Frame, so wird das Frame erst aktiv, wenn es im Teileprogramm explizit aktiviert wird.

Die Funktion setzt keine Alarmer, sondern gibt die Fehlercodes über den Returnwert zurück. Der Zyklus kann entsprechend der Fehlercodes reagieren.

## 9.5.8 Funktionen

### 9.5.8.1 Setzen von Nullpunkten, Werkstück- und Werkzeugvermessung

Das Istwertsetzen erfolgt über die HMI-Bedienung oder über Messzyklen. Das berechnete Frame wird in das Systemframe `SETFRAME` geschrieben werden. Beim Istwertsetzen kann die Sollposition einer Achse im WKS geändert werden.

Unter dem Begriff "Ankratzen" verstehen wir Werkstück- und Werkzeugvermessung. Bei der Werkstückvermessung kann die Lage des Werkstückes bzgl. einer Kante, einer Ecke oder einer Bohrung vermessen werden. Zur Festlegung der Nulllage des Werkstückes oder der Bohrung können die gemessenen Positionen dann mit Sollpositionen im WKS beaufschlagt werden. Die resultierenden Verschiebungen können dabei in ein ausgewähltes Frame eingetragen werden. Bei der Werkzeugvermessung kann die Länge oder der Radius eines Werkzeuges anhand eines vermessenen Referenzteiles gemessen werden.

Die Messungen können über die Bedienung oder über Messzyklen erfolgen. Als Kommunikation mit dem NCK dienen vordefinierte Systemvariablen. Die Berechnung erfolgt im NCK durch Aktivierung eines PI-Dienstes über die HMI-Bedienung oder über einen Teileprogrammbehehl aus den Messzyklen. Als Grundlage für die Berechnung kann ein Werkzeug und eine Ebene ausgewählt werden. Der berechnete Frame wird in das Ergebnis-Frame eingetragen.

### 9.5.8.2 Externe Nullpunktverschiebung über Systemframe

#### Nullpunktverschiebung über PLC oder Teileprogramm

Der Betrag für die externe Nullpunktverschiebung kann durch HMI und PLC über BTSS vorgegeben oder im Teileprogramm über die achsspezifische Systemvariable `$AA_ETRANS` [`<Achse>`] = `<Wert>` programmiert werden.

#### **Aktivierung**

Die Aktivierung der externen Nullpunktverschiebung erfolgt über das Nahtstellensignal: DB31, ... DBX3.0 (Externe Nullpunktverschiebung übernehmen)

#### **Verhalten**

Nach der Aktivierung wird der jeweilige achsspezifisch vorgegebene Betrag der externen Nullpunktverschiebung für jede Achse mit dem nächst möglichen Verfahrsatz herausgefahren.

Nächst möglich bedeutet, dass für die jeweilige Achse genügend Dynamikreserve zum Herausfahren der externen Nullpunktverschiebung vorhanden sein muss. Wird die Achse im nächsten Verfahrsatz nach der Aktivierung aufgrund ihrer Programmierung bereits mit maximaler Geschwindigkeit verfahren, ist keine Dynamikreserve zum Herausfahren der externen Nullpunktverschiebung mehr vorhanden.

Im Zusammenhang mit Bahnsteuerbetrieb `G64` kann sich das Herausfahren der Verschiebung auch über mehrere Teileprogrammsätze erstrecken.



## Nullpunktverschiebung über Systemframe

Die externe Nullpunktverschiebung kann alternativ zur oben beschriebenen Funktionalität auch über das Systemframe \$P\_EXTFRAME verwaltet und aktiviert werden.

### Projektierung

Die Projektierung der externe Nullpunktverschiebung über das Systemframe \$P\_EXTFRAME erfolgt über Bit1 = 1 im Maschinendatum: MD28082  
\$MC\_MM\_SYSTEM\_FRAME\_MASK = 'B0010'

Der Betrag für die externe Nullpunktverschiebung kann manuell über die HMI-Bedienoberfläche und das PLC-Anwenderprogramm über BTSS vorgegeben oder im Teileprogramm über die achsiale Systemvariable \$AA\_ETRANS[<Achse>] programmiert werden.

### Aktivierung

Die Aktivierung der externen Nullpunktverschiebung erfolgt über das Nahtstellensignal: DB31, ... DBX3.0 (Externe Nullpunktverschiebung übernehmen)

### Verhalten

Bei Aktivierung der externen Nullpunktverschiebung werden die Verfahrbewegungen aller Achsen, außer Kommando- und PLC-Achsen, sofort gestoppt und der Vorlauf reorganisiert. Die Grobverschiebung des aktuellen Systemframe und des Systemframe in der Datenhaltung wird auf den Wert der achsialen Systemvariablen \$AA\_ETRANS[<Achse>] gesetzt. Anschließend wird zuerst die Verschiebung herausgefahren und dann die unterbrochene Bewegung fortgesetzt.

### Verhalten bei Kettenmaßangabe

Bei aktiver Kettenmaßangabe <sup>G91</sup> und Maschinendatum: MD42440 \$MC\_FRAME\_OFFSET\_INCR\_PROG (Nullpunktverschiebungen in Frames) = 0 wird im Rahmen der externe Nullpunktverschiebung über Systemframe die Verschiebung, trotz gegenteiliger Projektierung des Maschinendatums, mit dem Anfahrsatz herausgefahren, obwohl sie durch einen Frame vorgegeben wird.

---

### Hinweis

Die externe Nullpunktverschiebung wirkt immer absolut.

---

### 9.5.8.3 Werkzeugträger

#### Translationen

Bei Kinematiken vom Typ "P" und "M" wird bei der Anwahl eines Werkzeugträgers ein additiver Frame aktiviert (Tischoffset des orientierbaren Werkzeugträgers), der die Verschiebung des Nullpunktes als Folge der Drehung des Tisches berücksichtigt. Die Nullpunktverschiebung wird in das Systemframe `$P_PARTFR` eingetragen. Dabei wird der translatorische Anteil dieses Frames überschrieben. Die anderen Framekomponenten bleiben erhalten.

Das Systemframe `$P_PARTFR` muss über folgendes Maschinendatum freigegeben werden: MD28082 `$MC_MM_SYSTEM_FRAME_MASK`, Bit 2 = 1 (Systemframe für TCARR und PAROT)

---

#### Hinweis

Alternativ kann die Verschiebung auch über das Maschinendatum zur Aufnahme des Tischoffsets parametrisiert werden:

MD20184 `$MC_TOCARR_BASE_FRAME_NUMBER` = <Nummer des Basisframes>

Diese Möglichkeit besteht nur aus Kompatibilitätsgründen zu älteren Softwareständen. Es wird dringend empfohlen diese Vorgehensweise **nicht** mehr zu verwenden.

---

Eine Frameverschiebung als Folge eines Werkzeugträgerwechsels wird sofort bei der Anwahl mit `TCARR=...` wirksam. Dagegen wird eine Änderung der Werkzeuglänge nur dann sofort wirksam, wenn ein Werkzeug aktiv ist.

Eine Framedrehung wird mit der Aktivierung nicht bewirkt, bzw. eine bereits wirksame Drehung wird nicht verändert. Die Stellung der Rundachsen, die für die Berechnung verwendet wird, wird wie im Fall T (nur das Werkzeug ist drehbar) abhängig vom Befehl `TCOFR / TCOABS` aus dem Drehanteil eines aktiven Frames bzw. aus den Einträgen `$TC_CARRn` ermittelt. Durch die Aktivierung eines Frames verändert sich die Position im Werkstückkoordinatensystem entsprechend, ohne dass es zu einer Ausgleichsbewegung der Maschine kommt.

Die Verhältnisse sind im folgenden Bild dargestellt:

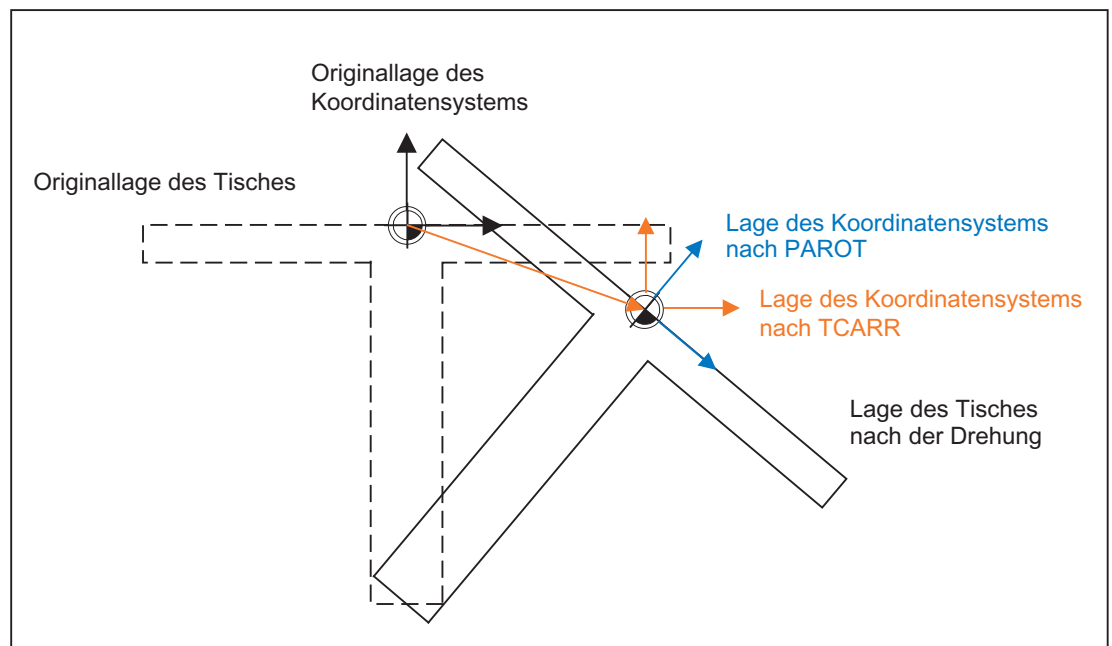


Bild 9-24 Frame bei Aktivierung eines drehbaren Werkzeugtisches mit TCARR

Bei Kinematiken des Typs M (Werkzeug und Tisch sind jeweils um eine Achse drehbar), bewirkt die Aktivierung eines Werkzeugträgers mit  $TCARR$  gleichzeitig eine entsprechende Änderung der effektiven Werkzeuglänge (falls ein Werkzeug aktiv ist) und der Nullpunktverschiebung.

## Drehungen

Je nach Bearbeitungsaufgabe muss beim Einsatz drehbarer Werkzeugträger bzw. Tische nicht nur eine Nullpunktverschiebung (sei es als Frame oder als Werkzeuglänge), sondern auch eine Drehung berücksichtigt werden. Die Aktivierung eines orientierbaren Werkzeugträgers führt jedoch in keinem Fall unmittelbar zu einer Drehung des Koordinatensystems.

Ist nur das Werkzeug drehbar, kann dafür ein Frame mittels  $TOFRAME$  bzw.  $TOROT$  usw. definiert werden.

Bei drehbaren Tischen (Kinematiktypen P und M) hat die Aktivierung mit  $TCARR$  zunächst ebenfalls keine Drehung des Koordinatensystems zur Folge, d.h. der Nullpunkt des Koordinatensystems verschiebt sich zwar in Bezug auf die Maschine und bleibt fest in Bezug auf den Nullpunkt des Werkstücks, die Orientierung bleibt jedoch unverändert raumfest.

Wird ein Koordinatensystem benötigt, das in Bezug auf das Werkstück fest ist, d.h. gegenüber der Originallage nicht nur verschoben sondern auch entsprechend der Tischdrehung gedreht ist, so kann analog zur Situation bei drehbarem Werkzeug mit  $PAROT$  eine entsprechende Drehung aktiviert werden.

Mit `PAROT` bleiben Translationen, Skalierungen und Spiegelungen im aktiven Frame erhalten, der Rotationsanteil wird jedoch durch den Rotationsanteil eines orientierbaren Werkzeugträgers, der dem Tisch entspricht gedreht. Dabei bleibt der gesamte programmierbare Frame, einschließlich seines Drehanteils, unverändert.

Der Rotationsanteil, der die Drehung des Werkzeigtisches beschreibt, wird dann entweder in den Systemframe `$PARTFR` oder in den durch MD20184

`$MC_TOCARR_BASE_FRAME_NUMBER` parametrisierten Basisframe eingetragen:

`$MC_MM_SYSTEM_FRAME_MASK`, Bit 2 = <Wert>

Wert	Bedeutung
1	Rotationsanteil → <code>\$PARTFR</code>
0	Rotationsanteil → MD20184 <code>\$MC_TOCARR_BASE_FRAME_NUMBER</code>

Entsprechend dem Hinweis bei der Beschreibung der Tischverschiebung gilt auch hier, dass empfohlen wird, die zweite Alternative für Neuanlagen nicht mehr zu verwenden.

Der Rotationsanteil des Partframes kann mit `PAROTOF` gelöscht werden, unabhängig davon, ob dieser Frame in einem Basis- oder in einem Systemframe steht.

Der Translationsanteil wird gelöscht, indem ein Werkzeugträger aktiviert wird, der keine Verschiebung bewirkt, bzw. durch Abwahl eines eventuell aktiven orientierbaren Werkzeugträgers mit `TCARR=0`.

`PAROT` bzw. `TOROT` berücksichtigen in den Fällen, in denen der Tisch bzw. das Werkzeug mit zwei Drehachsen orientiert werden, die gesamte Orientierungsänderung. Bei gemischten Kinematiken wird jeweils nur der entsprechende, durch eine Drehachse verursachte Anteil berücksichtigt. Damit ist es z. B. bei der Verwendung von `TOROT` möglich, ein Werkstück so zu drehen, dass eine schräge Ebene parallel zur raumfesten X-Y-Ebene liegt, bei der Bearbeitung aber eine Drehung des Werkzeugs zu berücksichtigen, wenn beispielsweise Bohrungen angebracht werden müssen, die nicht senkrecht zu dieser Ebene sind.

### Beispiel

Bei einer Maschine zeige die Drehachse des Tisches in die positive Y-Richtung. Der Tisch ist um +45 Grad gedreht. Mit `PAROT` wird dann ein Frame definiert, der ebenfalls eine Drehung von 45 Grad um die Y-Achse beschreibt. Das gegenüber der Außenwelt nicht gedrehte Koordinatensystem (im Bild mit "Lage des Koordinatensystems nach TCARR" gekennzeichnet) ist gegenüber dem mitgeführten Koordinatensystem (Lage nach `PAROT`) aber um -45 Grad gedreht. Wird dieses Koordinatensystem z. B. mit `ROT Y-45` definiert und anschließend der Werkzeugträger bei aktivem `TCOFR` angewählt, wird für die Drehachse des Werkzeugträgers ein Winkel von +45 Grad ermittelt.

Der Sprachbefehl `PAROT` wird nicht abgelehnt, wenn kein orientierbarer Werkzeugträger aktiv ist. Allerdings bewirkt ein solcher Aufruf dann keine Frameänderung.

## Bearbeitung in Richtung der Werkzeugorientierung

Vor allem an Maschinen mit orientierbarem Werkzeug soll mitunter in Werkzeugrichtung verfahren werden (typischerweise beim Bohren), ohne dass ein Frame aktiviert wird (z. B. mittels `TOFRAME` oder `TOROT`), bei dem eine der Achsen in Richtung des Werkzeugs zeigt. Das Problem stellt sich auch dann, wenn bei einer Schrägbearbeitung ein Frame aktiv ist, der die schräge Ebene definiert, das Werkzeug aber nicht exakt senkrecht eingestellt werden kann, weil wegen eines indexierten Werkzeugträgers (Hirth-Verzahnung) die Werkzeugorientierung nicht beliebig einstellbar ist. In diesen Fällen muss dann - abweichend von der eigentlich verlangten Bewegung senkrecht zur Ebene - in Werkzeugrichtung gebohrt werden, da andernfalls der Bohrer nicht in Richtung seiner Längsachse geführt würde (Werkzeugbruch).

Der Endpunkt einer solchen Bewegung wird mit `MOVT=...` programmiert.

Der programmierte Wert wirkt standardmäßig inkrementell in Werkzeugrichtung.

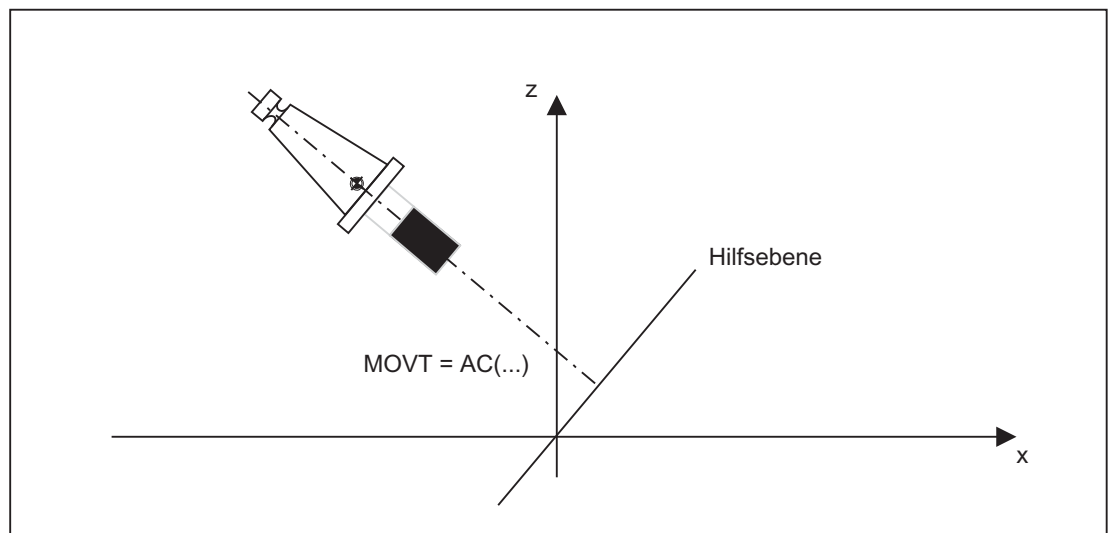
Die positive Richtung ist dabei von der Werkzeugspitze zur Werkzeugaufnahme definiert.

Der Inhalt von `MOVT` ist bei der Zustellbewegung (beim Bohren) deshalb in der Regel negativ, bei der Rückzugsbewegung dagegen positiv. Das entspricht den Verhältnissen bei normaler achsparalleler Bearbeitung z. B. mit `G91 Z...`

Anstelle von `MOVT=...` kann auch `MOVT=IC(...)` geschrieben werden, wenn deutlich sichtbar zum Ausdruck gebracht werden soll, dass `MOVT` inkrementell wirkt. Ein funktioneller Unterschied zwischen beiden Schreibweisen besteht nicht.

Wird die Bewegung in der Form `MOVT=AC(...)` programmiert, wirkt `MOVT` absolut.

Dazu wird eine Ebene definiert, die durch den aktuellen Nullpunkt verläuft, und deren Flächennormalenvektor parallel zur Werkzeugorientierung ist. `MOVT` gibt dann die Lage bezüglich dieser Ebene an (siehe Bild). Die Bezugsebene dient nur zur Berechnung der Endposition. Aktive Frames werden durch diese interne Berechnung nicht beeinflusst.



Die Programmierung mit `MOVT` ist unabhängig vom Vorhandensein eines orientierbaren Werkzeugträgers. Die Richtung der Bewegung ist von der aktiven Ebene abhängig. Sie verläuft in Richtung der Applikate, d.h. bei `G17` in Z-Richtung, bei `G18` in Y-Richtung und bei `G19` in X-Richtung. Dies gilt sowohl für den Fall, dass kein orientierbarer Werkzeugträger aktiv ist, als auch für den Fall eines orientierbaren Werkzeugträgers ohne drehbares Werkzeug oder mit drehbarem Werkzeug in Grundstellung.

MOV<sub>T</sub> wirkt bei aktiver Orientierungstransformation (3-4-5-Achs-Transformation) in gleicher Weise.

Wird in einem Satz mit MOV<sub>T</sub> gleichzeitig die Werkzeugorientierung verändert (z. B. bei aktiver 5-Achstransformation durch gleichzeitige Interpolation der Rundachsen), ist die Orientierung am Satzanfang für die Bewegungsrichtung von MOV<sub>T</sub> maßgebend.

Bei aktiver 5-Achstransformation wird die Bahn des "tool center points" (TCP) durch die Orientierungsänderung nicht beeinflusst, d. h. die Bahn bleibt eine Gerade, deren Richtung durch die Werkzeugorientierung im Satzangang bestimmt ist.

Wird MOV<sub>T</sub> programmiert, muss Linear- oder Spline-Interpolation (G<sub>0</sub>, G<sub>1</sub>, ASPLINE, BSPLINE, CSPLINE) aktiv sein. Andernfalls wird ein Alarm ausgegeben.

Ist eine Spline-Interpolationsart aktiv, ist die resultierende Bahn im allgemeinen keine Gerade, da der von MOV<sub>T</sub> ermittelte Endpunkt so behandelt wird, als sei er explizit mit X, Y, Z programmiert worden.

In einem Satz mit MOV<sub>T</sub> dürfen keine Geometrieachsen programmiert sein (Alarm 14157).

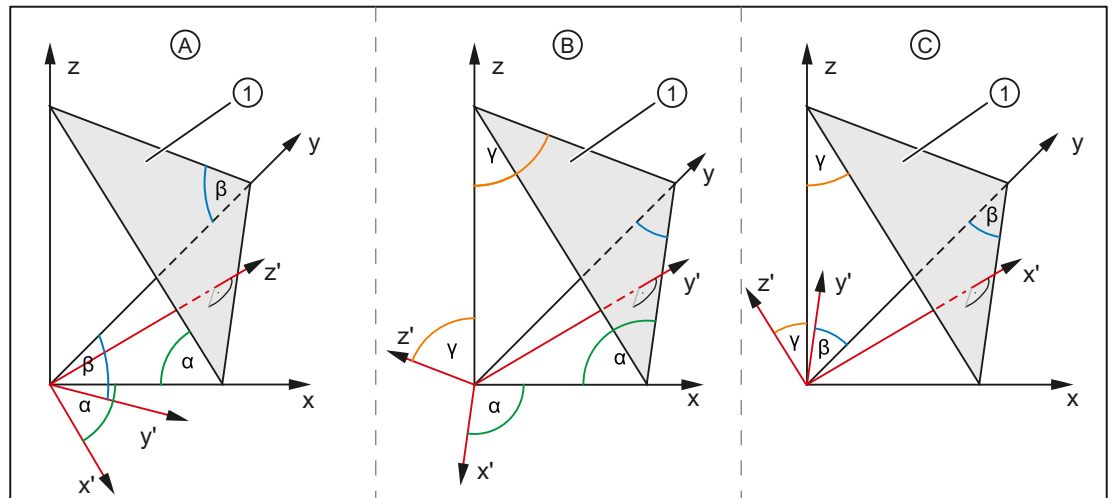
### Definition von Framedrehungen mit Raumwinkeln

Soll ein Frame, der eine Drehung um mehr als eine Achse beschreibt, definiert werden, so geschieht das durch die Verkettung von Einzeldrehungen. Dabei erfolgt eine neue Drehung immer in dem bereits gedrehten Koordinatensystem.

Das gilt sowohl bei Programmierung in einem Satz z. B. mit ROT X... Y... Z..., als auch beim Aufbau eines Frames in mehreren Sätzen, z. B. in der Form:

```
N10 ROT Y...  
N20 AROT X...  
N30 AROT Z...
```

In Werkstückzeichnungen werden zur Beschreibung schräger Flächen jedoch oft die Raumwinkel angegeben, d.h. die Winkel die die Schnittgeraden der schrägen Ebene mit den Hauptebenen (X-Y, Y-Z, Z-X-Ebene) bilden (siehe Bild). Die Umrechnung dieser Raumwinkel in die Drehwinkel einer Verkettung von Einzeldrehungen ist dem Maschinenbediener nicht zuzumuten.



① Schräge Ebene

Bild 9-25 Drehung mit Raumwinkeln

Es werden deshalb die Sprachbefehle `ROTS`, `AROTS` und `CROTS` eingeführt, mit denen die Drehungen unmittelbar als Raumwinkel beschrieben werden können.

Die Orientierung einer Ebene im Raum ist durch die Angabe zweier Raumwinkel eindeutig bestimmt. Der dritte Raumwinkel ergibt sich aus den beiden ersten. Es dürfen deshalb maximal 2 Raumwinkel programmiert werden, z. B. in der Form `ROTS X10 Y15`. Wird ein dritter Raumwinkel angegeben, wird ein Alarm ausgegeben.

Die Angabe eines einzelnen Raumwinkels ist zulässig. Die Drehungen, die in diesem Fall mit `ROTS` bzw. `AROTS` ausgeführt werden, sind identisch zu denen bei `ROT` bzw. `AROT`.

Eine Erweiterung gegenüber der bisherigen Funktionalität gibt es deshalb nur für den Fall, dass genau zwei Raumwinkel programmiert werden.

Die beiden programmierten Achsen definieren eine Ebene, die jeweils nicht programmierte Achse definiert die zugehörige dritte Achse eines rechtshändigen Koordinatensystems. Damit ist für die beiden programmierten Achsen auch eindeutig definiert, welches die erste und welches die zweite Achse ist (die Definition entspricht den Festlegungen der Ebenendefinition bei `G17/G18/G19`). Der mit dem Achsbuchstaben einer Achse der Ebene programmierte Winkel gibt dann die Achse an, um die man die andere Achse der Ebene drehen muss, um diese in die Schnittgerade zu überführen, die die gedrehte Ebene mit der aus der anderen und der dritten Achse aufgespannten Ebene bildet. Mit dieser Definition ist gewährleistet, dass für den Fall, dass einer der beiden programmierten Winkel gegen Null geht, die derart definierte Ebene in die Ebene übergeht, die sich ergibt, wenn nur eine Achse (z. B. auch mit `ROT` oder `AROT`) programmiert ist.

Im Bild ist als Beispiel der Fall dargestellt, dass X und Y programmiert sind. Hier gibt Y den Winkel an, um den man die X-Achse um die Y-Achse drehen muss, damit die X-Achse in die Schnittgerade überführt wird, die die schräge Ebene mit der X-Z-Ebene bildet. Entsprechendes gilt für den programmierten Wert von X.

---

#### Hinweis

Bei der dargestellten Lage der schrägen Ebene ist der Wert von Y positiv, der von X dagegen negativ.

---

Mit der Angabe der Raumwinkel ist die Orientierung des zweidimensionalen Koordinatensystems innerhalb der Ebene (d.h. der Drehwinkel um den Flächennormalenvektor) nicht definiert. Die Lage des Koordinatensystems wird deshalb so festgelegt, dass die gedrehte erste Achse in der Ebene liegt, die von der ersten und dritten Achse des nicht gedrehten Koordinatensystems aufgespannt wird.

Das bedeutet:

- Bei Programmierung von X und Y liegt die neue X-Achse in der alten Z-X-Ebene.
- Bei Programmierung von Z und X liegt die neue Z-Achse in der alten Y-Z-Ebene.
- Bei Programmierung von Y und Z liegt die neue Y-Achse in der alten X-Y-Ebene.

Wird eine von dieser Voreinstellung abweichende Lage des Koordinatensystems benötigt, muss eine zusätzliche Drehung mit `AROT...` ausgeführt werden.

Die programmierten Raumwinkel werden bei der Eingabe abhängig vom Maschinendatum: MD10600 \$MN\_FRAME\_ANGLE\_INPUT\_MODE in die äquivalenten Euler-Winkel nach zy'x"-Konvention (RPY-Winkel) oder zx'z"-Konvention umgerechnet. Diese erscheinen auch in der Anzeige.

### Framedrehung in Werkzeugrichtung

Mit dem bereits in älteren Softwareständen vorhandenen Sprachbefehl `TOFRAME` besteht die Möglichkeit, einen Frame zu definieren, dessen Z-Achse in Werkzeugrichtung zeigt. Ein vorhandener programmierter Frame wird dabei durch einen Frame überschrieben, der eine reine Drehung beschreibt. Im vorher aktiven Frame eventuell vorhandene Nullpunktverschiebungen, Spiegelungen oder Skalierungen werden gelöscht. Mitunter ist dieses Verhalten störend. Insbesondere ist es oft sinnvoll, eine Nullpunktverschiebung, mit der der Bezugspunkt im Werkstück definiert wird, beizubehalten.

Es wird deshalb zusätzlich der Sprachbefehl `TOROT` eingeführt, der im programmierten Frame nur den Rotationsanteil überschreibt, die übrigen Komponenten aber unverändert lässt. Die durch `TOROT` definierte Drehung ist die gleiche wie bei `TOFRAME`.

`TOROT` ist ebenso wie `TOFRAME` unabhängig vom Vorhandensein eines orientierbaren Werkzeugträgers. Insbesondere ist dieser Sprachbefehl auch bei 5-Achs-Transformationen verwendbar.

Mit dem neuen Sprachbefehl `TOROT` wird außerdem eine konsistente Programmierung bei aktiven orientierbaren Werkzeugträgern für jeden Kinematiktyp erreicht.



Mit `TOFRAME` bzw. `TOROT` werden Frames definiert, deren Z-Richtung in Werkzeugrichtung zeigt. Diese Definition passt zu Fräsbearbeitungen, bei denen typischerweise `G17` aktiv ist. Insbesondere bei Drehbearbeitungen oder allgemein bei aktivem `G18` oder `G19` ist es jedoch wünschenswert, dass Frames definiert werden können, bei denen die Ausrichtung an der X- bzw. Y-Achse erfolgt. Dazu sind in der G-Gruppe 53 folgende Befehle vorhanden:

- `TOFRAMEX`, `TOFRAMEY`, `TOFRAMEZ`
- `TOROTX`, `TOROTY`, `TOROTZ`

Mit diesen Befehlen ist es möglich, entsprechende Frames zu definieren. Dabei ist die Funktionalität von `TOFRAME` und `TOFRAMEZ` bzw. von `TOROT` und `TOROTZ` jeweils identisch.

Die durch `TOROT` bzw. `TOFRAME` entstehenden Frames können in einen eigenen Systemframe (`$P_TOOLFR`) geschrieben werden. Der programmierbare Frame bleibt dann unverändert erhalten.

- Voraussetzung: MD28082 `$MC_MM_SYSTEM_FRAME_MASK`, Bit 3 = 1

Bei der Programmierung von `TOROT` bzw. `TOFRAME` usw. ist das Verhalten mit bzw. ohne Systemframe identisch. Unterschiede ergeben sich dann, wenn der programmierbare Frame weiter bearbeitet wird.

---

### Hinweis

Es wird empfohlen, für die von den Befehlen der G-Gruppe 53 erzeugten Frames bei Neuanlagen nur noch den dafür vorgesehenen Systemframe zu verwenden.

---

### Beispiel

Nach `TOROT` wird `TRANS` programmiert. `TRANS` ohne Angabe von Parametern löscht den programmierbaren Frame. In der Variante ohne Systemframe wird damit auch der durch `TOROT` verursachte Frameanteil des programmierbaren Frames gelöscht, steht der `TOROT`-Anteil im Systemframe, bleibt er erhalten.

`TOROT` bzw. `TOFRAME` usw. werden mit dem Sprachbefehl `TOROTOF` ausgeschaltet. `TOROTOF` löscht den gesamten Systemframe `$P_TOOLFR`. Beschreiben die Befehle `TOFRAME` usw. nicht das Systemframe, sondern das programmierbare Frame (alte Variante), löscht `TOROT` nur den Rotationsanteil und lässt die übrigen Frameanteile unverändert.

Ist vor der Aktivierung der Sprachbefehle `TOFRAME` oder `TOROT` bereits ein drehender Frame aktiv, besteht oft die Forderung, dass der neu definierte Frame vom alten Frame möglichst wenig abweicht. Das ist z.B. dann der Fall, wenn eine Framedefinition leicht modifiziert werden muss weil die Werkzeugorientierung wegen Hirth-verzahnter Rundachsen nicht beliebig eingestellt werden kann. Durch die genannten Sprachbefehle wird die Z-Richtung des neuen Frames eindeutig definiert.

**Framdefinition bei TOFRAME, TOROT und PAROT**

Mit dem folgenden Maschinendatum kann für die Lage von X- und Y-Achse zwischen zwei Varianten gewählt werden. In beiden Fällen besteht jedoch kein Bezug zum vorher aktiven Frame.

MD21110 \$MC\_X\_AXIS\_IN\_OLD\_X\_Z\_PLANE (Koordinatensystem bei automatischer Framedefinition)

Es wird deshalb empfohlen, stattdessen folgendes Settingdatum zu verwenden, mit dem das Verhalten von TOFRAME und TOROT gezielt gesteuert werden kann.

SD42980 \$SC\_TOFRAME\_MODE (Framdefinition bei TOFRAME, TOROT und PAROT)

Wert	Bedeutung
1	Die neue X-Richtung wird so gewählt, dass sie im alten Koordinatensystem in der X-Z-Ebene liegt. Bei dieser Einstellung wird die Winkeldifferenz zwischen alter und neuer Y-Achse minimal.
2	Die neue Y-Richtung wird so gewählt, dass sie im alten Koordinatensystem in der Y-Z-Ebene liegt. Bei dieser Einstellung wird die Winkeldifferenz zwischen alter und neuer X-Achse minimal.
3	Es wird der Mittelwert der beiden Einstellungen, die sich nach 1 bzw. 2 ergeben gewählt.
Eine ausführliche Beschreibung aller Parametriermöglichkeiten findet sich in: <b>Literatur</b> Ausführliche Maschinendaten Beschreibung	

**Besonderheiten und Ergänzungen**

Das Verhalten bei Wert = 1 und 2 wird erreicht, indem ausgehend von einer beliebigen Lage der X- und Y-Achse, das Koordinatensystem um die neue Z-Achse so gedreht wird, bis die verlangte Einstellung erreicht ist.

Das Verhalten von Wert = 3 wird erreicht, indem eine Drehung ausgeführt wird, deren Betrag gleich dem Mittelwert dieser beiden Winkel ist. Dies gilt allerdings nur für den Fall, dass alte und neue Z-Richtung einen Winkel von weniger als 90 Grad einschließen.

Bei Wert = 1 bilden neue und alte X-Achse einen Winkel von weniger als 90 Grad, bei Wert = 2 gilt das gleiche für die Y-Achse (die betreffenden Achsen zeigen "ungefähr" in die gleiche Richtung). Bilden jedoch die beiden Z-Richtungen einen Winkel von mehr als 90 Grad, kann die Bedingung eines Winkels < 90 Grad zwischen alten und neuen Achsen für X und Y nicht mehr gleichzeitig erfüllt werden. In diesem Fall wird der X-Richtung der Vorzug gegeben, d. h. es wird der Mittelwert gebildet, der sich aus der Richtung nach 1 und der negativen Richtung nach 2 ergibt.

Wird an Stelle von `TOFRAME(Z)` bzw. `TOROT(Z)` einer der Befehle `TOFRAMEX`, `TOFRAMEY`, `TOROTX`, `TOROTY` programmiert, gelten die Beschreibungen für die Anpassungen der Achsrichtungen senkrecht zur Hauptrichtung entsprechend für die zyklisch vertauschten Achsen. Es gelten dann die Zuordnungen der nachfolgenden Tabelle:

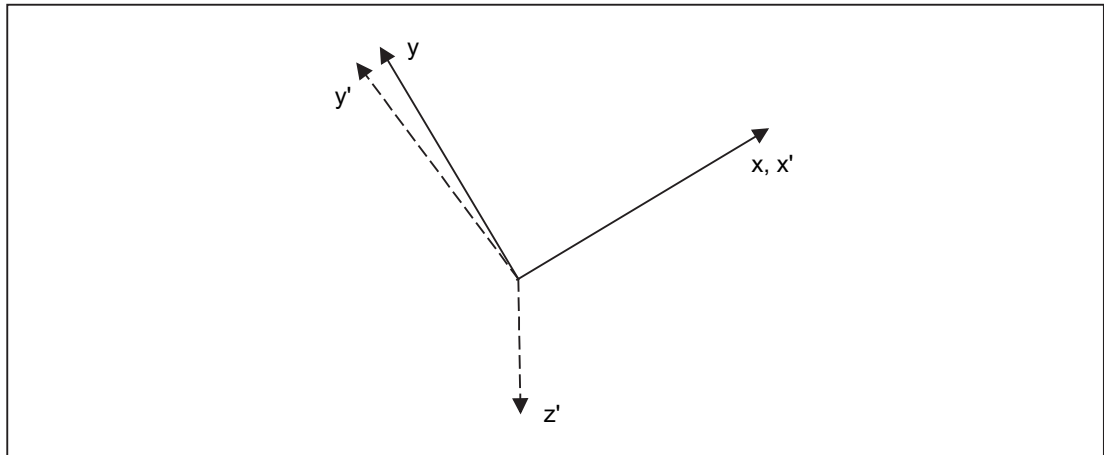
	TOFRAME, TOFRAMEZ TOROT, TOROTZ	TOFRAMEY TOROTY	TOFRAMEX TOROTX
Werkzeugrichtung (Applikate)	Z	Y	X
Nebenachse (Abszisse)	X	Z	Y
Nebenachse (Ordinate)	Y	X	Z

### Beispiel

```
N90 $SC_TOFRAME_MODE=1
N100 ROT Z45
N110 TCARR=1 TCOABS T1 D1
N120 TOROT
```

In `N100` wird eine Drehung in der X-Y-Ebene um 45 Grad beschrieben. Es wird angenommen, dass der in `N110` aktivierte Werkzeugträger das Werkzeug um 30 Grad um die X-Achse dreht, d.h. das Werkzeug liegt in der Y-Z-Ebene und ist gegen die Z-Achse um 30 Grad gedreht. Damit zeigt auch die Z-Achse des in `N120` neu definierten Frames in diese Richtung (unabhängig vom Wert des Settingdatums `SD42980 $SC_TOFRAME_MODE` in `N90`).

SD42980 \$SC\_TOFRAME\_MODE == 1:



Die alte und die neue X-Achse X und X' fallen in der Projektion in Richtung der alten Z-Achse zusammen. Die alte und die neue Y-Achse Y und Y' bilden einen Winkel von 8,13 Grad (in der Projektion bleiben rechte Winkel im Allgemeinen nicht erhalten!).

SD42980 \$SC\_TOFRAME\_MODE == 2:

Y und Y' fallen zusammen und X und X' bilden einen Winkel von 8,13 Grad.

SD42980 \$SC\_TOFRAME\_MODE == 3:

X und X' und auch Y und Y' bilden jeweils einen Winkel von 4,11 Grad.

---

#### Hinweis

Die genannten Winkel (8,13 bzw. 4,11 Grad) sind die Winkel, welche die Projektionen der Achsen in der X-Y-Ebene bilden. Es sind nicht die räumlichen Winkel dieser Achsen.

---

### TCARR (Werkzeugträger anfordern) und PAROT (Werkstückkoordinatensystem am Werkstück ausrichten)

TCARR nutzt das durch das folgende Maschinendatum bezeichnete Basisframe:  
MD20184 \$MC\_TOCARR\_BASE\_FRAME\_NUMBER.

Um Konflikte mit Systemen zu vermeiden, die bereits alle Basisframes nutzen, kann für TCARR und PAROT ein eigenes Systemframe angelegt werden.

PAROT, TOROT und TOFRAME verändern bisher den Rotationsanteil des programmierbaren Frames. Ein separates Abschalten von PAROT oder TOROT ist in diesem Falle nicht möglich. Bei RESET wird das programmierbare Frame gelöscht, was bedeutet, dass nach Betriebsartenwechsel nach JOG der Rotationsanteil von PAROT und TOROT nicht mehr vorhanden ist. Auch muss das programmierbare Frame dem Anwender uneingeschränkt zur Verfügung stehen. Die durch PAROT und TOROT erzeugten Frames müssen über die Datensicherung archivierbar und wieder ladbar sein.

Das Systemframe für `TCARR` und `PAROT` wird projiziert mit:  
`MD28082 $MC_MM_SYSTEM_FRAME_MASK, Bit 2 = 1`

Folgendes Maschinendatum wird dann nicht mehr ausgewertet:  
`MD20184 $MC_TOCARR_BASE_FRAME_NUMBER`

Ist das Systemframe für `TCARR` projiziert, so beschreibt `TCARR` und `PAROT` das entsprechende Systemframe, ansonsten wird das durch Maschinendatum `MD20184` bezeichnete Basisframe beschrieben.

`TCARR` trägt bei Kinematiken des Typs P und des Typs M den Tischoffset des orientierbaren Werkzeugträgers (Verschiebung des Nullpunktes als Folge der Drehung des Tisches), als Translation in das Systemframe ein. `PAROT` rechnet das Systemframe so um, dass sich ein werkstückbezogenes Werkstückkoordinatensystem ergibt.

Die Systemframes werden remanent gespeichert und bleiben deshalb nach Reset erhalten. Auch bei Betriebsartenwechsel bleiben die Systemframes aktiv.

Für die Anzeige werden die Befehle `PAROT` und `TOROT`, `TOFRAME` jeweils einer eigenen G-Gruppe zugeordnet.

## PAROTOF

`PAROTOF` ist der Ausschaltbefehl zu `PAROT`. Dieser Befehl löscht die Drehungen im Systemframe für `PAROT` ab. Es werden dabei die Drehungen im aktuellen `$P_PARTFRAME` und im Datenhaltungsframe `$P_PARTFR` gelöscht. Damit wird die Lage des Koordinatensystems nach `TCARR` wieder hergestellt. `PAROTOF` ist in der gleichen G-Gruppe, wie `PAROT` und erscheint deshalb in der G-Code-Anzeige.

## TOROT (Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten) und TOFRAME (dto.)

Das Systemframe für `TOROT` und `TOFRAME` wird über folgendes Maschinendatum aktiviert:  
`MD28082 $MC_MM_SYSTEM_FRAME_MASK, Bit 3 = 1`

Dieses Systemframe liegt in der Framekette vor dem programmierbaren Frame.  
 Das ENS-Koordinatensystem liegt entsprechend vor dem programmierbaren Frame.

## TOROTOF

`TOROTOF` ist der Ausschaltbefehl zu `TOROT` und `TOFRAME`. Dieser Befehl löscht den entsprechenden Systemframe. Dabei wird der aktuelle `$P_TOOLFRAME` und der Datenhaltungsframe `$P_TOOLFR` gelöscht. `TOROTOF` ist in der gleichen G-Gruppe wie `TOROT` und `TOFRAME` und erscheint deshalb in der G-Code-Anzeige.

**Beispiel**

Beispiel für die Verwendung eines orientierbaren Werkzeugträgers mit aufgelöster Kinematik.

```

N10          $TC_DP1[1,1] = 120
N20          $TC_DP3[1,1] = 13           ; Werkzeuglänge 13mm
; Definition des Werkzeugträgers 1:
N30          $TC_CARR1[1] = 0           ; X-Komponente des 1.
                                           ; Offsetvektors
N40          $TC_CARR2[1] = 0           ; Y-Komponente des 1.
                                           ; Offsetvektors
N50          $TC_CARR3[1] = 0           ; Z-Komponente des 1.
                                           ; Offsetvektors
N60          $TC_CARR4[1] = 0           ; X-Komponente des 2.
                                           ; Offsetvektors
N70          $TC_CARR5[1] = 0.         ; Y-Komponente des 2.
                                           ; Offsetvektors
N80          $TC_CARR6[1] = -15.       ; Z-Komponente des 2.
                                           ; Offsetvektors
N90          $TC_CARR7[1] = 1           ; X-Komponente der 1. Achse
N100         $TC_CARR8[1] = 0           ; Y-Komponente der 1. Achse
N110         $TC_CARR9[1] = 0           ; Z-Komponente der 1. Achse
N120         $TC_CARR10[1] = 0          ; X-Komponente der 2. Achse
N130         $TC_CARR11[1] = 1          ; Y-Komponente der 2. Achse
N140         $TC_CARR12[1] = 0          ; Z-Komponente der 2. Achse
N150         $TC_CARR13[1] = 30.        ; Drehwinkel der 1. Achse
N160         $TC_CARR14[1] = -30.      ; Drehwinkel der 2. Achse
N170         $TC_CARR15[1] = 0          ; X-Komponente des 3.
                                           ; Offsetvektors
N180         $TC_CARR16[1] = 0          ; Y-Komponente des 3.
                                           ; Offsetvektors
N190         $TC_CARR17[1] = 0          ; Z-Komponente des 3.
                                           ; Offsetvektors
N200         $TC_CARR18[1] = 0          ; X-Komponente des 4.
                                           ; Offsetvektors
N210         $TC_CARR19[1] = 0          ; Y-Komponente des 4.
                                           ; Offsetvektors
N220         $TC_CARR20[1] = 15.       ; Z-Komponente des 4.
                                           ; Offsetvektors
N230         $TC_CARR21[1] = A          ; Bezug fuer 1. Achse
N240         $TC_CARR22[1] = B          ; Bezug fuer 2. Achse
N250         $TC_CARR23[1] = "M"       ; Typ des Werkzeugtraegers
N260         X0 Y0 Z0 A0 B45 F2000
N270         TCARR=1 X0 Y10 Z0 T1 TCOABS ; Anwahl des orient.
    
```

```

; Werkzeugträgers
N280      PAROT      ; Drehung des Tisches
N290      TOROT     ; Drehung der z-Achse in
; WZ-Orient.
N290      X0 Y0 Z0
N300      G18 MOVT=AC(20) ; Bearbeitung in G18 Ebene
N310      G17 X10 Y0 Z0 ; Bearbeitung in G17 Ebene
N320      MOVT=-10
N330      PAROTOF   ; Drehung des Tisches abschalten
N340      TOROTOF   ; WKS nicht mehr am WZ
; ausrichten.
N400      M30

```

### 9.5.9 Unterprogramme mit Attribut SAVE

Bei verschiedenen Frames ist das Verhalten bezüglich Unterprogrammen mit dem Attribut SAVE einstellbar.

#### Einstellbare Frames G54 - G599

Mit MD10617 \$MN\_FRAME\_SAVE\_MASK.BIT0 kann das Verhalten der einstellbaren Frames eingestellt werden:

- BIT0 = 0

Werden durch das Unterprogramm nur die Werte des aktiven einstellbaren Frames über die Systemvariable \$P\_IFRAME verändert, die G-Funktion aber beibehalten, bleibt die Veränderung auch nach Unterprogrammende erhalten.

- BIT0 = 1

Mit dem Unterprogrammende wird der vor dem Unterprogrammaufruf aktive einstellbare Frame, G-Funktion und Werte, reaktiviert.

#### Basisframes \$P\_CHBFR[ ] und \$P\_NCBFR[ ]

Mit MD10617 \$MN\_FRAME\_SAVE\_MASK.BIT1 kann das Verhalten der Basisframes eingestellt werden:

- BIT1 = 0

Wird durch das Unterprogramm der aktive Basisframe verändert, bleibt die Veränderung auch nach Unterprogrammende erhalten.

- BIT1 = 1

Mit dem Unterprogrammende wird der vor dem Unterprogrammaufruf aktive Basisframe reaktiviert.

### Programmierbarer Frame

Mit dem Unterprogrammende wird der vor dem Unterprogrammaufruf aktive programmierbare Frame reaktiviert.

### Systemframes

Werden durch das Unterprogramm Systemframes verändert, bleibt die Veränderung nach Unterprogrammende erhalten.

## 9.5.10 Datensicherung

Die Archivierung der Systemframes erfolgt über die Datenbausteine `_N_CHANx_UFR`.

Zwischen Sichern und Wiedereinspielen von gesicherten Systemframes sollte sich das Maschinendatum:

`MD28082 $MC_MM_SYSTEM_FRAME_MASK`

nicht geändert haben, da sonst evtl. gesicherte Systemframes nicht mehr geladen werden können.

Der Ladevorgang erzeugt in diesem Fall einen Alarm.

Die Datensicherung erfolgt immer nach den aktuell gültigen Geometrieachs-Zuordnungen und nicht nach den in den Maschinendaten eingestellten Achskonstellationen.

Über das Maschinendatum:

`$MC_MM_SYSTEM_DATAFRAME_MASK`

können Datenhaltungsframes für die Systemframes projiziert werden.

Möchte man kein Datenhaltungsframe für ein Systemframe haben, so kann das Frame auch nicht gesichert werden. Bei `G500, G54 .. G599` bleibt das aktive Frame erhalten.

Die Archivierung der globalen Frames erfolgt über einen eigenen Datenbaustein `_N_NC_UFR`.

Der von der HMI angeforderte Baustein wird angelegt, wenn das Maschinendatum:

`MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES`

bzw.

`MD18602 $MN_MM_NUM_GLOBAL_BASE_FRAMES`

einen Wert größer Null hat.

Die kanalspezifischen Frames werden in den Datenbausteinen `_N_CHANx_UFR` gesichert.

Beim Wiedereinspielen von gesicherten Daten können unter Umständen Alarmer auftreten, wenn über Maschinendaten die Frame-Zugehörigkeiten, NCU-global oder kanalspezifisch, geändert wurden.

Die Datensicherung erfolgt immer nach den in den Maschinendaten eingestellten Achskonstellationen und nicht nach den aktuell gültigen Geometrieachs-Zuordnungen.



## 9.5.11 Positionen in den Koordinatensystemen

Die aktuellen Sollwert-Positionen in den Koordinatensystemen können über folgende Systemvariablen gelesen werden. Über PLC können wahlweise die Istwerte im WKS, ENS, BNS oder MKS angezeigt werden. Es gibt dazu einen Softkey Istwertanzeige im MKS/WKS. Der Maschinenhersteller kann von PLC-Seite vorgeben, welches Koordinatensystem bei seiner Maschine dem Werkstückkoordinatensystem entspricht. Der HMI fordert die entsprechenden Istwerte von NCK an.

### **\$AA\_IM[Achse]**

Die Sollwerte im Maschinenkoordinatensystem können für jede Achse mit der Variablen \$AA\_IM[Achse] gelesen werden.

### **\$AA\_IEN[Achse]**

Die Sollwerte im einstellbaren Nullpunkt-Koordinatensystem ENS können für jede Achse mit der Variablen \$AA\_IEN[Achse] gelesen werden.

### **\$AA\_IBN[Achse]**

Mit \$AA\_IBN[Achse] können die Sollwerte im Basisnullpunkt-Koordinatensystem BNS gelesen werden.

### **\$AA\_IW[Achse]**

Mit \$AA\_IW[Achse] können die Sollwerte im Werkstück-Koordinatensystem WKS gelesen werden.

## 9.5.12 Steuerungsverhalten

### 9.5.12.1 POWER ON

#### Framezustände nach POWER ON

Frame	Zustand nach POWER ON
Programmierbarer Frame	Gelöscht.
Einstellbare Frames	Bleiben erhalten, abhängig von: MD24080 \$MC_USER_FRAME_POWERON_MASK Bit 0 = 1 MD20152 \$MC_GCODE_RESET_MODE[7] = 1
Gesamt-Basisframe	Bleibt erhalten, abhängig von: MD20110 \$MC_RESET_MODE_MASK Bit 0 und Bit 14 Mit MD10615 \$MN_NCBFRAME_POWERON_MASK und MD24004 \$MC_CHBFRAME_POWERON_MASK können einzelne Basisframes gelöscht werden.
Systemframes	Bleiben erhalten. Abhängig von: MD24008 \$MC_CHSFRAME_POWERON_MASK können einzelne Systemframes bei POWER ON gelöscht werden. Das Löschen von Systemframes wird erstrangig in der Datenhaltung durchgeführt.
Externe Nullpunktverschiebung	Permanent, muss aber erneut aktiviert werden. Das Systemframe bleibt erhalten.
DRF-Verschiebung	Gelöscht.

### 9.5.12.2 Betriebsartenwechsel

#### Systemframes

Über Betriebsartenwechsel bleiben die Systemframes erhalten und aktiv.

#### Betriebsart JOG

In JOG werden nur die Framekomponenten des aktuellen Frames für die Geometrieachsen berücksichtigt, wenn eine Rotation aktiv ist. Alle anderen axialen Frames werden nicht berücksichtigt.

#### PLC- und Kommando-Achsen

Für PLC- und Kommando-Achsen ist das Verhalten einstellbar über das Maschinendatum:

MD32074 \$MA\_FRAME\_OR\_CORRPOS\_NOTALLOWED (Frame oder HL-Korrektur sind unzulässig)

### 9.5.12.3 RESET, Teileprogrammende

#### RESET-Verhalten der Basisframes

Das RESET-Verhalten der Basisframes wird eingestellt über das Maschinendatum:

MD20110 \$MC\_RESET\_MODE\_MASK (Festlegung der Steuerungs-Grundstellung nach RESET/TP-Ende)

#### RESET-Verhalten der Systemframes

Die Systemframes bleiben auch nach RESET in der Datenhaltung erhalten.

Die Aktivierung der einzelnen Systemframes kann über die folgenden Maschinendaten projektiert werden:

MD24006 \$MC\_CHSFRAME\_RESET\_MASK (Aktive Systemframes nach RESET)

Bit	Bedeutung
0	Systemframe für Istwertsetzen und Ankratzen ist nach RESET aktiv.
1	Systemframe für externe Nullpunktverschiebung ist nach RESET aktiv.
2	Wird nicht ausgewertet.
3	Wird nicht ausgewertet.
4	Systemframe für Werkstückbezugspunkte ist nach RESET aktiv.
5	Systemframe für Zyklen ist nach RESET aktiv.
6	Reserviert, RESET-Verhalten abhängig von MD20110 \$MC_RESET_MODE_MASK.
7	Systemframe \$P_ISO1FR ist nach RESET aktiv.
8	Systemframe \$P_ISO2FR ist nach RESET aktiv.
9	Systemframe \$P_ISO3FR ist nach RESET aktiv.
10	Systemframe \$P_ISO4FR ist nach RESET aktiv.
11	Systemframe \$P_RELFR ist nach RESET aktiv.

#### RESET-Verhalten der Systemframes für TCARR, PAROT, TOROT und TOFRAME

Das RESET-Verhalten der Systemframes für TCARR, PAROT, TOROT und TOFRAME ist von der G-Code-RESET-Einstellung abhängig.

Die Einstellung erfolgt mit den Maschinendaten:

MD20110 \$MC\_RESET\_MODE\_MASK (Festlegung der Steuerungs-Grundstellung nach RESET/TP-Ende)

MD20152 \$MC\_GCODE\_RESET\_MODE[ ] (RESET-Verhalten der G-Gruppen)

MD20150 \$MC\_GCODE\_RESET\_VALUES[ ] (Löschstellung der G-Gruppen)

MD20110	Bedeutung	
Bit 0 = 0	TCARR und PAROT-Systemframe bleibt wie vor RESET erhalten.	
Bit 0 = 1	MD20152 \$MC_GCODE_RESET_MODE[51] = 0	
	MD20150 \$MC_GCODE_RESET_VALUES[51] = 1	PAROTOF
	MD20150 \$MC_GCODE_RESET_VALUES[51] = 2	PAROT
	MD20152 \$MC_GCODE_RESET_MODE[51] = 1	
	TCARR und PAROT-Systemframe bleibt wie vor RESET erhalten.	
	MD20152 \$MC_GCODE_RESET_MODE[52] = 0	
	MD20150 \$MC_GCODE_RESET_VALUES[52] = 1	TOROTOF
	MD20150 \$MC_GCODE_RESET_VALUES[52] = 2	TOROT
	MD20150 \$MC_GCODE_RESET_VALUES[52] = 3	TOFRAME
	MD20152 \$MC_GCODE_RESET_MODE[52] = 1	
TOROT und TOFRAME-Systemframe bleibt wie vor RESET erhalten.		

TCARR und PAROT sind zwei eigenständige Funktionen, die das gleiche Frame beschreiben. Bei RESET wird mit PAROTOF auch der Anteil von TCARR nicht aktiviert.

MD20110	Bedeutung	
Bit 0 = 1 und Bit 14 = 0	Verkettetes Gesamtbasisframe wird gelöscht.	
Bit 0 = 1 und Bit 14 = 1	Das Gesamtbasisframe ergibt sich anhand von: MD24002 \$MC_CHBFRAME_RESET_MASK (Aktive kanalspezifische Basisframes nach RESET) und MD10613 \$MN_NCBFRAME_RESET_MASK (Aktive NCU-globale Basisframes nach RESET)	
	MD24002 \$MC_CHBFRAME_RESET_MASK	
	Bit 0 = 1	1. Kanal-Basisframe wird in das verkettete Gesamtbasisframe eingerechnet.
	Bit 7 = 1	8. Kanal-Basisframe wird in das verkettete Gesamtbasisframe eingerechnet.
	MD10613 \$MN_NCBFRAME_RESET_MASK	
	Bit 0 = 1	1. NCU-globales Basisframe wird in das verkettete Gesamtbasisframe eingerechnet.
	Bit 7 = 1	8. NCU-globales Basisframe wird in das verkettete Gesamtbasisframe eingerechnet.

## Framezustände nach RESET / Teileprogrammende

Frame	Zustand nach RESET / Teileprogrammende
Programmierbarer Frame	Gelöscht.
Einstellbare Frames	Bleiben erhalten, abhängig von: MD20110 \$MC_RESET_MODE_MASK und MD20152 \$MC_GCODE_RESET_MODE.
Gesamt-Basisframe	Bleibt erhalten, abhängig von: MD20110 \$MC_RESET_MODE_MASK Bit 0 und Bit 14, MD10613 \$MN_NCBFRAME_RESET_MASK und MD24002 \$MC_CHBFRAME_RESET_MASK.
Systemframes	Bleiben erhalten, abhängig von: MD24006 \$MC_CHSFRAME_RESET_MASK und MD20150 \$MC_GCODE_RESET_VALUES[ ].
Externe Nullpunktverschiebung	Bleibt erhalten.
DRF-Verschiebung	Bleibt erhalten.

## Löschen von Systemframes

Die Systemframes in der Datenhaltung können bei RESET über folgendes Maschinendatum gelöscht werden:

MD24007 \$MC\_CHSFRAME\_RESET\_CLEAR\_MASK (Löschen von Systemframes bei RESET)

Bit	Bedeutung
0	Systemframe für Istwertsetzen und Ankratzen wird bei RESET gelöscht.
1	Systemframe für externe Nullpunktverschiebung wird bei RESET gelöscht.
2	Reserviert, TCARR und PAROT siehe MD20150 \$MC_GCODE_RESET_VALUES[ ].
3	Reserviert, TOROT und TOFRAME siehe MD20150 \$MC_GCODE_RESET_VALUES[ ].
4	Systemframe für Werkstückbezugspunkte wird bei RESET gelöscht.
5	Systemframe für Zyklen wird bei RESET gelöscht.
6	Reserviert, Reset-Verhalten abhängig von MD20110 \$MC_RESET_MODE_MASK.
7	Systemframe \$P_ISO1FR wird bei RESET gelöscht.
8	Systemframe \$P_ISO2FR wird bei RESET gelöscht.
9	Systemframe \$P_ISO3FR wird bei RESET gelöscht.
10	Systemframe \$P_ISO4FR wird bei RESET gelöscht.
11	Systemframe \$P_RELFR wird bei RESET gelöscht.

### 9.5.12.4 Teileprogrammstart

#### Framezustände nach Teileprogrammstart

Frame	Zustand nach Teileprogrammstart
Programmierbarer Frame	Gelöscht.
Einstellbare Frames	Bleiben erhalten, abhängig von: MD20112 \$MC_START_MODE_MASK
Gesamt-Basisframe	Bleibt erhalten.
Systemframes	Bleiben erhalten.
Externe Nullpunktverschiebung	Bleibt erhalten.
DRF-Verschiebung	Bleibt erhalten.

### 9.5.12.5 Satzsuchlauf

#### Satzsuchlauf

Beim Satzsuchlauf mit Berechnung werden auch Datenhaltungsframes modifiziert.

#### Abbruch des Satzsuchlaufs

Wird der Satzsuchlauf mit RESET abgebrochen, so kann mit dem Maschinendatum:  
MD28560 \$MC\_MM\_SEARCH\_RUN\_RESTORE\_MODE  
projektiert werden, dass alle Datenhaltungsframes auf den Wert vor dem Satzsuchlauf  
zurückgesetzt werden:

Bit	Bedeutung
0	Alle Frames in der Datenhaltung werden restauriert.

Bei kaskadierten Satzsuchläufen, werden die Frames auf den Zustand des Vorgänger-  
Satzsuchlaufes gesetzt.

### SERUPRO

Die Funktion "SERUPRO" wird nicht unterstützt.

### 9.5.12.6 REPOS

Es gibt keine Sonderbehandlung für Frames. Wird ein Frame in einem ASUP geändert, so  
bleibt es im Programm erhalten. Beim Wiederanfahren mit REPOS wird ein geändertes  
Frame berücksichtigt, sofern die Änderung im ASUP aktiviert wurde.

## 9.6 Werkstücknahes Istwertsystem

### 9.6.1 Übersicht

#### Definition

Unter dem Begriff "Werkstücknahes Istwertsystem" werden eine Reihe von Funktionen zusammengefasst, die dem Anwender folgendes Vorgehen ermöglichen:

- Nach Hochlauf auf ein über Maschinendaten definiertes Werkstückkoordinatensystem aufsetzen.  
Merkmale:
  - keine zusätzlichen Bedienhandlungen nötig
  - in der Betriebsart JOG und AUTOMATIK wirksam
- Gültige Einstellungen bei Teileprogrammende für das folgende Teileprogramm beibehalten bezüglich:
  - aktiver Ebene
  - einstellbarer Frame (G54-G57)
  - kinematischer Transformation
  - aktiver Werkzeugkorrektur
- Über die Bedienoberfläche zwischen Werkstückkoordinatensystem und Maschinenkoordinatensystem wechseln.
- Das Werkstückkoordinatensystem per Bedienung ändern (z.B. Änderung des einstellbaren Frame oder der Werkzeugkorrektur).

### 9.6.2 Benutzung des werkstücknahen Istwertsystems

#### Voraussetzungen, Grundsätzliches

Für das System wurden die im vorigen Abschnitt beschriebenen Einstellungen getroffen. Die Voreinstellung nach Hochlauf der HMI-Software ist MKS.

#### Umschalten auf WKS

Die Umschaltung auf das WKS über die Bedienoberfläche bewirkt, dass die Achspositionen bezüglich des Ursprungs des WKS angezeigt werden.

#### Umschalten auf MKS

Die Umschaltung auf das MKS über die Bedienoberfläche bewirkt, dass die Achspositionen bezüglich des Ursprungs des MKS angezeigt werden.

### Zusammenhänge zwischen Koordinatensystemen

Das folgende Bild stellt die Zusammenhänge vom Maschinenkoordinatensystem MKS bis zum Werkstückkoordinatensystem WKS dar.

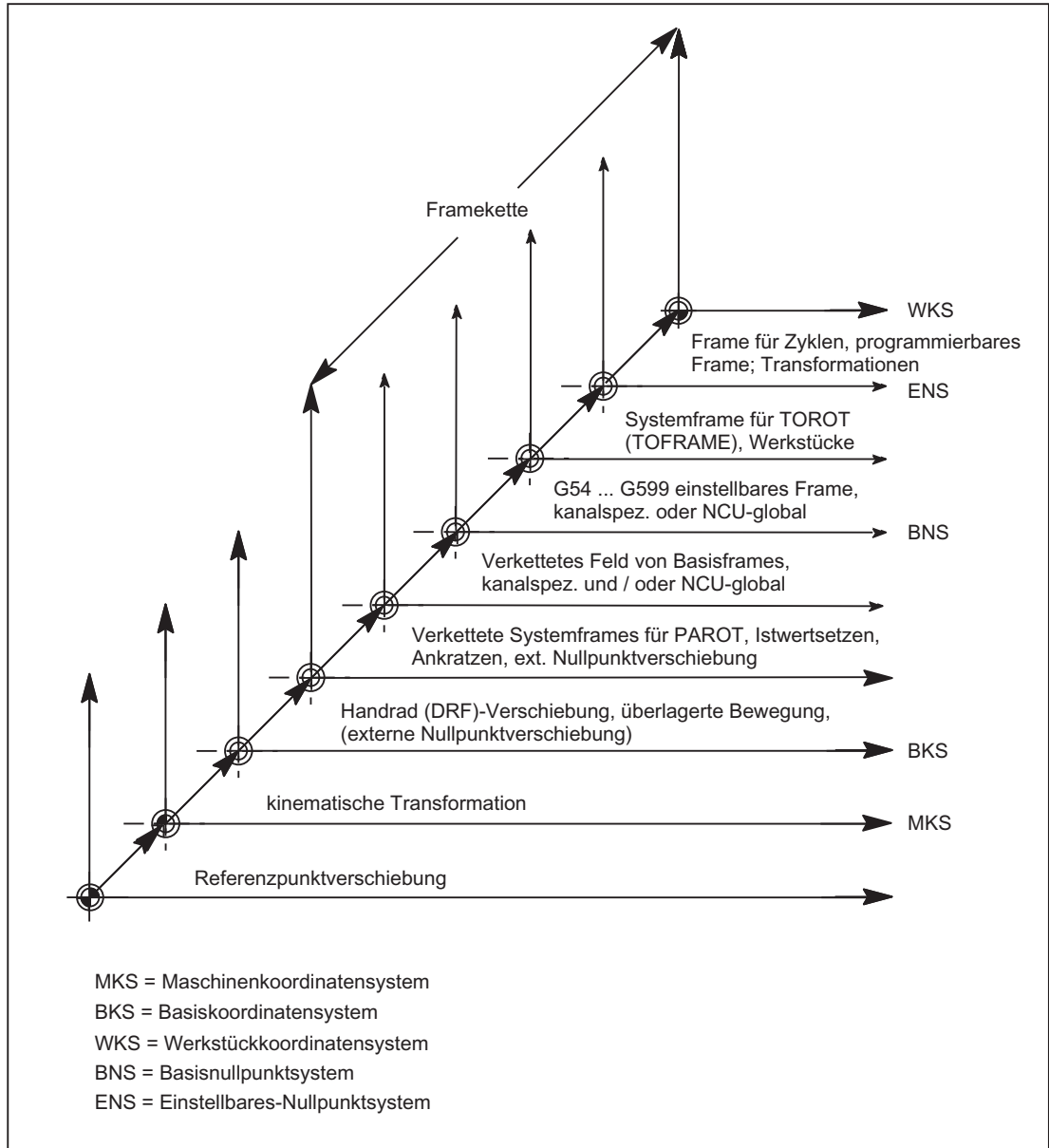


Bild 9-26 Zusammenhang Koordinatensysteme



Für weitere Informationen siehe "H2: Hilfsfunktionsausgaben an PLC (Seite 395)" und "W1: Werkzeugkorrektur (Seite 1567)".

**Literatur:**

- Programmieranleitung Grundlagen
- Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformation (M1)
- Funktionshandbuch Sonderfunktionen; Achskopplungen und ESR (M3); Kapitel: Mitschleppen, Kapitel: Leitwertkopplung
- Funktionshandbuch Sonderfunktionen; Tangentialsteuerung (T3)

### 9.6.3 Besondere Reaktionen

#### Überspeichern

Überspeichern im `RESET`-Zustand von:

- Frames (Nullpunktverschiebungen)
- Aktiver Ebene
- Aktivierter Transformation
- Werkzeugkorrektur

wirkt sofort auf die Istwert-Anzeige aller Achsen im Kanal.

#### Eingabe über Bedientafelfront

Werden die Werte für:

"Aktiver Frame" (Nullpunktverschiebungen, Bedienbereich Parameter)

und

"Aktive Werkzeuglängenkorrektur" (Bedienbereich Parameter)

durch Bedienung an der Bedientafelfront geändert, so werden sie in der Anzeige durch eine der folgenden Maßnahmen wirksam:

- Betätigen der `RESET`-Taste
- Neuanwahl von:
  - Nullpunktverschiebung durch das Teileprogramm
  - Werkzeugkorrektur durch das Teileprogramm
- Neueinstellung von:
  - Nullpunktverschiebung durch Überspeichern
  - Werkzeugkorrektur durch Überspeichern
- Teileprogramm-Start

## MD9440

Ist das HMI-Maschinendatum für die Bedientafelfront:

MD9440 ACTIVATE\_SEL\_USER\_DATA

gesetzt, so werden die eingegebenen Werte im Reset-Zustand sofort aktiv.

Bei erfolgter Eingabe in gestoppter Teileprogrammbearbeitung werden die Werte mit dem Fortsetz-Start aktiv.

## Istwertlesen

Wird aus \$AA\_IW nach dem Aktivieren eines Frames (Nullpunktverschiebung) oder einer Werkzeugkorrektur der Istwert im WKS gelesen, so sind die aktivierten Änderungen im gelesenen Ergebnis bereits enthalten, auch wenn die Achsen noch nicht mit den aktivierten Änderungen bewegt wurden.

Die Istwerte im Einstellbaren-Nullpunktsystem ENS können für jede Achse mit der Variablen \$AA\_IEN[Achse] aus dem Teileprogramm gelesen werden.

Mit \$AA\_IBN[Achse] können die Istwerte im Basisnullpunkt-Koordinatensystem BNS aus dem Teileprogramm gelesen werden.

## Istwertanzeige

Im WKS wird immer die programmierte Kontur angezeigt.

Auf das MKS werden folgende Verschiebungen aufgerechnet:

- Kinematische Transformation
- DRF-Verschiebung / Externe Nullpunktverschiebung
- Aktiver Frame
- Aktive Werkzeugkorrektur des aktuellen Werkzeugs

## Umschalten durch PLC

Über die PLC können wahlweise die Istwerte im WKS, ENS, BNS oder MKS angezeigt werden. Von PLC-Seite kann vorgegeben werden, welches Koordinatensystem bei einer Maschine dem Werkstückkoordinatensystem entspricht.

Nach Hochlauf ist standardmäßig das MKS angewählt.

Durch das Signal DB19 DBB0.7 "MKS/WKS-Umschaltung" kann auch von der PLC aus auf das WKS umgeschaltet werden.

## Übergaben an PLC

In Abhängigkeit vom Maschinendatum:

MD20110 / MD20112, Bit 1

werden bei der Anwahl der Werkzeuglängenkorrektur die Hilfsfunktionen (D, T, M) an die PLC ausgegeben (oder nicht).

---

### Hinweis

Ist von der PLC das WKS angewählt, kann per Bedienung für die jeweilige Betriebsart dennoch zwischen WKS und MKS umgeschaltet werden.

Bei Betriebsarten- oder/und Bereichswechsel wird jedoch das von der PLC angewählte WKS ausgewertet und aktiviert (siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)").

---

## **9.7 Randbedingungen**

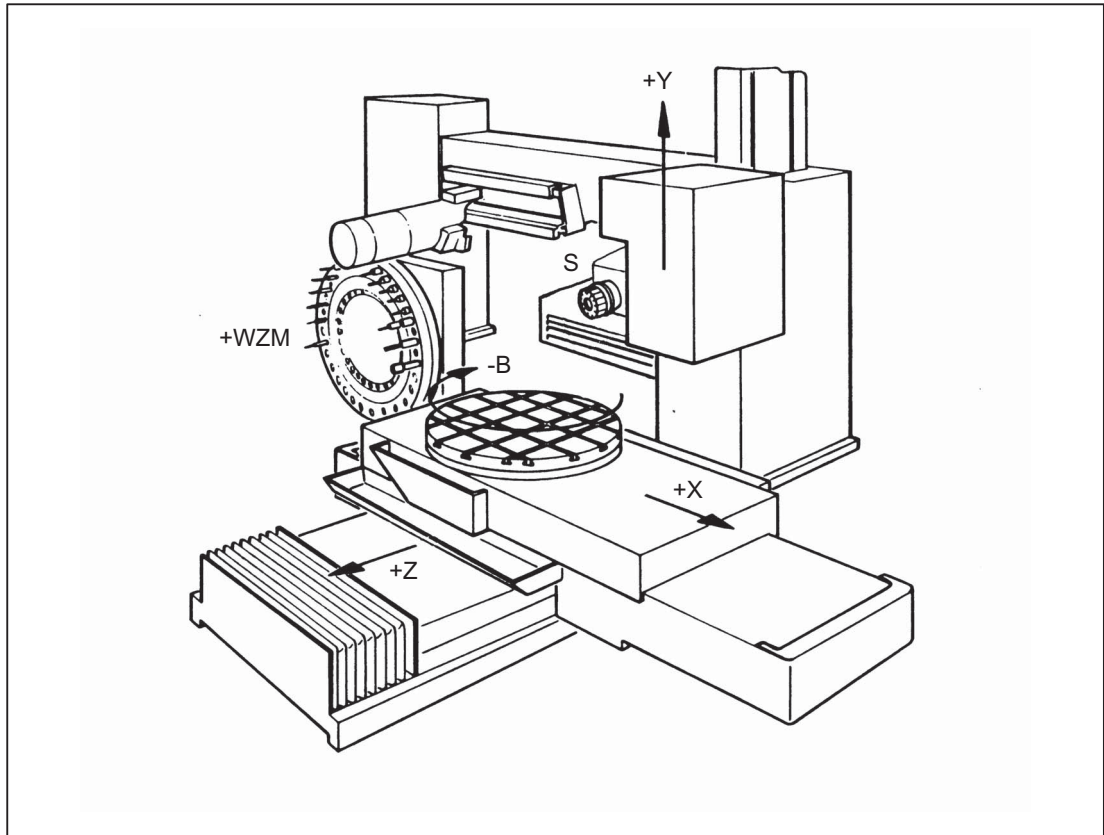
Es sind keine Randbedingungen zu beachten.

## 9.8 Beispiele

### 9.8.1 Achsen

#### Achskonfiguration für eine 3-Achs-Fräsmaschine mit Rundtisch

1. Maschinenachse: X1	Linearachse
2. Maschinenachse: Y1	Linearachse
3. Maschinenachse: Z1	Linearachse
4. Maschinenachse: B1	Rundtisch (zum Drehen für Mehrseitenbearbeitung)
5. Maschinenachse: W1	Rundachse für Werkzeugmagazin (WZ-Teller)
6. Maschinenachse: C1	(Spindel)
1. Geometrieachse: X	(1. Kanal)
2. Geometrieachse: Y	(1. Kanal)
3. Geometrieachse: Z	(1. Kanal)
1. Zusatzachse: B	(1. Kanal)
2. Zusatzachse: WZM	(1. Kanal)
1. Spindel: S1/C	(1. Kanal)



Parametrierung der Maschinendaten

Maschinendatum	Wert
MD10000 AXCONF_MACHAX_NAME_TAB[0]	= X1
MD10000 AXCONF_MACHAX_NAME_TAB[1]	= Y1
MD10000 AXCONF_MACHAX_NAME_TAB[2]	= Z1
MD10000 AXCONF_MACHAX_NAME_TAB[3]	= B1
MD10000 AXCONF_MACHAX_NAME_TAB[4]	= W1
MD10000 AXCONF_MACHAX_NAME_TAB[5]	= C1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[0]	= 1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[1]	= 2
MD20050 AXCONF_GEOAX_ASSIGN_TAB[2]	= 3
MD20060 AXCONF_GEOAX_NAME_TAB[0]	= X
MD20060 AXCONF_GEOAX_NAME_TAB[1]	= Y
MD20060 AXCONF_GEOAX_NAME_TAB[2]	= Z
MD20070 AXCONF_MACHAX_USED[0]	= 1
MD20070 AXCONF_MACHAX_USED[1]	= 2

Maschinendatum	Wert
MD20070 AXCONF_MACHAX_USED[2]	= 3
MD20070 AXCONF_MACHAX_USED[3]	= 4
MD20070 AXCONF_MACHAX_USED[4]	= 5
MD20070 AXCONF_MACHAX_USED[5]	= 6
MD20080 AXCONF_CHANAX_NAME_TAB[0]	= X
MD20080 AXCONF_CHANAX_NAME_TAB[1]	= Y
MD20080 AXCONF_CHANAX_NAME_TAB[2]	= Z
MD20080 AXCONF_CHANAX_NAME_TAB[3]	= B
MD20080 AXCONF_CHANAX_NAME_TAB[4]	= WZM
MD20080 AXCONF_CHANAX_NAME_TAB[5]	= S1
MD30300 IS_ROT_AX[3]	= 1
MD30300 IS_ROT_AX[4]	= 1
MD30300 IS_ROT_AX[5]	= 1
MD30310 ROT_IS_MODULO[3]	= 1
MD30310 ROT_IS_MODULO[4]	= 1
MD30310 ROT_IS_MODULO[5]	= 1
MD30320 DISPLAY_IS_MODULO[3]	= 1
MD30320 DISPLAY_IS_MODULO[4]	= 1
MD20090 SPIND_DEF_MASTER_SPIND	= 1
MD35000 SPIND_ASSIGN_TO_MACHAX[AX1]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX2]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX3]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX4]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX5]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX6]	= 1

### 9.8.2 Koordinatensysteme

#### Projektierung eines globalen Basisframes

Vorausgesetzt ist eine NC mit 2 Kanälen. Dabei gilt:

- Beide Kanäle können den globalen Basisframe schreiben.
- Der jeweils andere Kanal erkennt die Änderung nach erneuter Aktivierung des globalen Basisframes.
- Beide Kanäle können den globalen Basisframe lesen.
- Beide Kanäle können für sich den globalen Basisframe aktivieren

#### Maschinendaten

Maschinendatum	Wert
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[0]	= X1
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[1]	= X2
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[2]	= X3
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[3]	= X4
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[4]	= X5
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[5]	= X6
MD18602 \$MN_MM_NUM_GLOBAL_BASE_FRAMES	= 1
MD28081 \$MC_MM_NUM_BASE_FRAMES	= 1

Maschinendaten für Kanal 1	Wert	Maschinendaten für Kanal 1	Wert
\$MC_AXCONF_CHANAX_NAME_TAB[0]	= X	\$MC_AXCONF_CHANAX_NAME_TAB[0]	= X
\$MC_AXCONF_CHANAX_NAME_TAB[1]	= Y	\$MC_AXCONF_CHANAX_NAME_TAB[1]	= Y
\$MC_AXCONF_CHANAX_NAME_TAB[2]	= Z	\$MC_AXCONF_CHANAX_NAME_TAB[2]	= Z
\$MC_AXCONF_MACHAX_USED[0]	= 1	\$MC_AXCONF_MACHAX_USED[0]	= 4
\$MC_AXCONF_MACHAX_USED[1]	= 2	\$MC_AXCONF_MACHAX_USED[1]	= 5
\$MC_AXCONF_MACHAX_USED[2]	= 3	\$MC_AXCONF_MACHAX_USED[2]	= 6
\$MC_AXCONF_GEOAX_NAME_TAB[0]	= X	\$MC_AXCONF_GEOAX_NAME_TAB[0]	= X
\$MC_AXCONF_GEOAX_NAME_TAB[1]	= Y	\$MC_AXCONF_GEOAX_NAME_TAB[1]	= Y
\$MC_AXCONF_GEOAX_NAME_TAB[2]	= Z	\$MC_AXCONF_GEOAX_NAME_TAB[2]	= Z
\$MC_AXCONF_GEOAX_ASSIGN_TAB[0]	= 1	\$MC_AXCONF_GEOAX_ASSIGN_TAB[0]	= 4
\$MC_AXCONF_GEOAX_ASSIGN_TAB[1]	= 2	\$MC_AXCONF_GEOAX_ASSIGN_TAB[1]	= 5
\$MC_AXCONF_GEOAX_ASSIGN_TAB[2]	= 3	\$MC_AXCONF_GEOAX_ASSIGN_TAB[2]	= 6



### Teilprogramm im 1. Kanal

Code (Ausschnitt)	Kommentar
. . .	
N100 \$P_NCBFR[0] = CTRANS( x, 10 )	; Aktivierung des NC-globalen Basisframes
. . .	
N130 \$P_NCBFRAME[0] = CROT(X, 45)	; Aktivierung des NC-globalen Basisframes mit Drehung => Alarm 18310, da Drehungen bei NC-globalen Frames nicht erlaubt sind
. . .	

### Teilprogramm im 2. Kanal

Code (Ausschnitt)	Kommentar
. . .	
N100 \$P_NCBFR[0] = CTRANS( x, 10 )	; Der NC-globale Basisframe wirkt auch im 2.Kanal
. . .	
N510 G500 X10	; Basisframe aktivieren
N520 \$P_CHBFRAME[0] = CTRANS( x, 10 )	; Aktueller Frame des 2. Kanals wird mit einer Verschiebung aktiviert
. . .	

## 9.8.3 Frames

### Beispiel 1

Die Kanalachse soll durch einen Geometrieachsen-Tausch zur Geometrieachse werden.

Durch den Tausch soll der programmierbare Frame einen Translationsanteil von 10 in der x-Achse haben.

Der aktuelle einstellbare Frame soll erhalten bleiben:

```
FRAME_GEOX_CHANGE_MODE = 1
```

\$P_UIFR[1] =	; Frame bleibt nach Geometrieachsen- Tausch
CROT(x,10,y,20,z,30)	erhalten
G54	; einstellbarer Frame wird aktiv
TRANS a10	; achsiale Verschiebung von a wird getauscht
GEOAX(1,a)	; a wird zur x-Achse
	; \$P_ACTFRAME= CROT(x,10,y,20,z,30):CTRANS(x10)

Bei Transformationswechsel können gleichzeitig mehrere Kanalachsen zu Geometrieachsen werden.

## Beispiel 2

Durch eine 5-Achs-Orientierungs-Transformation werden die Kanalachsen 4, 5 und 6 zu Geometrieachsen der Transformation. Die Geometrieachsen vor der Transformation werden also alle ersetzt.

Beim Einschalten der Transformation ändern sich alle aktuellen Frames.

Zur Berechnung des neuen WKS werden die achsialen Frame-Anteile der Kanalachsen, die zu Geometrieachsen werden, berücksichtigt. Programmierte Drehungen vor der Transformation werden beibehalten. Nach dem Ausschalten der Transformation wird das alte WKS wieder hergestellt.

Der häufigste Anwendungsfall wird wohl der sein, dass sich die Geometrieachsen vor und nach der Transformation nicht ändern und die Frames so beibehalten werden sollen, wie sie vor der Transformation waren.

### Maschinendaten:

```
$MN_FRAME_GEOAX_CHANGE_MODE = 1
```

```
$MC_AXCONF_CHANAX_NAME_TAB[0] = "CAX"
```

```
$MC_AXCONF_CHANAX_NAME_TAB[1] = "CAY"
```

```
$MC_AXCONF_CHANAX_NAME_TAB[2] = "CAZ"
```

```
$MC_AXCONF_CHANAX_NAME_TAB[3] = "A"
```

```
$MC_AXCONF_CHANAX_NAME_TAB[4] = "B"
```

```
$MC_AXCONF_CHANAX_NAME_TAB[5] = "C"
```

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
```

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 2
```

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3
```

```
$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"
```

```
$MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"
```

```
$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"
```

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0] = 4
```

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1] = 5
```

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2] = 6
```

```
$MC_TRAFO_AXES_IN_1[0] = 4
```

```
$MC_TRAFO_AXES_IN_1[1] = 5
```

```
$MC_TRAFO_AXES_IN_1[2] = 6
```

```
$MC_TRAFO_AXES_IN_1[3] = 1
```

```
$MC_TRAFO_AXES_IN_1[4] = 2
```

**Programm:**

```

$P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
$P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
$P_IFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(z,45)
$P_PFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(x,10,y,20,z,30)

TRAORI ; Trafo setzt GeoAx(4,5,6)
; $P_NCBFRAME[0] =
;   ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3)
; $P_ACTBFRAME =
;   ctrans(x,8,y,10,z,12,cax,2,cay,4,caz,6)
; $P_PFRAME = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3) :
; crot(x,10,y,20,z,30)
; $P_IFRAME =
;   ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3):crot(z,45)

TRAFOOF; ; Ausschalten der Transformation setzt GeoAx(1,2,3)
; $P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
; $P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
; $P_IFRAME =
;   ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(z,45)
; $P_PFRAME =
;   ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(x,10,y,20,z,30)

```

## 9.9 Datenlisten

### 9.9.1 Maschinendaten

#### 9.9.1.1 Anzeige-Maschinendaten

Nummer	Bezeichner: \$MM_	Beschreibung
SINUMERIK Operate		
9242	MA_STAT_DISPLAY_BASE	Zahlenbasis für die Anzeige der Gelenkstellung <i>STAT</i>
9243	MA_TU_DISPLAY_BASE	Zahlenbasis für die Anzeige der Rundachsstellung <i>TU</i>
9244	MA_ORIAXES_EULER_ANGLE_NAME	Anzeige der Orientierungsachsen als Euler-Winkel
9245	MA_PRESET_FRAMEIDX	Wertablage Ankratzen und Istwertsetzen
9247	USER_CLASS_BASE_ZERO_OFF_PA	Verfügbarkeit der Basisverschiebung in Bedienbereich "Parameter"
9248	USER_CLASS_BASE_ZERO_OFF_MA	Verfügbarkeit der Basisverschiebung in Bedienbereich Maschine
9424	MA_COORDINATE_SYSTEM	Koordinatensystem für Istwertanzeige
9440	ACTIVE_SEL_USER_DATA	Aktive Daten (Frames) werden nach Änderung sofort wirksam
9449	WRITE_TOA_LIMIT_MASK	Wirksamkeit von MD9203 bezüglich Schneidendaten und ortsabhängige Korrekturen
9450	MM_WRITE_TOA_FINE_LIMIT	Grenzwert für Verschleiß fein
9451	MM_WRITE_ZOA_FINE_LIMIT	Grenzwert für Verschiebung fein

## 9.9.1.2 NC-spezifischen Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10000	AXCONF_MACHAX_NAME_TAB	Maschinenachsname
10600	FRAME_ANGLE_INPUT_MODE	Drehreihenfolge in Frame
10602	FRAME_GEOAX_CHANGE_MODE	Frames beim Umschalten von Geometrieachsen
10610	MIRROR_REF_AX	Bezugsachse für das Spiegeln
10612	MIRROR_TOGGLE	Mirror umschalten
10613	NCBFRAME_RESET_MASK	Aktive NCU-globalen Basisframes nach Reset
10615	NCBFRAME_POWERON_MASK	Globalen Basisframes nach Power On zurücksetzen
10617	FRAME_SAVE_MASK	Verhalten von Frames bei SAVE- Unterprogrammen
10650	IPO_PARAM_NAME_TAB	Name der Interpolationsparameter
10660	INTERMEDIATE_POINT_NAME_TAB	Name der Zwischenpunktkoordinaten bei G2/G3
11640	ENABLE_CHAN_AX_GAP	Kanalachslücken werden erlaubt
18600	MM_FRAME_FINE_TRANS	Feinverschiebung bei FRAME (SRAM)
18601	MM_NUM_GLOBAL_USER_FRAMES	Anzahl der globalen vordefinierten Anwender-Frames (SRAM)
18602	MM_NUM_GLOBAL_BASE_FRAMES	Anzahl der globalen Basisframes (SRAM)

## 9.9.1.3 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20050	AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrieachse zu Kanalachse
20060	AXCONF_GEOAX_NAME_TAB	Geometrieachsname im Kanal
20070	AXCONF_MACHAX_USED	Maschinenachsnummer gültig im Kanal
20080	AXCONF_CHANAX_NAME_TAB	Kanalachsname im Kanal
20110	RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach RESET / TP-Ende
20118	GEOAX_CHANGE_RESET	Automatischen Geometrieachswechsel erlauben
20126	TOOL_CARRIER_RESET_VALUE	Wirksamer Werkzeugträger bei RESET
20140	TRAFO_RESET_VALUE	Transformationsdatensatz im Hochlauf (RESET / TP-Ende)
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen
20152	GCODE_RESET_MODE	RESET-Verhalten der G-Gruppen
20184	TOCARR_BASE_FRAME_NUMBER	Nummer des Basiframes für Aufnahme des Tischoffsets
21015	INVOLUTE_RADIUS_DELTA	Endpunktüberwachung bei Evolvente
22532	GEOAX_CHANGE_M_CODE	M-Code bei Umschaltung der Geometrieachsen
22534	TRAFO_CHANGE_M_CODE	M-Code bei Transformationswechsel
24000	FRAME_ADD_COMPONENTS	Framekomponenten für G58 und G59
24002	CHBFRAME_RESET_MASK	RESET-Verhalten der kanalspezifischen Basisframes

9.9 Datenlisten

Nummer	Bezeichner: \$MC_	Beschreibung
24004	CHBFRAME_POWERON_MASK	Kanalspezifische Basisframes nach Power On zurücksetzen
24006	CHSFRAME_RESET_MASK	Aktive Systemframes nach Reset
24007	CHSFRAME_RESET_CLEAR_MASK	Löschen von Systemframes bei RESET
24008	CHSFRAME_POWERON_MASK	Systemframes nach POWER ON zurücksetzen
24010	PFRAME_RESET_MODE	RESET-Mode für programmierbaren Frame
24020	FRAME_SUPPRESS_MODE	Positionen bei Frameunterdrückung
24030	FRAME_ACT_SET	Einstellung des ENS-Koordinatens
24040	FRAME_ADAPT_MODE	Anpassungen von aktiven Frames
24050	FRAME_SAA_MODE	Speichern und Aktivieren von Datenhaltungsframes
24805	TRACYL_ROT_AX_FRAME_1	Rundachs-Verschiebung TRACYL 1
24855	TRACYL_ROT_AX_FRAME_2	Rundachs-Verschiebung TRACYL 2
24905	TRANSMIT_ROT_AX_FRAME_1	Rundachsverschiebung TRANSMIT 1
24955	TRANSMIT_ROT_AX_FRAME_2	Rundachsverschiebung TRANSMIT 2
28080	MM_NUM_USER_FRAMES	Anzahl der einstellbaren Frames (SRAM)
28081	MM_NUM_BASE_FRAMES	Anzahl Basisframes
28082	MM_SYSTEM_FRAME_FRAMES	Systemframes (SRAM)
28560	MM_SEARCH_RUN_RESTORE_MODE	Restore von Daten nach einer Simulation

9.9.1.4 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32074	FRAME_OR_CORRPOS_NOTALLOWED	FRAME oder HL-Korrektur ist unzulässig
35000	SPIND_ASSIGN_TO_MACHAX	Zuordnung Spindel zu Maschinenachse

9.9.2 Settingdaten

9.9.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42440	FRAME_OFFSET_INCR_PROG	Nullpunktverschiebungen in Frames
42980	TOFRAME_MODE	Frame-Definition bei TOFRAME, TOROT und PAROT

### 9.9.3 Systemvariablen

Bezeichner	Beschreibung
\$AA_ETRANS[Achse]	Externe Nullpunktverschiebung
\$AA_IBN [Achse]	Istwert im Basisnullpunkt-Koordinatensystem (BNS)
\$AA_IEN [Achse]	Aktueller Istwert im einstellbaren Nullpunkt-Koordinatensystem (ENS)
\$AA_OFF[Achse]	Überlagerte Bewegung für die programmierte Achse
\$AC_DRF[Achse]	Handradüberlagerung einer Achse
\$AC_JOG_COORD	Koordinatensystem für das handverfahren
\$P_ACSFRAME	Aktives Frame zwischen BKS und ENS
\$P_ACTBFRAME	Aktives Gesamt-Basisframe
\$P_ACTFRAME	Aktives Gesamtframe
\$P_BFRAME	1. aktives Basisframe im Kanal. Entspricht \$P_CHBFRAME
\$P_CHBFR[n]	Aktives Basisframe im Kanal, über G500, G54...G599 aktivierbar
\$P_CHBFRAME[n]	Aktives Basisframe im Kanal 0 bis 15 NCU-Basisframes sind einstellbar über: MD28081 MM_NUM_BASE_FRAMES
\$P_CHBFRMASK	Basisframemaske im Kanal
\$P_CHSFRMASK	Systemframe-Maske
\$P_CYCFR	Aktives Systemframe für Zyklen
\$P_CYCFRAME	Aktives Systemframe für Zyklen
\$P_EXTFR	Systemframe für Externe Nullpunktverschiebung in der Datenhaltung
\$P_EXTFRAME	Aktives Systemframe für Externe Nullpunktverschiebung
\$P_IFRAME	Aktives einstellbares Frame
\$P_ISO1FR	Datenhaltungsframe für ISO G51.1 Spiegeln
\$P_ISO2FR	Datenhaltungsframe für ISO G68 2DROT
\$P_ISO3FR	Datenhaltungsframe für ISO G68 3DROT
\$P_ISO4FR	Datenhaltungsframe für ISO G51 Scale
\$P_ISO1FRAME	Aktives Systemframe für ISO G51.1 Spiegeln
\$P_ISO2FRAME	Aktives Systemframe für ISO G68 2DROT
\$P_ISO3FRAME	Aktives Systemframe für ISO G68 3DROT
\$P_ISO4FRAME	Aktives Systemframe für ISO G51 Scale
\$P_NCBFR[n]	Globales Basisframein der Datenhaltung, über G500, G54...G599 aktivierbar
\$P_NCBFRAME[n]	Aktueller NCU-Basisframe 0 bis 15 NCU-Basisframes sind einstellbar über: MD18602 MM_NUM_GLOBAL_BASE_FRAMES
\$P_NCBFRMASK	Globale Basisframe-Maske
\$P_PARTFR	Datenhaltungsframe für TCARR und PAROT
\$P_PARTFRAME	Aktives Systemframe für TCARR und PAROT bei orientierbarem Werkzeugträger
\$P_PFRAME	Programmierbarer Frame
\$P_SETFR	Datenhaltungsframe für Istwertsetzen

Bezeichner	Beschreibung
\$P_SETFRAME	Aktives Systemframe für Istwertsetzen
\$P_TOOLFR	Datenhaltungsframe für TOROT und TOFRAME
\$P_TOOLFRAME	Aktives Systemframe für TOROT und TOFRAME
\$P_TRAFR	Datenhaltungsframe für Transformationen
\$P_TRAFRAME	Aktives Systemframe für Transformationen
\$P_UBFR	1. Basisframe im Kanal in der Datenhaltung, das nach G500, G54...G599 aktiviert wird. Entspricht \$P_CHBFR[0].
\$P_UIFR[n]	Einstellbare Datenhaltungsframes, über G500, G54...G599 aktivierbar
\$P_UIFRNUM	Nummer des aktiven einstellbaren Frame \$P_UIFR
\$P_WPFR	Datenhaltungsframe für das Werkstück
\$P_WPFRAME	Aktives Systemframe für das Werkstück

**\$AA\_ETRANS[X]**

\$AA\_ETRANS[X] ist eine achsspezifische Systemvariable vom Typ DOUBLE. Sie ist systemseitig mit dem Wert Null vorbesetzt.

Vom Anwender gesetzte Werte werden aktiviert durch das NC/PLC-Nahtstellensignal:  
DB31, ... DBX3.0 (Externe Nullpunktverschiebung)



## 9.9.4 Signale

### 9.9.4.1 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
T-Funktionsänderung	DB21, ... .DBX61.0-.2	-
D-Funktionsänderung	DB21, ... .DBX62.0-.2	-
T-Funktion 1	DB21, ... .DBB118-119	DB2500.DBD2000
D-Funktion 2	DB21, ... .DBB129	DB2500.DBD5000
Nummer aktive Fkt. G-Gruppe 1 8 (Bit -Int)	DB21, ... .DBB208	DB3500.DBB0
Nummer aktive Fkt. G-Gruppe 2 8 (Bit -Int)	DB21, ... .DBB209	DB3500.DBB1
...	...	...
Nummer aktive Fkt. G-Gruppe 29 8 (Bit -Int)	DB21, ... .DBB236	DB3500.DBB28

### 9.9.4.2 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Externe Nullpunktverschiebung übernehmen	DB31, ... .DBX3.0	-

### 9.9.4.3 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Spindel / keine Achse	DB31, ... .DBX60.0	DB390x.DBX0.0



## N2: Not-Halt

### 10.1 Kurzbeschreibung

#### Funktion

Die Steuerung unterstützt den Maschinenhersteller bei der Realisierung der Not-Halt-Funktion durch folgende Funktionen:

- An allen SINUMERIK-Maschinensteuertafeln ist ein Not-Halt-Taster für den Maschinenbediener leicht erreichbar angebracht. Der Funktionalität des Not-Halt-Tasters umfasst die Zwangsöffnung der elektrischen Schaltkontakte und eine mechanisch selbsttätige Verrastung/Verriegelung.
- Die Not-Halt-Anforderung an die NC erfolgt über die PLC über die NC/PLC-Nahtstelle.
- Bei Not-Halt werden durch die NC alle Achsen und Spindeln mit Sollwert = 0 schnellstmöglich, d. h. an der Stromgrenze der Antriebe, abgebremst.
- Alle durch die PLC gesteuerten Maschinenfunktionen können bei Not-Halt einen, durch den Maschinenhersteller einstellbaren, sicheren Zustand einnehmen.
- Ein Entriegeln des Not-Halt-Tasters bewirkt kein Aufheben des Not-Halt-Zustandes bzw. löst keinen Wiederanlauf aus.
- Nach dem Aufheben des Not-Halt-Zustandes ist kein Referenzieren der Maschinenachsen bzw. Synchronisieren der Spindeln nötig. Die Istpositionen der Maschinenachsen werden während des Not-Halt-Ablaufs kontinuierlich nachgeführt.

## 10.2 Normen

### Relevante Normen

Im Rahmen der Not-Halt-Funktion sind folgende Normen unbedingt zu beachten:

- EN ISO 12000-1
- EN ISO 12000-2
- EN 418
- EN 60204

### Not-Halt

Entsprechend EN 418 ist Not-Halt eine Funktion, die:

- aufkommende oder bestehende Gefahren für Personen, Schäden an der Maschine oder dem Arbeitsgut abwenden oder vermindern soll.
- durch eine einzige Handlung einer Person ausgelöst wird, wenn die normale Haltefunktion dafür nicht angemessen ist.

### Gefahren

Gefahren im Sinne der EN 418 sind solche, die herrühren können von:

- funktionalen Unregelmäßigkeiten (Fehlfunktionen der Maschine, nicht hinnehmbare Eigenschaften des bearbeiteten Materials, menschliche Fehler, ...).
- normalem Betrieb.

## Norm EN ISO 12000-2

Gemäß einer grundlegenden Sicherheitsanforderung der EG-Richtlinie Maschinen hinsichtlich Not-Halt müssen Maschinen mit einer Not-Halt-Einrichtung versehen sein.

### Ausnahmen

Keine Not-Halt-Einrichtung wird bei Maschinen benötigt:

- an denen eine Not-Halt-Einrichtung das Risiko nicht verringern würde, weil dadurch entweder die Stillsetzzeit nicht verringert würde oder weil die dafür zu ergreifenden Maßnahmen nicht geeignet wären, das Risiko zu beeinflussen.
- die von Hand getragen und geführt werden.

<b>ACHTUNG</b>
Der Maschinenhersteller wird auf die Einhaltung der nationalen und internationalen Normen ausdrücklich hingewiesen. Die SINUMERIK-Steuerungen unterstützen den Maschinenhersteller bei der Realisierung der Not-Halt-Funktion entsprechend den Festlegungen der folgenden Funktionsbeschreibung. Die Verantwortung für die Not-Halt-Funktion (Auslösung, Ablauf, Quittierung) liegt aber ausschließlich beim Maschinenhersteller selbst.

## 10.3 Not-Halt-Stellteile

### Not-Halt-Stellteile

Nach EN 418 müssen Not-Halt-Stellteile so konstruiert sein, dass sie mechanisch selbsttätig verrasten und für die Bedienperson und andere im Notfall leicht zu betätigen sind.

Folgende Typen von Stellteilen können unter anderem eingesetzt werden:

- Pilztaster (drucktastenbetätigter Schalter)
- Drähte/Drahtseile, Leinen, Stangen
- Griffe
- In besonderen Fällen: Fußschalter ohne Schutzhaube

### Not-Halt-Taster und Steuerung

Die Betätigung des Not-Halt-Tasters oder ein direkt daraus abgeleitetes Signal muss als PLC-Eingang zur Steuerung (PLC) geführt werden. Im PLC-Anwenderprogramm muss dieser PLC-Eingang weitergeleitet werden an die NC auf das Nahtstellensignal:

DB10 DBX56.1 (Not-Halt)

Das Rückstellen des Not-Halt-Tasters oder ein direkt daraus abgeleitetes Signal muss als PLC-Eingang zur Steuerung (PLC) geführt werden. Im PLC-Anwenderprogramm muss dieser PLC-Eingang weitergeleitet werden an die NC auf das Nahtstellensignal:

DB10 DBX56.2 (Not-Halt quittieren)

### Anschlussbedingungen

Zum Anschluss des Not-Halt-Tasters siehe:

**Literatur:**

Gerätehandbuch Bedienkomponenten

## 10.4 Not-Halt-Ablauf

Nach Betätigung des Not-Halt-Stellteils muss die Not-Halt-Einrichtung in einer Weise arbeiten, dass die Gefahr automatisch auf die bestmögliche Weise abgewendet oder verringert wird.

"Auf bestmögliche Weise" bedeutet, dass die günstigste Verzögerungsrate gewählt und die richtige Stopp-Kategorie (definiert in EN 60204) entsprechend einer Risikoabschätzung festgelegt werden kann.

### Not-Halt-Ablauf in der NC

Der nach EN 418 vorbestimmte Ablauf interner Funktionen zum Not-Halt-Zustand sieht in der Steuerung wie folgt aus:

1. Die Teileprogrammabarbeitung wird unterbrochen.

Alle Maschinenachsen werden abgebremst in der jeweiligen achsspezifisch parametrisierten Zeit:

MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME (Zeit der Bremsrampe bei Fehlern)

Die maximale Bremsrampe, die dabei erzielt werden kann, wird definiert durch den maximalen Bremsstrom des jeweiligen Antriebs. Der maximale Bremsstrom wird erreicht durch Vorgabe von Sollwert = 0 (Schnellbremsung).

2. Zurücksetzung des Nahtstellensignals:

DB11 DBX6.3 (BAG-betriebsbereit)

3. Setzen des Nahtstellensignals:

DB10 DBX106.1 (Not-Halt aktiv)

4. Alarm 3000 "Not-Halt" wird angezeigt.

5. Nach Ablauf einer parametrierbaren Verzögerungszeit werden die Reglerfreigaben der Maschinenachsen zurückgesetzt.

Die Einstellung der Verzögerungszeit erfolgt im Maschinendatum:

MD36620 \$MA\_SERVO\_DISABLE\_DELAY\_TIME (Abschaltverzögerung Reglerfreigabe)

Folgende Einstellregel ist zu beachten:  $MD36620 \geq MD36610$

6. Alle Maschinenachsen werden steuerungsintern in den Nachführbetrieb geschaltet.

Die Maschinenachsen sind dabei nicht mehr in Lageregelung.

### Not-Halt-Ablauf an der Maschine

Der Not-Halt-Ablauf an der Maschine wird ausschließlich vom Maschinenhersteller bestimmt. Dabei ist in Verbindung mit dem Ablauf in der NC Folgendes zu beachten:

- Der Ablauf in der NC wird gestartet mit dem Nahtstellensignal:

DB10 DBX56.1 (Not-Halt)

Nachdem die Maschinenachsen im Stillstand sind, muss nach EN 418 die Energiezufuhr unterbrochen werden.

---

#### Hinweis

Das Unterbrechen der Energiezufuhr liegt in der Verantwortung des Maschinenherstellers.

---

- Die digitalen und analogen Ausgänge der PLC-Peripherie werden vom Not-Halt-Ablauf in der NC nicht beeinflusst.

Sollen einzelne Ausgänge bei Not-Halt einen bestimmten Zustand oder Spannungspegel einnehmen, ist dies vom Maschinenhersteller im PLC-Anwenderprogramm vorzunehmen.

- Die schnellen digitalen Ausgänge der NCK-Peripherie werden vom Not-Halt-Ablauf in der NC nicht beeinflusst.

Sollen einzelne Ausgänge bei Not-Halt einen bestimmten Zustand einnehmen, muss der Maschinenhersteller im PLC-Anwenderprogramm den gewünschten Zustand an die NC übergeben über die Nahtstellensignale:

DB10 DBB4-7

---

#### Hinweis

Soll bei Not-Halt der Ablauf in der NC nicht wie oben beschrieben ablaufen, darf bis zum Erreichen eines durch den Maschinenhersteller im PLC-Anwenderprogramm festgelegten Not-Halt-Zustandes das Nahtstellensignal DB10 DBX56.1 (Not-Halt) nicht gesetzt werden.

Solange das Nahtstellensignal nicht gesetzt ist und kein anderer Alarm ansteht, sind in der NC alle Nahtstellensignale wirksam. Dadurch kann jeder herstellenspezifische Not-Halt-Zustand (auch Achs-, Spindel- und Kanal-spezifisch) eingenommen werden.

---



## 10.5 Not-Halt-Quittierung

Nach EN 418 darf das Rückstellen des Not-Halt-Stellteils nur als Ergebnis einer von Hand ausgeführten Handlung am Not-Halt-Stellteil möglich sein.

Das Rückstellen des Not-Halt-Stellteils allein darf keinen Wiederanlauf-Befehl auslösen.

Der Wiederanlauf der Maschine darf nicht möglich sein, bis alle betätigten Not-Halt-Stellteile von Hand einzeln und bewusst rückgestellt worden sind.

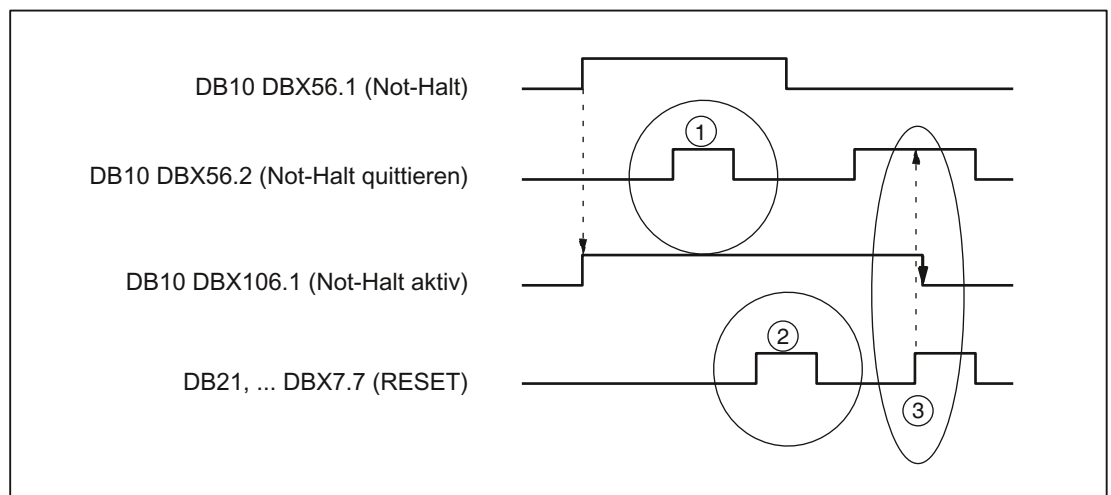
### Not-Halt-Quittierung

Der Not-Halt-Zustand wird nur dann wieder zurückgesetzt, wenn zuerst das Nahtstellensignal DB10 DBX56.2 (Not-Halt quittieren) und anschließend das Nahtstellensignal DB11 DBX0.7 (BAG-Reset) gesetzt wird.

Dabei ist zu beachten, dass das Nahtstellensignal DB10 DBX56.2 (Not-Halt quittieren) und das Nahtstellensignal DB21, ... DBX7.7 (Reset) gemeinsam mindestens so lange gesetzt sind, bis das Nahtstellensignal DB10 DBX106.1 (Not-Halt aktiv) zurückgesetzt wurde.

#### Hinweis

Allein mit dem Nahtstellensignal DB21, ... DBX7.7 (Reset) kann der Not-Halt-Zustand nicht zurückgesetzt werden.



- (1) DB10 DBX56.2 (Not-Halt quittieren) ist wirkungslos
- (2) DB21, ... DBX7.7 (Reset) ist wirkungslos
- (3) DB10 DBX56.2 und DB21, ... DBX7.7 setzen DB10 DBX106.1 (Not-Halt aktiv) zurück

Bild 10-1 Not-Halt-Zustand rücksetzen

## Auswirkungen

Durch Rücksetzen des Not-Halt-Zustandes werden:

- steuerungsintern für alle Maschinenachsen:
  - die Reglerfreigaben gesetzt.
  - der Nachführbetrieb aufgehoben.
  - die Lageregelung aktiviert.
- die folgenden Nahtstellensignale gesetzt:  
DB31, ... DBX60.5 (Lageregelung aktiv)  
DB11 DBX6.3 (BAG betriebsbereit)
- das folgende Nahtstellensignal zurückgesetzt:  
DB10 DBX106.1 (Not-Halt aktiv)
- der Alarm 3000 "Not-Halt" gelöscht.
- die Teileprogrammbearbeitung in allen Kanälen der NC abgebrochen.

## PLC- und NCK-Peripherie

Die PLC- und NCK-Peripherie müssen durch das PLC-Anwenderprogramm wieder in den Zustand zum Betrieb der Maschine versetzt werden.

## POWER OFF/ON (Netz aus/ein)

Der Not-Halt-Zustand kann auch durch Aus- und Einschalten der Steuerung (POWER OFF/ON) zurückgesetzt werden.

### Voraussetzung:

Beim Hochlauf der Steuerung darf das Nahtstellensignal DB10 DBX56.1 (Not-Halt) nicht gesetzt sein.

## 10.6 Datenlisten

### 10.6.1 Maschinendaten

#### 10.6.1.1 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
36610	AX_EMERGENCY_STOP_TIME	Zeitdauer der Bremsrampe bei Fehlerzuständen
36620	SERVO_DISABLE_DELAY_TIME	Abschaltverzögerung Reglerfreigabe

### 10.6.2 Signale

#### 10.6.2.1 Signale an NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Not-Halt	DB10.DBX56.1	DB2600.DBX0.1
Not-Halt quittieren	DB10.DBX56.2	DB2600.DBX0.2

#### 10.6.2.2 Signale von NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Not-Halt aktiv	DB10.DBX106.1	DB2700.DBX0.1

#### 10.6.2.3 Signale an BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
BAG-Reset	DB11.DBX0.7	DB3000.DBX0.7



## P1: Planachsen

### 11.1 Kurzbeschreibung

#### Planachse

Mit Planachse wird im Rahmen der Technologie "Drehen" die Maschinenachse bezeichnet, die senkrecht zur Symmetrieachse der Spindel bzw. zur Längsachse Z verfährt.

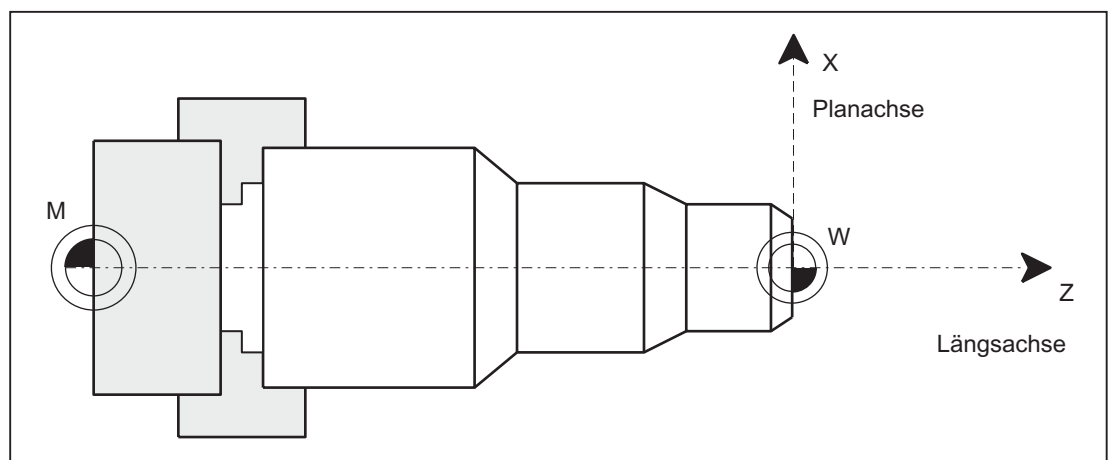


Bild 11-1 Lage der Planachse im Maschinenkoordinatensystem

#### Eigenschaften

- Jede Geometrieachse eines Kanals kann als Planachse definiert werden.
- Eine Planachse ist eine Linearachse, für die folgende Funktionen gleichzeitig oder getrennt zugelassen und aktivierbar sind:
  - Programmierung und Anzeige im Durchmesser
  - Bezugsachse für konstante Schnittgeschwindigkeit G96/G961/G962

**Mehrere Planachsen im Kanal**

Die Einführung mehrerer Planachsen im Kanal bringt eine funktionelle Entkopplung von Durchmesserprogrammierung und Bezugsachse für G96/G961/G962 mit sich. Durchmesserprogrammierung und Bezugsachse für G96/G961/G962 können für unterschiedliche Planachsen aktiv sein (siehe folgende Tabelle).

	Programmierung und Anzeige im Durchmesser			Bezugsachse für G96 / G961 / G962	
<b>Zulässiger Achsentyp:</b>	Geometrieachse		Lineare Kanalachsen	Geometrieachse	
<b>Auswahl im Kanal:</b>	eine	m aus 3	m aus n	eine	eine aus 3
<b>Spezifische Wirkung:</b>	Kanal	Achse		Kanal	
<b>Maschinendaten:</b>	MD20100	MD30460		MD20100	
<b>Programmierung:</b>	DIAM* kanalspezifisch modal G-Gruppe 29			SCC [AX] kanalspezifisch modal Bezugsachse für G96/G961/G962	
<b>Übernahme bei Achstausch:</b>	DIAM*A [AX] achsspezifisch modal				
<b>Achsspezifische nicht modale Durchmesser- / Radiusprogrammierung</b>	DAC, DIC; RAC, RIC achsspezifisch satzweise nur Programmierung				
DIAM*: DIAMOF, DIAMON, DIAM90, DIAMCYCOF DIAM*A[AX]: DIAMOFA[AX], DIAMONA[AX], DIAM90A[AX], DIACYCOFA[AX], DIAMCHANA[AX] AX: Achsbezeichner für Geo-/Kanal- oder Maschinenachsbezeichner					

**Hinweis**

Rundachsen sind nicht als Planachsen zugelassen.

## Programmierung der Verfahrswege

Die im Teileprogramm programmierten Verfahrswege einer Planachse können wahlweise Radius- oder Durchmesser-bezogen erfolgen. Die Umschaltung der beiden Bezugsarten erfolgt über die Teileprogrammbefehle `DIAMON` (DIAMeter ON = Durchmesser) und `DIAMOF` (DIAMeter OF = Radius). Somit können Maßangaben ohne Umrechnung direkt aus der technischen Zeichnung entnommen werden.

### Aktives Teileprogramm

Bei aktivem Teileprogrammbefehl `DIAMON` (Maßangaben als Durchmesser) gilt für die Planachse:

- Die Anzeige der Soll- und Istwerte, die sich auf das Werkstückkoordinatensystem beziehen, erfolgt als Durchmesserangabe.
- Systemvariablen für Soll- und Istwerte, die sich auf das Werkstückkoordinatensystem beziehen, enthalten Durchmesserangaben.
- Verschiebungsangaben werden als Radius eingegeben, programmiert und angezeigt.
- Programmierte Endpositionen werden intern in Radiuswerte umgerechnet.
- Die absoluten Interpolationsparameter (z. B. I, J, K) der Kreisinterpolation (G2 und G3) werden intern in Radiuswerte umgerechnet.
- Messergebnisse, die mit schaltendem Messtaster im Werkstückkoordinatensystem ermittelt wurden, werden als Durchmesser abgelegt.
- Soll- und Istwerte mit Systemvariablen im WKS können im Durchmesser gelesen werden.

Bei aktivem Teileprogrammbefehl `DIAMOF` (Maßangaben als Radius) werden die oben genannten Daten immer im Radius eingegeben, programmiert, intern abgelegt, gelesen oder angezeigt.

## 11.2 Geometrieachse als Planachse definieren

### Definition einer Planachse im Kanal

Die Definition einer Geometrieachse als Planachse erfolgt mit dem Maschinendatum:

MD20100 \$MC\_DIAMETER\_AX\_DEF (Geometrieachse mit Planachsfunktion)

Beispiel:

MD20100 \$MC\_DIAMETER\_AX\_DEF="X" ; Die Geometrieachse X ist Planachse im Kanal.

Für diese Achse ist sowohl Durchmesserprogrammierung als auch die Zuordnung einer Konstanten Schnittgeschwindigkeit mit G96/G961/G962 zugelassen.

### Mehrere Planachsen im Kanal

Das achsspezifische Maschinendatum:

MD30460 BASE\_FUNCTION\_MASK (Achsfunktionen)

ermöglicht die Definition weiterer Planachsen, für die die Funktionalität der achsspezifischen Durchmesserprogrammierung zugelassen ist:

Bit	Wert	Bedeutung
2	0	Achsspezifische Durchmesserprogrammierung ist <b>nicht</b> zugelassen.
	1	Achsspezifische Durchmesserprogrammierung ist zugelassen.

### Hinweis

Die Einstellung MD30460 Bit 2 = 1 ist nur für Linearachsen möglich.

Eine Achse darf gleichzeitig im MD20100 und im MD30460 (Bit 2) definiert sein. Dabei hat das kanalspezifische MD20100 gegenüber dem achsspezifischen MD30460 eine höhere Priorität.

Mit:

- MD20100 wird der Planachse im Hochlauf die Funktion G96/G961/G962 zugeordnet.
- MD20100 wird der Planachse im Hochlauf die kanalspezifische Durchmesserprogrammierung DIAMON, DIAMOF, DIAM90, DIAMCYCOF zugeordnet.  
Diese Achse besitzt nach Hochlauf die achsspezifische Grundstellung DIAMCHANA[AX].
- MD30460 Bit2 die zusätzliche Freigabe der achsspezifischen Anweisungen DIAMONA[AX], DIAMOFA[AX], DIAM90A[AX], DIACYCOFA[AX], DIMCHANA[AX].



### Kanalspezifische Grundstellung nach Hochlauf, RESET

Die kanalspezifische Grundstellung nach Hochlauf bzw. RESET oder Teileprogrammende der G-Gruppe 29: DIAMON, DIAM90, DIAMOF, DIAMCYCOF definieren das

MD20150 \$MC\_GCODE\_RESET\_VALUE

und abhängig von

MD20110 \$MC\_RESET\_MODE\_MASK / Bit0 das MD20152 \$MC\_GCODE\_RESET\_MODE.

Der Anwender kann über einen ereignisgesteuerten Programmaufruf (Prog-Event) den jeweils gewünschten Zustand einstellen.

Ist G96/G961/G962 Grundstellung nach Hochlauf, muss mit MD20100 \$MC\_DIAMETER\_AX\_DEF eine Planachse definiert sein, anderenfalls erfolgt die Alarmmeldung 10870.

Bezugsachse für G96/G961/G962 beibehalten:

MD20110 \$MC\_RESET\_MODE\_MASK, Bit 18=1 bei RESET oder Teileprogrammende

MD20112 \$MC\_START\_MODE\_MASK, Bit 18=1 bei Teileprogrammstart

Eine Bezugsachse für G96/G961/G962 kann auch ohne Applizierung einer Planachse in MD20100 über SCC[AX] zugeordnet werden. Für diesen Fall darf die Konstante Schnittgeschwindigkeit nicht mit G96 eingeschaltet sein. Weitere Informationen siehe:

#### Literatur

Programmierhandbuch Grundlagen, Vorschubregelung und Spindelbewegung

"Konstante Schnittgeschwindigkeit (G96, G961, G962, G97, G971, LIMS, ACC[AX])"

## 11.3 Maßangaben von Planachsen

Planachsen können sowohl Durchmesser- als auch Radius-bezogen programmiert werden. In der Regel werden sie Durchmesser-bezogen, d. h. mit doppeltem Wegmaß programmiert, um aus technischen Zeichnungen die entsprechenden Maßangaben direkt in das Teileprogramm übernehmen zu können.

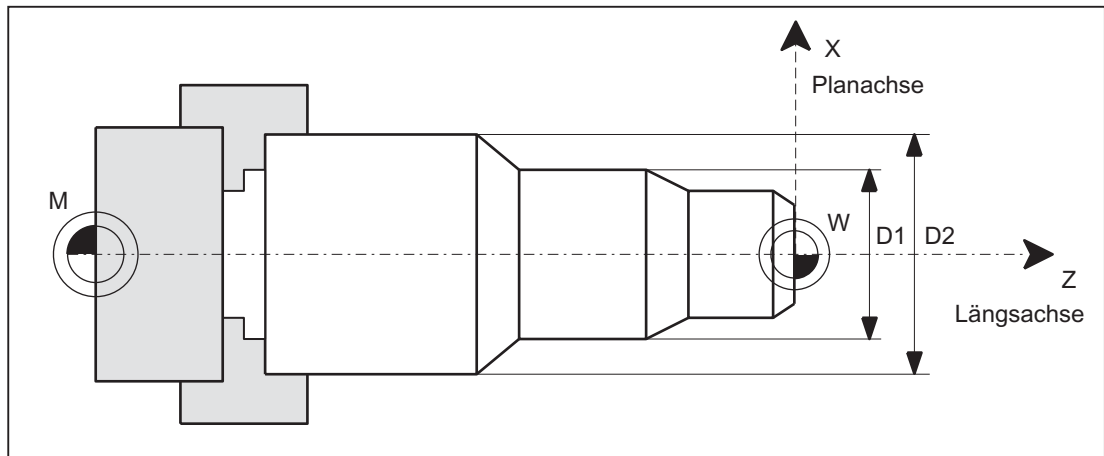


Bild 11-2 Planachse mit Durchmesserangabe (D1, D2)

### Ein/Ausschalten der Durchmesserprogrammierung

#### Kanalspezifische Durchmesserprogrammierung

Das Ein bzw. Ausschalten der Durchmesserprogrammierung erfolgt über die modal wirksamen Teileprogrammanweisungen der G-Gruppe 29:

- **DIAMON**: Durchmesserprogrammierung EIN
- **DIAMOF**: Durchmesserprogrammierung AUS bzw. Radiusprogrammierung EIN
- **DIAM90**: Durchmesser- bzw. Radiusprogrammierung abhängig vom Bezugsmodus:
  - Durchmesserprogrammierung EIN im Zusammenhang mit Absolutmaßangabe  $G_{90}$
  - Radiusprogrammierung EIN im Zusammenhang mit Kettenmaßangabe  $G_{91}$
- **DIAMCYCOF**: Radiusprogrammierung für  $G_{90}$  und  $G_{91}$  EIN, zum HMI bleibt der zuletzt aktive G-Code dieser Gruppe weiterhin aktiv

Es ist ausschließlich auf die Planachse des Kanals bezogen.

## Achsspezifische Durchmesserprogrammierung für mehrere Planachsen im Kanal

### Hinweis

Die zusätzlich angegebene Achse muss über das MD30460 \$MA\_BASE\_FUNCTION\_MASK mit Bit2=1 aktiviert werden.

Die angegebene Achse muss eine im Kanal bekannte Achse sein. Zugelassen sind Geometrie- Kanal- oder Maschinenachsen.

Eine Programmierung in Synchronaktionen ist nicht zugelassen.

Folgende achsspezifische modale Anweisungen können mehrmals in einem Teileprogrammsatz programmiert werden:

- DIAMONA [Achse]: Durchmesserprogrammierung für G90, G91 AC und IC EIN
- DIAMOFA [Achse]: Durchmesserprogrammierung AUS bzw. Radiusprogrammierung EIN
- DIAM90A [Achse]: Durchmesser- bzw. Radiusprogrammierung abhängig vom Bezugsmodus:
  - Durchmesserprogrammierung EIN im Zusammenhang mit Absolutmaßangabe G90 und AC
  - Radiusprogrammierung EIN im Zusammenhang mit Kettenmaßangabe G91 und IC
- DIACYCOFA [Achse]: Radiusprogrammierung für G90 und G91 EIN, zum HMI bleibt der zuletzt aktive G-Code dieser Gruppe weiterhin aktiv
- DIAMCHANA [Achse]: Übernahme Kanalzustand Durchmesserprogrammierung
- DIAMCHAN: alle Achsen mit MD30460, Bit2=1 übernehmen den Kanalzustand Durchmesserprogrammierung

Achsspezifische modale Anweisungen sind dominant gegenüber der Kanaleinstellung.

## Übername der zusätzlichen Planachse im Kanal

Aufgrund einer GET Anforderung aus dem Teileprogramm wird beim Achstausch mit RELEASE[Achse] der Zustand Durchmesserprogrammierung für eine zusätzliche Planachse im neuen Kanal übernommen.

### Achstausch in Synchronaktionen

Bei Achstausch in Synchronaktionen nimmt eine Planachse den Zustand der achsspezifischen Durchmesserprogrammierung in den neuen Kanal mit, falls für die Planachse:

- mit MD30460, Bit2=1 achsspezifische Durchmesserprogrammierung zugelassen ist.
- nicht der kanalspezifischen Durchmesserprogrammierung im abgebenden Kanal unterstellt ist.

Über die Systemvariable \$AA\_DIAM\_STAT[AX] kann die aktive Maßangabe abgefragt werden.

### Achstausch über Achscontainer-Drehung

Durch Achscontainer-Drehung kann sich die Zuordnung einer Kanalachse zu einer Maschinenachse ändern. Der aktuelle Zustand der Durchmesserprogrammierung bleibt aber für die Kanalachse nach der Drehung erhalten. Dies betrifft auch den aktuellen Kanal- und Achs-Zustand, da zum Zeitpunkt von Maschinendaten "wirksam Setzen" der Zustand aus dem MD30460 \$MA\_BASE\_FUNCTION\_MASK für alle Achsen des Achscontainers gleich ist.

### Löschstellung

Die Parametrierung der Löschstellung erfolgt über das Maschinendatum:

MD20150 \$MC\_GCODE\_RESET\_VALUES [28] (Löschstellung der G-Gruppen)  
und abhängig von MD20110 \$MC\_RESET\_MODE\_MASK bei Bit0 das  
MD20152 \$MC\_GCODE\_RESET\_MODE

### Durchmesser-bezogene Daten

Nach dem Einschalten der Durchmesserprogrammierung beziehen sich folgende Daten auf Durchmesserangaben:

### DIAMON/DIAMONA[AX]

- Anzeigedaten der Planachse im Werkstückkoordinatensystem:
  - Soll- und Istposition
  - Restweg
  - REPOS-Verschiebung
- Betriebsart "JOG":
  - Inkremente für Schrittmaß (INC) und Handradfahren (abhängig von wirksamen MD)
- Teileprogrammprogrammierung:
  - Endpositionen, unabhängig vom Bezugsmodus ( $G_{90}$  /  $G_{91}$ )
  - Interpolationsparameter der Kreisprogrammierung ( $G_{2}$  /  $G_{3}$ ), falls diese mit Teileprogrammanweisung:  $AC$  absolut programmiert sind.
- Mit Bezug auf das Werkstückkoordinatensystem (WKS) gelesene Istwerte:
  - \$AA\_MW[Planachse]  
Systemvariable der Messfunktionen  $MEAS$  (Messen mit Restweglöschen) und  $MEAW$  (Messen ohne Restweglöschen)
  - \$P\_EP[Planachse]
  - \$AA\_IW[Planachse]

## DIAM90/DIAM90A[AX]

Nach dem Einschalten der Bezugsmodus-abhängigen Durchmesserprogrammierung werden folgende Daten unabhängig vom Bezugsmodus ( $G_{90}$  /  $G_{91}$ ) immer Durchmesser-bezogen angezeigt:

- Istwert
- Mit Bezug auf das Werkstückkoordinatensystem (WKS) gelesene Istwerte:
  - \$AA\_MW[Planachse]  
Systemvariable der Messfunktionen MEAS (Messen mit Restweglöschen) und MEAW (Messen ohne Restweglöschen)
  - \$P\_EP[Planachse]
  - \$AA\_IW[Planachse]

## DIAMCYCOF/DIACYCOFA[AX]

Analog zu DIAMCYCOF erfolgt bei DIACYCOFA[AX] steuerungsintern eine Umschaltung auf Radiusprogrammierung. Es wird der vor DIAMCYCOF bzw. DIACYCOFA[AX] aktive Zustand bezüglich Durchmesserprogrammierung weiterhin zum HMI angezeigt.

## Fest Radius-bezogene Daten

Für Planachsen werden folgende Daten **immer** Radius-bezogen eingegeben, programmiert und angezeigt:

- Verschiebungen:
  - Werkzeugkorrekturen
  - programmierbare und einstellbare Frames
  - externe Nullpunktverschiebung
  - DRF- und Preset-Verschiebung
  - etc.
- Arbeitsfeldbegrenzung
- SW-Endschalter
- Vorschub
- Anzeigedaten bezogen auf das Maschinenkoordinatensystem
- Anzeigedaten der Servicebilder für Achse, VSA und HSA

**Funktionserweiterungen für immer Radius-bezogene Daten:**

Für PLC-Achsen, über FC18 oder ausschließlich vom PLC kontrollierte Achsen gilt:

- Die Maßangabe für PLC-Achsen im Radius gilt auch für mehrere Planachsen mit Durchmesserfunktion und ist unabhängig von kanalspezifischer oder achsspezifischer Durchmesserprogrammierung.
- In der Betriebsart JOG (Inc) unterliegt eine PLC-Achse dem Kanalzustand. Ist Durchmesserprogrammierung aktiv und MD20624 \$MC\_HANDWH\_CHAN\_STOP\_COND Bit 15 = 0, wird nur der halbe Weg des vorgegebenen Inkrements verfahren.

**Radiusprogrammierung** aus MD20100 \$MC\_DIAMETER\_AX\_DEF und MD30460 \$MA\_BASE\_FUNCTION\_MASK Bit 2 wird abhängig von MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK wie folgt berücksichtigt:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK		
Bit	Wert	Bedeutung
3	0	Nullpunktverschiebung \$P_EXTFRAME und Frames Für Planachsen werden Nullpunktverschiebungen in Frames immer als Radiuswerte eingerechnet.
5	0	Externe Nullpunktverschiebung (Achsüberlagerung) Für Planachsen wird die externe Nullpunktverschiebung immer als Radiuswerte eingerechnet.
8	1	Anzeige Restweg im WKS immer als Radius
9		Für alle Planachsen wird mit MD11346 \$MN_HANDWH_TRUE_DISTANCE==1
	0	<ul style="list-style-type: none"> <li>• der halbe Weg des vorgegebenen Handradinkrements verfahren, falls für diese Achse kanalspezifische oder achsspezifische Durchmesserprogrammierung aktiv ist.</li> </ul>
	1	<ul style="list-style-type: none"> <li>• immer der halbe Weg des vorgegebenen Handradinkrementes verfahren.</li> </ul>
13	1	Beim Joggen von Kreisen ist die Kreismittelpunktskoordinate immer ein Radiuswert, siehe SD42690 \$SC_JOG_CIRCLE_CENTRE
14	1	Bei Zyklenmasken sind Absolutwerte der Planachse im Radius.

**Positionswerte im Durchmesser anzeigen**

Positionswerte der Planachse werden immer als Durchmesserwert angezeigt, wenn von MD27100 \$MC\_ABSBLOCK\_FUNCTION\_MASK das Bit 0 = 1 gesetzt wird.

### Maßangabe auf mehrere Planachsen fest Durchmesser bezogene Daten

Mehrere mit MD30460 \$MA\_BASE\_FUNCTION\_MASK, Bit 2 = 1 zugelassene Planachsen verhalten sich nicht abweichend gegenüber einer mit MD20100 \$MC\_DIAMETER\_AX\_DEF definierten Planachse. Durchmesserwerte werden weiterhin in Radiuswerte umgerechnet.

Per MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK sind für alle im Kanal definierten Planachsen die folgenden Funktionen als Durchmesser aktivierbar:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK		
Bit	Wert	Bedeutung
1	1	Werkzeuglänge Planachse als Durchmesser
2	1	Alarm bei Verschleiß oder Werkzeuglänge als Durchmesser und Ebenenwechsel
3	1	Nullpunktverschiebung (NPV) in Frames der Planachse als Durchmesser
4	1	Preset-Wert als Durchmesser
5	1	Externe NPV Planachse als Durchmesser
6	1	Istwerte der Planachse als Durchmesser
7	1	Anzeige Istwerte der Planachse als Durchmesserwert
10	1	Werkzeuganteil eines aktiven orientierbaren Werkzeugträgers wenn kein Werkzeug aktiv ist
11	1	Bewertung von \$TC_DP6 als Durchmesser
12	1	Bewertung von \$TC_DP15 als Verschleiß des Werkzeugdurchmessers
15	1	Inkrementalwerte der Planachse bei Zyklenmasken als Durchmesser

#### Nullpunktverschiebung \$P\_EXTFRAME und Frames

Bit 3 = 1: Für alle Planachsen werden Nullpunktverschiebungen in Frames immer als Durchmesserwerte eingerechnet. Der Frame legt die Nullpunktverschiebungen intern als Radiuswert ab. Es erfolgt keine Umrechnung bei Wechsel von Durchmesser- auf Radiusprogrammierung bzw. umgekehrt.

#### Externe Nullpunktverschiebung

Bit 5 = 1: Für alle Planachsen wird externe Nullpunktverschiebung immer als Durchmesserwerte eingerechnet. Es erfolgt keine Umrechnung bei Wechsel von Durchmesser- auf Radiusprogrammierung bzw. umgekehrt.

### Einstellbares Verhalten von Geometrieachsen beim Handradfahren

Falls die Geometrieachse als Planachse im Kanal verfahren wird, kann beim Handradfahren MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE == 1 das Verhalten des Handradfahrens über MD20624 \$MC\_HANDWH\_CHAN\_STOP\_COND, Bit15 verändert werden:

Bit 15 = 0: Es wird nur der halbe Weg des vorgegebenen Inkrements verfahren.

Bit 15 = 1: Es wird das vorgegebene Inkrement vollständig verfahren.

### Anwendungsbeispiele

X ist eine über MD20100 \$MC\_DIAMETER\_AX\_DEF definierte Planachse.  
 Y ist eine Geometrieachse und U eine Zusatzachse. Diese beiden Achsen sind weitere im MD30460 \$MA\_BASE\_FUNCTION\_MASK mit Bit2=1 definierte Planachsen mit Durchmesserangabe. DIAMON ist nach Hochlauf nicht aktiv.

```

N10 G0 G90 X100 Y50           ;keine Durchmesserprogrammierung aktiv
N20 DIAMON                    ;Durchmesserprogrammierung kanalspezifisch, wirksam
                               für X
N30 Y200 X200                 ;Maßangabe: X im Durchmesser, Y im Radius
N40 DIAMONA[Y]                ;achsspezifische modale Durchmesserprogrammierung,
                               ;wirksam für Y
N50 Y250 X300                 ;Maßangabe: X und Y im Durchmesser
N60 DIAM90                     ;Maßangabe: X G90/AC im Durchmesser, G91/IC im Radius
N70 Y200                       ;Y: weiterhin achsspezif. modale
                               Durchmesserprogrammierung
N75 G91 Y20 U=DIC(40)         ;Maßangabe: Y im Durchmesser, U satzweise IC im
                               Durchmesser
N80 X50 Y100                  ;Maßangabe: X im Radius (G91), Y im Durchmesser
N85 G90 X100 U200             ;Maßangabe: X im Durchmesser, U im Radius
N90 DIAMCHANA[Y]              ;Y übernimmt den Kanalzustand DIAM90
N95 G91 X100 Y100             ;Maßangabe: X und Y im Radius(G91)
N100 G90 X200 Y200            ;Maßangabe: X und Y im Durchmesser
    
```

### Beispiel mit Achstausch

Planachsen mit Durchmesserangabe wie im vorherigen Beispiel appliziert.  
 X und Y befinden sich im Kanal 1 und sind zusätzlich im Kanal 2 bekannt, d.h. für Achstausch zugelassen.

```

Kanal 1
N10 G0 G90 X100 Y50           ;keine Durchmesserprogrammierung aktiv
N20 DIAMON                    ;Durchmesserprogrammierung kanalspezifisch für X
N30 Y200 X200                 ;Maßangabe: X im Durchmesser, Y im Radius
N40 DIAMONA[Y]                ;Y achsspezifische modale Durchmesserprogrammierung
N50 Y250 X300                 ;Maßangabe: X und Y im Durchmesser
N60 SETM(1)                   ;Synchronmarke 1
N70 WAIT(1,2)                 ;warten auf Synchronmarke 1 im Kanal 2
Kanal 2
...
N50 DIAMOF                     ;Kanal 2 keine Durchmesserprogrammierung aktiv
...
N100 WAIT(1,1)                ;warten auf Synchronmarke 1 im Kanal 1
N110 GETD(Y)                  ;Achstausch direkt Y
N120 Y100                     ;Y wird der kanalspezifischen Durchmesserprogrammierung
                               ;im Kanal 2 unterstellt, d.h. Maßangabe im Radius
    
```



## 11.4 Datenlisten

### 11.4.1 Maschinendaten

#### 11.4.1.1 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20050	AXCONF_GEOAX_ASSIGN_TAB[n]	Zuordnung Geometrieachse zu Kanalachse
20060	AXCONF_GEOAX_NAME_TAB[n]	Geometrieachsname im Kanal
20100	DIAMETER_AX_DEF	Geometrieachse mit Planachsfunktion
20110	RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach Hochlauf und RESET/Teileprogrammende
20112	START_MODE_MASK	Festlegung der Steuerungs-Grundstellung bei NC-Start
20150	GCODE_RESET_VALUES[n]	Löschstellung der G-Gruppen
20152	GCODE_RESET_MODE[n]	G-Code Grundstellung bei RESET/Teileprogrammende
20360	TOOL_PARAMETER_DEF_MASK	Definition der WZ-Parameter
20624	HANDWH_CHAN_STOP_COND	Festlegung des Verhaltens des Handradfahrens
27100	ABSBLOCK_FUNCTION_MASK	Satzanzeige mit Absolutwerten parametrieren

#### 11.4.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30460	BASE_FUNCTION_MASK	Achsfunktionen



## 12.1 Kurzbeschreibung

### Allgemeines

Das PLC-Grundprogramm organisiert den Austausch von Signalen und Daten zwischen dem PLC-Anwenderprogramm und dem NCK-, HMI- und MCP-Bereich. Bei den Signalen und Daten wird zwischen folgenden Gruppen unterschieden:

- Zyklischer Signalaustausch
- Ereignisgesteuerter Signalaustausch
- Meldungen

### Zyklischer Signalaustausch

Die Signale, die zyklisch ausgetauscht werden, bestehen im Wesentlichen aus Bitfeldern.

- Sie enthalten **Befehle**, die von der PLC an den NCK übergeben werden (z. B. Start, Stopp etc.) und **Status-Informationen** des NCK (z. B. Programm läuft, Programm unterbrochen etc.).
- Die Bitfelder sind gegliedert in Signale für:
  - BAG
  - Kanäle
  - Achsen/Spindeln
  - allgemeine NCK-Signale

Der zyklische Signalaustausch wird vom Grundprogramm am Zyklusanfang der PLC (OB 1) durchgeführt. Dadurch ist sichergestellt, dass z. B. die Signale vom NCK über einen Zyklus konstant bleiben.

### Ereignisgesteuerter Signalaustausch NCK → PLC

PLC-Funktionen, die abhängig vom Werkstückprogramm ausgeführt werden müssen, werden über Hilfsfunktionen im Werkstückprogramm angestoßen. Gelangt ein Satz mit Hilfsfunktionen zur Ausführung, so hängt es von der Art der Hilfsfunktion ab, ob der NCK die Ausführung dieser Funktion abwarten muss (z. B. Werkzeugwechsel) oder ob diese Funktion begleitend zur Werkstückbearbeitung zur Ausführung kommt (z. B. Werkzeugbereitstellung bei Fräsmaschinen mit Kettenmagazinen).

Damit die NCK-Bearbeitung möglichst wenig beeinflusst wird, muss die Datenübergabe möglichst schnell und trotzdem sicher erfolgen. Sie erfolgt deshalb alarm- und quittungsgesteuert. Das Grundprogramm wertet die Signale und Daten aus, quittiert dies zum NCK und überträgt die Daten am Zyklusanfang zur Anwendernahtstelle. Erfordern die Daten keine Anwenderquittung, so wird die NCK-Bearbeitung dadurch nicht beeinflusst.

### Ereignisgesteuerter Signalaustausch PLC → NCK

Immer dann, wenn die PLC an den NCK einen Auftrag übergibt (z. B. Verfahren einer Hilfsachse), findet ein "ereignisgesteuerter Signalaustausch PLC → NCK" statt. Auch hier erfolgt die Datenübergabe quittungsgesteuert. Vom Anwenderprogramm aus wird ein derartiger Signalaustausch über einen FB bzw. FC angestoßen.

Die zugehörigen FBs (Funktionsbausteine) und FCs (Function Calls) werden zusammen mit dem Grundprogramm ausgeliefert.

### Meldungen

Die Erfassung und Aufbereitung der Anwendermeldungen erfolgt durch das Grundprogramm. Über ein vereinbartes Bitfeld werden die Meldesignale an das Grundprogramm übergeben. Dort werden diese Signale ausgewertet und bei Auftreten der Meldeereignisse über die ALARM S/SQ-Funktionen in den Alarm-Puffer der PLC eingetragen. Ist ein HMI (z. B. SINUMERIK Operate) vorhanden, so werden die Meldungen an den HMI übertragen und angezeigt.

### Datenaustausch PLC / HMI

Bei diesem Datenaustausch übernimmt der HMI die Initiative als sogenannter Client am Bussystem. Der HMI fordert Daten an bzw. beschreibt Daten. Die PLC wickelt über das Betriebssystem diese Aufträge zum Zykluskontrollpunkt ab. Das PLC-Grundprogramm ist an diesen Übertragungen nicht beteiligt.

---

#### Hinweis

Die Funktion der Maschine wird maßgebend vom PLC-Programm bestimmt. Jedes im Arbeitsspeicher vorhandene PLC-Anwenderprogramm kann mit dem Programmiergerät geändert werden.

---

### Know How-Schutz für Anwenderbausteine

Zum Schutz des in Anwenderbausteinen (OB, FB und FC) enthaltenen Know How, können diese mit dem in SIMATIC STEP 7 enthaltenen SBP-Tool (SIMATIC Block Protection) verschlüsselt werden. Ohne Angabe des bei der Verschlüsselung verwendeten Passworts, ist dann für diese Bausteine Öffnen, Debuggen und Modifizieren nicht mehr möglich.

Bei der Verschlüsselung ist das Automatisierungssystem anzugeben, auf deren PLC-CPU die Bausteine ausführbar sein sollen: SIMATIC und/oder SINUMERIK PLC-CPU.

Die Handhabung der Bausteine, z.B. Laden in die CPU, wird durch die Verschlüsselung nicht beeinflusst.

#### Voraussetzung

SIMATIC STEP 7 ab Version 5.5 SP3

## 12.2 Eckdaten der PLC-CPU

### Eckdaten der PLC-CPU

Die Übersicht der Eckdaten der in die SINUMERIK NCU integrierten PLC-CPU findet sich in:

#### Literatur

Gerätehandbuch NCU 7x0.3 PN, Kapitel "Technische Daten"

<b>ACHTUNG</b>
<b>E/A-Adressen für integrierte Antriebe</b>
Die E/A-Adressen oberhalb 4096 sind für die integrierten Antriebe der NCU reserviert und dürfen nicht anderweitig belegt werden.

### Funktionen des PLC-Grundprogramms

<b>Umfang</b>	
Achsen / Spindeln	31
Kanäle	10
BAGs	10
<b>Funktionen</b>	
Status- / Steuersignale	+
M-Decoder (M00-99)	+
G-Gruppen-Decoder	+
Hilfsfunktionsverteiler	+
Hilfsfunktionsübergabe alarmgesteuert	+
M-Dekodierung nach Liste	+
Achsen / Spindeln bewegen von PLC	+
ASUP-Schnittstelle	+
Fehler- / Betriebsmeldungen	+
MCP- und BHG-Signale übertragen	+
Display-Steuerung Bedienhandgerät	+
Lesen / Schreiben von NCK-Variablen und GUDs	+
PI-Dienste	+
Werkzeugverwaltung	+
Stern / Dreieck Umschaltung	+
<b>M zu N</b>	+
<b>Safety Integrated</b>	+
<b>Programm-Diagnose</b>	+

## 12.3 PLC-Betriebssystemversion

Die Version des PLC-Betriebssystems wird angezeigt unter:

- Bedienoberfläche von SINUMERIK Operate: "Bedienbereichumschaltung" > "Diagnose" > "Version" ⇒ Versionsdaten /Systemsoftware NCU: Auswahl "PLC" > "Details" ⇒ Versionsdaten /Systemsoftware NCU/PLC: Die PLC-Betriebssystemversion wird in der erste Zeile unter "PLC 3xx..." angezeigt.

### **Hinweis**

Die angezeigte Version ist SINUMERIK-spezifisch. Sie ist nicht kompatibel zur zugrunde liegenden SIMATIC CPU.

- SIMATIC STEP 7, HW Konfig: In den Eigenschaften der PLC-CPU im SINUMERIK Baugruppenträger: "Eigenschaften – CPU 3xx..." > "Bestell-Nr. / Firmware": xxxx / **Vx.y.z**

### **Hinweis**

Es wird die Version der zugrunde liegenden SIMATIC CPU angezeigt.

## 12.4 PLC-Betriebsartenschalter

Der PLC-Betriebsartenschalter befindet sich auf Frontseite der NCU-Baugruppe. Über den PLC-Betriebsartenschalter können folgende PLC-Betriebsarten eingestellt werden:

S <sup>1)</sup>	Bedeutung	Bemerkung
0	RUN-P	Änderungen am PLC-Programm können ohne Aktivierung des Kennwortes erfolgen
1	RUN	Über ein Programmiergerät (PG) sind nur lesende Zugriffe möglich. Erst nach dem Setzen des Kennwortes, können Änderungen am PLC-Programm durchgeführt werden.
2	STOP	Die Bearbeitung des PLC-Anwenderprogramms ist angehalten und alle PLC-Ausgänge auf Ersatzwerte gesetzt.
3	MRES	Die PLC wird in den STOP-Zustand versetzt und anschließend PLC-Urlöschen (Default Data) durchgeführt.
1) Schalterstellung des PLC-Betriebsartenschalters		

### Literatur

Eine ausführliche Beschreibung der Lage des PLC-Betriebsartenschalters auf Frontseite der NCU-Baugruppe, sowie seine Verwendung im Zusammenhang mit NCK- und PLC-Urlöschen findet sich in:

Inbetriebnahmehandbuch CNC: NCK, PLC, Antrieb:

- Kapitel "Einschalten Hochlauf" > "Bedien- und Anzeigeelemente für den Hochlauf"
- Kapitel "Einschalten Hochlauf" > "NCK- und PLC-Urlöschen"
- Kapitel "Allgemeine Tipps" > "Getrenntes NCK- und PLC-Urlöschen"

## 12.5 Ressourcen (Timer, Zähler, FC, FB, DB, Peripherie) reservieren

### Reservierung von Ressourcen (Timer, Zähler, FC, FB, DB, Peripherie)

Folgende Komponenten sind für das Grundprogramm reserviert:

- **Timer**

Keine Reservierung.

- **Zähler**

Keine Reservierung.

- **FC, FB, DB**

Für das Grundprogramm sind FC 0 bis FC 29 und FB 0 bis FB 29 reserviert. Für FC, FB ist zusätzlich das Nummernband von 1000 bis 1023 reserviert. Bei Datenbausteinen sind DB 1 bis DB 62, DB 71 bis DB 80 reserviert. Für DB ist zusätzlich das Nummernband 1000 bis 1099 reserviert. Datenbausteine von nicht aktivierten Kanälen, Achsen / Spindeln, Werkzeugverwaltung sind frei für den Anwender.

- **Peripheriebereich**

Die PLC haben ein Peripherie-Adressvolumen von je 8192 Byte für Ein- / Ausgänge. Hiervon sind für integrierte Antriebe die Adressbereiche ab 4096 / 4096 reserviert bzw. belegt. Diagnoseadressen für Baugruppen können aber in dem obersten Adressbereich platziert werden, so wie es STEP 7 vorschlägt. Weiterhin ist der Adressbereich 256 bis 287 belegt für NCK, CP und HMI im Rack 0 der SIMATIC 300 Station.



## 12.6 Inbetriebnahme Hardware-Konfiguration der PLC-CPU

Die Inbetriebnahme der PLC-CPU ist ausführlich beschrieben in:

### Literatur

Inbetriebnahmehandbuch CNC: NCK, PLC, Antrieb:

- Kapitel: "PG/PC mit PLC verbinden"
- Kapitel: "Inbetriebnahme PLC"
- Kapitel: "Grundlagen" > "PLC-Programm"
- Kapitel: "Allgemeine Tipps" > "Getrenntes NCK- und PLC-Urlöschen"
- Kapitel: "Allgemeine Tipps" > "Einbringen PG/PC ins Netzwerk (NetPro)"

## 12.7 Inbetriebnahme PLC-Programm

### 12.7.1 Installation des Grundprogramms

Die Installation des Grundprogramms ist ausführlich beschrieben in:

#### Literatur

Inbetriebnahmehandbuch CNC: NCK, PLC, Antrieb; Kapitel: "Inbetriebnahme PLC" > "PLC-Programm erstellen"

---

#### Hinweis

##### Installation / Update

Vor Installation der Toolbox für SINUMERIK 840D sl muss SIMATIC STEP 7 installiert sein.

Es wird nach einem Update von STEP 7 empfohlen, die Hardware-Ergänzungen für STEP 7 aus der Toolbox neu zu installieren.

#### Inhalt

Die Quellprogramme der OBs inklusive Standard-Parametrierung, die Nahtstellensymbolik und DB-Vorlagen für Bedienhandgerät und M-Dekodierung sind im Grundprogramm enthalten.

---

### 12.7.2 Anwendung des Grundprogramms

Für jede Anlage (Maschine) ist ein neues CPU-Programm in einem Projekt über die STEP 7-Software anzulegen (z. B. "Drehma1").

#### Anmerkung

Die Katalogstrukturen eines Projekts und die Vorgehensweise zur Erstellung von Projekten und Anwenderprogrammen sind in der zugehörigen SIMATIC-Dokumentation enthalten.

## Vorgehensweise

Das Kopieren der Grundprogrammteile erfolgt über den SIMATIC Manager mit "Datei" > "Öffnen" > "Bibliothek".

Aus der Bibliothek müssen folgende Teile kopiert werden:

- Aus dem Baustein-Container: FCs, FBs, DBs, OBs, SFC, SFB, UDT
- Die Source\_files (aus dem Source-Container): GPOB840D
- Gegebenenfalls MDECLIST, BHG\_DB und weitere
- Die Symboltabelle (aus dem Symbol-Container)

## Kompatibilität zu STEP 7

Es sind keine Abhängigkeiten des Grundprogramms zu den derzeitigen STEP 7-Versionen vorhanden.

### 12.7.3 Versionskennzeichnungen

#### Grundprogramm

Die Version des Grundprogramms einschließlich des Steuerungstyps wird im Versionsbild der Bedienoberfläche angezeigt.

Der Steuerungstyp ist folgendermaßen verschlüsselt:

Linksbündige Dekade von DB 17 DBD 0 (Byte 0)	Steuerungstyp
03	SINUMERIK 840D sl (NCU 7x0)

#### Anwenderprogramm

Der Anwender kann auch seine eigene PLC-Versionskennzeichnung im Versionsbild darstellen. Hierzu ist in einem beliebigen Datenbaustein ein Datum vom Datentyp STRING mit max. 54 Zeichen zu definieren. Als Inhalt kann ein beliebiger Text stehen. Die Parametrierung auf diesen String erfolgt über einen Zeiger am FB 1. Hierzu muss der Datenbaustein symbolisch definiert sein.

Siehe hierzu Kapitel "Bausteinbeschreibungen" > "FB 1: RUN\_UP Grundprogramm, Anlaufteil (Seite 999)".

## 12.7.4 Maschinenprogramm

Das Maschinenprogramm wird durch den Maschinenhersteller unter Zuhilfenahme der Bibliotheksroutinen des Grundprogramms erstellt. Im Maschinenprogramm sind die logischen Verknüpfungen und Abläufe der Maschine enthalten. Weiterhin werden die Nahtstellensignale zur NCK bedient. Komplexere Kommunikationsfunktionen zur NCK wie z. B. NC-Daten lesen, schreiben, Werkzeugverwaltungs-Quittungen, usw. werden über die Bausteine FCs, FBs des Grundprogramms angestoßen und ausgeführt.

Das Maschinenprogramm kann in unterschiedlichen STEP 7-Erstellungssprachen wie z. B. AWL, KOP, FUP, S7-HIGRAPH, S7GRAPH, SCL erstellt werden. Das komplette Maschinenprogramm muss in der richtigen Reihenfolge erstellt und kompiliert sein.

Das bedeutet, dass Bausteine die von anderen Bausteinen aufgerufen werden, generell **vor diesen** Bausteinen kompiliert werden müssen.

Wenn diese aufgerufenen Bausteine nachträglich im Zuge der Programmentwicklung in der Schnittstelle (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR) geändert werden, dann sind auch der rufende Baustein und alle damit verbundenen Bausteine anschließend zu kompilieren. Diese Vorgehensweise gilt sinngemäß auch für Instanz-Datenbausteine für FBs. Wenn diese Reihenfolge nicht eingehalten wird, treten Zeitstempel-Konflikte bei der Rückübersetzung in STEP 7 auf. Somit ist die Rückübersetzbarkeit der Bausteine nicht sichergestellt und bei der Funktion "Status Baustein" treten unter anderen unnötige Konflikte auf. Es ist außerdem empfehlenswert die im Kontaktplan oder in Einzelanweisung (inkrementeller Modus) erstellten Bausteine in ASCII-AWL durch den STEP 7-Editor zu generieren.

## 12.7.5 Datensicherung

Die PLC-CPU speichert keine symbolischen Namen ab, sondern nur die Datentypbeschreibungen der Bausteinparameter VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR und die Datentypen der globalen Datenbausteine.

---

### Hinweis

Ohne das zugehörige Projekt für diese Maschine ist keine vernünftige Rückübersetzung möglich. Dies betrifft insbesondere z. B. die Funktion "Status Baustein" oder nachträglich notwendige Änderungen der PLC-CPU-Programme. Es ist deshalb erforderlich, das sich in der PLC-CPU befindende STEP 7-Projekt an der Maschine unbedingt aufzubewahren. Für einen Servicefall ist dies eine große Hilfe und erspart einen unnötigen Zeitaufwand das ursprüngliche zugehörige Projekt wieder herzustellen.

---

Wenn das STEP 7-Projekt vorhanden ist und nach den genannten Regeln erstellt wurde, kann an dieser Maschine in der PLC-CPU symbolisch gearbeitet werden. Gegebenfalls sind auch die Quell-Programme der Maschine als ".awl"-Dateien mit zu hinterlegen für eine eventuell notwendige Hochrüstung.

Als Minimum sollten die Quellprogramme aller Organisationsbausteine und auch aller Instanz-Datenbausteine vorhanden sein.

## 12.7.6 PLC-Serieninbetriebnahme, PLC Archive

Nach dem Laden der Bausteine in die PLC-CPU kann über die Bedienoberfläche HMI ein Serien-Archiv erzeugt werden zur Datensicherung an der Maschine. Die Datensicherung soll direkt nach dem Laden der Bausteine im PLC-Stop-Zustand erfolgen um Konsistenz der Daten zu erreichen. Diese Datensicherung ersetzt nicht die Sicherung des SIMATIC-Projekts, da im Serienarchiv nur Binärdaten abgespeichert werden. Hier stehen z. B. keine symbolischen Informationen zur Verfügung. Weiterhin werden keine CPU-DBs (SFC22-DBs) und in der CPU erzeugte SDBs gesichert.

### Auswahl des SINUMERIK Archivierungsprogramms

Alternativ kann das PLC-Serienarchiv direkt aus dem SIMATIC-Projekt erzeugt werden:

- Öffnen sie im SIMATIC Manager den Dialog "Einstellungen": Menüleiste "Extras" > "Einstellungen"
- Öffnen sie die Registerkarte "Archivieren"
- Wählen sie im Dropdown-Listefeld "Bevorzugtes Archivier-Programm" das SINUMERK Archivierungsprogramm "SINUMERIK (\*.arc)" aus.

### Start des SINUMERIK Archivierungsprogramms

Das SINUMERIK Archivierungsprogramm wird im SIMATIC Manager über den Menüpunkt "Datei" > "Archivieren" gestartet.

Nach der Vergabe des Archivnamens wird das PLC-Archiv erzeugt. Enthält das Projekt mehrere Programmpfade, kann im Dialog das S7-Programm ausgewählt werden, für welches das PLC-Archiv angelegt wird. Es werden alle Bausteine archiviert, die im ausgewählten Programmpfad enthalten sind (außer Datenbausteine, die mit SFC 22 (online) in der CPU erstellt wurden).

Für das Archivier-Programm kann die Funktion "SDB Archiv" aktiviert bzw. deaktiviert werden. Ist "SDB Archiv" aktiviert, wird ein PLC-Archiv erstellt, in dem sich nur die Systemdatenbausteine (SDB) des ausgewähltem Programmpfades befinden.

## Automatisierung

Die Erzeugung des Serienarchivs ist auch automatisierbar (vergleichbar mit der Kommandoschnittstelle von STEP 7). Diese Erzeugung stellt eine Erweiterung der Kommandoschnittstelle dar.

Folgende Funktionen stehen in dieser Erweiterung zur Verfügung:

Die Funktionen (hier mit VB-Script dargestellt) sind erst nach Aufruf von Server-Instanzierungen und Magic-Aufruf verfügbar:

```
Const S7BlockContainer = 1138689, S7PlanContainer = 17829889
Const S7SourceContainer = 1122308
set S7 = CreateObject("Simatic.Simatic.1")
rem Kommandoschnittstelle von STEP 7 instanzieren
Set S7Ext = CreateObject("SimaticExt.S7ContainerExt")
Call S7Ext.Magic("")
```

Funktionen:

- Function Magic(bstrVal As String) As Long
- Function MakeSerienIB (FileName As String, Option As Long, Container As S7Container) As Long

**Beschreibung**

Function **Magic**(bstrVal As String) As Long

Über Aufruf wird Zugang zu bestimmten Funktionen erreicht. Die Funktion muss nach Serverinstanzierung einmalig aufgerufen werden. Der Wert von bstrVal kann leer sein. Hiermit wird die korrekte STEP 7-Version und Path-Angabe in Autoexec geprüft. Bei Rückgabe von 0 sind die Funktionen freigeschaltet.

Rückgabe (-1) = falsche STEP 7-Version

Rückgabe (-2) = kein Eintrag in Autoexec.bat

Function **MakeSerienIB**(FileName As String, Option As Long, Container As S7Container) As Long

**Parameter "Option":**

0:	Normale Serien-IBN-Datei mit Urlöschen.
Bit 0 = 1:	Serien-IBN-Datei ohne Urlöschen. Wenn SDBs im Projekt sind, ist diese Option nicht wirksam. Es wird dann immer ein Urlöschen durchgeführt.
Bit 1 = 1:	Serien-IBN-Datei mit PLC-Neustart

**Rückgabe-Wert:**

0	= OK
-1	= Funktion nicht verfügbar, Funktion "Magic" vorher aufrufen
-2	= Dateiname nicht erzeugbar
-4	= Parameter Container ungültig oder Baustein Container leer
-5	= interner Fehler (Speicheranforderung durch Windows abgelehnt)
-6	= interner Fehler (Problem im STEP 7-Projekt)
-7	= Schreibfehler beim Erzeugen der Serien-IBN-Datei (z. B. Diskette voll)

## Nutzung im Script

### Programmcode

```
If S7Ext.Magic("") < 0 Then
  Wscript.Quit(1)
End If
Set Proj1 = s7.Projects("neu")
set S7Prog = Nothing
Set s7prog = Proj1.Programs.Item(1) 'wenn es nur ein Programm gibt
For Each cont In s7prog.Next
  If (Cont.ConcreteType = S7BlockContainer) Then
    'Bausteincontainer prüfen
    Exit For
  End if
  Cont = Nothing
Next
Fehler = S7Ext.MakeSerienIB("f:\dh\arc.dir\PLC.arc", 0, Cont)
'Jetzt Fehlerauswertung
```

Der oben programmierte For Each ... Next -Block kann in der Programmiersprache Delphi wie folgt programmiert werden (ähnliche Programmierung gilt auch für die Programmiersprache C, C++):

### Programmcode

```
Var
  EnumVar: IEnumVariant;
  rgvar: OleVariant;
  fetched: Cardinal;

//For Each Next
EnumVar := (S7Prog.Next._NewEnum) as IEnumVariant;
While (EnumVar.Next(1,rgvar,fetched) = S_OK) Do Begin
  Cont := IS7Container(IDispatch(rgvar)); // Bausteincontainer,
  Quellen prüfen
  If (Cont.ConcreteType = S7BlockContainer) Then Break;
  Cont := NIL;
End;
```

### 12.7.7 Software-Hochrüstung

Für einen definierten Ausgangszustand ist vor einer PLC-Software-Hochrüstung ein PLC-Urlöschen durchzuführen. Dabei werden unter anderem alle Anwenderdaten (Programm- und Datenbausteine) gelöscht. Das PLC-Urlöschen ist beschrieben in:

**Literatur:**

Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb; Allgemeine Tipps, Kapitel: PLC-Urlöschen

### Neues SIMATIC S7-Projekt erzeugen

Im Regelfall ist bei einem neuen NCU-Softwarestand das neue PLC-Grundprogramm mit einzubinden. Hierzu ist die Übertragung der Grundprogramm-Bausteine in das Anwenderprojekt erforderlich. Befinden sich folgende Programm- und Datenbausteine schon im Anwenderprojekt sollten sie nicht mit den Bausteinen des PLC-Grundprogramms übertragen werden: OB 1, OB 40, OB 82, OB 86, OB 100, FC 12 und DB 4. Diese sind evtl. vom Anwender verändert worden und sollten nicht überschrieben werden. Das neue Grundprogramm ist mit dem Anwenderprogramm zusammenzubinden. Folgende Vorgehensweise ist dabei zu beachten:

1. Vor dem Kopieren des PLC-Grundprogramms von allen Anwender-Bausteinen die Text- bzw. Quell-Datei generieren.
2. Die neuen Grundprogramm-Bausteine in das SIMATIC S7-Projekt kopieren (Beschreibung siehe Kapitel "Anwendung des Grundprogramms (Seite 914)")
3. Alle Anwenderprogramme "\*.awl" in korrekter Reihenfolge neu zu übersetzen! (siehe auch " Maschinenprogramm (Seite 916)")
4. Dieses neu kompilierte SIMATIC S7-Projekt ist dann mit STEP 7 in die PLC zu laden.

Im Normalfall reicht jedoch ein neues Übersetzen der Organisationsbausteine (OBs) und der Instanz-Datenbausteine des S7-Projekts aus. D. h. es brauchen vor der Hochrüstung nur Quellen für die Organisationsbausteine und die Instanz-Datenbausteine erzeugt werden.

### NC-Variablen

Für jeden NC-Softwarestand (auch ältere Versionen) kann der neueste NC-VAR-Selector verwendet werden. Für ältere NC-Softwarestände können die Variablen auch aus der neuesten Gesamtliste selektiert werden. Der Informationsinhalt im DB 120 (Standard-DB für Variablen) ist nicht abhängig vom Softwarestand. D. h. selektierte Variable in einem älteren Softwarestand müssen bei einer Software-Hochrüstung nicht neu selektiert werden.

### 12.7.8 Peripheriebaugruppen (FM-, CP-Baugruppen)

Für komplexere Peripheriebaugruppen (FM-, CP-Baugruppen) werden in der Regel Zusatzpakete zu STEP 7 benötigt. In diesen Zusatzpaketen werden Unterstützungsbausteine (FC / FB) mitgeliefert. Diese Bausteine beinhalten spezifische Funktionen zum Betrieb der jeweiligen Baugruppe. Diese Funktionen können im Anwenderprogramm parametrisiert und aufgerufen werden.



### Nummerngleichheit

Bei Nummerngleichheit von Handierungs- und Grundprogrammbausteinen müssen die Baustein-Nummern des Grundprogramms unverändert bleiben. Die Baustein-Nummern der Handierungsbausteine müssen über STEP 7 auf freie Nummern umbenannt werden.

## 12.7.9 Fehler-Beseitigung

Dieser Abschnitt soll Hinweise zu Problemfällen und deren Beseitigung bzw. auch Ursachen geben, bevor ein Hardware-Tausch erfolgt.

Fehler, Ursache/Beschreibung und Abhilfe			
lfd. Nr. Fehler- hinweis	Fehler	Ursache / Beschreibung	Abhilfe
1	Keine Verbindung über MPI zur PLC.	Das MPI-Kabel ist nicht gesteckt oder defekt. Möglicherweise ist auch die STEP 7-Software für die MPI-Karte nicht richtig konfiguriert.	Versuch: Mit dem PG im STEP 7-Editor über die Verbindung "Direct_PLC" ankoppeln. Hierbei müssen einige Teilnehmeradressen eingeblendet werden. Fehlen die, dann MPI-Kabel defekt / nicht gesteckt.
2	Trotz Urlöschen der PLC ist kein Zugriff zur PLC möglich.	Ein Systemdatenbaustein SDB 0 wurde mit einer geänderten MPI-Adresse geladen. Hierdurch tritt ein MPI-Buskonflikt wegen doppelt vergebenen Adressen auf.	Alle MPI-Kabel zu anderen Komponenten trennen. Mit dem PG die Verbindung "Direct_PLC" aufbauen. Durch Taufe der MPI-Adresse die MPI-Adresse korrigieren.
3	Alle 4 LEDs der PLC blinken (DI-Katastrophe)	Ein Systemfehler in der PLC ist aufgetreten. <b>Maßnahmen:</b> Zur genaueren Analyse des Systemfehlers ist der Diagnosepuffer der PLC auszulesen. Hierzu ist die PLC zuerst in den Stoppzustand zu bringen (z. B. Schalter "PLC" auf Stellung 2. Anschließend ist ein Hardware-RESET erforderlich. Danach kann mit STEP 7 der Diagnosepuffer ausgelesen werden. Die Informationen aus dem Diagnosepuffer sind der Hotline / Entwicklung zu melden. Wenn nach dem Hardware-RESET zusätzlich ein Urlöschen angefordert wird, ist erst das Urlöschen durchzuführen. Anschließend kann dann im Stoppzustand der Diagnosepuffer gelesen werden.	Nach RESET bzw. nach Neuladen des PLC-Programms läuft das System unter Umständen normal weiter. In jedem Fall ist der Diagnosepuffer der Entwicklung zur Verfügung zu stellen.

## 12.8 Ankopplung der PLC-CPU

### 12.8.1 Allgemeines

Als PLC wird bei SINUMERIK 840D sl eine CPU des Automatisierungssystems S7-300 verwendet. Die PLC-CPU ist als Submodul in die NCU-Komponente integriert. Ein Verweis auf die Leistungsdaten der PLC-CPU findet sich im Kapitel "Eckdaten der PLC-CPU (Seite 909)".

### 12.8.2 Eigenschaften der PLC-CPU

Die in die SINUMERIK 840D sl integrierte PLC hat im Allgemeinen die gleiche Funktionalität wie die entsprechende SIMATIC S7-300 PLC.

Zu Abweichungen, siehe Verweis in Kapitel "Eckdaten der PLC-CPU (Seite 909)".

Durch das teilweise unterschiedliche Speicherkonzept gegenüber einer SIMATIC S7-300 PLC sind bestimmte Funktionen nicht vorhanden (z. B. Bausteine auf Memory Card, Projekt auf Memory Card speichern).

---

#### Hinweis

Bei einer SIMATIC S7-300 PLC wird, wie auch bei der in SINUMERIK integrierten PLC, bei Auslösen eines "PLC-Stop" durch Bedienung am Programmiergerät kein automatisches Starten der PLC nach Spannungsausfall und -wiederkehr durchgeführt. Die PLC verbleibt aus Sicherheitsgründen mit entsprechendem Diagnoseeintrag im Stoppzustand. Ein Starten der PLC ist nur durch Bedienung am Programmiergerät "Neustart ausführen" oder über den Betriebsartenschalter "Stop" > "Run" (Warmstart) möglich.

---

### 12.8.3 Nahtstelle mit integrierter PLC

#### Physikalische Nahtstellen

Die in die NCU integrierte PLC bietet bei SINUMERIK 840D sl die Möglichkeit, den Austausch der Signale zwischen NCK und PLC direkt über ein Dual-Port-RAM vorzunehmen.

#### Datenaustausch mit der Bedientafel

Der Datenaustausch mit der Bedientafel (z. B. TCU / OP) kann über Ethernet oder PROFIBUS erfolgen. Bei Anschluss über Ethernet erfolgt die Kommunikation über den integrierten Kommunikationsprozessor (CP 840D sl).

Der Datenaustausch mit Maschinensteuertafel (MCP) und Bedienhandgerät (BHG) kann über MPI, PROFIBUS oder Ethernet erfolgen.

Programmiergeräte werden vorzugsweise über Ethernet oder über MPI (Multi-Point-Interface) direkt an die PLC angeschlossen.

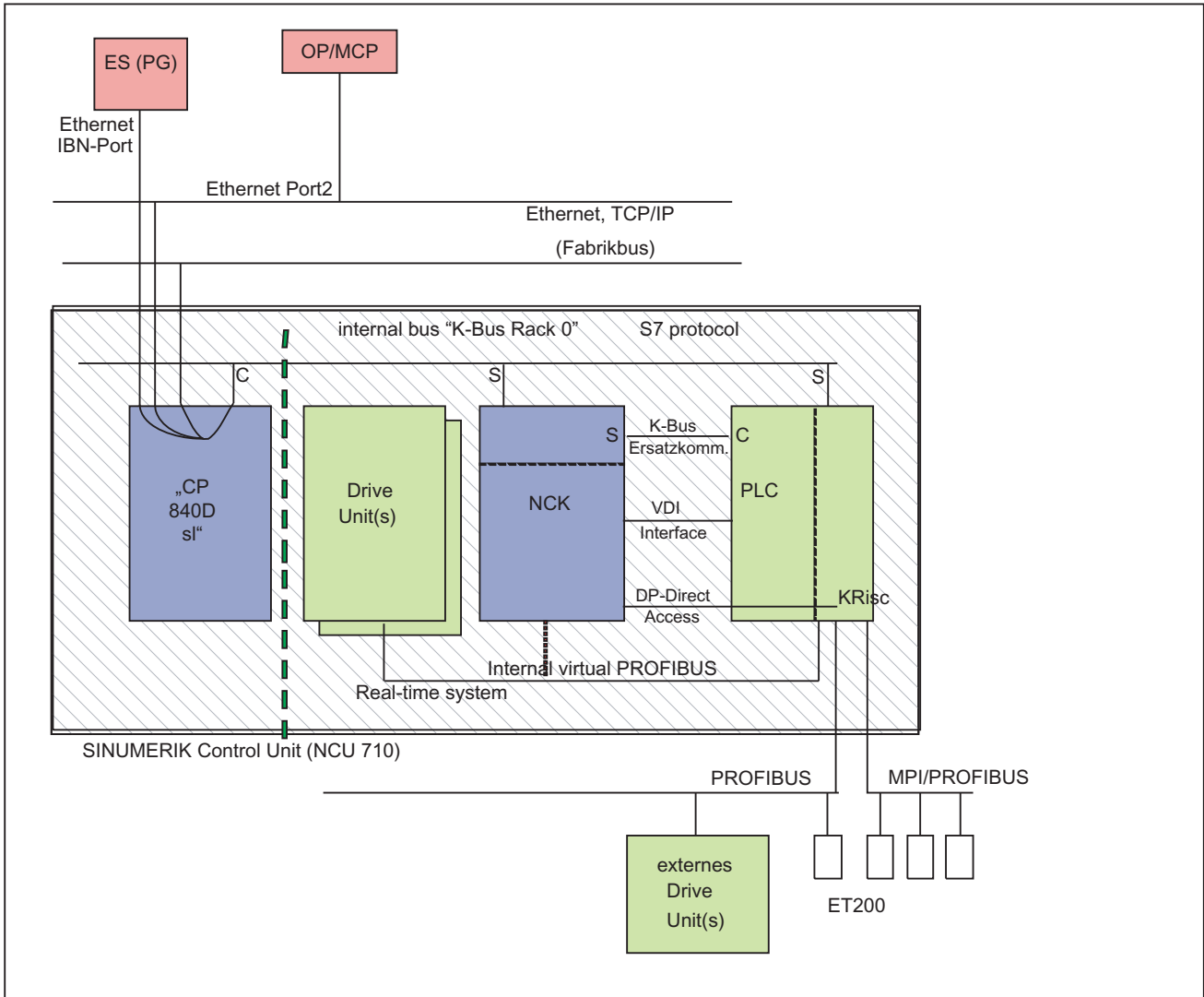


Bild 12-1 NCK-PLC-Kopplung bei SINUMERIK 840D sl (integrierte PLC)

### Nahtstelle: NCK / PLC

Der Datenaustausch zwischen NCK und PLC wird auf PLC-Seite vom Grundprogramm organisiert. Die von der NCK in der NCK/PLC-Nahtstelle abgelegten **Statusinformationen**, wie z. B. "Programm läuft", werden vom Grundprogramm am Zyklusanfang (OB 1) in Datenbausteine kopiert, auf die im Anwenderprogramm dann zugegriffen werden kann (Anwendernahtstelle). Die vom Anwender in die Nahtstellen-DB eingetragenen **Steuersignale** an die NCK, wie z. B. "NC-Start", werden ebenso am Zyklusanfang ins interne DPR zur NCK übertragen.

Die abhängig vom Werkstückprogramm an die PLC übergebenen **Hilfsfunktionen** werden zunächst alarmgesteuert vom Grundprogramm ausgewertet und dann am Anfang von OB 1 an die Anwendernahtstelle übergeben. Sind im betreffenden NC-Satz Hilfsfunktionen enthalten, die eine Unterbrechung der NCK-Bearbeitung erfordern (wie z. B. M06 für Werkzeugwechsel), wird die NCK-Satz-Decodierung vom Grundprogramm zunächst für eine PLC-Zykluszeit angehalten. Der Anwender kann dann über das Nahtstellensignal "Einlesesperre" die Decodierung so lange anhalten, bis z. B. der Werkzeugwechsel abgeschlossen ist. Sind dagegen im betreffenden NC-Satz nur Hilfsfunktionen enthalten, die keine Unterbrechung der Decodierung erfordern (wie z. B. M08 für Kühlmittel Ein), so wird die Übergabe dieser "schnellen" Hilfsfunktionen direkt im OB 40 quittiert, sodass die Decodierung von der Übergabe an die PLC nur unwesentlich beeinflusst wird.

Die vom NCK übergebenen **G-Funktionen** werden ebenfalls alarmgesteuert ausgewertet und quittiert, dann jedoch direkt an die Anwendernahtstelle übergeben. Wird eine G-Funktion an mehreren Stellen im PLC-Programm ausgewertet, können sich Unterschiede in der Information der G-Funktion innerhalb eines PLC-Zyklus ergeben.

Bei **NCK-Aktionen**, die vom PLC aus angestoßen und parametrierbar werden (z. B. konkurrierende Achsenverfahren), erfolgt der Anstoß und die Parametrierung nicht über Nahtstellen-DBs, sondern über FCs und FBs. Die zu den NC-Aktionen gehörenden FCs werden zusammen mit dem Grundprogramm ausgeliefert. Die jeweils benötigten FCs und FBs müssen vom Anwender geladen und in dem PLC-Programm des Maschinenherstellers (Maschinenprogramm) aufgerufen werden. Eine Übersicht der Bausteine FC, FB und DB getrennt nach Basisfunktion und erweiterter Funktion finden Sie im Kapitel "Inbetriebnahme PLC-Programm".

### Nahtstelle: HMI / PLC

Der Datenaustausch HMI / PLC erfolgt über den integrierten CP, wobei der HMI immer der aktive Partner (Client) und die PLC immer der passive Partner (Server) ist. Die vom HMI angeforderten bzw. übergebenen Daten werden vom PLC-Betriebssystem aus dem Nahtstellenbereich HMI / PLC ausgelesen bzw. in ihn eingetragen (Zeitpunkt: Zykluskontrollpunkt). Aus PLC-Anwendersicht verhalten sich die Daten wie E/A-Signale.

#### Nahtstelle: MCP / PLC bzw. BHG / PLC (Anschluss: Ethernet)

Der Datenaustausch MCP / PLC, BHG (HT2) / PLC erfolgt über den integrierten CP. Vom CP werden die MCP / BHG-Signale in das interne DPR (Dual Port RAM) der PLC abgelegt bzw. von ihm abgeholt. Auf PLC-Seite übernimmt das Grundprogramm den Austausch mit der Anwendernahtstelle. Über die Parameter des Grundprogramms (FB 1, DB 7) werden die Operandenbereiche (z. B. E/A-Bereiche) und die Anfangsadressen festgelegt.

**Nahtstelle: MCP / PLC (Anschluss: PROFIBUS)**

Der Datenaustausch MCP / PLC erfolgt über den PROFIBUS der PLC. Die E/A-Adressen der MCP sind in den Prozess-Abbildbereich der PLC über die HW-Projektierung in STEP 7 zu legen. Die Pointer Variablen MCP\*In, MCP\*Out sind auf die gleichen Adressen zu setzen. In MCP\*BusAdr ist die eingestellte DP-Slave-Nummer einzutragen.

**Nahtstelle: BHG / PLC (Anschluss: MPI)**

Der Datenaustausch BHG / PLC erfolgt über die MPI-Nahtstelle der PLC. Es wird hierfür der Dienst "Kommunikation mit Globaldaten (GD)" verwendet (siehe hierzu STEP7 Benutzerhandbuch). Das PLC-Betriebssystem übernimmt den Transfer der Signale von und zur Anwendernahtstelle. Über das STEP7-Projektierwerkzeug "Communication-Configuration" werden neben den GD-Parametern auch die Operandenbereiche (z. B. E/A-Bereiche) und deren Anfangsadressen festgelegt.

## 12.8.4 Diagnosepuffer der PLC

Im Diagnosepuffer der PLC (auslesbar mit STEP 7) werden Diagnoseinformationen des PLC-Betriebssystems eingetragen.

## 12.9 Struktur der Nahtstelle

### Nahtstellen-DBs

Aufgrund der Vielzahl der Signale zwischen NCK und PLC ist die Abbildung in Nahtstellen-DBs notwendig. Aus PLC-Programmsicht sind dies globale DBs. Das Grundprogramm erzeugt diese DBs beim Systemanlauf anhand von aktuellen NCK-Maschinendaten (Anzahl der Kanäle, Achsen etc.). Dies hat den Vorteil, dass nur soviel PLC-RAM belegt wird, wie für die aktuelle Maschinenkonfiguration benötigt wird.

### 12.9.1 Nahtstelle PLC/NCK

#### Allgemeines

Die Nahtstelle PLC / NCK wird einerseits durch eine Datenschnittstelle und andererseits durch eine Funktionsschnittstelle gebildet. In der Datenschnittstelle sind Status- und Steuersignale, Hilfs- und G-Funktionen enthalten, während über die Funktionsschnittstelle Aufträge von der PLC an den NCK übergeben werden.

#### Datenschnittstelle

Die Datenschnittstelle wird in folgende Gruppen unterteilt:

- NCK-spezifische Signale
- BAG-spezifische Signale
- Kanal-spezifische Signale
- Achs-/Spindel-/Antriebsspezifische Signale

### Funktionsschnittstelle

Die Funktionsschnittstelle wird durch FBs und FCs gebildet. Das folgende Bild zeigt die generelle Struktur der Nahtstelle zwischen PLC und NCK.

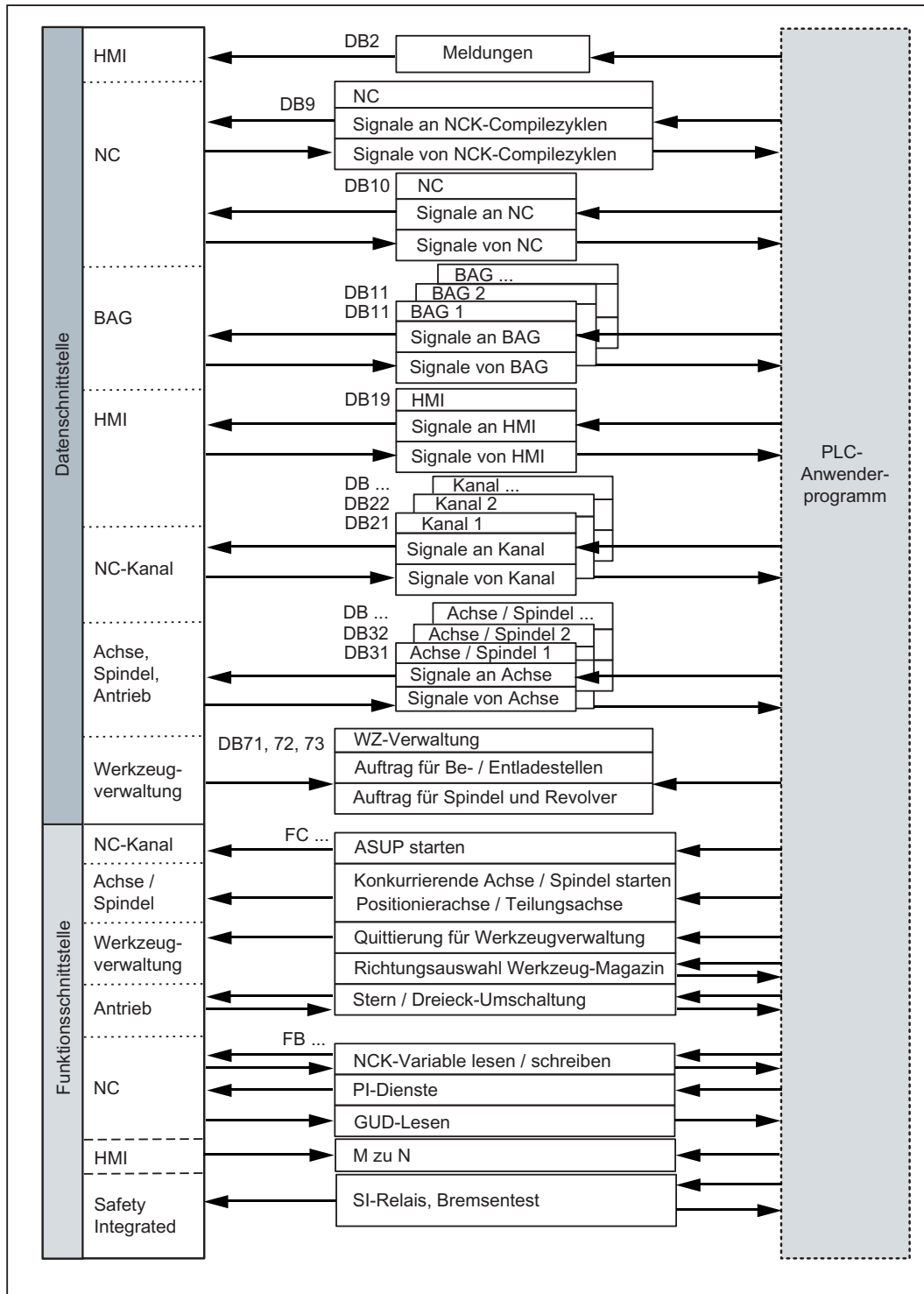


Bild 12-2 Anwendernahtstelle PLC / NCK

### Compilezyklen-Signale

Neben den standardmäßig vorhandenen Signalen zwischen PLC und NCK wird bei Bedarf ein Nahtstellen-DB für Compile-Zyklen erzeugt (DB 9). Die zugehörigen Signale, die abhängig von den jeweiligen Compile-Zyklen sind, werden zyklisch zu Beginn des OB 1 übertragen. Die Übertragung erfolgt von niedriger nach höherer Adresse durch das Grundprogramm. Zuerst werden die Signale von PLC an NCK übertragen, danach die Signale von NCK an PLC. Notwendige Synchronisationen zwischen NCK und PLC sind durch den Anwender (z. B. durch Semaphorentechnik) vorzusehen. Die Signalübertragungen erfolgen asynchron zwischen NCK und PLC. Hierbei kann z. B. eine begonnene Datenübertragung des NCK vom PLC unterbrochen werden. Damit sind die Daten unter Umständen nicht konsistent.

### Signale PLC / NCK

In die Gruppe der Signale von PLC an NCK fallen:

- Signale für die Beeinflussung der digitalen und analogen E-/A-Signale der NCK
- Schlüsselschalter- und Not-Halt-Signale

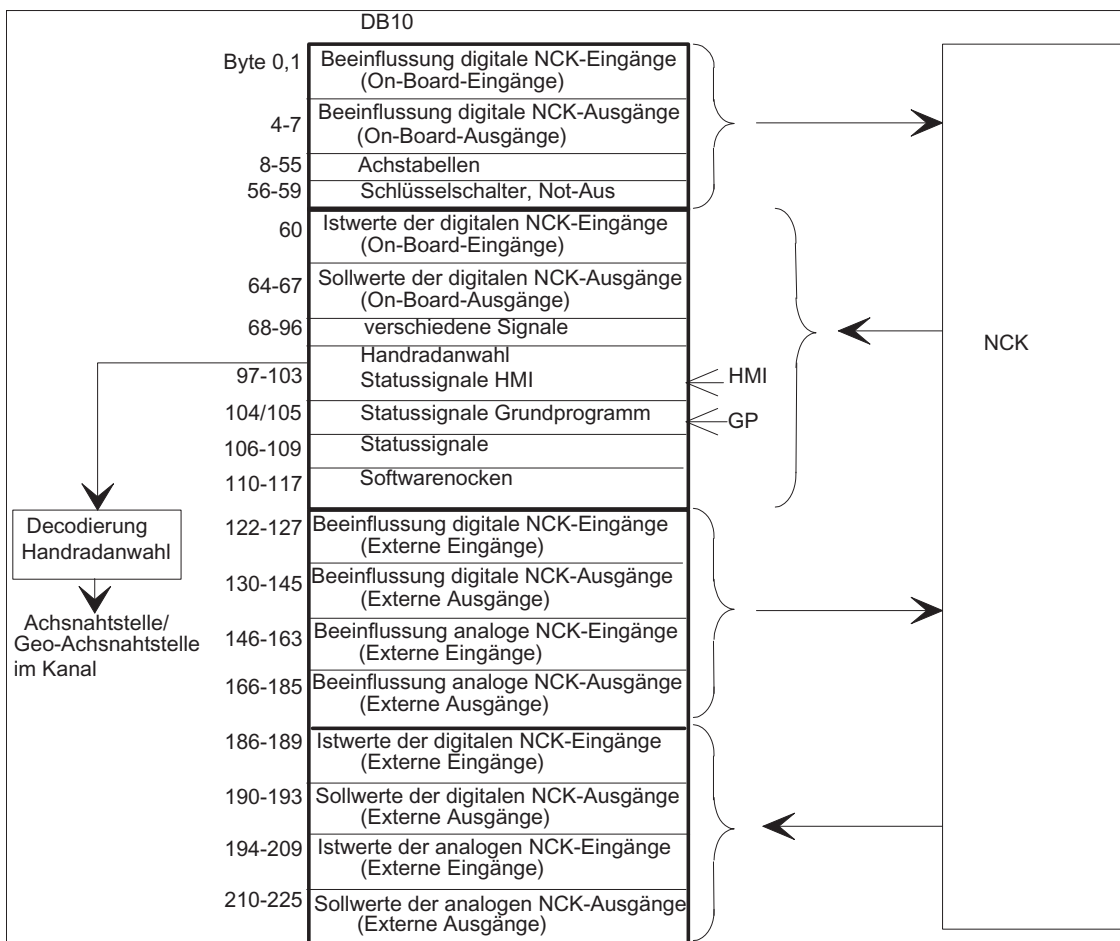


Bild 12-3 Nahtstelle PLC / NCK



## Signale NCK / PLC

In die Gruppe der Signale von NCK an PLC fallen:

- Istwerte der digitalen und analogen E/A-Signale der NCK
- Bereitschafts- und Statussignale der NCK

Weiterhin sind hier auch die Handradanwahlsignale und die Kanal-Statussignale vom HMI abgelegt.

Die Signale zur Handradanwahl werden vom Grundprogramm decodiert und in die maschinen- bzw. kanalachsspezifische Nahtstelle eingetragen.

## Digitale/Analoge Ein-/Ausgänge des NCK

Bei den digitalen und analogen Ein- und Ausgängen des NCK ist Folgendes zu beachten:

### Eingänge:

- Alle Eingangssignale bzw. Eingangswerte des NCK werden auch an die PLC übergeben.
- Die Weitergabe der Signale an das NC-Teilprogramm kann von der PLC unterdrückt werden. Stattdessen kann ein Signal bzw. Wert von der PLC vorgegeben werden.
- Es kann auch dann ein Signal bzw. Wert von der PLC an den NCK übergeben werden, wenn auf NCK-Seite für diesen Kanal keine Hardware vorhanden ist.

### Ausgänge:

- Alle auszugebenden Signale bzw. Werte werden auch an die PLC übergeben.
- Es können auch dann Signale bzw. Werte vom NCK an die PLC übergeben werden, wenn auf NCK-Seite für diesen Kanal keine Hardware vorhanden ist.
- Die vom NCK vorgegebenen Werte können von der PLC überschrieben werden.
- Von der PLC aus können auch direkt Signale bzw. Werte über die NCK-Peripherie ausgegeben werden.

---

### Hinweis

Bei der Realisierung von digitaler und analoger NCK-Peripherie sind die Informationen in folgender Dokumentation zu berücksichtigen:

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Digitale und analoge NCK-Peripherie (A4)

---

### Signale PLC / BAG

Zur Betriebsartengruppe (BAG) des NCK werden die von der Maschinensteuertafel oder vom HMI vorgegebenen Betriebsartensignale übertragen. Diese gelten für alle NCK-Kanäle. Im NCK können optional mehrere BAGs definiert sein.

Von der BAG wird deren aktueller Zustand an die PLC gemeldet.

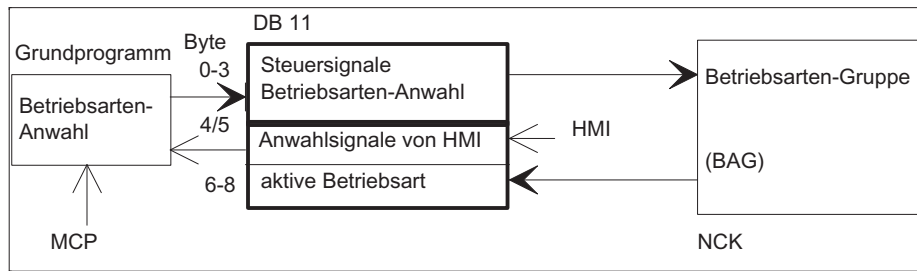


Bild 12-4 Nahtstelle PLC / BAG

### Signale PLC / NCK-Kanäle

Bei der Nahtstelle sind folgende Signalgruppen zu betrachten:

- Steuer- / Status-Signale
- Hilfs- / G-Funktionen
- Signale der Werkzeugverwaltung
- NCK-Funktionen.

Die **Steuer- / Status-Signale** werden zyklisch am Anfang von OB 1 übertragen. Auch die vom HMI in die kanalspezifische Nahtstelle eingetragenen Signale (der Eintrag der HMI-Signale erfolgt durch das Betriebssystem der PLC) werden zu diesem Zeitpunkt übertragen, wenn diese Signale nicht über die MCP, sondern über die HMI-Bedientafel vorgegeben werden.

Die **Hilfs- und G-Funktionen** werden auf zweierlei Weise in die Nahtstellen-DBs eingetragen. Zunächst werden sie zusammen mit den Änderungssignalen eingetragen.

- Die **M-Signale** M00 - M99 (sie werden vom NCK mit der erweiterten Adresse 0 übergeben) werden zusätzlich decodiert und die zugehörigen Nahtstellenbits für eine Zyklusdauer gesetzt.
- Bei den **G-Funktionen** werden nur die durch Maschinendaten ausgewählten Gruppen in den Nahtstellen-DB eingetragen.
- Die **S-Werte** werden zusätzlich zusammen mit den zugehörigen M-Signalen (M03, M04, M05) in die spindelspezifische Nahtstelle eingetragen. Ebenso werden achsspezifische Vorschübe in die entsprechende achsspezifische Nahtstelle eingetragen.

Bei aktivierter **Werkzeugverwaltung (Magazinverwaltung)** im NCK wird die Belegung von Spindel bzw. Revolver und der Be-/Entladestellen in eigene Nahtstellen-DBs eingetragen (DB 71 - 73).

Die **NCK-Funktionen** werden über PLC-Funktionsaufrufe angestoßen und parametrierbar. Es stehen z. B. folgende Funktionsaufrufe zur Verfügung:

- Positionieren einer Linear- bzw. Rundachse
- Positionieren einer Teilungsachse
- Starten eines vorbereiteten asynchronen Unterprogramms (ASUP)
- Lesen/Schreiben von NC-Variablen
- Magazin- und Werkzeugbewegung aktualisieren

Die vorgenannten Funktionen sind zum Teil in eigenen Funktionsdokumentationen beschrieben.

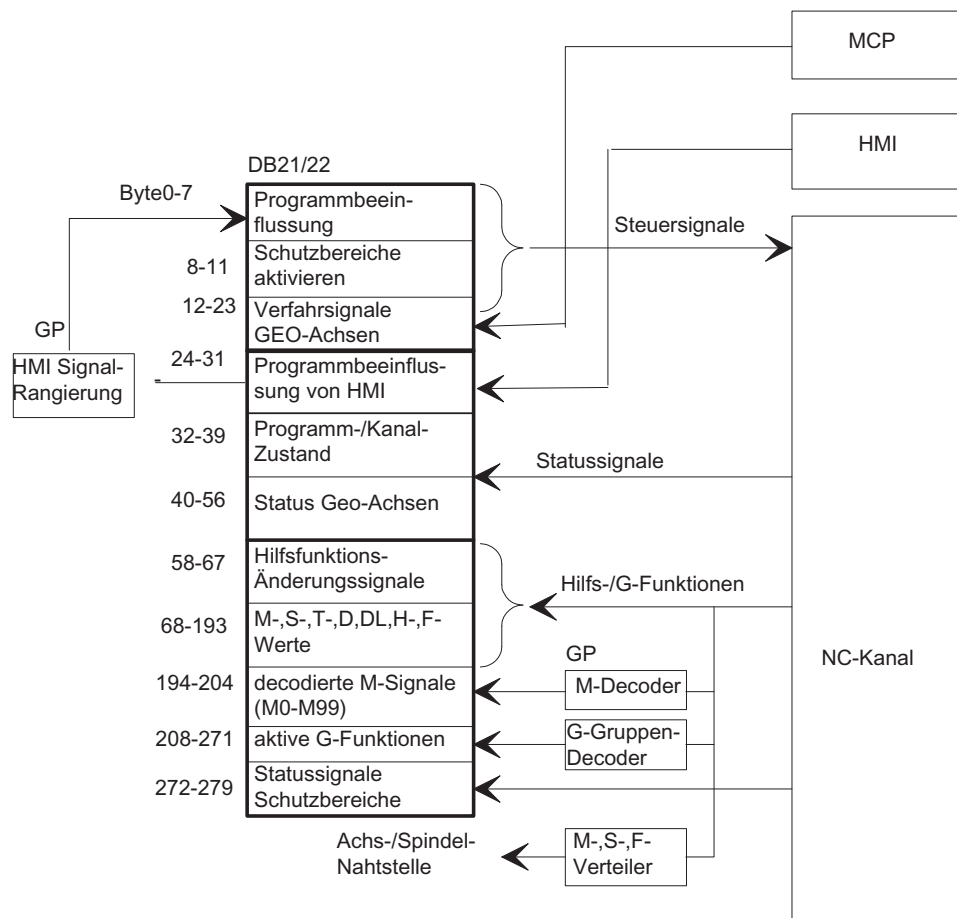


Bild 12-5 Nahtstelle PLC/NCK-Kanal

### Signale PLC / Achsen, Spindel, Antrieb

Die achs- und spindelspezifischen Signale sind in folgende Gruppen aufgeteilt:

- Gemeinsame Achs- / Spindelsignale
- Achssignale
- Spindelsignale
- Antriebssignale

Die Signale werden bis auf die im Folgenden beschriebenen Ausnahmen zyklisch am Anfang des OB 1 übertragen.

Zu den Ausnahmen gehören:

- axialer F-Wert
- M-Wert
- S-Wert

Ein **axialer F-Wert** wird über den M-, S-, F-Verteiler des Grundprogramms dann eingetragen, wenn er im Zuge der NC-Programmbearbeitung an die PLC übertragen wird.

**M- und S-Wert** wird ebenfalls dann über den M-, S-, F-Verteiler des Grundprogramms eingetragen, wenn einer oder beide Werte zur Abarbeitung kommen.

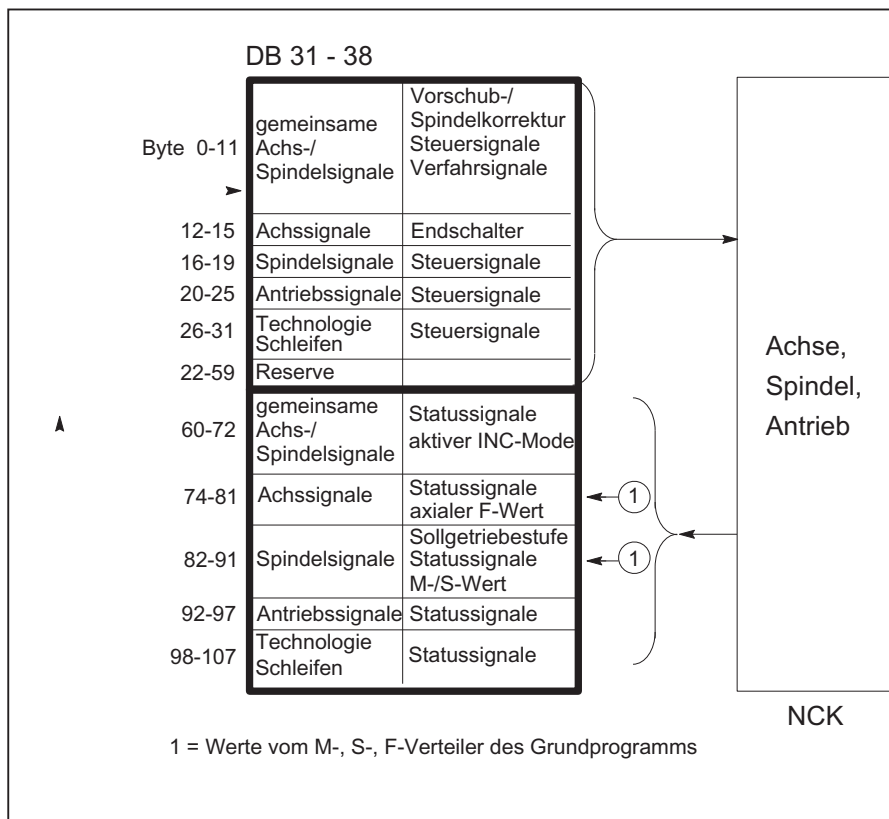


Bild 12-6 Nahtstelle PLC / Achsen, Spindeln, Antriebe

## 12.9.2 Nahtstelle PLC/HMI

### Allgemeines

Bei der Nahtstelle PLC/HMI müssen folgende Funktionskomplexe betrachtet werden:

- Steuersignale
- Maschinenbedienung
- PLC-Meldungen
- PLC-Status-Anzeige

### Steuersignale

Bei den Steuersignalen handelt es sich um Signale, die u. a. von der Maschinensteuertafel vorgegeben werden und vom HMI berücksichtigt werden müssen. Zu diesen Signalen gehören z. B. Istwerte im MKS oder WKS anzeigen, Tastensperre usw. Diese Signale werden über einen eigenen Nahtstellen-DB (DB 19) mit dem HMI ausgetauscht.

### Maschinenbedienung

Alle Bedienhandlungen, die zu Aktionen an der Maschine führen, werden von der PLC kontrolliert. Im Normalfall werden diese über die Maschinensteuertafel (MCP) vorgenommen. Es ist jedoch auch möglich, ein Teil der Bedienhandlungen wahlweise vom HMI aus vorzunehmen wie z. B. Betriebsartenwahl.

Die vom HMI kommenden Bediensignale werden vom PLC-Betriebssystem direkt in die Nahtstellen-DBs eingetragen. Im Standardfall werden diese Bediensignale so vom Grundprogramm rangiert, dass - sofern eine gleichartige Bedienmöglichkeit von der MCP besteht - die Bedienung wahlweise vom HMI **oder** von der MCP vorgenommen werden kann. Bei Bedarf kann die Bedienung über HMI durch einen Parameter "MMCToIF" des FB 1 vom Anwender ausgeschaltet werden.

## PLC-Meldungen

Basis der Meldefunktionen sind die in das Betriebssystem der AS 300 integrierten Systemdiagnose-Funktionen, die folgende Eigenschaften besitzen:

- In einer **Diagnose-Zustandsliste** werden vom PLC-Betriebssystem alle wichtigen Systemzustände sowie Zustandsübergänge eingetragen. Weiterhin werden Kommunikationsereignisse und (Peripherie-) Baugruppen-Diagnosedaten (bei Baugruppen mit Diagnosefunktionen) eingetragen.
- Zusätzlich werden die Diagnose-Ereignisse, die zum Systemstopp führen, in einen **Diagnosepuffer** (Ringpuffer) in der zeitlichen Reihenfolge des Auftretens mit Uhrzeitstempel eingetragen.
- Die in den Diagnosepuffer eingetragenen Ereignisse werden automatisch an Bedien-/Beobachtungssysteme (OP bzw. HMI) über die Bussysteme versandt, sofern diese sich als empfangsbereit angemeldet haben (Meldedienst). Die Übertragung an den angemeldeten Teilnehmer ist eine Funktion des Betriebssystems der PLC. Der Empfang und die Interpretation der Meldungen werden durch die HMI-Software sichergestellt.
- Vom PLC-Anwenderprogramm können über SFCs (System-Function-Calls) ebenfalls Meldungen in den Diagnosepuffer bzw. ALARM\_S/ ALARM\_SQ -Puffer eingetragen werden.
- Die Ereignisse werden in den Alarm-Puffer eingetragen.

Die zugehörigen Texte müssen im OP bzw. HMI vorgehalten werden.

Zusammen mit dem Grundprogramm wird ein FC zur Meldeerfassung (FC 10) bereitgestellt, der die zu meldenden Ereignisse - in Signalgruppen eingeteilt - erfasst und über den Alarm-Puffer zum HMI meldet.

Die Struktur der Meldeerfassung ist in Bild "Erfassung und Meldung von PLC-Ereignissen" dargestellt. Sie weist folgende Merkmale auf:

- Die Bitfelder für Ereignisse, die die NC/PLC-Nahtstelle betreffen, sind zusammen mit den Bitfeldern für die Anwendermeldungen in dem DB 2 zusammengefasst.
- Die Bitfelder werden mehrfach durch den FC 10 ausgewertet.
  - **Auswertung 1; Erfassung von Sammelsignalen**

Je Signalgruppe wird ein Sammelsignal erzeugt, wenn mindestens ein Bit Signal "1" führt. Dieses Signal wird im Normalfall auf das zugehörige Sperrsignal der NC/PLC-Nahtstelle (bei Baugruppen mit Diagnosefunktionen) geführt. Die Sammelsignale werden zyklisch komplett erfasst.
  - **Auswertung 2; Erfassung von Alarmmeldungen**

Es gibt eine feste Vorgabe, welche Signale einer Gruppe bei ihrem Wechsel von "0" nach "1" eine Alarmmeldung erzeugen.
  - **Auswertung 3; Erfassung von Betriebsmeldungen**

Es gibt eine feste Vorgabe, welche Signale einer Gruppe eine Betriebsmeldung erzeugen.
- Der Umfang der Anwender-Bitfelder (Bereich User) ist standardmäßig auf 10 Bereiche zu je 8 Byte festgelegt, kann jedoch über Grundprogramm-Parameter am FB 1 bezüglich der Anzahl der Bereiche auf die Bedürfnisse des Maschinenherstellers angepasst werden.

## Quittungskonzept

Bei den Fehler- und Betriebsmeldungen werden folgende Quittungskonzepte verfolgt:

Die **Betriebsmeldungen** zeichnen sich dadurch aus, dass mit ihnen normale Betriebszustände der Maschine als Information für den Bediener angezeigt werden sollen. Deshalb entfällt bei ihnen die Notwendigkeit von Quittungssignalen. Bei ihnen wird sowohl das Kommen als auch das Gehen des Ereignisses erfasst und ein Eintrag in die Diagnosezustandsliste vorgenommen. Der HMI erzeugt anhand der Kennungen "BM gekommen" und "BM gegangen" ein aktuelles Abbild der anstehenden Betriebsmeldungen.

Mit **Alarmmeldungen** werden Fehlerzustände an der Maschine angezeigt, die im Normalfall zum Maschinenstillstand führen. Treten mehrere Fehler "gleichzeitig" auf, ist es für die Fehlersuche wichtig, die Reihenfolge des Auftretens zu kennen. Dies wird einerseits durch die Reihenfolge des Eintrags in den Diagnosepuffer angezeigt und andererseits durch den Uhrzeitstempel, den jeder Eintrag erhält.

Verschwundet die Fehlerursache, so wird die zugehörige Alarmmeldung erst dann gelöscht, wenn eine Anwenderquittung vorliegt (z. B. Kundentaste an der MCP). Als Reaktion auf dieses Signal untersucht der FC "Meldungserfassung", welche der bereits gemeldeten Fehler verschwunden sind und trägt diese mit der Kennung "Alarm gegangen" in den Diagnosepuffer ein. Somit kann der HMI auch bei den Alarmmeldungen ein aktuelles Abbild der anstehenden Meldungen erzeugen. Bei den noch anstehenden Meldungen bleibt die Uhrzeit ihres Entstehens erhalten (im Gegensatz zu einer Neuabfrage).

## STEP 7

Im SIMATIC Manager kann mit dem Menüpunkt "Zielsystem" > "CPU-Meldungen" ein Tool gestartet werden. Mit dem Tool können die Alarmer und Meldungen nummernmäßig angezeigt werden. Hierzu ist der Tabulator "Alarm" zu aktivieren und ein Haken im oberen Bildteil unter "A" zu setzen.

### Anwenderprogramm

Das Anwender PLC-Programm muss im zyklischen Programmteil lediglich den Grundprogrammbaustein FC 10 mit entsprechender Parametrierung aufrufen und die Bitfelder im DB 2 setzen bzw. rücksetzen. Alles Weitere wird durch das Grundprogramm und den HMI erledigt.

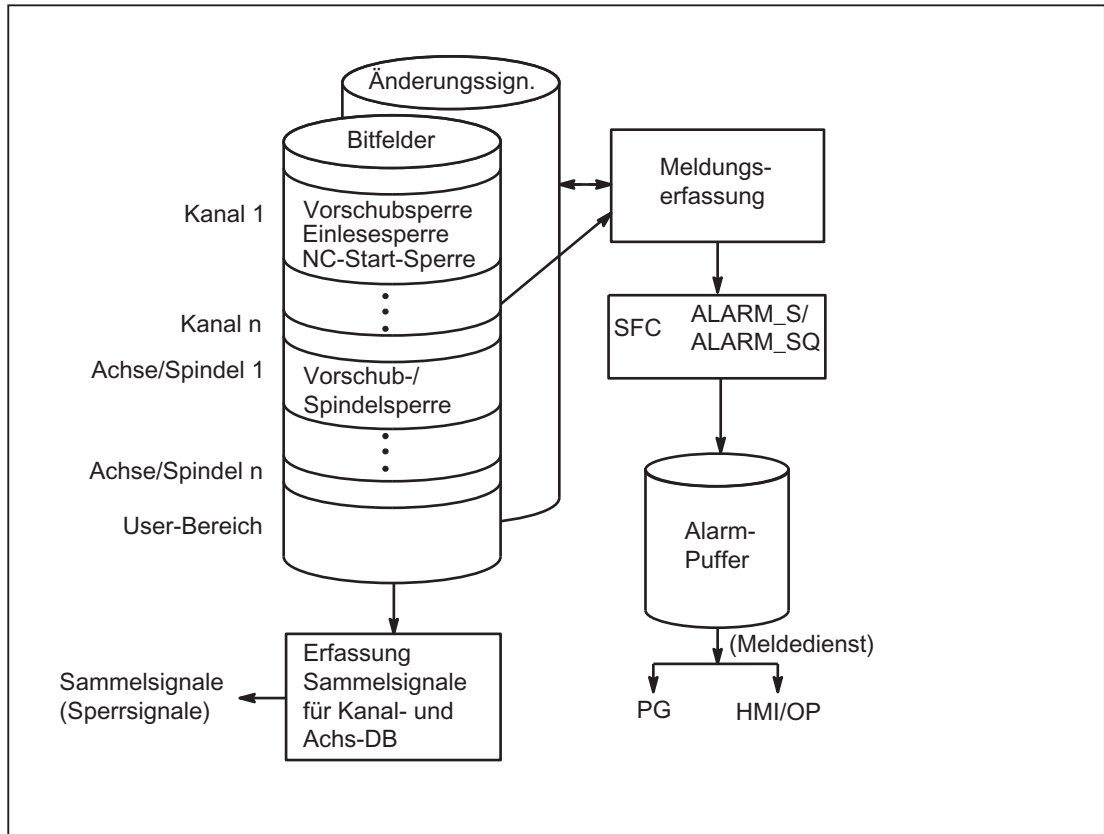


Bild 12-7 Erfassung und Meldung von PLC-Ereignissen

### Erweiterungen der PLC-Alarme über den Baustein FC 10

Mit dem FB 1-Parameter "ExtendAIMsg" wird eine Auswahl über den PLC-Alarmmechanismus getroffen.

Bei "ExtendAIMsg:= FALSE" ist das bisherige Verfahren des FC 10 mit dem DB 2 als Bitfeld-Datenbaustein aktiv. Es gelten die bekannten Einschränkungen bezüglich Kanal- und Achs-Anzahl.

Dagegen wird bei "ExtendAIMsg:= TRUE" die Erweiterung des FC 10 aktiv. Der DB 2 und der DB 3 werden wie bisher angelegt. Der Anwender muss im DB 2 die Bits setzen bzw. rücksetzen. Die Parametrierung über Meldung und Alarm und einer Parametrierung des Zahlenwerts der 2. Dekade der Useralarme sind im DB 5 enthalten.



Die Erweiterungen sind:

- Unterstützung für 10 Kanäle, 31 Achsen.
- Es stehen Bereiche für Vorschubhalt, Einleesperre usw. ohne Meldungen zur Verfügung. Die Informationen aus diesem Bereich werden in Abhängigkeit vom FC 10-Parameter "ToUserIF" zusammen mit den zugehörigen Meldungsbits als Sammelsignale auf die Nahtstelle im DB 21, DB 31 gelegt. Somit entfällt das bisher recht umständliche Handling dieser Signale.
- Die Alarmer / Meldungen bekommen für den Userbereich 0 zusätzlich zur Alarmnummer auch 16 Bit Integer Zusatzwerte (%Z Parameter im Alarmtext). Die 16 Bit Integerwerte sind vom Anwender in den DB 2 in die Array-Variable ZInfo0 parallel zum Setzen eines Alarmbits zu schreiben. Für jedes Bit im User Bereich 0 ist ein Integerwert vorhanden, siehe UDT1002 im Grundprogramm.
- Die Usermeldungen sind in der 2. Dekade der Meldenummer im Zahlenbereich 0 bis 9 parametrierbar. Der Anzeigewert der 2. Dekade muss vom Anwender in den DB 5 in die Array-Variable UserDek2No geschrieben werden. Für jeden Userbereich kann eine Zahl festgelegt werden, siehe DB 5 im Grundprogramm.

Standardmäßig ist der Wert 0 für die 2. Dekade vorbelegt.

Die Strukturierung des DB 2 ist im UDT1002 (Grundprogramm) erkennbar. Der UDT1002 sollte bei den neuen Alarmfunktionen in der Symboltabelle dem DB 2 symbolisch zugeordnet werden.

Am Beginn des DB 2 liegen Bitfelder für Signale ohne Meldungserzeugung. Danach folgt ein 64 Integer großes Array für die Zusatzinfo des Userbereichs 0.

Anschließend folgen die Bereiche, die auch Alarmer / Meldungen absetzen (siehe Listenhandbuch). Diese Bereiche sind ergänzt auf 10 Kanäle, 31 Achsen.

### Einfache Umsetzung eines Anwenderprogramms auf die neuen Alarmer

Im Source Container des Grundprogramms gibt es das File "udt2\_for\_Convert.awl", das folgende Strukturelemente aus UDT1002 enthält:

- ChanA als Array von 1 ... 8
- AxisA als Array von 1 ... 18
- UserA als Array von 1 ... 31

Dieser UDT2 ist über den KOP/FUP/AWL-Editor zu übersetzen. Der UDT2 muss in der Symboltabelle dem DB 2 zugewiesen sein.

Von den Bausteinen, die Zuweisungen auf den DB 2 haben, sollen Quellen erzeugt werden. Alternativ können natürlich auch von allen Bausteinen Quellen erstellt werden. Der UDT1002 muss nun noch in der Symboltabelle dem DB 2 zugeordnet werden. Anschließend sind die Quellen neu zu übersetzen.

Jetzt sind alle Alarmzuweisungen auf die neuen Datenbereiche im DB 2 zugeordnet und es muss nur noch der FB 1-Parameter "ExtendAIMsg" auf TRUE gesetzt werden.

Nach einem Power-On-Reset ist das Alarmverhalten identisch wie bisher.

### 12.9.3 Nahtstelle PLC/MCP/BHG

#### Allgemeines

Es gibt verschiedene Anschlussmöglichkeiten für die Maschinensteuertafel (MCP) und das Bedienhandgerät (BHG). Dies resultiert teilweise aus der Historie der MCP und des BHG. In dieser Beschreibung ist vorrangig der Anschluss der Ethernet-Komponenten beschrieben.

Die Maschinensteuertafel (MCP) und das Bedienhandgerät (BHG) sind bei SINUMERIK 840D sl über den Ethernet-Bus angekoppelt, der auch die TCU mit der NCU verbindet. Dies hat den Vorteil, dass nur eine Busleitung zur Bedieneinheit verlegt werden muss.

#### Topologie SINUMERIK 840D sl

Bei 840D sl wird die Maschinensteuertafel und auch das Bedienhandgerät an den Ethernet-Bus des CP 840D sl angeschlossen (siehe folgendes Bild). Sollen weitere Tasten und Anzeigen für Kundenbedientafeln angeschlossen werden, so kann dies über ein weiteres Tastatur-Interface (MCP ohne Bedienteil) erfolgen. Je Tastatur-Interface können über Flachbandkabel 64 Taster, Schalter etc. und 64 Anzeigeelemente angeschlossen werden.

Die von der MCP kommenden Signale werden vom integrierten Ethernet des CP-840D sl ins DPR (Dual-Port-RAM) zur PLC kopiert. Das Grundprogramm der PLC trägt die ankommenden Signale in das am FB 1 projektierte Eingangsabbild ein. Im Normalfall werden die NCK-relevanten Signale vom Grundprogramm auf die NC/PLC-Nahtstelle verteilt. Bei Bedarf können die Signale vom Anwender modifiziert werden.

Die Signale von der PLC zur MCP (Anzeigen) nehmen den umgekehrten Weg.

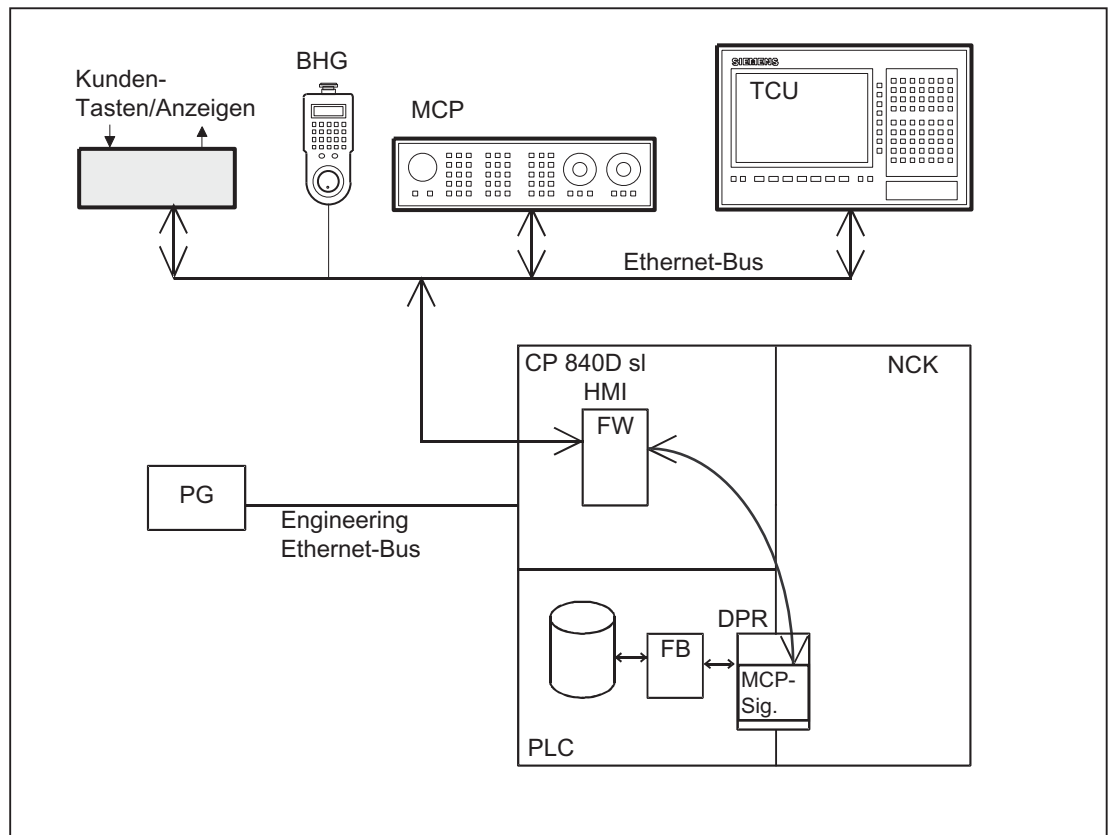


Bild 12-8 Ankopplung der Maschinensteuertafel bei 840D sl

## Busadressen

Bei den Ethernet-Komponenten sind MAC- und IP-Adressen bzw. logische Namen für die Kommunikation bestimmend. Die Umsetzung von logischen Namen in die MAC bzw. IP Adressen erfolgt durch die Systemprogramme der Steuerung. In der PLC wird der numerische Teil des logischen Namens für die Kommunikation verwendet. Dieser numerische Teil wird vom Anwender über den Parameter "MCPxBusAdr" an den FB 1 übergeben.

Der logische Name einer MCP oder BHG beginnt immer mit "DIP". Danach folgt eine Nummer, die der Schalterstellung der MCP-Komponente entspricht (z. B. DIP 192, DIP 17).

### MCP-Nahtstelle in der PLC

Die Signale der Maschinensteuertafel werden standardmäßig über die E-/A-Nahtstelle in den PLC-Bereich geführt. Es ist dabei zwischen den NC- und den Maschinenspezifischen Signalen zu unterscheiden. Die NC-spezifischen Tastensignale werden standardmäßig vom FC 19 (oder FC 24, FC 25, FC 26 je nach MCP-Variante) auf die jeweilige BAG-, NCK-, Achs- und Spindel-spezifische Nahtstelle verteilt. Umgekehrt werden die zugehörigen Statussignale auf die MCP-Nahtstelle rangiert. Der FC 19 oder die weiteren oben erwähnten Bausteine müssen dafür im **Anwenderprogramm** aufgerufen werden.

Die Kundentasten, mit denen diverse Maschinenfunktionen ausgelöst werden können, müssen direkt vom Anwenderprogramm ausgewertet werden. Ebenso sind von diesem auch die zugehörigen Statussignale auf den Ausgabebereich für die LEDs zu rangieren.

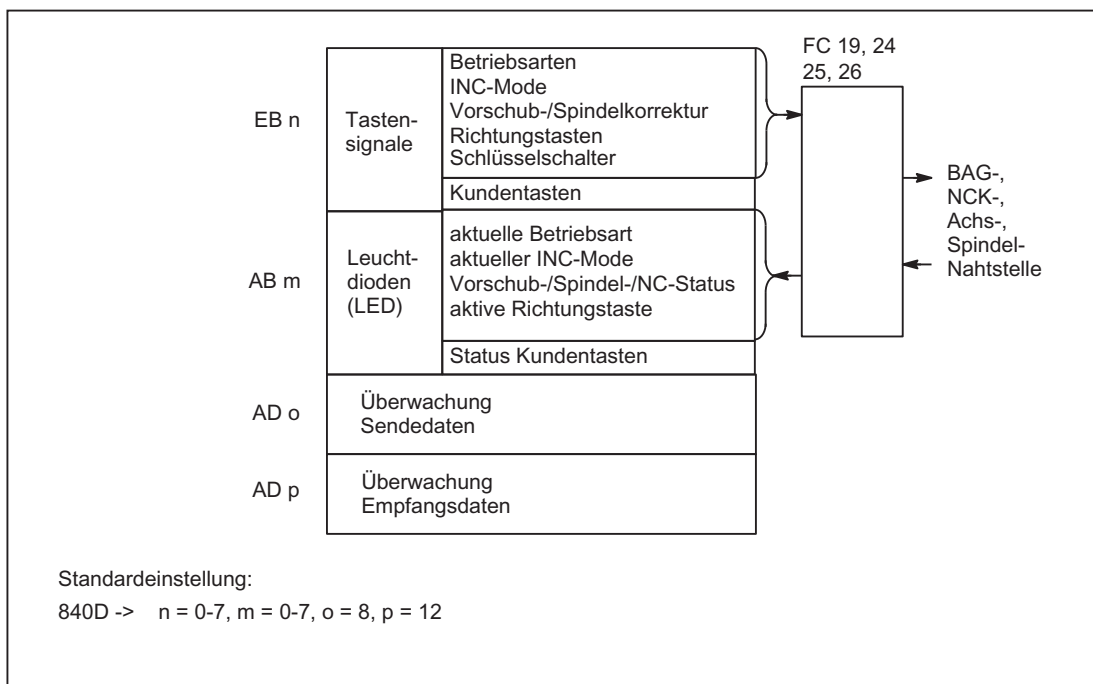


Bild 12-9 Nahtstelle von und zur Maschinensteuertafel

## 12.10 Struktur und Funktionen des Grundprogramms

### Allgemeines

Das PLC-Programm ist modular aufgebaut. Die Organisationsbausteine (OB) bilden die Schnittstelle zwischen dem Betriebssystem und dem Grund- und Anwenderprogramm.

- Neustart (Warmstart) mit Anlauf und Synchronisation (OB 100)
- Zyklischer Betrieb (OB 1)
- Prozessalarme (OB 40)
- Asynchrone Fehler: Diagnose-Alarm (OB 82), Baugruppenausfall (OB 86)

In den Organisationsbausteinen (OB) müssen vom Anwender die Aufrufe der Funktionsbausteine des Grund- und Anwenderprogramms programmiert werden.

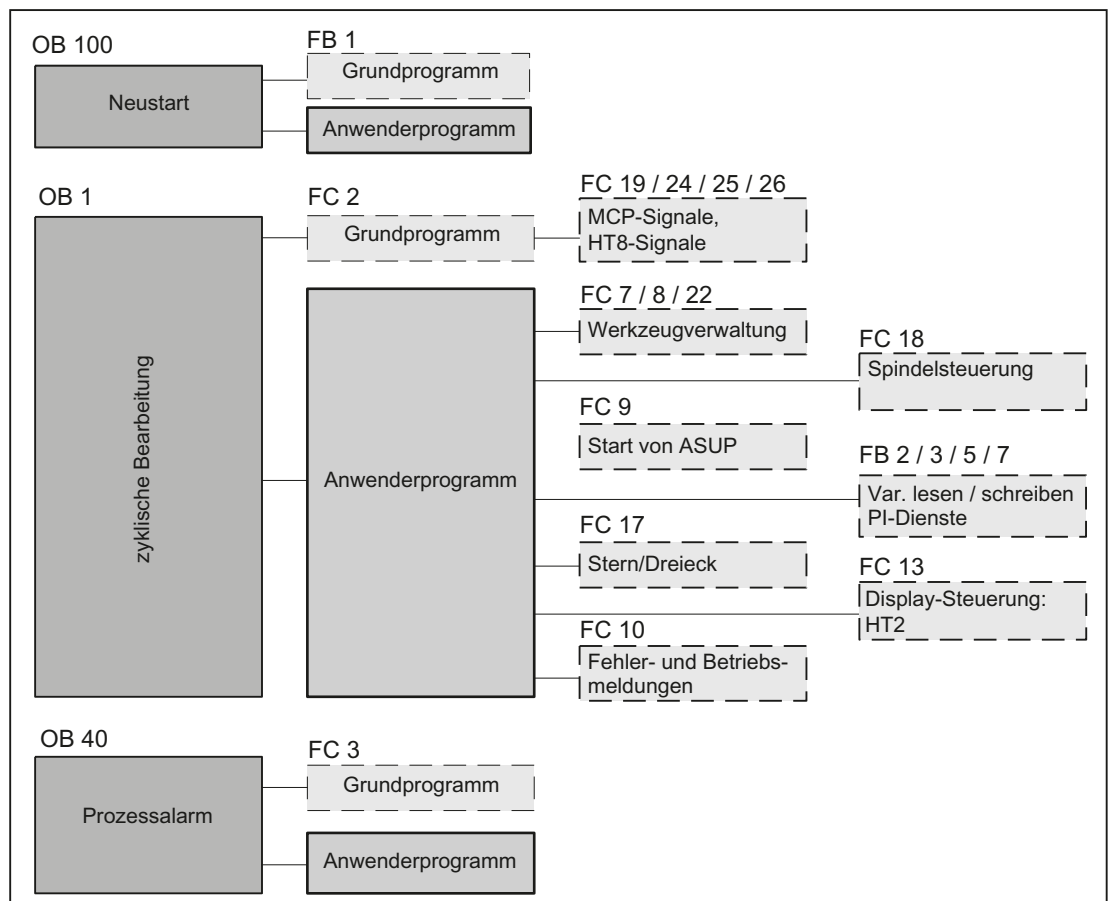


Bild 12-10 Struktur des Grundprogramms (Prinzip)

## 12.10.1 Anlauf und Synchronisation NCK-PLC

### Laden des Grundprogramms

Das Laden des Grundprogramms mit dem S7-Tool muss im Stopp-Zustand der PLC erfolgen. Es wird so sichergestellt, dass alle Bausteine des Grundprogramms beim nächsten Anlauf richtig initialisiert werden. Es kann sonst zu undefinierten Zuständen der PLC kommen (z. B. Blinken aller PLC-LEDs).

### Anlauf

Im Anlauf erfolgt die Synchronisation zwischen NCK und PLC. Es werden die System- und Anwenderdatenbausteine auf Vollständigkeit und die wichtigsten GP-Parameter auf Plausibilität geprüft. Im Fehlerfall meldet das Grundprogramm einen Alarm (am HMI sichtbar) und überführt die PLC in den Stopp-Zustand.

Das Betriebssystem durchläuft nach der Systeminitialisierung den Organisationsbaustein OB 100 und beginnt den zyklischen Betrieb immer am Anfang des OB 1.

### Synchronisation

Beim Hochlauf synchronisiert sich die PLC mit HMI, NCK und CP.

### Lebenszeichen

Nach ordnungsgemäßem Anlauf und erstem vollständigen OB 1-Zyklus (Grundstellungszyklus) tauschen PLC und NCK fortlaufend Lebenszeichen aus. Bleibt das Lebenszeichen des NCK aus, wird die PLC/NCK-Nahtstelle initialisiert und das Signal "NCK-CPU-ready" im DB 10 wird auf FALSE gesetzt.

## 12.10.2 Zyklischer Betrieb (OB 1)

### Allgemeines

Die komplette Bearbeitung der NCK/PLC-Schnittstelle erfolgt ausschließlich im zyklischen Betrieb. Das Grundprogramm läuft - zeitlich gesehen - vor der Bearbeitung des Anwenderprogramms. Um die Grundprogrammlaufzeit gering zu halten, werden nur die Steuer-/Statussignale zyklisch übertragen, die Hilfs- und G-Funktionsübergabe wird nur auf Anforderung bearbeitet.

Folgende Funktionen werden im zyklischen Teil des Grundprogramms ausgeführt:

- Übertragung der Steuer-/Statussignale
- Verteilung der Hilfsfunktionen
- M-Decodierung (M00 - M99)
- M-, S-, F-Verteilung
- Übertragung der MCP-Signale über NCK
- Erfassung und Aufbereitung der Anwender-Fehler- und Betriebsmeldungen

### Steuer-/ Statussignale

Gemeinsames Merkmal der Steuer- und Statussignale ist, dass es sich um Bitfelder handelt. Das Grundprogramm aktualisiert sie am Anfang des OB 1.

Die Signale sind in folgende Gruppen eingeteilt:

- Allgemeine Signale
- BAG-spezifische Signale (wie z. B. Betriebsarten)
- Kanalspezifische Signale (wie z. B. Programm- und Vorschub-Beeinflussung)
- Achs- und Spindel-spezifische Signale (wie z. B. Vorschub-Sperre)

### Hilfs- und G-Funktionen

Die Hilfs- und G-Funktionen weisen folgende Merkmale auf:

- Sie werden jeweils satzsynchron zur PLC übertragen (bezogen auf einen Teileprogrammsatz).
- Die Übergabe erfolgt quittungsgesteuert.
- Die Quittungszeiten beeinflussen unmittelbar die Satzausführungszeit von NC-Sätzen mit quittierungspflichtigen Hilfsfunktionen.

Der Wertebereich ist in folgender Tabelle dargestellt:

Funktion	Struktur		Wertebereich		Datentyp	
	1. Wert	2. Wert	1. Wert	2. Wert	1. Wert	2. Wert
G-Funktion		G-Funktion		255 <sup>1)</sup>		Byte
M-Wort	M-Gruppe	M-Wort	99	99.999.999	Word	DWord
S-Wort	Spindel-Nr.	S-Wort	6	Gleitpunkt <sup>2)</sup>	Word	DWord
T-Wort	Magazin-Nr.	T-Wort	99	65535	Word	Word
D-Wort	-	D-Wort	99	255	Byte	Byte
H-Wort	H-Gruppe	H-Wort	99	Gleitpunkt	Word	DWord
F-Wort	Achs-Nr.	F-Wort	18	Gleitpunkt	Word	DWord

1) relative Nummer, die je G-Gruppe übergeben wird

2) entsprechend STEP 7-Format (24 Bit Mantisse, 8 Bit Exponent)

Über den Hilfs-/G-Funktionsverteiler werden die vom NCK übergebenen M-, S-, T-, H-, D-, F-Werte zusammen mit den zugehörigen Änderungssignalen auf die Nahtstelle **KANAL-DB** (siehe Listenhandbuch) ausgegeben. Dabei werden jeweils die beiden Werte der Hilfsfunktion in die entsprechenden Datenworte übertragen. Das dazugehörige Änderungssignal wird für einen PLC-Zyklus auf 1 aktiv geschaltet. Mit dem Rücksetzen des Änderungssignals erfolgt die Quittierung an den NCK. Die Quittierung von schnellen Hilfsfunktionen erfolgt gleich nachdem das Grundprogramm die Hilfsfunktion erkannt hat durch das Grundprogramm.

Zusätzlich zur Verteilung der Hilfs- und G-Funktionen erfolgt für ausgewählte Signale eine weitere Aufbereitung wie im Folgenden ausgeführt.

### M-Decoder

Mit M-Funktionen können sowohl Schaltbefehle als auch Festpunkt-Werte übergeben werden. Für die Standard M-Funktionen (Bereich M00 - M99) werden ausdecodierte dynamische Signale auf die Nahtstelle **KANAL-DB** ausgegeben (Signaldauer = 1 Zykluszeit).



## G-Gruppen-Decoder

Bei den vom NCK übergebenen G-Funktionen werden die zugehörigen Gruppen ausdecodiert und die jeweils aktuelle G-Nummer in das entsprechende Nahtstellenbyte des KANAL-DB eingetragen, d. h. in den Kanal-DBs sind alle aktiven G-Funktionen eingetragen. Die eingetragenen G-Funktionen bleiben auch nach NC-Programm-Ende oder Programm-Abbruch erhalten.

---

### Hinweis

Bei Systemanlauf wird in alle G-Gruppenbytes der Wert "0" eingetragen.

---

## M-, S-, F-Verteiler

Mit dem M-, S-, F-Verteiler werden spindelspezifische M-Worte M(1..6)=[3,4,5], S-Worte und F-Worte für axiale Vorschübe in die zugehörigen **Spindel- und Achs-DBs** eingetragen. Kriterium für die Verteilung ist die erweiterte Adresse, die bei M-Worten, S-Worten und axialen F-Worten an die PLC übergeben wird.

## MCP-Signal-Übertragung

Die MCP-Signale werden abhängig vom Bus-Anschluss entweder direkt zur PLC oder indirekt über interne Verfahrensweisen mit Hilfe des Grundprogramms in die parametrisierten E/A Bereiche übertragen.

## Anwendermeldungen

Die Erfassung und Aufbereitung der Anwender-Fehler- und Betriebsmeldungen erfolgt durch einen FC des Grundprogramms.

### 12.10.3 Zeitalarm Bearbeitung (OB 35)

Für die Zeitalarmbearbeitung ist der **OB 35** vom Anwender zu programmieren. Das Zeitraster des OB 35 ist im Standard auf 100 ms eingestellt. Durch das STEP 7-Tool "HW-Konfig" kann jedoch ein anderes Zeitraster gewählt werden. Allerdings darf eine Zeiteinstellung des OB 35 nicht kleiner als etwa 3 ms verwendet werden, da es sonst zum Stopp der PLC-CPU kommt. Der Stopp wird verursacht durch das Lesen der Systemzustandsliste (SZL) des HMI bei dessen Hochlauf. Dieses Lesen bedeutet eine Blockierung der Ablafebene-Steuerung (AES) für ca. 2 ms. Dadurch wird der OB 35 bei kleineren Zeiteinstellungen nicht mehr korrekt bearbeitet.

#### 12.10.4 Prozessalarm Bearbeitung (OB 40)

Ein Prozessalarm **OB 40** (Interrupt) kann z. B. durch entsprechend projektierte Peripherie oder durch bestimmte NC-Funktionen ausgelöst werden. Wegen der unterschiedlichen Herkunft des Interrupts muss das PLC-Anwenderprogramm im OB 40 zuerst die Interrupt-Ursache interpretieren. Die Interrupt-Ursache ist in den Lokaldaten des OB 40 enthalten.

**Literatur:**

SIMATIC STEP 7-Beschreibung oder Online-Hilfe von STEP 7

#### 12.10.5 Diagnosealarm, Baugruppenausfall Bearbeitung (OB 82, OB 86)

##### Allgemeines

Eine Baugruppendiagnose bzw. Baugruppenausfall von einer Peripherie-Baugruppe löst den OB 82 bzw. OB 86 aus. Diese Bausteine werden vom Grundprogramm mitgeliefert. In diesen OBs wird der Grundprogramm-Baustein FC 5 aufgerufen. Dieser ist standardmäßig so beschaltet, das er einen PLC-Stop bei einer Fehlererkennung auslöst.

Ein PLC-Stop:

- wird nicht ausgelöst, wenn die Fehlerursache behoben ist.
- erfolgt nicht bei dem an den FB 1-Parametern angegebenen PROFIBUS-MCPs.

Durch Änderung der Parametrierung des FC 5-Parameters kann das Verhalten geändert werden.

##### PROFIBUS-Diagnose

Die Slaves der beiden PROFIBUS-Anschlüsse MPI/DP bzw. DP1 werden durch das Grundprogramm als Sammelmeldung jedes Bussystems in Form eines Ready-Signals dem Anwenderprogramm gemeldet:

DB10 DBX92.0 (MPI/DP Bus Slaves OK) und

DB10 DBX92.1 (DP1 Bus Slaves OK)

Die Sammelmeldung wird abgeleitet aus dem LED-Status des jeweiligen PROFIBUS (Systemzustandsliste SZL 0x174).

Wird eine Störung oder Ausfall eines Slaves erkannt, wird zusätzlich der Alarm 400551 bzw. 400552 gemeldet. Bei Wegfall der Störung wird der Alarm automatisch gelöscht. Eine detaillierte Information über die Störung bzw. den Ausfall wird bei den Alarmmeldungen in einem zukünftigen HMI-Softwarestand im Diagnosebild als genaue Störungsursache angezeigt. Für dieses Diagnosebild ist kein FC 125, FB 125 erforderlich, da die Informationen direkt vom HMI über die entsprechenden SZL-Funktionen des PLC-Betriebssystems über den Kommunikationskanal zur PLC abgerufen werden. Dies erspart einerseits den relativ großen Baustein in der PLC und verringert auch die PLC-Zykluszeit. Da der FB 125 während seiner Bearbeitung unter Interrupt-Sperre arbeitet, können höherpriorie OBs schneller zur Ausführung kommen.

---

#### Hinweis

Bis der HMI-Softwarestand mit den Diagnosebildern für den PROFIBUS verfügbar ist, wird für die PROFIBUS-Diagnose der Einsatz von FC 125 empfohlen. Dieser Baustein belegt relativ wenig Zykluszeit und über die vorher genannten DB 10-Nahtstellensignale kann eine Aktualisierung der Slave-Stati mittels Aktivieren des Bausteins FC 125 erfolgen.

---

## 12.10.6 Verhalten bei NCK-Ausfall

### Allgemeines

Während des zyklischen Betriebs erfolgt eine ständige Überwachung der NCK-Bereitschaft durch das PLC-Grundprogramm mittels Abfrage eines Lebenszeichens. Reagiert der NCK nicht mehr, so wird die Nahtstelle NCK/PLC initialisiert und das Signal **NCK-CPU-Ready** im Bereich der **Signale von NC (DB10.DBX104.7)** wird zurückgesetzt. Des Weiteren werden die Signale, die vom NCK an die PLC und von PLC an NCK übergeben werden, in einen Grundzustand gesetzt.

Die PLC selbst bleibt aktiv, so dass weiterhin Maschinenfunktionen von ihr gesteuert werden können. Es bleibt aber dem Anwenderprogramm überlassen, einen sicheren Zustand der Maschine einzuleiten.

### Signale NCK → PLC

Bei den Signalen, die von NCK an PLC übergeben werden, müssen folgende Gruppen unterschieden werden:

- Statussignale von NCK, Kanälen, Achsen und Spindeln
- Änderungssignale der Hilfsfunktionen
- Werte der Hilfsfunktionen
- Werte der G-Funktionen

#### Statussignale:

Die Statussignale von NCK, den Kanälen, Achsen und Spindeln werden zurückgesetzt.

**Änderungssignale der Hilfsfunktionen:**

Die Änderungssignale der Hilfsfunktionen werden ebenfalls zurückgesetzt.

**Werte der Hilfsfunktionen:**

Die Werte der Hilfsfunktionen bleiben erhalten, so dass rekonstruierbar ist, welche Funktionen als letzte vom NCK angestoßen wurden.

**Werte der G-Funktionen:**

Die Werte der G-Funktionen werden rückgesetzt (d. h. es wird jeweils der Wert 0 eingetragen).

**Signale PLC → NCK**

Bei den Signalen, die von der PLC an den NCK übergeben werden, wird zwischen Steuersignalen und Aufträgen, die per FCs an den NCK übergeben werden, unterschieden.

**Steuersignale:**

Die Steuersignale von PLC an NCK werden eingefroren; die zyklische Aktualisierung durch das PLC-Grundprogramm wird unterbrochen.

**Aufträge von PLC an NCK:**

Die FCs und FBs, mit denen Aufträge an den NCK übergeben werden, dürfen durch das PLC-Anwenderprogramm nicht mehr bearbeitet werden, da unter Umständen fehlerhafte Rückmeldungen auftreten können. Beim Hochlauf der Steuerung muss im Anwenderprogramm auch solange gewartet werden mit einer Auftragsaktivierung (z. B. NCK-Daten lesen), bis das Signal **NCK-CPU ready** gesetzt ist.

## 12.10.7 Funktionen des Grundprogramms mit Aufruf vom Anwenderprogramm

### Allgemeines

Zusätzlich zu den Modulen des Grundprogramms, die am Anfang vom OB 1, OB 40 und OB 100 aufzurufen sind, werden Funktionen bereitgestellt, die an geeigneter Stelle im Anwenderprogramm aufgerufen und parametrisiert werden müssen.

Mit diesen Funktionen können z. B. folgende Aufträge von PLC an den NCK übergeben werden:

- Verfahren von konkurrierenden Achsen (FC 18)
- Starten asynchroner Unterprogramme (ASUPs) (FC 9)
- Anwahl von NC-Programmen (FB 4)
- Steuern der Spindel (FC 18)
- Lesen/Schreiben von Variablen (FB 2, FB 3)

---

### Hinweis

#### Kontrolle und Diagnose eines Funktionsaufrufs des PLC-Grundprogramms

Zur Vereinfachung der Kontrolle und Diagnose eines Funktionsaufrufs (FB oder FC) des PLC-Grundprogramms, die über einen Anstoß (z. B. über Parameter Req, Start) gesteuert werden und die als Ausgangsparameter eine Ausführungsquittierung (z. B. über Parameter Done, NDR, Error) liefern, ist wie folgt vorzugehen.

Es sollte eine von anderen Signalen zusammengefasste Variable gesetzt werden, die den Anstoß für den Funktionsaufruf herbeiführen. Das Rücksetzen der Startbedingungen darf nur von den Parametern Done, NDR, Error abgeleitet werden.

Das entsprechende Steuerwerk kann entweder vor oder hinter dem Funktionsaufruf platziert werden. Wenn das Steuerwerk hinter dem Aufruf platziert ist, können die Ausgangsvariablen als lokale Variablen definiert sein (Vorteil: Reduzierung der globalen Variablen Merker, Datenvariable und zeitliche Vorteile gegenüber Datenvariable).

Der Anstoßparameter muss eine globale Variable sein (z. B. Merker, Datenvariable).

Im OB 100 sind vom Anwenderprogramm noch aktive Aufträge zurückzusetzen (Parameter Req, Start, etc. von TRUE ⇒ FALSE). Durch ein Aus- und Wiedereinschalten kann ein Zustand mit noch aktivierten Aufträgen entstehen.

---

### Konkurrierende Achsen

Konkurrierende Achsen zeichnen sich durch folgende Eigenschaften aus:

- Sie müssen als solche per NC-Maschinendatum definiert sein.
- Sie können entweder von der PLC oder von NC mittels JOG-Tasten verfahren werden.
- Der Start von der PLC ist in den NC-Betriebsarten MDA und AUTOMATIK per FC möglich.
- Der Start erfolgt unabhängig von NC-Satzgrenzen.

Es stehen FCs für Positionierachsen, Teilungsachsen und Spindeln (FC 18) zur Verfügung.

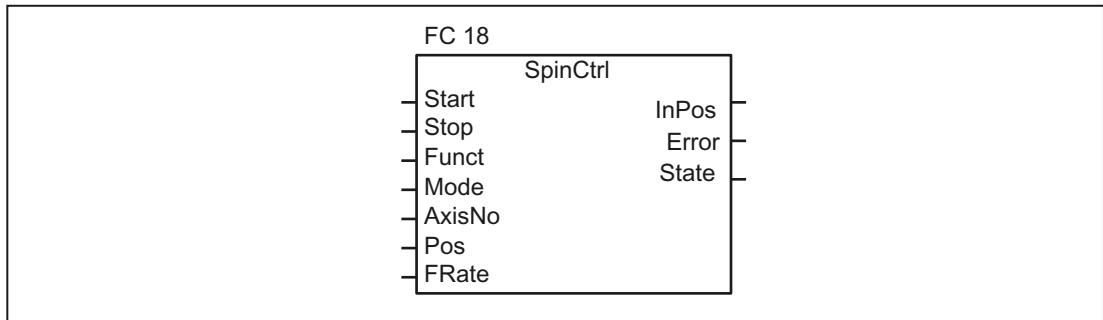
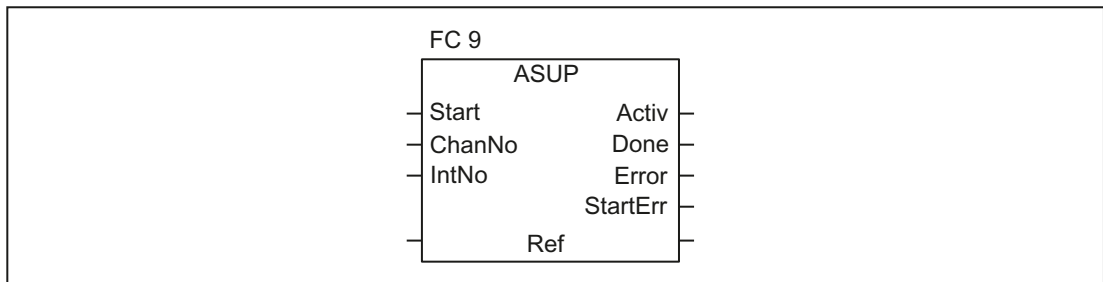


Bild 12-11 FC 18 Ein-/Ausgabe-Parameter

### Asynchrone Unterprogramme (ASUP)

Mit ASUP können beliebige Funktionen in der NCK ausgelöst werden. Voraussetzung dafür, dass ein ASUP von der PLC aus gestartet werden kann, ist dessen Existenz und Vorbereitung vom NC-Programm bzw. durch FB 4 PI-Dienste (ASUP).

Ein derart vorbereitetes ASUP kann von der PLC zu einem beliebigen Zeitpunkt gestartet werden. Das in einem parametrisierten Kanal des FC 9 laufende NC-Programm wird durch das ASUP unterbrochen. Der Start eines ASUP erfolgt durch Aufruf des FC 9 vom Anwenderprogramm durch den Parameter Start = 1.



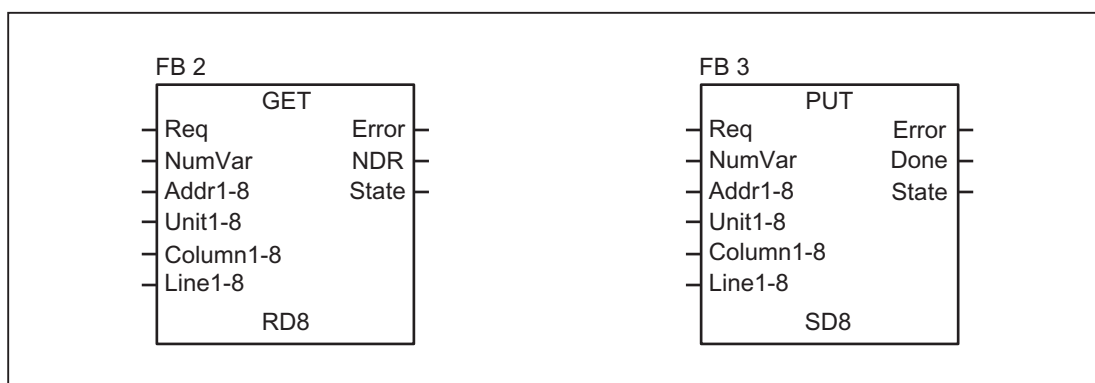
#### Hinweis

Ist ein ASUP noch nicht von einem NC-Programm oder vom FB 4 (ASUP) vorbereitet worden (z. B. keine Interrupt-Nr. vergeben), wird Start-Fehler (StartErr = TRUE) gemeldet.

## NC-Variable lesen/schreiben

Mit dem FB GET können Variable des NCK gelesen und mit dem FB PUT können Werte in NCK-Variable eingetragen werden. Die NCK-Variablen werden über Bezeichner an den Eingängen Addr1 bis Addr8 adressiert. Die Bezeichner (Symbole) zeigen auf Adressangaben, die in einem globalen DB hinterlegt sein müssen. Zur Erzeugung dieses DBs wird zusammen mit dem Grundprogramm eine PC-Software (NC-Var-Selector) ausgeliefert, mit der aus einer ebenfalls mitgelieferten Tabelle die gewünschten Variablen selektiert werden können. Die selektierten Variablen werden zunächst in einer zweiten, projektabhängigen Liste gesammelt. Mit dem Kommando **DB erzeugen** wird eine "\*.AWL"-Datei erzeugt, die in die Programmdatei für die betreffende Maschine einzubinden und mit dem Maschinenprogramm zusammen zu kompilieren ist.

Es können 1 bis 8 Werte mit einem Lese- oder Schreibauftrag gelesen bzw. beschrieben werden. Die Werte werden - wenn notwendig - konvertiert [z. B. werden die NCK-Gleitpunktwerte (64 Bit) in das PLC-Format (32 Bit mit 24 Bit Mantisse und 8 Bit Exponent) gewandelt und umgekehrt]. Durch die Wandlung von 64 Bit auf 32 Bit REAL geht Genauigkeit verloren. Die max. Genauigkeit der 32 Bit REAL-Zahlen liegt etwa bei  $10^7$ .



## Funktionen AG\_SEND / AG\_RECV

Die Funktionen AG\_SEND / AG\_RECV entsprechen den Funktionen der Bibliothek "SIMATIC\_NET\_CP" der S7-300 CPU in STEP 7. Generell gilt für diese Funktionen die Online-Hilfe zu diesen Funktionen.

Die Funktionen AG\_SEND / AG\_RECV können über die integrierte "CP 840D sl" zum Datenaustausch mit einer weiteren Station genutzt werden. Eine Beschreibung der Funktionen ist im Kapitel "Bausteinbeschreibungen (Seite 999)" zu finden.

---

### Hinweis

Weitere Kommunikationsbausteine (z. B. BSEND, USEND), die über eine CP343-1 verfügbar sind, werden bei der SINUMERIK 840D sl nicht unterstützt.

---

## 12.10.8 Symbolische Programmierung des Anwenderprogramms mit Nahtstellen-DB

### Allgemeines

---

#### Hinweis

Auf der Liefer-CD des der Toolbox 840D sl in der Bibliothek des Grundprogramms werden die Dateien NST\_UDTB.AWL und TM\_UDTB.AWL mitgeliefert.

---

In dem CPU-Programm des Grundprogramms sind die kompilierten UDT-Bausteine aus diesen beiden Dateien abgelegt.

Ein UDT ist ein vom Anwender definierter Datentyp, der z. B. einem Datenbaustein zugeordnet werden kann, der in der CPU erzeugt wurde.

In diesen UDT-Bausteinen sind symbolische Namen nahezu aller Nahtstellensignale definiert.

Verwendet werden die UDT-Nummern 2, 10, 11, 19, 21, 31, 71, 72, 73.

Folgende Zuordnung ist hierbei getroffen worden:

UDT-Zuordnungen		
UDT-Nummer	Zuordnung Nahtstellen-DB	Bedeutung
UDT 2	DB 2	Alarmer / Meldungen
UDT 10	DB 10	NCK-Signale
UDT 11	DB 11	BAG-Signale
UDT 19	DB 19	HMI-Signale
UDT 21	DB 21 bis DB 30	Kanal-Signal
UDT 31	DB 31 bis 61	Achs/Spindel-Signale
UDT 71	DB 71	Werkzeugverwaltung: Be-/Endladestellen
UDT 72	DB 72	Werkzeugverwaltung: Wechsel in Spindel
UDT 73	DB 73	Werkzeugverwaltung: Wechsel in Revolver
UDT 77	DB 77	MCP- und BHG-Signale mit Standard SDB210
UDT 1002	DB 2	erweiterte Alarmer / Meldungen (FB 1-Parameter "ExtendAlMsg:=TRUE")

Für eine symbolische Programmierung der Nahtstellensignale müssen zuerst die Datenbausteine der Nahtstelle mit dem Symboleditor symbolisch zugeordnet werden.

Hierzu wird z. B. das Symbol "AchseX" dem Operand DB 31 mit dem Datentyp UDT 31 in der Symboldatei zugeordnet.



Nach dieser Eingabe kann die Programmierung des STEP 7-Programms für diese Nahtstelle symbolisch erfolgen.

---

#### Hinweis

Bisher erstellte Programme, die mit diesen Nahtstellen-DBs arbeiten, können auch in diese symbolische Form umgesetzt werden. Hierzu ist dann im bisher erstellten Programm eine voll qualifizierte Anweisung beim Datenzugriff z. B. "U DB31.DBX60.0" für (Spindel/ keine Achse) notwendig. Dieser Befehl wird bei Einschaltung der Symbolik im Editor in "AchseX.E\_SpKA" umgesetzt.

---

### Beschreibung

In den beiden AWL-Dateien NST\_UDTB.AWL und TM\_UDTB.AWL sind abgekürzte symbolische Namen der Nahtstellensignale definiert.

Um den Bezug auf die Namen der Nahtstellensignale herzustellen, ist hinter jedem Signal der Name im Kommentar mit aufgeführt.

Die Namen sind englischsprachig abgeleitet. Die Kommentare sind englischsprachig.

Über den STEP 7-Editor können beim Aufschlagen des UDT-Bausteins die symbolischen Namen, Kommentare und Absolutadressen sichtbar gemacht werden.

---

#### Hinweis

Die nicht verwendeten Bits und Bytes werden z. B. mit der Bezeichnung "f56\_3" aufgeführt:

- "56": Byteadresse des jeweiligen Datenbausteins
  - "3": Bit-Nummer in diesem Byte
- 

## 12.10.9 M-Dekodierung nach Liste

### Funktionsbeschreibung

Mit dem Aktivieren der Funktion **M-Dekodierung nach Liste** über den GP-Parameter des FB 1 "ListMDecGrp" (Anzahl M-Gruppen zur Dekodierung) können bis zu 256 M-Funktionen mit erweiterter Adresse vom Grundprogramm dekodiert werden.

Die Zuordnung von M-Funktion mit erweiterter Adresse und zu setzendem Bit in der Signalliste wird in der Dekodierliste festgelegt. Dabei erfolgt eine Einteilung in Gruppen.

Es gibt in der Signalliste 16 Gruppen mit jeweils 16 Bit als dekodierte Signale.

Die Dekodier- und Signalliste existieren nur ein Mal, also kanalunabhängig.

Die M-Funktionen werden dekodiert und wenn sie in der Dekodierliste enthalten sind, dann wird das zugeordnete Bit in der Signalliste gesetzt.

Mit dem Setzen in der Signalliste erfolgt gleichzeitig ein Setzen der Einlesesperre im zugehörigen NCK-Kanal durch das Grundprogramm.

Ein Rücksetzen der Einlesesperre im Kanal erfolgt dann, wenn vom Anwender alle von diesem Kanal ausgegeben Bits in der Signalliste zurückgesetzt und damit quittiert wurden.

Bei Ausgabe einer aus Liste dekodierten M-Funktion als schnelle Hilfsfunktion erfolgt keine Einlesesperre.

Folgendes Bild stellt die Struktur der **M-Dekodierung nach Liste** dar:

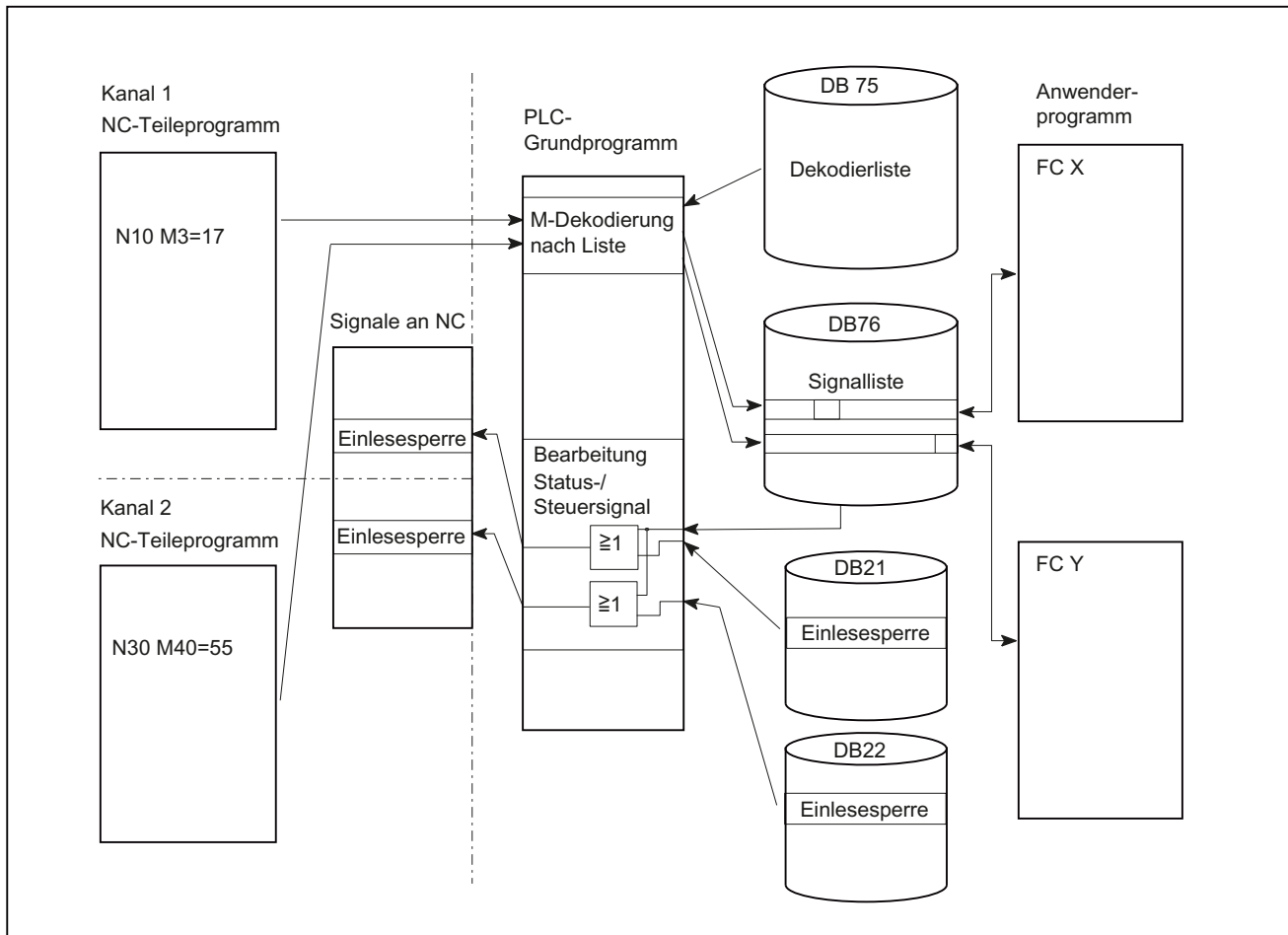


Bild 12-12 M-Dekodierung nach Liste

### Aktivierung der Funktion

Die Anzahl der auszuwertenden / dekodierenden Gruppen wird im Grundprogramm Parameter "ListMDecGrp" beim Aufruf des FB 1 im OB 100 angegeben (siehe auch "FB 1: RUN\_UP Grundprogramm, Anlaufteil (Seite 999)"). Die M-Dekodierung wird aktiviert, wenn dieser Wert zwischen 1 und 16 liegt. Vor Aktivierung der Funktion muss die Dekodierliste DB 75 in das AG übertragen und ein Neustart durchgeführt werden.

## Aufbau der Dekodierliste

Die Quelldatei für die Dekodierliste (MDECLIST.AWL) wird mit dem Grundprogramm ausgeliefert. Nach der Übersetzung der AWL-Quelle entsteht der DB 75.

Für jede zu dekodierende Gruppe von M-Funktionen muss ein Eintrag in der Dekodierliste DB 75 enthalten sein.

Höchstens 16 Gruppen können gebildet werden.

In jeder Gruppe stehen in der Liste der dekodierten Signale 16 Bit zur Verfügung.

Die Zuordnung von M-Funktion mit erweiterter Adresse und zu setzendem Bit in der Signalliste wird über erste und letzte M-Funktion in der Dekodierliste angegeben.

Dabei wird die Bitadresse entsprechend von der ersten M-Funktion ("MFirstAdr") bis zur letzten M-Funktion ("MLastAdr") von Bit 0 bis maximal Bit 15 für jede Gruppe gebildet.

Jeder Eintrag in der Dekodierliste besteht aus 3 Parametern, die jeweils einer Gruppe zugeordnet sind.

Zuordnung der Gruppen			
Gruppe	Erweiterte M-Adresse	erste M-Adresse in Gruppe	letzte M-Adresse in Gruppe
1	MSigGrp[1].MExtAdr	MSigGrp[1].MFirstAdr	MSigGrp[1].MLastAdr
2	MSigGrp[2].MExtAdr	MSigGrp[2].MFirstAdr	MSigGrp[2].MLastAdr
...	...	...	...
16	MSigGrp[16].MExtAdr	MSigGrp[16].MFirstAdr	MSigGrp[16].MLastAdr

Typ und Wertebereich für die Signale			
Signal	Typ	Wertebereich	Bemerkung
MExtAdr	Int	0 bis 99	Erweiterte M-Adresse
MFirstAdr	DInt	0 bis 99.999.999	erste M-Adresse in Gruppe
MLastAdr	Dint	0 bis 99.999.999	letzte M-Adresse in Gruppe

## Signalliste

Der Datenbaustein DB 76 wird bei Aktivierung der Funktion vom Grundprogramm eingerichtet.

Für ein nach Liste dekodiertes M-Signal wird im DB 76 in der entsprechenden Gruppe ein Bit gesetzt.

Gleichzeitig erfolgt in dem Kanal, in dem die M-Funktion ausgegeben wurde, die Beeinflussung der Einlesesperre.

**Beispiel**

Im folgenden Beispiel sollen 3 Gruppen an M-Befehlen dekodiert werden:

- M2 = 1 bis M2 = 5
- M3 = 12 bis M3 = 23
- M40 = 55

Aufbau der Dekodierliste im DB 75:

Beispielparameter				
Gruppe	Dekodierliste (DB 75)			Signalliste
	Erweiterte M-Adresse	Erste M-Adresse in Gruppe	Letzte M-Adresse in Gruppe	DB 76
1	2	1	5	DBX 0.0 ... DBX 0.4
2	3	12	23	DBX 2.0 ... DBX 3.3
3	40	55	55	DBX 4.0

```

DATA_BLOCK DB 75
TITLE =
VERSION : 0.0
    STRUCT
        MSigGrp : ARRAY [1 .. 16 ] OF STRUCT
            MExtAdr : INT ;
            MFirstAdr : DINT ;
            MLastAdr : DINT ;
        END_STRUCT ;
    END_STRUCT ;
BEGIN
    MSigGrp[1].MExtAdr := 2;
    MSigGrp[1].MFirstAdr := L#1;
    MSigGrp[1].MLastAdr := L#5;
    MSigGrp[2].MExtAdr := 3;
    MSigGrp[2].MFirstAdr := L#12;
    MSigGrp[2].MLastAdr := L#23;
    MSigGrp[3].MExtAdr := 40;
    MSigGrp[3].MFirstAdr := L#55;
    MSigGrp[3].MLastAdr := L#55;
END_DATA_BLOCK
    
```

## Aufbau des FB 1 im OB 100

(Anzahl M-Gruppen zur Dekodierung eintragen und damit Funktionalität aktivieren):

```
Call FB 1, DB 7(  
...  
ListMDecGrp := 3,           // M-Dekodierung von 3 Gruppen  
...  
);
```

Nachdem der Eintrag im OB 100 angefügt ist und der DB 75 (Dekodierliste) in das AG übertragen wurde, muss ein Neustart erfolgen. Das Grundprogramm richtet nun im Neuanlauf den DB 76 (Signalliste) ein.

Wird nun das NC-Programm gestartet und die erweiterte M-Funktion (z. B.  $M_{3=17}$ ) vom NCK abgearbeitet, dann wird diese M-Funktion dekodiert und im DB 76 wird das Bit 2.5 gesetzt (siehe Dekodierliste DB 75). Gleichzeitig setzt das Grundprogramm die Einleesesperre und die Abarbeitung des NC-Programms wird gestoppt (im entsprechenden NC-Kanal-DB erfolgt ebenfalls der Eintrag der "erweiterten Adresse M-Funktion" und die "M-Funktions-Nr.").

Die Einleesesperre im Kanal wird zurückgesetzt, wenn der Anwender alle von diesem Kanal ausgegebenen Bits in der Signalliste (DB 76) zurückgesetzt und damit quittiert hat.

## 12.10.10 PLC-Maschinendaten

### Allgemeines

Es besteht für den Anwender die Möglichkeit, PLC-spezifische Maschinendaten in der NCK zu speichern. Diese Maschinendaten können nach Hochlauf der PLC (OB 100) vom Anwender verarbeitet werden. Dadurch können z. B. Anwender-Optionen, Maschinenausbaustufen, Maschinenkonfiguration, etc. realisiert werden.

Die Nahtstelle zum Lesen dieser Daten liegt im DB 20. Der DB 20 wird aber nur bei Verwendung der Anwender-Maschinendaten, d. h. Summe aus den GP-Parametern "UDInt", "UDHex" und "UDReal" ist größer NULL, vom Grundprogramm im Hochlauf angelegt.

Die Größen der einzelnen Bereiche, und damit auch die Gesamtlänge des DB 20, werden durch folgende PLC-Maschinendaten eingestellt:

MD14504 \$MN\_MAXNUM\_USER\_DATA\_INT

MD14506 \$MN\_MAXNUM\_USER\_DATA\_HEX

MD14508 \$MN\_MAXNUM\_USER\_DATA\_FLOAT

Dem Anwender werden diese Einstellungen in den GP-Parametern "UDInt", "UDHex" und "UDReal" angegeben.

Die Datenablage im DB 20 erfolgt durch das GP bündig in der Reihenfolge:

1. Integer-MD
2. Hexa-Felder-MD
3. Real-MD

Die Integer- und Realwerte werden im S7-Format im DB 20 abgelegt.

Die Hexa-Daten werden in der Reihenfolge der Eingabe (Nutzung als Bitfelder) im DB 20 gespeichert.

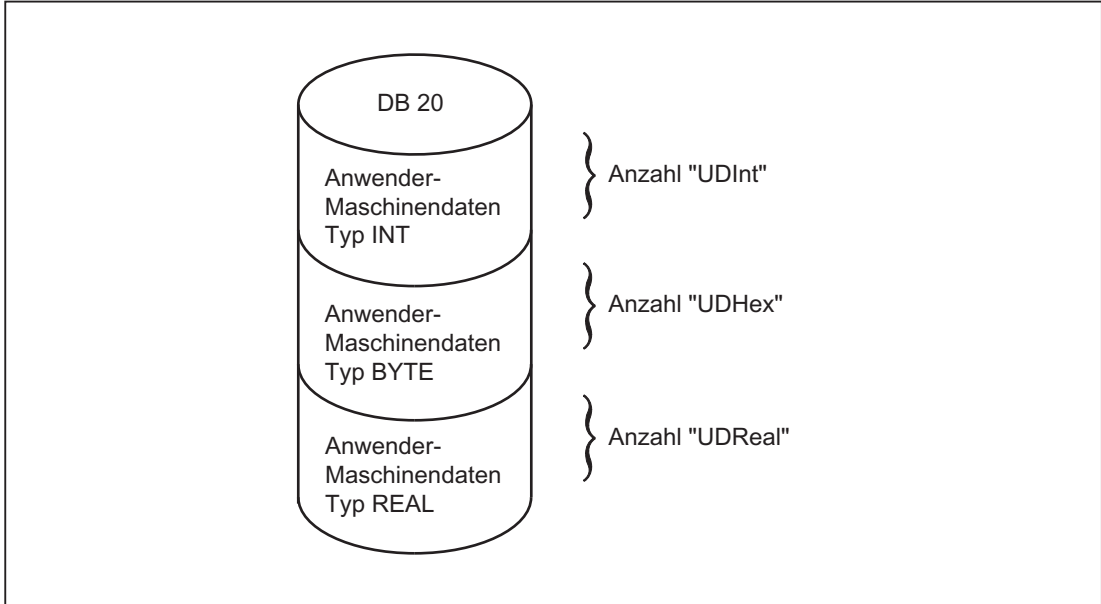


Bild 12-13 DB 20

**Hinweis**

Soll die Anzahl der genutzten PLC-Maschinendaten später erhöht werden, ist es notwendig, den DB 20 vorher zu löschen. Damit solche Erweiterungen keine Auswirkungen auf das bestehende Anwenderprogramm haben, sollten die Zugriffe auf die Daten im DB 20 möglichst symbolisch erfolgen, z. B. über eine Strukturbeschreibung im UDT.

Alarmer	
<b>400120</b>	<b>DB 20 im PLC löschen und Neustart</b>
Erläuterung	DB-Länge ist unterschiedlich zur geforderten DB-Länge
Reaktion	Alarmanzeige und PLC-Stop
Abhilfe	DB 20 löschen und anschließender RESET
Fortsetzung	nach Neustart

## Beispiel

Für das Projekt im Beispiel werden 4 INTEGER-Werte, 2 Hexa-Felder mit Bitinformationen und 1 Real-Wert benötigt.

Maschinendaten:

MD14510 \$MN_USER_DATA_INT[0]	123
MD14510 \$MN_USER_DATA_INT[1]	456
MD14510 \$MN_USER_DATA_INT[2]	789
MD14510 \$MN_USER_DATA_INT[3]	1011
...	
MD14512 \$MN_USER_DATA_HEX[0]	12
MD14512 \$MN_USER_DATA_HEX[1]	AC
...	
MD14514 \$MN_USER_DATA_FLOAT[0]	123.456

GP-Parameter (OB 100):

```
CALL FB 1, DB 7(  
    MCPNum := 1,  
    MCP1In := P#E0.0,  
    MCP1Out := P#A0.0,  
    MCP1StatSend := P#A8.0,  
    MCP1StatRec := P#A12.0,  
    MCP1BusAdr := 6,  
    MCP1Timeout := S5T#700MS,  
    MCP1Cycl := S5T#200MS,  
    NCCyclTimeout := S5T#200MS,  
    NCRunupTimeout := S5T#50S;
```

GP-Parameter (zur Laufzeit abfragen):

```
1 gp_par.UDInt; //4,  
1 gp_par.UDHex; //2,  
1 gp_par.UDReal; //1 )
```

Im Hochlauf der PLC wurde der DB 20 mit einer Länge von 28 Byte erstellt:

DB 20	
Adresse	Daten
0.0	123
2.0	456
4.0	789
6.0	1011
8.0	b#16#12
9.0	b#16#AC
10.0	1.234560e+02

Die Struktur der genutzten Maschinendaten wird in einem UDT angegeben:

```
TYPE UDT 20
  STRUCT
    UDInt :    ARRAY [0 .. 3] OF INT;
    UDHex0 :  ARRAY [0 .. 15] OF BOOL;
    UDReal  :  ARRAY [0 .. 0] OF REAL;      //Beschreibung als Feld, für
                                           //spätere Erweiterungen
  END_STRUCT;
END_TYPE
```

---

### Hinweis

ARRAY OF BOOL werden immer an geradzahligen Adressen ausgerichtet. Deshalb ist in der Definition des UDT generell ein Array-Bereich von 0 bis 15 zu wählen bzw. alle Booleschen Variablen sind einzeln aufzuführen.

---

Obwohl im Beispiel zunächst nur ein REAL-Wert genutzt wird, wurde für die Variable ein Feld (mit einem Element) angelegt. Dies sichert eine einfache Erweiterung zu einem späteren Zeitpunkt, ohne dass sich die symbolische Adresse ändert.



## Symbolische Zugriffe

Für den symbolischen Zugriff erfolgt ein Eintrag in der Symbolliste:

Symbol	Operand	Datentyp
UData	DB 20	UDT 20

Zugriffe im Anwenderprogramm (nur symbolische Lesezugriffe dargestellt):

```
...  
L      "UData".UDInt [0] ;  
L      "UData".UDInt [1] ;  
L      "UData".UDInt [2] ;  
L      "UData".UDInt [3] ;  
  
U      "UData".UDHex0 [0] ;  
U      "UData".UDHex0 [1] ;  
U      "UData".UDHex0 [2] ;  
U      "UData".UDHex0 [3] ;  
U      "UData".UDHex0 [4] ;  
U      "UData".UDHex0 [5] ;  
U      "UData".UDHex0 [6] ;  
U      "UData".UDHex0 [7] ;  
...   ...  
U      "UData".UDHex0 [15] ;  
  
L      "UData".UDReal [0] ;  
...
```

## 12.10.11 Projektierung von Maschinensteuertafel, Bedienhandgerät, Direktasten

### Allgemeines

Es ist ein gleichzeitiger Betrieb von maximal 2 Maschinensteuertafeln und einem Bedienhandgerät möglich. Für die Maschinensteuertafel (MCP) und Bedienhandgerät (BHG) gibt es verschiedene Anschlussmöglichkeiten (Ethernet, PROFIBUS). Hierbei ist es möglich, dass 2 MCP an unterschiedlichen Bussystemen (nur bei Ethernet und PROFIBUS ist Mischbetrieb zulässig) angeschlossen werden können. Erreicht werden kann dies durch den FB 1-Parameter "MCPBusType". Bei diesem Parameter ist die rechte Dekade für die 1. MCP und die linke Dekade für die 2. MCP zuständig.

Die Parametrierung der Komponenten erfolgt grundsätzlich über den Aufruf des Grundprogrammbausteins FB 1 im OB 100. Der FB 1 speichert seine Parameter im zugehörigen Instanz-DB (DB 7, symbolisch "gp\_par"). Hierzu existieren für jede Maschinensteuertafel und das Bedienhandgerät eigene Parametersätze. In diesen Parametersätzen ist die Ein-, Ausgangsadresse vom Anwender zu definieren. Diese Ein- und Ausgangsadressen werden auch im FC 19, FC 24, FC 25, FC 26 und FC 13 verwendet. Weiterhin sind noch Adressen für Statusinformationen, PROFIBUS bzw. Ethernet zu definieren. Die Zeiteinstellungen für Timeout und zyklische Zwangsnachtriggerung sollen auf dem voreingestellten Wert bleiben. Weitere Informationen zu den MCP, BHG-Komponenten sind im Handbuch Bedienkomponenten zu finden.

### Aktivierung

Die jeweilige Komponente wird entweder über die Anzahl der Maschinensteuertafeln (Parameter MCPNum) oder beim Bedienhandgerät über den Parameter BHG aktiviert. Die Festlegung über den Anschluss der MCP, BHG wird über die FB 1-Parameter "MCPMPI", "MCPBusType" bzw. "BHG", "BHGMPPI" vorgenommen.

### Bedienhandgerät (HT2)

Beim Bedienhandgerät wird die Adressierung über einen Parameter des GD-Parametersatzes vorgenommen. Dieses war aus Kompatibilitätsgründen der Parameternamen erforderlich.

### Projektierung

Grundsätzlich existieren verschiedene Kommunikationsmechanismen, bedingt durch den Bus-Anschluss von MCP und BHG, zur Übertragung der Daten zwischen MCP/BHG und PLC. In einem Fall (Ethernet) werden die Daten durch das "CP 840D sl" transportiert.

Die Parametrierung erfolgt dabei vollständig über die MCP/BHG Parameter im FB 1.

Im anderen Fall erfolgt die Übertragung durch das PLC-Betriebssystem durch die PROFIBUS-Konfiguration.

Die Parametrierung erfolgt über STEP 7 in HW-Config. Für den Zugriff des Grundprogramms auf diese Daten und Ausfallüberwachung von MCP/BHG müssen die eingestellten Adressen in FB 1-Parametern dem Grundprogramm bekannt gemacht werden.

Im Weiteren folgt eine Übersicht über die verschiedenen Kopplungsmöglichkeiten. Hierbei kann auch ein Mischbetrieb projektiert werden.

Wird ein Fehler aufgrund einer Zeitüberwachung erkannt, erfolgt ein Eintrag im Alarmpuffer der PLC-CPU (Alarm 400260 bis 400262). In diesem Fall werden die Eingangssignale von der MCP bzw. vom Bedienhandgerät (MCP1In/MCP2In bzw. BHGIn) mit 0 initialisiert. Sollte eine Neusynchronisation zwischen PLC und MCP/BHG möglich sein, wird die Kommunikation automatisch wieder aufgenommen und die Fehlermeldung vom GP gelöscht.

---

#### Hinweis

Die Abkürzung "(n.r.)" in den unten angegebenen Tabellen bedeutet "nicht relevant".

---

### Ethernet-Ankopplung

Kommunikation erfolgt ohne weitere Projektierung direkt vom PLC-GP aus über die CP 840D sl. Die Parametrierung erfolgt über die unten genannten Parameter des FB 1.

In "MCP1 BusAdr", "MCP2 BusAdr" bzw. "BHGRecGDNo" (entspricht der Busadresse des Teilnehmers) ist der numerische Teil des logischen Namen der Komponente einzutragen. Der logische Name wird über die Schalter an der MCP bzw. Anschlussbox eingestellt.

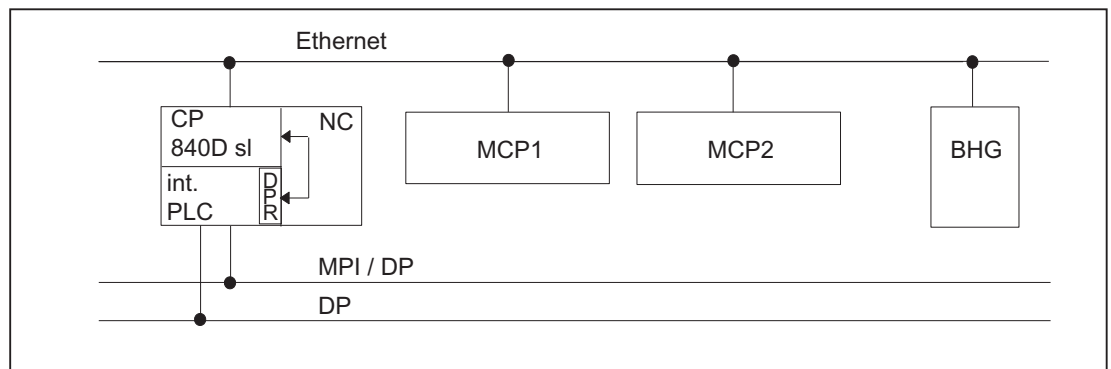


Bild 12-14 Ethernet-Ankopplung

Relevante Parameter (FB 1)		
MCP		BHG
MCPNum=1 oder 2 (Anzahl MCP)		BHG = 5 (über CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend	MCP2StatSend	BHGStatSend
MCP1StatRec (n.r.)	MCP2StatRec (n.r.)	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen (n.r.)
MCP1Timeout (n.r.)	MCP2Timeout (n.r.)	BHGOutLen (n.r.)
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout (n.r.)
<b>MCPMPI = FALSE</b>		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCP1 NotSend	MCP2 NotSend	BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPBusType = b#16#55 (über CP 840D sl)		BHGSendGDNo (n.r.)
		BHGSendGBZNo (n.r.)
MCPsDB210 = FALSE		BHGSendObjNo (n.r.)
MCPCopyDB77 = FALSE		<b>BHGMPI = FALSE</b>
		BHGStop
		BHG NotSend

Für die Zeitüberwachungen wird ein Fehlereintrag im Alarmpuffer der PLC generiert. Daraus entstehen am HMI die Fehlermeldungen:

- 400260: MSST 1 ausgefallen  
oder
- 400261: MSST 2 ausgefallen
- 400262: BHG ausgefallen

Ein Ausfall von MCP oder BHG wird auch gleich nach dem Neustart erkannt, auch dann, wenn noch keine Daten zwischen MCP/BHG und PLC ausgetauscht wurden. Die Überwachungsfunktion wird aktiviert, sobald alle Komponenten nach dem Hochlauf "Ready" melden.

### Beispiel: OP mit Direkttasten

Die Direkttasten der OPs am Ethernet-Bus sollen in die PLC übertragen werden. Bisher sind die Direkttasten über den PROFIBUS bzw. über eine spezielle Kabelverbindung zwischen OP und MCP an die PLC geführt worden.

Für die Ankopplung der Direkttasten über Ethernet, dies betrifft z. B. das "OP08T", gibt es eine Parametrierung im Grundprogramm für die Aktivierung des Datentransports. Die zugehörigen Parameter liegen im Instanz-DB des FB 1 (OpKeyNum bis OpKeyBusType, siehe Datentabelle). Die Parameter werden im Anlauf-OB 100 durch das Beschalten der Parameter beim FB 1-Aufruf durch den Anwender versorgt. Die Busadresse und Op1/2KeyStop können auch im zyklischen Betrieb über Beschreiben des FB 1-Instanz-DB DB 7 verändert werden.

Der Transport der Nutzdaten der Direkttasten verläuft gleichartig wie bei der Ethernet-MCP. Der Datentransport kann über das Beschreiben des DB 7-Parameters "Op1/2KeyStop" auch gestoppt werden und wieder neu gestartet werden. Während der Stopp-Phase kann auch die Adresse des Direkttasten-Moduls (TCU-Index bzw. MCP-Adresse) geändert werden.

Nach Rücksetzen des Stopp-Signals wird eine Verbindung zu der neuen Adresse aufgebaut.

Der Status der jeweiligen Direkttasten-Schnittstelle ist ablesbar im Nahtstellensignal:

DB10.DBX104.3 (OP1Key bereit)

bzw.

DB10.DBX104.4 (OP2Key bereit)

### Adresse Direkttasten

Für den Parameter Op1/2KeyBusAdr ist im Regelfall der TCU-Index zu verwenden. Dieses trifft für die OPs wie z. B. OP08T, OP12T zu, die für die Direkttasten **keine** spezielle Kabel-Verbindung zu einer Ethernet-MCP haben.

Verfügen OPs mit Direkttasten über eine spezielle Kabelverbindung und sind diese mit einer Ethernet-MCP verbunden, so ist für den Parameter Op1/2KeyBusAdr die Adresse der MCP (DIP-Schalterstellung der MCP) zu verwenden. Über die Direkttasten-Schnittstelle wird nur der Datenstrom der Direkttasten (2 Bytes) übertragen.

### Alarm Direkttasten

Für die Zeitüberwachungen wird ein Fehlereintrag im Alarmpuffer der PLC generiert. Daraus entstehen am HMI die Fehlermeldungen:

- 400274: Direkttasten 1 ausgefallen  
 oder
- 400275: Direkttasten 2 ausgefallen

### Bedieneinheitenumschaltung bei Direkttasten

Der Anwender beschaltet Op1/2KeyBusAdr mit 0xFF und Stopp = TRUE im Anlaufbaustein OB 100. Über den M-zu-N-Baustein FB 9 wird die Direkttasten-Adresse der M-zu-N-Schnittstelle an den Parameter "Op1KeyBusAdr" gelegt.

Relevante Parameter (FB 1)		
Direkttasten		z. B. Direkttasten OP08T
OpKeyNum = 1 oder 2 (Anzahl OPs mit Direkttasten)		
Op1KeyIn	Op2KeyIn	
Op1KeyOut	Op2KeyOut	
OpKey1BusAdr	Op2KeyBusAdr	Adresse: TCU-Index
Op1KeyStop	Op2KeyStop	
Op1KeyNotSend	Op2KeyNotSend	
OpKeyBusType = b#16#55 (über CP 840D sl)		

## MCP-Identifizierung

Über die Identify-Schnittstelle im DB 7 ist es möglich, mit den relevanten Parametern am Ein-/Ausgang den Typ einer Ethernet-Komponente (MCP, HT2, HT8 oder Direkttasten) im zyklischen Betrieb zu erfragen:

- Relevante Parameter am Eingang:  
"IdentMcpBusAdr", "IdentMcpProfilNo", "IdentMcpBusType", "IdentMcpStrobe"
- Relevante Parameter am Ausgang:  
"IdentMcpType", "IdentMcpLengthIn", "IdentMcpLengthOut"

Hierbei wird die DIP-Device-Adresse bzw. TCU-Index am Parameter "IdentMcpBusAdr" vom Anwenderprogramm in Verbindung mit dem Setzen des Strobe-Signals angeschaltet.

Der Eingangsparameter "IdentMcpProfilNo" ist im Regelfall auf den Wert "0" zu setzen. Nur bei der Identifikation von Direkttasten ist dieser Parameter auf den Wert "1" zu setzen. Der Parameter "IdentMcpBusType" hat für ein Anwenderprogramm derzeit keine Bedeutung und ist auf dem Standardwert zu lassen.

Nach dem Rücksetzen des Strobe-Signals durch das Grundprogramm stehen gültige Ausgangsinformationen für den Anwender zur Verfügung. Das Rücksetzen des Strobe-Signals vom Grundprogramm kann mehrere PLC-Zyklen (bis zu 2 Sekunden) dauern.

Die Ausgangsparameter sollen dem Anwender anzeigen, wie groß die Datenbereiche für das angesprochene Gerät sind. Weiterhin kann hierüber z. B. festgestellt werden, ob an einer Anschlussbox ein HT2 oder ein HT8 oder kein Gerät angeschlossen ist. Mit der Information kann dann der MCP-Kanal oder BHG-Kanal aktiviert werden. Im zyklischen Betrieb können die Parameter vom Anwenderprogramm symbolisch beschrieben und gelesen werden über den Symbolnamen des DB 7 (gp\_par).

Relevante Parameter (FB 1)		
MCP-Geräte Identifizierung		Eingangsparameter z. B. OP08T
Eingang	Ausgang	Werte bei Direkttasten
IdentMcpBusAdr	IdentMcpType	IdentMcpBusAdr = TCU-Index
IdentMcpBusProfilNo	IdentMcpLengthIn	IdentMcpBusProfilNo = Wert 1
IdentMcpBusType	IdentMcpLengthOut	IdentMcpBusType = Defaultwert
IdentMcpStrobe		
IdentMcpBusProfilNo		Wert
MCP, BHG, HT8, HT2		B#16#0
Direktstasten wie z.B. OP08T, OP12T		B#16#1
IdentMcpType (Mcp-Type)		
kein Gerät verbunden		0
MCP 483C IE (Compact)		B#16#80
MCP 483C IE		B#16#81
MCP 310		B#16#82
MCP OEM		B#16#83
MCP DMG		B#16#84
HT8		B#16#85
TCU_DT (Direktstasten)		B#16#86
MCP_MPP		B#16#87
HT2		B#16#88
OP08T (Direktstasten)		B#16#89

## PROFIBUS-Ankopplung am DP-Anschluss

Bei der PROFIBUS-Ankopplung der MCP muss diese Komponente in der Hardware-Projektierung von STEP 7 berücksichtigt werden. Die MCP wird auf dem Standard-DP-Bus der PLC angekoppelt (**nicht am MPI/DP**). Die Adressen sind in den Eingangs- und Ausgangs-Abbildbereich zu legen. Diese Anfangs-Adressen sind auch in den Pointer-Parametern des FB 1 zu hinterlegen. Die weitere Parametrierung erfolgt über die unten aufgeführten Parameter des FB 1.

Eine PROFIBUS-Variante des BHG existiert nicht. Deshalb ist in diesem Bild ein Ethernet-Anschluss für das BHG dargestellt. In den Parameter "MCP1BusAdr" und "MCP2BusAdr" muss die PROFIBUS-Slave-Adresse hinterlegt werden. "MCPxStatRec" ist der Pointer auf die projektierte Diagnoseadresse (z. B. P#A8190.0) einzutragen.

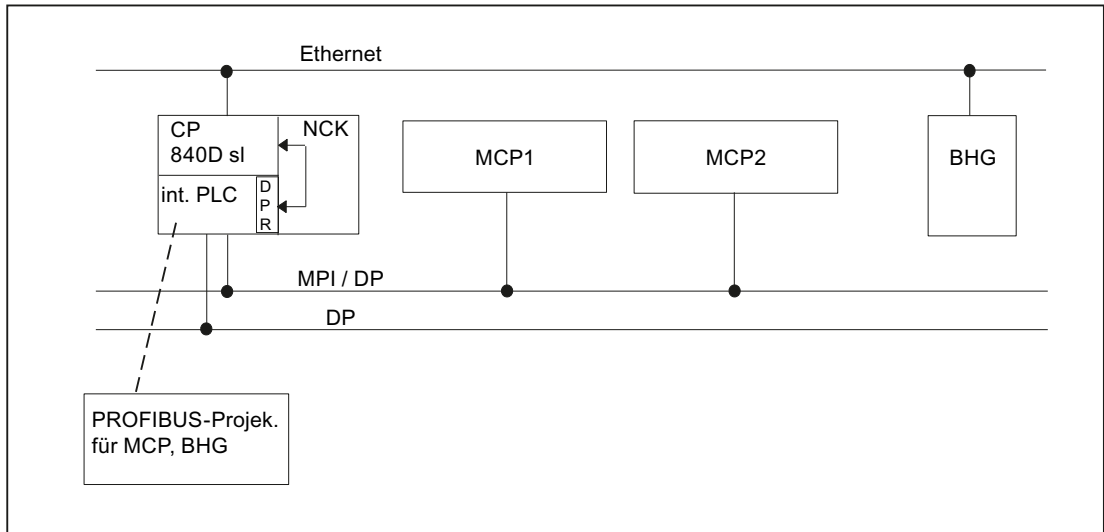


Bild 12-15 PROFIBUS-Ankopplung

Relevante Parameter (FB 1)		
MCP		BHG
MCPNum = 1 oder 2 (Anzahl MCP)		BHG = 5 (über CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOOutLen
MCP1Cycl (n.r.)	MCP2Cycl	BHGTimeout (n.r.)
<b>MCPMPI = FALSE</b>		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
<b>MCPBusType = b#16#33</b>		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPSDB210 = FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)
		<b>BHGMPI = FALSE</b>
		BHGStop

Ein Ausfall der MCP führt die PLC normalerweise in den Stopp-Zustand. Falls dieses nicht gewünscht ist, dann kann durch OB 82, OB 86 ein Stopp vermieden werden. Das Grundprogramm bringt standardmäßig den OB 82 und OB 86 Aufruf mit. In diesen OBs wird der FC 5 aufgerufen. Dieser FC 5 prüft, ob der ausgefallene Slave eine MCP ist. Falls dies zutrifft wird kein Stopp der PLC ausgelöst. Durch "MCPxStop" = TRUE wird die MCP vom Grundprogramm über SFC 12 als Slave abgeschaltet. Falls die PLC bei Ausfall oder Störung der MCP nicht in Stopp-Zustand gebracht wird, wird eine Alarmmeldung durch das Grundprogramm erzeugt. Bei Stationswiederkehr wird der Alarm gelöscht.



## PROFIBUS-Ankopplung am MPI/DP-Anschluss

Bei der PROFIBUS-Ankopplung der MCP muss diese Komponente in der Hardwareprojektierung von STEP 7 berücksichtigt werden. Die MCP wird auf dem MPI/DP-Bus der PLC angekoppelt.

Die Adressen sind in den Eingangs- und Ausgangs-Abbildbereich zu legen. Diese Anfangsadressen sind auch in den Pointer-Parametern des FB 1 zu hinterlegen. Die weitere Parametrierung erfolgt über die unten aufgeführten Parameter des FB 1. Eine PROFIBUS-Variante des BHG existiert nicht. Deshalb ist in diesem Bild ein Ethernet-Anschluss für das BHG gezeichnet. Im MCP1BusAdr und MCP2BusAdr muss die PROFIBUS-Slave Adresse hinterlegt werden. In MCPxStatRec ist der Pointer auf die projektierte Diagnoseadresse einzutragen (z. B. P#A8190.0).

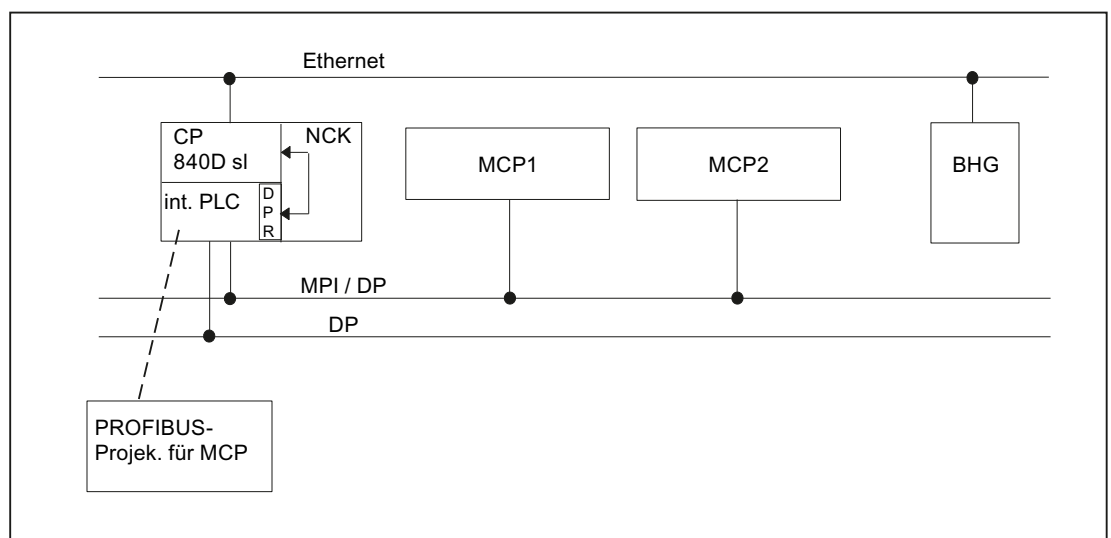


Bild 12-16 PROFIBUS-Ankopplung am MPI/DP-Anschluss

Relevante Parameter (FB1)		
MCP		BHG
MCPNum = 1 oder 2 (Anzahl MCP)		BHG = 5 (über CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOOutLen
MCP1Cycl (n.r.)	MCP2Cycl	BHGTimeout (n.r.)
<b>MCPMPI = FALSE</b>		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
<b>MCPBusType = b#16#44</b>		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPsDB210 = FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)
		<b>BHGMPI = FALSE</b>
		BHGStop

Ein Ausfall der MCP führt die PLC normalerweise in den Stopp-Zustand. Falls dieses nicht gewünscht ist, kann durch OB 82, OB 86 ein PLC-Stopp vermieden werden. Das Grundprogramm bringt standardmäßig den OB 82 und OB 86 Aufruf mit. In diesen OBs wird der FC 5 aufgerufen. Dieser FC 5 prüft, ob der ausgefallene Slave eine MCP ist. Falls dieses zutrifft, wird kein Stopp der PLC ausgelöst. Durch MCPxStop:=True wird die MCP vom Grundprogramm über SFC 12 als Slave abgeschaltet. Falls die PLC bei Ausfall oder Störung der MCP nicht in den Stopp-Zustand gebracht wird, wird eine Alarmmeldung durch das Grundprogramm erzeugt. Bei Stations-Wiederkehr wird der Alarm gelöscht.

### PROFINET-Ankopplung

Bei der PROFINET-Ankopplung der MCP muss diese Komponente in der Hardware-Projektierung von STEP7 parametrisiert werden. Die MCP wird gekoppelt mit dem PROFINET-Modul der CPU.

Bei der Parametrierung der MCP in HW-Konfig sind die Adressen in den Eingangs- und Ausgangs-Abbildbereich zu legen. Diese Anfangsadressen sind auch in den Pointer-Parametern (MCPxIn und MCPxOut) des FB 1 zu hinterlegen. Denn der Signaltransfer zwischen MCP und Grundprogramm erfolgt über diese Parameter. Über den Parameter MCPxIn wird die MCP auch überwacht. Deshalb ist der Parameter MCPxBusAdr bei dieser MCP-Variante nicht relevant.

In MCPxStatRec ist der Pointer auf die projektierte Diagnoseadresse einzutragen (z. B. P#A8190.0).

Die PROFINET-MCP hat einen eigenen Typ, der beim Parameter MCPBusType anzutragen ist.

Die weitere Parametrierung erfolgt über die unten aufgeführten Parameter des FB 1. Eine Profibus-Variante des BHG existiert nicht. Im Bild ist ein Ethernet-Anschluss für das BHG gezeichnet.

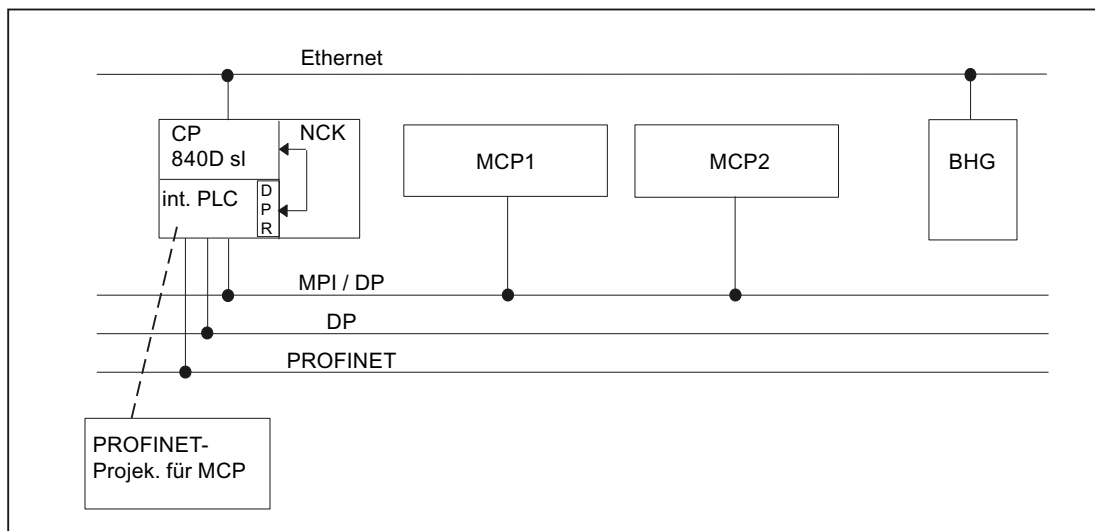


Bild 12-17 PROFINET-Ankopplung

Relevante Parameter (FB1)		
MCP		BHG
MCPNum = 1 oder 2 (Anzahl MCP)		BHG = 5 (über CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec (n.r.)	MCP2StatRec (n.r.)	BHGStatRec
MCP1BusAdr (n.r.)	MCP2BusAdr (n.r.)	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl	MCP2Cycl	BHGTimeout (n.r.)
<b>MCPMPI = FALSE</b>		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
<b>MCPBusType = b#16#36</b>		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPsDB210 = FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)
		<b>BHGMPI = FALSE</b>
		BHGStop

Ein Ausfall der MCP führt die PLC normalerweise in den Stopp-Zustand. Falls dieses nicht gewünscht ist, dann kann durch OB 82, OB 86 ein PLC-Stop vermieden werden. Das Grundprogramm bringt standardmäßig den OB 82- und OB 86-Aufruf mit. In diesen OBs wird der FC 5 aufgerufen. Dieser FC 5 prüft, ob der ausgefallene Slave eine MCP ist. Falls dies zutrifft wird kein PLC-Stop ausgelöst. Für die Überwachung bei einem MCP-Ausfall ist die Eingangsadresse am Parameter MCPxIn von Bedeutung.

Durch MCPxStop := True wird die MCP vom Grundprogramm über SFC 12 als Slave abgeschaltet. Falls die PLC bei Ausfall oder Störung der MCP nicht in Stopp-Zustand gebracht wird, wird eine Alarmmeldung durch das Grundprogramm erzeugt. Bei Stations-Wiederkehr wird der Alarm gelöscht.

### 12.10.12 Umschaltung von Maschinensteuertafel, Bedienhandgerät

Eine Umschaltung bzw. Abschaltung von Maschinensteuertafel (MCP) oder Bedienhandgerät (BHG) wird vom Standard nur bei den Ethernet-Varianten unterstützt. Bei den PROFIBUS-Varianten ist diese Funktionalität nur in einem begrenzten Umfang mit zusätzlichem Anwenderaufwand möglich. Z. B. bei der PROFIBUS-Variante der MCP kann der für MCP 1, MCP 2 oder BHG spezifizierte Datenbereich des DB 77 für die Pointer der MCP am FB 1 verwendet werden. Die Slave-Busadresse der MCP muss unter MCPxBusAdr korrekt gesetzt sein, da hierüber die Überwachung realisiert ist. Durch eine Kopieroutine des Anwenderprogramms müssen die Signale der aktiven MCP von dem in HW-Config projektierten E/A-Bereich in den DB 77 kopiert werden. Somit können mehrere MCPs am PROFIBUS signalmäßig umgeschaltet werden. Für die Umschaltphase von einer zur anderen MCP ist der Parameter MCPxStop auf TRUE zu setzen.

Hier soll nun ein Weg dargestellt werden mit den Ethernet-Varianten von MCP und BHG.

#### Steuersignale

Mit Hilfe der Parameter MCP1Stop, MCP2Stop und BHGStop ist ein Anhalten der Kommunikation zu den einzelnen Komponenten möglich (Wertzuweisung = 1). Diese Funktion steht nur bei den Ethernet-Varianten zur Verfügung. Dieses Stoppen bzw. auch Aktivieren der Kommunikation ist im laufenden Zyklus möglich. Allerdings darf die Wertänderung nicht über einen erneuten Aufruf des FB 1 erfolgen, sondern durch die symbolische Schreibweise der Parameter.

Beispiel für Stoppen der Übertragung der 1. Maschinensteuertafel:

```
SET;  
S gp_par.MCP1Stop;
```

Bei gesetzten Parametern MCP1Stop, MCP2Stop, BHGStop erfolgt auch eine Unterdrückung bzw. Löschung der Alarme 400260 bis 400262.

## Umschaltung Bus-Adresse

Eine bestehende Verbindung zu einer Maschinensteuertafel (MCP) oder Bedienhandgerät (BHG) kann abgebaut werden. Eine andere MCP/BHG-Komponente, die sich bereits am Bus befindet (andere Adresse), kann anschließend aktiviert werden. Für diese Umschaltung ist folgende Vorgehensweise erforderlich:

1. Anhalten der Kommunikation der abzukoppelnden Komponente über den Parameter MCP1Stop bzw. MCP2Stop bzw. BHGStop = 1.
2. Nach Rückmeldung im DB 10 Byte 104 (relevante Bits 0, 1, 2 auf Zustand "0"), erfolgt die Änderung der Busadresse (bei MCP sind dies die FB 1-Parameter "MCP1BusAdr" bzw. "MCP2BusAdr". Bei BHG wird die Busadresse bei der Ethernet-Variante am FB 1-Parameter "BHGRecGDNo" eingestellt) dieser Einheit auf die neue Komponente.
3. In diesem PLC-Zyklus kann nun die Kommunikation der neuen Komponente wieder aktiviert werden über den Parameter MCP1Stop bzw. MCP2Stop bzw. BHGStop = 0.
4. Die Kommunikation mit der neuen Komponente läuft, wenn Rückmeldung im DB 10 Byte 104 vorhanden ist (relevante Bits 0, 1, 2 auf Zustand "1").

## Abschalten des Blinkens der MCP IE

### Allgemeines Verhalten und Randbedingungen

Nach POWER ON blinken die MCP's immer und zeigen so die Beendigung des Hochlaufs und das Warten auf einen Verbindungsaufbau an (Defaulteinstellung des MCP).

Bei einem Verbindungsaufbau zum MCP (z. B. PLC verbindet sich mit dem entsprechenden MCP) kann für den anschließenden kommunikationslosen Zustand das Verhalten der LEDs eingestellt werden.

Eine remanente Speicherung dieses Verhaltens auf dem MCP ist derzeit nicht möglich!

### Einstellung über die PLC:

Ab der MCP-Firmware V02.02.04 kann das Blinken im Offline-Modus unterdrückt werden.

Beim Offline-Modus findet keine Kommunikation statt (z. B. auch bei Ausfall der MCP-Verbindung).

Um das Blinken zu deaktivieren, ist es erforderlich, dass die Bits im FB1-Parameter MCPxStatSend (Status-Doppelwort für Senden) vor Kommunikationsstart mit der MCP folgendermaßen gesetzt sind:

Bit 30 = FALSE und Bit 31 = TRUE

Vor Kommunikationsstart zum MCP bedeutet:

- Im OB100 oder
- Im OB1 vor MCP-Stopp/-Starten

(DB7-Parameterwechsel MCPxStop = TRUE → MCPxStop = FALSE)

Eine Rückmeldung des Status gibt es zurzeit nicht.

**Beispiel:**

Auszug aus OB100: (am Beispiel fuer MCP1)

```
CALL "RUN_UP" , "gp_par"
```

```
...
```

```
MCP1StatSend      := P#A 8.0
```

```
...
```

```
//MCP-Blinken deaktivieren
```

```
SET
```

```
R      A 11.6
```

```
S      A 11.7
```

```
...
```

## **12.11 SPL für Safety Integrated**

Die SPL ist keine Funktion des Grundprogramms, sondern des Anwenders. Das Grundprogramm stellt einen Datenbaustein DB 18 für Safety-SPL-Signale zur Verfügung und sichert über einen Datenvergleich die Information gegenüber dem SPL-Programm in der NCK ab.

**Literatur:**

Funktionshandbuch Safety Integrated

## 12.12 Belegungsübersicht

### 12.12.1 Belegung: NCK/PLC-Nahtstelle

Die Belegung der NC/PLC-Nahtstelle für SINUMERK 840D sl ist ausführlich beschrieben in:

**Literatur:**

Listen sl (Buch2)

### 12.12.2 Belegung: FB/FC

Nummer	Bedeutung
FB 15	Basis-Grundprogramm
FB 1, FC 2, FC 3, FC 5	Basis-Grundprogramm
FC 0 ... 29	reserviert für Siemens
FB 0 ... 29	reserviert für Siemens
FC 30 ... 999 <sup>1)</sup>	frei für Anwender
FB 30 ... 999 <sup>1)</sup>	frei für Anwender
FC 1000 ... 1023	reserviert für Siemens
FB 1000 ... 1023	reserviert für Siemens
FC 1024 ... Obergrenze	frei für Anwender
FB 1024 ... Obergrenze	frei für Anwender

<sup>1)</sup> Die tatsächliche Obergrenze der Baustein-Nummer (FB/FC) ist abhängig von der PLC-CPU, die in der gewählten NCU enthalten ist.

---

**Hinweis**

Belegung der FC, FB siehe " Speicherbedarf des PLC-Grundprogramms (Seite 979)".

---

### 12.12.3 Belegung: DB

---

**Hinweis**

Es werden nur so viele Datenbausteine eingerichtet, wie aufgrund der Projektierung in den NC-Maschinendaten erforderlich sind.

---



Übersicht der Datenbausteine			
DB-Nr.	Bezeichnung	Name	Paket
1		reserviert für Siemens	GP
2 ... 5	PLC-MELD	PLC-Meldungen	GP
6 ... 8		Grundprogramm	
9	NC-COMPILE	Nahtstelle für NC-Compile-Zyklen	GP
10	NC-NAHTSTELLE	Zentrale NCK-Nahtstelle	GP
11	BAG 1	Nahtstelle BAG	GP
12		Nahtstelle Rechnerkopplung und Transportsystem	
13 ... 14		reserviert für Grundprogramm	
15		Grundprogramm	
16		PI-Dienst Definition	
17		Versionskennung	
18		reserviert für Grundprogramm	
19		Nahtstelle HMI	
20		PLC-Maschinendaten	
21 ... 30	KANAL 1 ... n	Nahtstelle NC-Kanäle	GP
31 ... 61	ACHSE 1 ... m	Nahtstellen für Achsen/Spindeln bzw. frei für Anwender	GP
62 ... 70		frei für Anwender	
71 ... 74		Werkzeugverwaltung	GP
75 ... 76		M-Gruppen-Dekodierung	
77		DB für MCP Signale	
78 ... 80		reserviert für Siemens	
81 ... 999 <sup>1)</sup>		siehe unten: ShopMill, ManualTurn	
1000 ... 1099		reserviert für Siemens	
1100 ... Obergrenze		frei für Anwender	

<sup>1)</sup> Die tatsächliche Obergrenze der Baustein-Nummer (DB) ist abhängig von der PLC-CPU, die in der gewählten NCU enthalten ist. Datenbausteine von nicht aktivierten Kanälen, Achsen/Spindeln, Werkzeugverwaltung sind frei für den Anwender.

### Hinweis

Datenbausteine von nicht aktivierten Kanälen, Achsen/Spindeln, Werkzeugverwaltung können frei verwendet werden.

## 12.12.4 Belegung: Timer

Timer-Nr.	Bedeutung
T 0 ... T 512 <sup>1)</sup>	Anwenderbereich

- <sup>1)</sup> Die tatsächliche Obergrenze der Timer-Nummer (DB) ist abhängig von der PLC-CPU, die in der gewählten NCU enthalten ist.

## 12.13 Speicherbedarf des PLC-Grundprogramms

Das Grundprogramm besteht aus Basisfunktionen und optionalen Funktionen. Zu den **Basisfunktionen** gehört der zyklische Signalaustausch NC ↔ PLC. Zu den **Optionen** gehören z. B. die FCs, die bei Bedarf eingesetzt werden können.

In der folgenden Tabelle ist der Speicherbedarf für die Basisfunktionen und die Optionen aufgelistet. Die Angaben stellen Richtwerte dar, sie sind vom jeweils aktuellen Softwarestand abhängig.

Speicherbedarf der Bausteine bei SINUMERIK 840D sl			
Baustein Typ, Nr.	Funktion	Bemerkung	Baustein-Größe (Byte) Arbeits- speicher
<b>Grundprogramm-Basisfunktionen</b>			
FB 1, FB 15		muss geladen werden / auf CompactFlash Card	52182
FC 2, 3, 5, 12		müssen geladen werden	470
DB 4, 5, 7, 8		müssen geladen werden	1006
DB 2, 3, 17		werden vom GP erzeugt	632
OB 1, 40, 100, 82, 86		müssen geladen werden	398
		Summe	55698
<b>Nahtstelle PLC / NCK, PLC / HMI</b>			
DB 10	Signale PLC / NCK	muss geladen werden	262
DB 11	Signale PLC / BAG	wird vom GP erzeugt	56
DB 19	Signale PLC / HMI	wird vom GP erzeugt	434
DB 21 bis 30	Signale PLC / Kanal	werden vom GP abhängig von NC-MD erzeugt: Je DB	416
DB 31 bis 61	Signale PLC / Achse, Spindel	werden vom GP abhängig von NC-MD erzeugt: je DB	148

<b>Grundprogramm-Optionen</b>			
Maschinensteuertafel			
FC 19	Transfer MCP-Signale, M-Variante	muss bei M-Variante der MCP geladen werden	92
FC 25	Transfer MCP-Signale, T-Variante	muss bei T-Variante der MCP geladen werden	92
FC 24	Transfer MCP-Signale, schmale Variante	muss bei schmalen-Variante der MCP geladen werden	100
FC 26	Transfer MCP-Signale, HT8-Variante	muss bei HT8 geladen werden	68
Bedienhandgerät			
FC 13	Display-Steuerung BHG	kann bei Bedienhandgeräten geladen werden	144
Fehler-/Betriebsmeldungen			
FC 10	Erfassung FM/BM	Laden bei Anwendung von FM / BM	66
ASUP			
FC 9	ASUP-Start	Laden bei Anwendung von ASUPs von PLC	128
<b>Grundprogramm-Optionen</b>			
Stern- / Dreieck-Umschaltung			
FC 17	Stern- / Dreieck-Umschaltung für HSA	Laden bei Stern- / Dreieck-Umschaltung	114
Spindelsteuerung			
FC 18	Spindelsteuerung	Laden bei Spindelsteuerung von PLC	132
PLC-/NC-Kommunikation			
FB 2	NC-Variable lesen	Laden bei NC-Variable lesen	76
DB n	NC-Variable lesen	ein Instanz-DB je FB 2-Aufruf	270
FB 3	NC-Variable schreiben	Laden bei NC-Variable schreiben	76
DB m	NC-Variable schreiben	ein Instanz-DB je FB 3-Aufruf	270
FB 4	PI-Dienste	Laden bei PI-Diensten	76
DB o	PI-Dienste	ein Instanz-DB je FB 4-Aufruf	130
DB 16	PI-Dienste Beschreibung	Laden bei PI-Diensten	618
FB 5	GUD Variablen lesen	Laden bei PI-Diensten	76
DB p	GUD Variablen lesen	ein Instanz-DB je FB 5-Aufruf	166
DB 15	Allgemeine Kommunikation	Global-DB für Kommunikation	146
FB 7	PI-Dienste 2	Laden bei PI-Diensten	76
DB o	PI-Dienste 2	ein Instanz-DB je FB 4-Aufruf: je	144
FC 21	Transfer	Laden bei DualPortRAM, ...	164
M zu N			
FB 9	Umschaltung M zu N	Laden bei M zu N	58
Safety Integrated			

<b>Grundprogramm-Optionen</b>			
FB 10	Sicherheitsrelais	Laden bei Option Safety	74
FB 11	Bremsentest	Laden bei Option Safety	76
DB 18	Safety Daten	DB für Safety	308
Werkzeug-Verwaltung			
FC 7	Transfer-Funktion Revolver	Laden bei Option Werkzeugverwaltung	84
FC 8	Transfer-Funktion	Laden bei Option Werkzeugverwaltung	132
FC 22	Richtungsauswahl	Laden, wenn Richtungsauswahl benötigt	138
DB 71	Beladestellen	wird vom GP abhängig von NC-MD erzeugt	40+30*B
DB 72	Spindeln	wird vom GP abhängig von NC-MD erzeugt	40+48*Sp
DB 73	Revolver	wird vom GP abhängig von NC-MD erzeugt	40+44*R
DB 74	Basis-Funktion	wird vom GP abhängig von NC-MD erzeugt	100+(B+ Sp+R)*22
Compile-Zyklen			
DB 9	Nahtstelle PLC-Compile-Zyklen	wird vom GP abhängig von NC-Option erzeugt	436

**Beispiel:**

Anhand des in der vorigen Tabelle angegebenen Speicherbedarfs wurde für zwei Musterkonfigurationen der Speicherbedarf ermittelt (siehe folgende Tabelle).

Baustein Typ, Nr.	Funktion	Bemerkung	Baustein-Größe (Byte)
			Arbeits- speicher
<b>Minimal-Konfiguration (1 Spindel, 2 Achsen und T-MCP)</b>			
s. o.	Grundprogramm, Basis		54688
	Nahtstellen-DBs		1612
	MCP		92
		Summe	56392

Baustein Typ, Nr.	Funktion	Bemerkung	Baustein-Größe (Byte)
			Arbeits- speicher
Maximal-Konfiguration (2 Kanäle, 4 Spindeln, 4 Achsen,T-MCP)			
s. o.	Grundprogramm, Basis		54688
s. o.	Nahtstellen-DBs		2768
s. o.	MCP		92
s. o.	Fehler- / Betriebsmeldungen		66
s. o.	ASUPs	1 ASUP-Anstoß	128
s. o.	Konkurrierende Achse	für 2 Revolver	132
s. o.	PLC- / NC- Kommunikation	1 x Variable lesen und 1 x Variable schreiben	838
s. o.	Werkzeug-Verwaltung	2 Revolver mit je 1 Beladestelle	674
s. o.	Compile-Zyklen		436
		Summe	59822

## 12.14 Rahmenbedingungen und NC-VAR-Selector

### 12.14.1 Rahmenbedingungen

#### 12.14.1.1 Programmier- und Parametrierwerkzeuge

#### Hardware

Für die bei SINUMERIK 840D sl eingesetzten PLCs ist bei den Programmiergeräten oder PCs folgende Ausstattung erforderlich:

	Minimal	Empfehlung
Prozessor	Pentium	Pentium
RAM (MB)	256	512 oder mehr
Festplatte, freier Speicherplatz (MB)	500	> 500
Schnittstellen	MPI, Ethernet inkl. Kabel Memory-Card	
Grafik	SVGA (1024*768)	
Maus	ja	
Betriebssystem	Windows 2000 /XP Professional, ab STEP 7 Version 5.3 SP2	

Auf Geräten, die den oben aufgeführten Randbedingungen entsprechen, kann das erforderliche **STEP 7** installiert werden, sofern es nicht bereits zum Lieferumfang des PG gehört.

Mit diesem Paket sind folgende Funktionen möglich:

- Programmieren
  - Editoren und Compiler für AWL (kompletter Sprachumfang incl. SFB-/SFC-Aufrufe), KOP, FUP
  - Erstellen und Bearbeiten von Zuordnungslisten (Symbol-Editor)
  - DB-Editor
  - Ein- und Ausgabe von Bausteinen ON-/OFF-Line
  - Einfügen von Änderungen und Ergänzungen sowohl ON- als auch OFF-Line
  - Übertragen der Bausteine von PG nach PLC und zurück
- Parametrieren
  - Parametrierwerkzeug **HW Config** für CPU- und Peripherie-Parametrierung
  - Parametrierwerkzeug **NetPro** für Einstellung der CPU-Kommunikations-Parameter
  - Ausgabe von Systemdaten wie Hardware-, Softwarestand, Speicherausbau, Peripherieausbau/-belegung

- Test und Diagnose (ON-LINE)
  - Status/Steuern Variable (Ein-/Ausgänge, Merker, DB Inhalte, etc.)
  - Status einzelner Bausteine
  - Anzeige von Systemzuständen (USTACK, BSTACK, SZL)
  - Anzeige von Systemmeldungen
  - PLC-Stop/Neustart/Urlöschen auslösen von PG
  - PLC Komprimieren
- Dokumentation
  - Ausdruck einzelner oder aller Bausteine
  - Vergabe von symbolischen Namen (auch für Variable in DBs)
  - Ein- und Ausgabe von Kommentaren innerhalb jedes Bausteines
  - Ausdruck der Test und Diagnoseanzeigen
  - Hardcopy-Funktion
  - Querverweisliste
  - Programmübersicht
  - Belegungsplan E/A/M/T/Z/D
- Archivieren Dienstprogramme
  - Vergabe von Ausgabeständen einzelner Bausteine
  - Vergleichen von Bausteinen
  - Umverdrahten
  - Umsetzer STEP 5 → STEP 7
- Optionspakete
  - Programmierung in S7-HIGRAPH, S7-GRAPH, SCL.  
Diese Pakete sind über den SIMATIC-Vertrieb bestellbar.
  - Zusatzpakete für die Projektierung von Baugruppen (z. B CP3425 → NCM-Paket)

---

**Hinweis**

Weitergehende Informationen zu den möglichen Funktionen sind den SIMATIC-Katalogen und der STEP 7-Dokumentation zu entnehmen.

---



### **12.14.1.2 Notwendige SIMATIC-Dokumentation**

**Literatur:**

- Systembeschreibung SIMATIC S7
- Operationsliste S7-300
- Programmierung mit STEP 7
- Benutzerhandbuch STEP 7
- Programmierhandbuch STEP 7; Entwerfen von Anwenderprogrammen
- Referenzhandbuch STEP 7; Anweisungsliste AWL
- Referenzhandbuch STEP 7; Kontaktplan KOP
- Referenzhandbuch STEP 7; Standard- und Systemfunktionen
- Handbuch STEP 7: Konvertieren von STEP 5 Programmen
- STEP 7 Gesamtindex
- Handbuch CPU 317-2DP

### **12.14.1.3 Relevante SINUMERIK-Dokumente**

**Literatur:**

- Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb
- Gerätehandbuch Bedienkomponenten und Vernetzung
- Funktionshandbuch Grundfunktionen
- Funktionshandbuch Erweiterungsfunktionen
- Funktionshandbuch Sonderfunktionen
- Listen sl (Buch1)
- Listen sl (Buch2)

## 12.14.2 NC-VAR-Selector

### 12.14.2.1 Übersicht

#### Allgemeines

Mit der PC-Applikation "NC-VAR-Selector" werden Adressen von benötigten NC-Variablen beschafft und für den Zugriff im PLC-Programm (FB 2 / FB 3) aufbereitet. Damit wird es einem PLC-Programmierer ermöglicht, NCK- und Antriebs-Variablen aus dem Gesamtangebot der NCK- und Antriebs-Variablen auszuwählen, diese Auswahl an Variablen abzuspeichern und sie mittels eines Code-Generators für den STEP 7-Compiler aufzubereiten, um sie dann als ASCII-Datei (\*.AWL) im Maschinen-CPU-Programm abzuspeichern. Das Bild "NC-VAR-Selector" gibt eine Vorstellung dieser Bearbeitung.

Zur Ablage der vom NC-VAR-Selector erzeugten Dateien ist ein Katalog über den Windows-Explorer mit beliebigem Katalognamen einzurichten. In diesen angelegten Katalog werden die selektierten Daten des NC-VAR-Selectors abgespeichert (Dateien Daten.VAR und Daten.AWL). Anschließend ist über den Menüpunkt "Code" → "in STEP7 Projekt" die AWL-Datei zu übertragen und zu übersetzen. Alternativ ist über den STEP 7 Manager ein "Einfügen", "externe Quelle" für die Datei "Daten.AWL" in das STEP 7-Maschinenprojekt durchzuführen. Hierzu ist im Manager der Quell-Container anzuwählen. Hiermit wird diese Datei in die Projektstruktur abgelegt. Nach der Übertragung der Datei sind diese AWL-Dateien mit STEP 7 zu übersetzen.

---

#### Hinweis

Für jeden NC-Softwarestand (auch ältere Versionen) kann der neueste NC-VAR-Selector verwendet werden. Für ältere NC-Software-Stände können die Variablen auch aus der neuesten Gesamtliste selektiert werden. Der Informationsinhalt im DB 120 (Standard-DB für Variablen) ist nicht abhängig vom Softwarestand. D. h. selektierte Variablen in einem älteren Softwarestand müssen bei einer Software-Hochrüstung nicht neu selektiert werden.

---

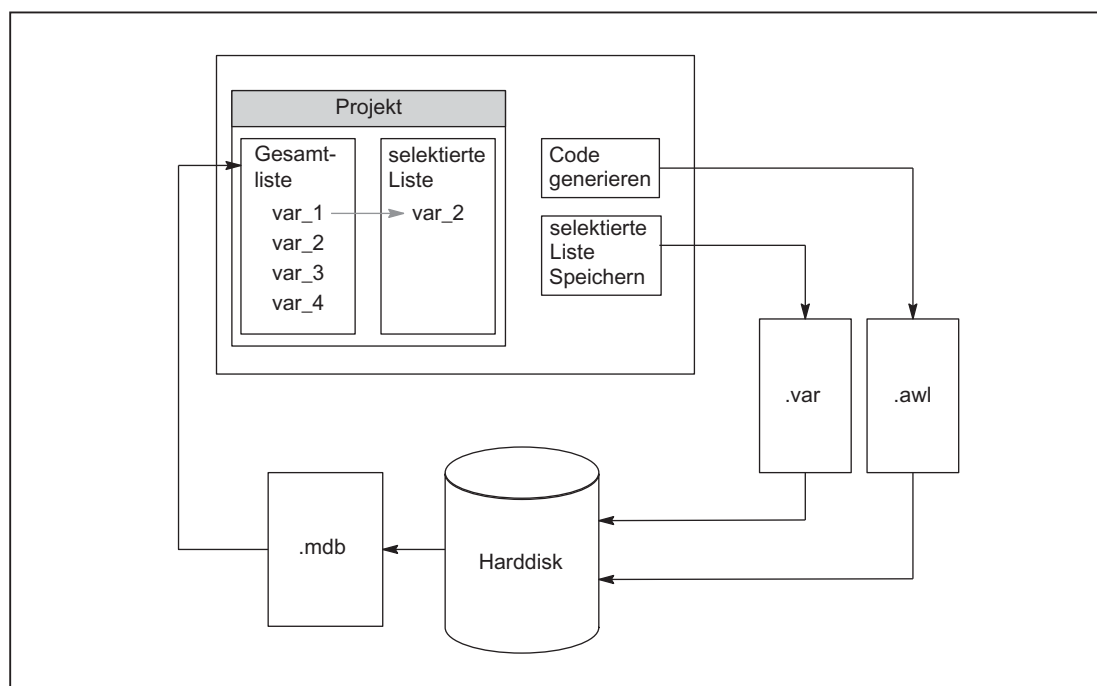


Bild 12-18 NC-VAR-Selector

Nach dem Starten der Applikation "NC-VAR-Selector" werden, nach Auswahl einer Variablenliste einer NC-Variante (Harddisk → File Ncv.mdb), alle in dieser Liste verfügbaren Variablen in einem Fenster angezeigt.

Es gibt folgende Variablen-Listen ncv\*.mdb:

Variablen	Liste
Variablen der NC inklusive Maschinen-, Setting-Daten:	ncv_NcData.mdb
Parameter des Antriebs:	ncv_SinamicsServo.mdb

Es können vom Bediener Variablen in eine zweite Liste (weiteres Fenster) übernommen werden. Diese selektierten Variablen können anschließend in einer ASCII-Datei abgelegt sowie als STEP 7-Source-Datei (.awl) aufbereitet und abgespeichert werden.

Mit der erzeugten STEP 7-Datei ist der PLC-Programmierer in der Lage, nach Generierung eines PLC-Datenbausteins durch den STEP 7-Compiler, NCK-Variablen zu lesen bzw. zu schreiben über die Grundprogramm-Funktionsbausteine "PUT" und "GET".

Die Liste der selektierten Variablen wird ebenfalls als ASCII-Datei abgelegt (Datei-Extension .var).

Die mit dem Werkzeug "NC-VAR-Selector" mitgelieferte Variablenliste ist passend zu dem aktuellen NC-Softwarestand. In dieser Liste sind keine vom Anwender definierten Variablen (GUD-Variablen) enthalten. Diese Variablen werden durch das Grundprogramm durch den Funktionsbaustein FB 5 behandelt.

---

#### **Hinweis**

Mit der neuesten Version des "NC-VAR-Selectors" können alle vorherigen NC-Software-Stände bearbeitet werden. Es ist somit keine parallele Installation von verschiedenen Versionen des "NC-VAR-Selectors" nötig.

---

### **Systemmerkmale, Rahmenbedingungen**

Die PC-Applikation "NC-VAR-Selector" setzt Windows 2000 oder ein höheres Betriebssystem voraus.

Die Namensgebung der Variablen ist beschrieben in:

**Literatur:**

Listen sl (Buch1); Kapitel: Variablen

bzw. auch in der Hilfedatei der Variablen (integriert im NC-VAR-Selector)

## 12.14.2.2 Funktionsbeschreibung

### Übersicht

Folgendes Bild verdeutlicht den Umfang des NC-VAR-Selectors beim Einsatz in der STEP 7-Umgebung.

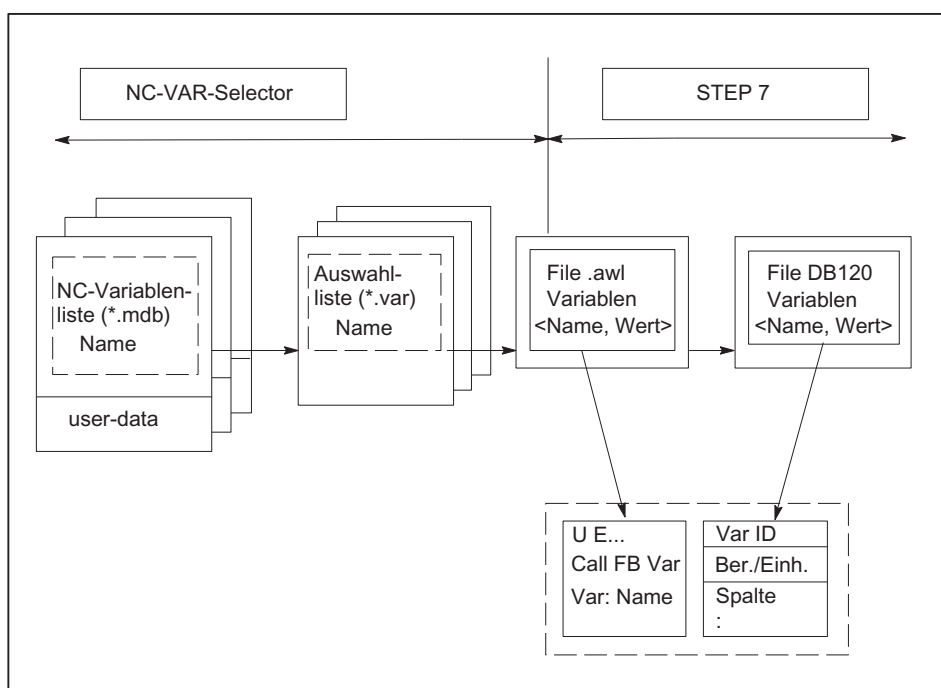


Bild 12-19 Einsatz des NC-VAR-Selectors in der STEP 7-Umgebung

Mit dem NC-VAR-Selector wird aus einer Variablenliste eine Liste selektierter Variablen erstellt und anschließend eine **.awl**-Datei erzeugt, die vom STEP 7-Compiler übersetzt werden kann.

- Ein \*.awl-File enthält sowohl die Namen bzw. ALIAS-Namen als auch die Informationen für die Adressparameter der NC-Variablen. Ein daraus erzeugter Datenbaustein enthält nur noch die Adressparameter (10 Byte je Parameter).
- Die erzeugten Datenbausteine sind stets in der maschinenspezifischen Dateiablage entsprechend der STEP 7-Festlegungen zu speichern.
- Damit die Parametrierung der Bausteine GET/PUT (FB 2/3) bezüglich der NC-Adressen symbolisch erfolgen kann, ist der frei vergebbare symbolische Name des erzeugten Datenbausteins in der STEP 7-Symboltabelle aufzunehmen.

### Grundbild / Grundmenü

Nach Anwahl (Starten) des NC-VAR-Selectors wird das Grundbild mit allen Bedienoptionen (obere Menüleiste) eingeblendet. Alle weiteren Fenster, die aufgeblendet werden, werden innerhalb des Gesamtfensters platziert.

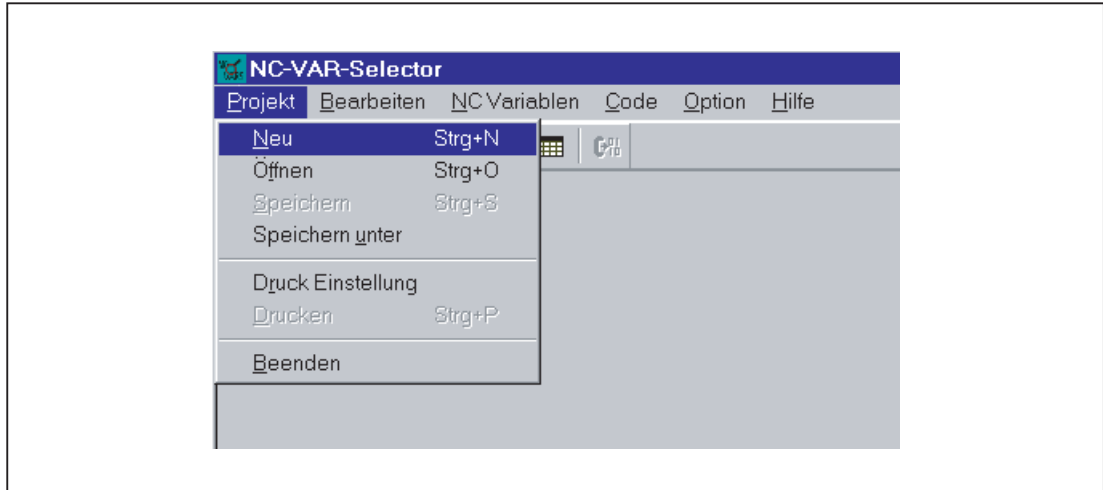


Bild 12-20 Grundbild mit Grundmenü

### Menüpunkt Projekt

Unter diesem Menüpunkt werden sämtliche Bedienhandlungen durchgeführt, die mit dem Projektfile (File der selektierten Variablen) zusammenhängen.

### Beenden der Applikation

Unter dem Menüpunkt "Projekt" wird über die Anwahl "Beenden" die Applikation beendet.

### Anlegen eines neuen Projekts

Über den Menüpunkt "Projekt" wird ein neues Projekt (neue Datei für selektierte Variablen) erstellt.

Die Anwahl "NEU" blendet ein Fenster für die selektierten Variablen auf. Danach wird nach einer Abfrage die Dateiauswahl für die NC-Variablen-Liste angeboten (nur wenn die NC-Variablen-Liste noch nicht geöffnet ist).

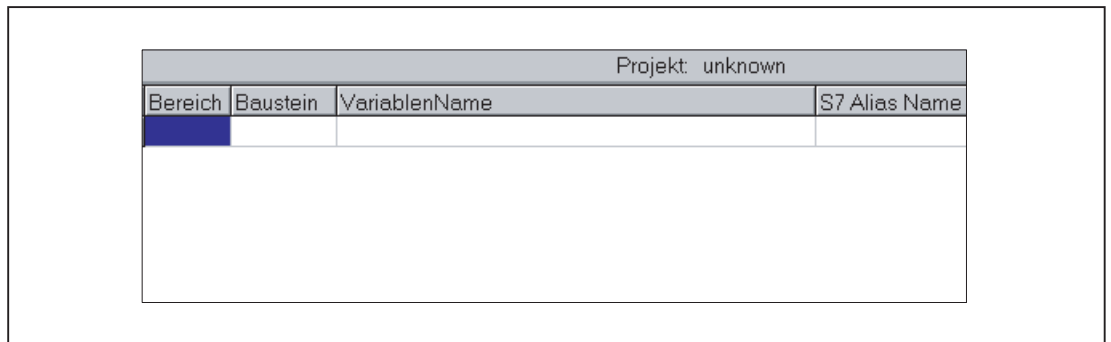


Bild 12-21 Fenster für selektierte Variable bei neuem Projekt

Die selektierten Variablen werden in einem Fenster dargestellt.

### Öffnen eines bereits existierenden Projekts

Unter dem Menüpunkt "Projekt" kann über die Anwahl "Öffnen" ein bereits existierendes Projekt (bereits selektierte Variable) geöffnet werden. Es erscheint ein Datei-Auswahlfenster, mit dem das entsprechende Projekt mit der Erweiterung ".var" ausgewählt werden kann.

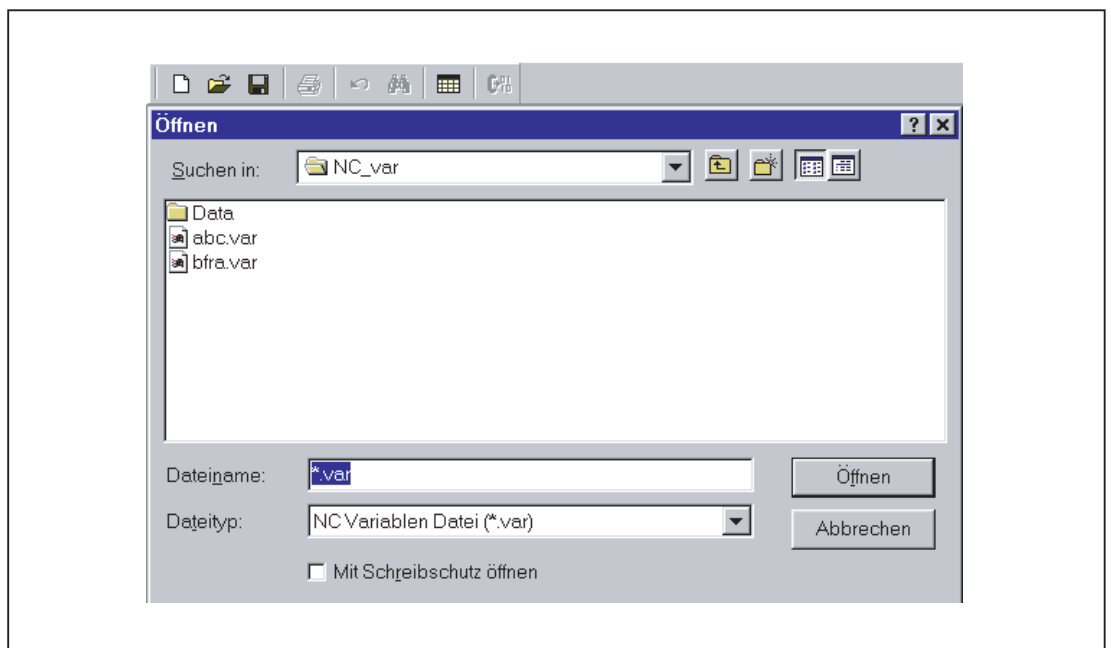


Bild 12-22 Auswahlfenster für vorhandene Projekte

Nach Auswahl des Projekts muss, falls Variable neu aufgenommen werden sollen, wiederum eine Gesamtliste von NC Variablen angewählt werden. Sollen nur Variable gelöscht werden, kann dies ohne Anwahl einer Gesamtliste erfolgen.

### Speichern eines Projekts

Über die Menüpunkte "Projekt" > "Speichern" bzw. "Speichern unter ..." wird die Variablen-Liste gespeichert.

"Speichern" legt die Variablen-Liste unter einem bereits bekannten Pfad ab. Ist der Projektpfad nicht bekannt, ist das Verhalten wie bei "Speichern unter ...".

"Speichern unter ..." blendet ein Fenster auf, um den Pfad für das zu speichernde Projekt angeben zu können.

### Drucken eines Projekts

Unter dem Menüpunkt "Projekt" kann über die Anwahl "Drucken" eine Projektdatei ausgedruckt werden. Die Zeilen je Seite werden durch den Menüpunkt "Druck Einstellung" vorgenommen. Standardeinstellung sind 77 Zeilen.

### Menüpunkt Bearbeiten

Hier können z. B. folgende Bedienaktionen direkt durchgeführt werden:

- Variablen übernehmen
- Variablen löschen
- Aliasnamen ändern
- Variablen suchen

Weiterhin können diese Aktionen rückgängig gemacht werden.

### Rückgängig machen

Bedienaktionen, die sich auf die Erstellung der Projektdatei beziehen (Variablen übernehmen, Variablen löschen, Aliasnamen ändern), können hier rückgängig gemacht werden.

### Menüpunkt NC Variablen

Die Ablage der Basisliste aller Variablen erfolgt unter dem NC-Var-Selector-Pfad Data\Swxy (xy steht für SW-Stand-Nr., z. B. SW 5.3:=xy=53). Diese Liste kann als NC-Variablen-Liste angewählt werden. Bei SINUMERIK 840D sl sind die Basislisten unter dem Pfad Data\Swxy\_sl enthalten.



## Wählen einer NC-Variablen-Liste

Mit dem Menüpunkt "NC Variablen Liste", "Wählen" wird nun eine Liste der NC-Variablen einer NC-Version ausgewählt und angezeigt.

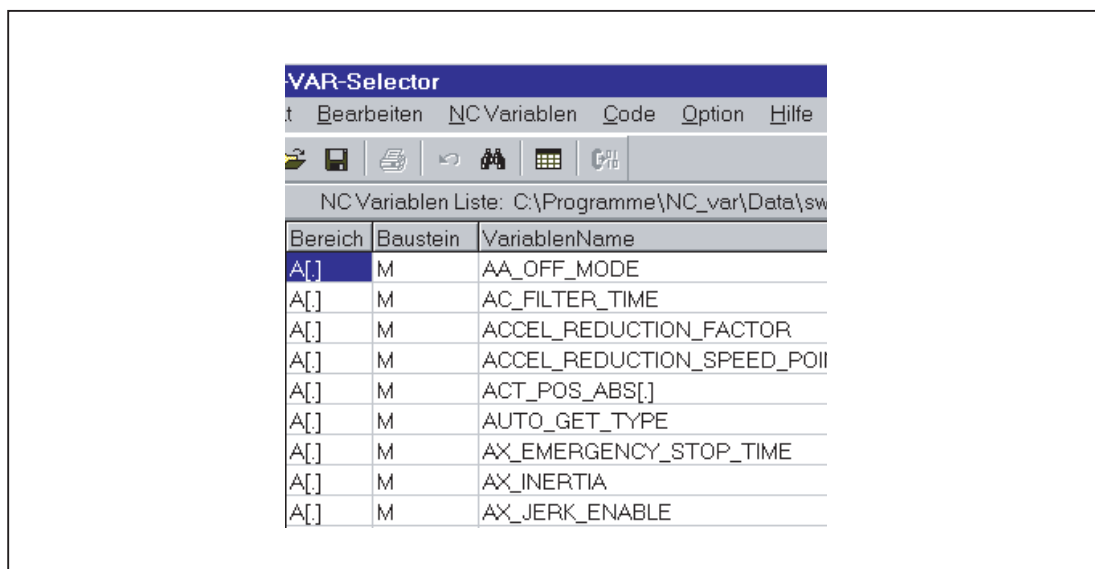


Bild 12-23 Fenster mit angewählter Gesamtliste

Die Feldvariablen (z. B. Achsbereich, T-Bereichsdaten usw.) werden mit Klammern ([.]) angedeutet. An dieser Stelle ist eine Zusatzinformation notwendig. Bei Übernahme der Variablen in die Projektliste wird die benötigte Zusatzinformation abgefragt.

## Teilmengen anzeigen

Mit Doppelklick auf ein beliebiges Tabellenfeld (Ausnahme: Variablenfeld!) wird ein Fenster eingeblendet, in dem Filterkriterien vorgegeben werden können.

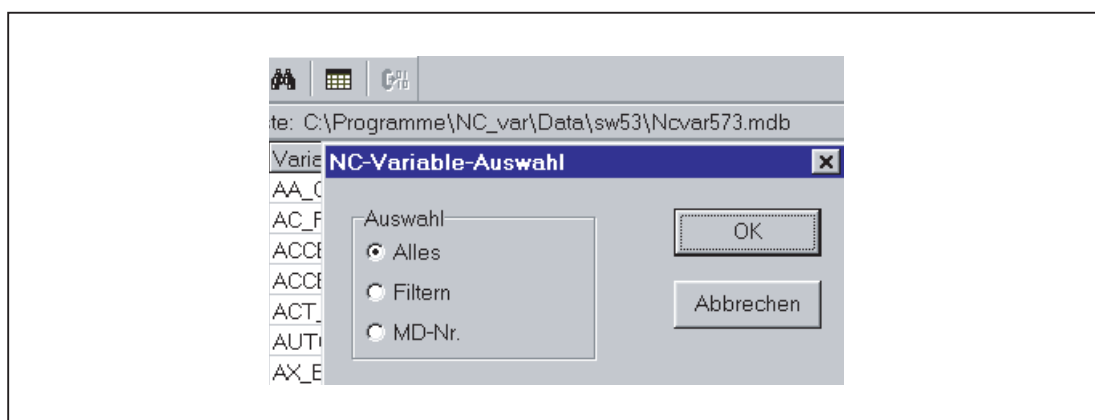


Bild 12-24 Fenster mit Filterkriterien für Anzeige der Variablen-Liste

Es stehen drei Optionen zur Verfügung:

- Alles anzeigen
- Bereich, Baustein und Name vorgeben (auch kombiniert)
- MD/SE-Daten-Nummer anzeigen

Es besteht auch die Möglichkeit, folgende Wildcards zu benutzen:

*	für eine beliebig lange Ergänzung des Suchkriteriums
---	--

### Beispiel für Suchkriterien

Suchkriterium Name: CHAN\* gefunden wird:

CHAN\_NAME  
chanAlarm  
chanStatus  
channelName  
chanAssignment

- Variablen selektieren

Eine Variable wird mittels einfachem Mausklick selektiert und mit einem Doppelklick in das Fenster der selektierten Variablen übernommen. Unter dem Menüpunkt "Bearbeiten" kann diese Aktion auch wieder rückgängig gemacht werden.

### Alias-Name

Die angebotenen Variablennamen können bis zu 32 Zeichen lang sein. Um Variablen im zu erzeugenden Datenbaustein eindeutig zu machen, wird der gewählte Name weiterhin um einige ASCII-Zeichen ergänzt. Der STEP 7-Compiler erkennt jedoch maximal 24 ASCII-Zeichen als eindeutige S7-Variable. Da nicht ausgeschlossen werden kann, dass Variablennamen sich nur in den letzten 8 Stellen unterscheiden, werden für die zu langen Namen **Alias**-Namen verwendet. Bei der Selektion einer Variablen wird deshalb geprüft, wie lang der zu verwendende S7-Name wird. Wird dieser länger als 24 Zeichen, muss über eine Zusatzeingabe ein vom Anwender vorzugebender Name (Alias-Name) eingesetzt werden.

**Hierbei muss der Anwender darauf achten, dass dieser eindeutig ist.**

Die Alias-Eingabe kann vom Anwender im Menü "Option" auch immer aktiviert werden. Die Alias-Eingabe ist dann bei jeder Übernahme einer Variablen möglich.

Weiterhin kann durch Doppelklick auf das S7-Variablennamen-Feld dieser Name nachträglich editiert werden. Unter dem Menüpunkt "Bearbeiten" kann diese Aktion auch wieder rückgängig gemacht werden.

7	A[.]	M	AX_EMERGENCY_STOP_TIME
8	A[.]	M	AX_INERTIA
9	A[.]	M	AX_JERK_ENABLE
10	A[.]	M	AX_JERK_TIME
Proje			
	Bereich	Baustein	VariablenName
1	A[.]	M	AA_OFF_MODE

Bild 12-25 Bild mit Gesamtliste und selektierten Variablen

## Scrollen

Können nicht alle Variablen im Fenster angezeigt werden, wird ein Scrollbar eingeblendet. Mit Scrollen (Page-Up/Down) können die restlichen Variablen erreicht werden.

## Variablen in mehrdimensionalen Strukturen

Werden Variable aus mehrdimensionalen Strukturen ausgewählt, so werden für die Adressierung dieser Variablen die Eingabe der Spalten- und/oder Zeilennummer sowie die Bereichs-Nummer abgefragt. Die erforderlichen Nummern können der NC-Variablen-Dokumentation entnommen werden.

### Literatur:

Listen sl (Buch1); Variablen

Die Eingabe einer Null (0) als Bereichs-Nr., Zeilen- oder Spaltenindex bewirkt, dass die Variable im S7-AG als Zeiger auf diese Daten benutzt werden kann. Beim Lesen bzw. Schreiben dieser Daten über die Funktion "PUT" und "GET" müssen dann die optionalen Parameter "UnitX", "ColumnX" und "LineX" mit den notwendigen Informationen aufgefüllt werden.

Eingabe von Zeile, Spalte und Bereichsnummer

Bereichs Nr. <input style="width: 40px;" type="text" value="1"/>	Variable ACT_POS_ABS[.] ergänzen mit: (siehe Hilfe) Bereichs Nr. = Achsnr. Zeile
Zeile <input style="width: 40px;" type="text" value="1"/>	
<input type="button" value="OK"/> <input style="margin-left: 100px;" type="button" value="Abbrechen"/> <input style="margin-left: 100px;" type="button" value="Hilfe"/>	

Bild 12-26 Eingabefeld für Zeile, Spalte und Baustein-Nr.

### Löschen von Variablen

Das Löschen von Variablen im Fenster der selektierten Variablen wird durch Auswahl (einfacher Mausklick) der Variablen und anschließendem Betätigen der Taste "Delete" ausgeführt. Für die Funktion Doppelklick gibt es keine Aktion. Eine Anwahl von mehreren Variablen zum Löschen ist möglich (siehe Abschnitt "Beispiel für Suchkriterien > Variablen selektieren").

Unter dem Menüpunkt "Bearbeiten" kann diese Aktion auch wieder rückgängig gemacht werden.

### Hinweis

Beim Löschen von Variablen ergibt sich eine Veränderung der Absolutadressen der Zeiger-Strukturen auf die Variablen. Es ist bei der Änderung der Variablen-Selektion also zwingend erforderlich, **vor der Änderung** eine oder mehrere **Textdateien aller Anwender-Bausteine zu erzeugen**. Nur so kann nach erneuter Kompilierung sichergestellt werden, dass die Zuordnung der Variablen im FB "GET" oder FB "PUT" noch übereinstimmt.

### Selektierte Liste speichern

Nach erfolgter Variablen Auswahl können diese unter einem Projektnamen angelegt werden. Die Ablage der Dateien erfolgt projektspezifisch.

Für die abzulegende Datei wird ein Fenster aufgeblendet, in welchem der Projektpfad und Name für diese Datei auszuwählen ist.

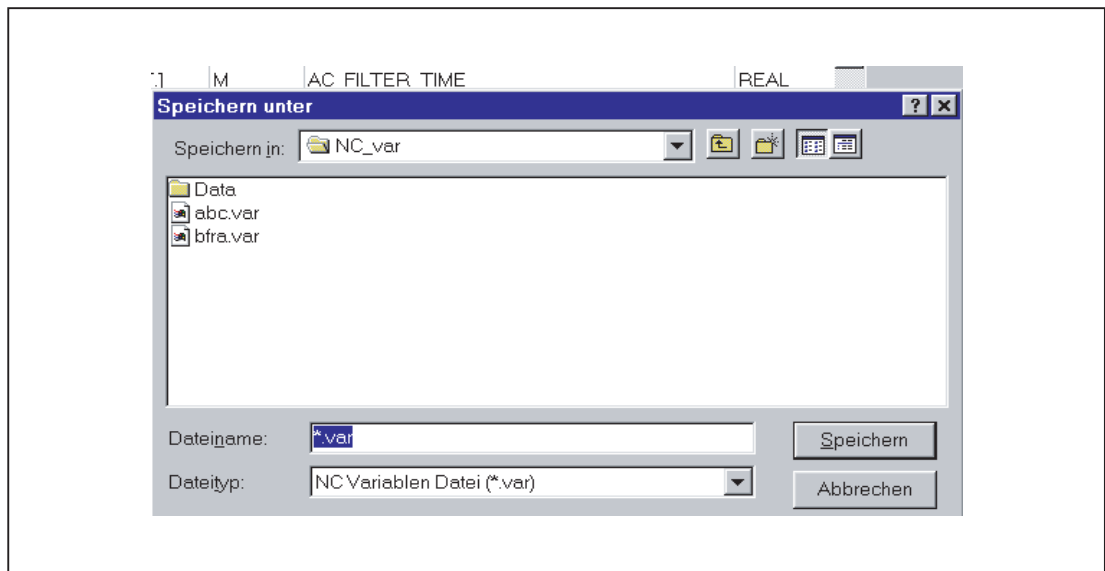


Bild 12-27 Fenster für Projektpfad und Name der abzulegenden Datei

## Code-Generierung

Unter diesem Menüpunkt stehen drei Unterpunkte zur Auswahl:

1. Einstellungen (Vorgabe der zu erzeugenden Datenbausteinnummer) und weitere Einstellungen
2. Generieren (Datenbaustein generieren)
3. In STEP 7 Projekt (Übernahme des Datenbausteins in ein STEP 7-Projekt)

## Einstellungen

Unter diesem Menüpunkt erfolgt die Eingabe der DB-Nummer und des Symbols für diese DB-Nummer, für den der Code erzeugt wird.

Unter dem Tabulator "Mass System" kann eine Auswahl getroffen werden, wie Maßsystem-behaftete Variablen in der PLC verrechnet werden.

Unter dem Tabulator "Generierung" kann die Erzeugung für das jeweilige Zielsystem festgelegt werden.

## Generieren

Unter diesem Menüpunkt erfolgt die Einstellung der STEP 7-Datei aus der selektierten Variablen Liste mit der Erweiterung ".awl".

Mit "Anwahl" wird eine Datei erzeugt:

Eine .awl-Datei, die als Input für den STEP 7-Compiler genutzt werden kann.

Für die zu speichernde Datei wird ein Fenster eingeblendet, in welchem Pfad und Name für die zu erzeugende .awl-Datei anzugeben ist.

## In STEP 7-Projekt

Die generierte AWL-Datei wird in ein auswählbares SIMATIC-Projekt (Programmpfad) übertragen und compiliert. Zusätzlich kann auch das Symbol übernommen werden. Diese Funktion ist ab STEP 7 Version 5.1 verfügbar. Dieser Vorgang dauert eine längere Zeit wegen Aufruf von STEP 7. Vor Übertragen einer neuen AWL-Datei ist im KOP/FUP/AWL-Editor das Dateifenster der AWL-Datei zu schließen.

## Menüpunkt Option

Unter dem Menüpunkt "Option" kann gewählt werden:

- Die aktuelle Sprache
- Der Modus für die Alias-Eingabe (immer / > 24 Zeichen)

### **Menüpunkt Hilfe**

Hier kann durch Auswahl des entsprechenden Untermenüpunktes nachgelesen werden:

- Die Bedienungsanleitung
- Die Variablenbeschreibung

Weiterhin werden hier das Copyright und die Versionsnummer angezeigt.

### **12.14.2.3 Inbetriebnahme, Installation**

Die Installation der Windows-Applikation "NC-VAR-Selector" erfolgt über das mitgelieferte SETUP-Programm.

## 12.15 Bausteinbeschreibungen

### 12.15.1 FB 1: RUN\_UP Grundprogramm, Anlaufteil

#### Funktion

Im Anlauf erfolgt die Synchronisation zwischen NCK und PLC. Es werden die Datenbausteine für die Anwendernachtstelle NC/PLC anhand der über Maschinendaten festgelegten NC-Konfiguration erzeugt und die wichtigsten GP-Parameter auf Plausibilität geprüft. Im Fehlerfall übergibt der FB 1 eine Fehlerkennung in den Diagnosepuffer und überführt die PLC in den Stopp-Zustand.

Für den ordnungsgemäßen Hochlauf der Steuerung ist es erforderlich, die für sich autark hochlaufenden Systeme NCK, PLC zu synchronisieren. Deshalb ist der Anlauf derart konzipiert, dass die CPUs jeweils für sich "Teilanlaufaktionen" ausführen und der jeweils anderen CPU durch Austausch von Kennungen den korrekten Funktionsablauf mitteilen.

Da der Anlauf asynchron erfolgt, ist es nicht vermeidbar, dass unter Umständen eine CPU "warten" muss, bis die andere vom Ablauf her "gleichgezogen" hat. Dieses wird automatisch durch das Grundprogramm realisiert.

Die integrierte PLC kennt nur die Anlaufart Neustart. Es existiert kein Wiederanlauf, d. h. das Betriebssystem durchläuft nach der Systeminitialisierung den Organisationsbaustein OB 100 und beginnt den zyklischen Betrieb immer am Anfang des OB 1.

Der Anwender braucht nur die für ihn relevanten Parameter des FB 1 zu versorgen. Die voreingestellten Werte im zugehörigen Instanz-DB 7 brauchen nicht zugewiesen zu werden. Der Baustein darf nur im OB 100 aufgerufen werden.

#### Ausgangsparameter

Die Ausgangsparameter im FB 1 dienen dem PLC-Anwender als Informationsquelle über die Konfiguration der Steuerung. Auf diese Daten kann auch im zyklischen Programmteil zugegriffen werden.

Hierzu gibt es 2 Möglichkeiten des Zugriffs:

1. Zugriff über einen Direktzugriff in den Datenbaustein DB 7 (Instanz des FB 1) in symbolischer Form (z. B. L gp\_par.MaxChan, gp\_par ist hier der symbolische Name des DB 7).
2. Zuweisung eines Merker, Datenelements bei der Parametrierung des FB 1 an den zugehörigen Parameter (z. B. MaxChan:=MW 20). Anschließend kann im weiteren Anwenderprogramm die Information über die maximale Anzahl Kanäle über das Merkerwort 20 abgefragt werden.

---

#### Hinweis

Für die Belegung der Parameter der MCP und des BHG siehe "Projektierung von Maschinensteuertafel, Bedienhandgerät, Direkttasten (Seite 962)".

---

**Deklaration SINUMERIK 840D sl**

```

FUNCTION_BLOCK FB 1
VAR_INPUT
    MCPNum :           INT:=1;           //0: keine MCP
                                           //1: 1 MCP (default)
                                           //2: 2 MCP

    MCP1In :           POINTER;          //Anfangsadr. Eingangssignale MCP 1
    MCP1Out :          POINTER;          //Anfangsadr. Ausgangssignale MCP 1
    MCP1StatSend :    POINTER;          //Status-DW für Senden MCP 1
    MCP1StatRec :     POINTER;          //Status-DW für Empfangen MCP 1
    MCP1BusAdr :      INT:=6;           //default
    MCP1Timeout :     S5TIME:=S5T#700MS;
    MCP1Cycl :        S5TIME:=S5T#200MS;

    MCP2In :           POINTER;          //Anfangsadr. Eingangssignale MCP 2
    MCP2Out :          POINTER;          //Anfangsadr. Ausgangssignale MCP 2
    MCP2StatSend :    POINTER;          //Status-DW für Senden MCP 2
    MCP2StatRec :     POINTER;          //Status-DW für Empfangen MCP 2
    MCP2BusAdr :      INT;
    MCP2Timeout :     S5TIME:=S5T#700MS;
    MCP2Cycl :        S5TIME:=S5T#200MS;
    MCPMPI :          BOOL:=FALSE;
    MCP1Stop :        BOOL:=FALSE;
    MCP2Stop :        BOOL:=FALSE;
    MCP1NotSend :     BOOL:=FALSE;
    MCP2NotSend :     BOOL:=FALSE;
    MCPSDB210 :       BOOL:=FALSE;
    MCPCopyDB77 :     BOOL:=FALSE;
    MCPBusType :      BYTE=B#16#0;

    BHG :             INT:=0;           //Bedienhandgerät-Schnittstelle
                                           //0: kein BHG
                                           //1: BHG an MPI
                                           //2: BHG an BTSS

    BHGIn :           POINTER;          //Sendedaten des Bedienhandgerätes
    BHGOut :          POINTER;          //Empfangsdaten des Bedienhandgerätes
    BHGStatSend :    POINTER;          //Status-DW für Senden BHG
    BHGStatRec :     POINTER;          //Status-DW für Empfangen BHG
    BHGInLen :       BYTE:=B#16#6;     //Input 6 Byte
    BHGOutLen :      BYTE:=B#16#14;    //Output 20 Byte
    BHGTimeout :     S5TIME:=S5T#700MS;
    BHGCycl :        S5TIME:=S5T#100MS;
    BHGRecGDNo :     INT:=2;
    BHGRecGBZNo :    INT:=2;
    BHGRecObjNo :    INT:=1;
    BHGSendGDNo :    INT:=2;
    
```



```
BHGSendGBZNo :      INT:=1;
BHGSendObjNo :      INT:=1;
BHGMPI :            BOOL:=FALSE;
BHGStop :           BOOL:=FALSE;
BHGNotSend :        BOOL:=FALSE;
NCCyclTimeout :     S5TIME:=S5T#200MS;
NCRunupTimeout :    S5TIME:=S5T#50S;
ListMDecGrp :       INT:=0;
NCKomm :            BOOL:=FALSE;
MMCToIF :           BOOL:=TRUE;
HWheelMMC :         BOOL:=TRUE;           //Handradanwahlen durch HMI
ExtendAlMsg :       BOOL;
MsgUser :           INT:=10;             //Anzahl der Userbereiche im DB 2
UserIR :            BOOL:=FALSE;        //Anwenderprogramme im OB 40,
                                           //Lokaldatenerweiterung beachten!
IRAuxfuT :          BOOL:=FALSE;        //T-Funktion in OB 40 auswerten
IRAuxfuH :          BOOL:=FALSE;        //H-Funktion in OB 40 auswerten
IRAuxfuE :          BOOL:=FALSE;        //DL-Funktion in OB 40 auswerten
UserVersion :       POINTER;            //Zeiger auf String-Variable, die im
                                           //Versionsbild angezeigt wird

OpKeyNum :          INT;
Op1KeyIn :          POINTER;
Op1KeyOut :         POINTER;
Op1KeyBusAdr :      INT;
Op2KeyIn :          POINTER;
Op2KeyOut :         POINTER;
Op2KeyBusAdr :      INT;
Op1KeyStop :        BOOL;
Op2KeyStop :        BOOL;
Op1KeyNotSend :     BOOL;
Op2KeyNotSend :     BOOL;
OpKeyBusType :      BYTE;
IdentMcpBusAdr :    INT;
IdentMcpProfilNo :  BYTE;
IdentMcpBusType :   BYTE;
IdentMcpStrobe :    BOOL;
END_VAR
```

```

VAR_OUTPUT
    MaxBAG :           INT;
    MaxChan :          INT;
    MaxAxis :          INT;
    ActivChan :        ARRAY[1..10] OF BOOL;
    ActivAxis :        ARRAY[1..31] OF BOOL;
    UDInt :            INT;
    UDHex :            INT;
    UDReal :           INT;
    IdentMcpType :     BYTE;
    IdentMcpLengthIn : BYTE;
    IdentMcpLengthOut : BYTE;
END_VAR
    
```

**Erläuterung der Formalparameter SINUMERIK 840D sl**

Die folgende Tabelle zeigt alle Formalparameter der Funktion RUN\_UP für 840D sl:

Signal	Art	Typ	Wertebereich	Bemerkung
MCPNum	E	INT	bis 2	Anzahl der aktiven MCP 0: keine MCP vorhanden
MCP1In MCP2In	E	POINTER	E0.0 bis E120.0 oder M0.0 bis M248.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für die Eingangssignale der betr. Maschinensteuertafel
MCP1Out MCP2Out	E	POINTER	A0.0 bis A120.0 oder M0.0 bis M248.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für die Ausgangssignale der betr. Maschinensteuertafel
MCP1StatSend MCP2StatSend	E	POINTER	A0.0 bis A124.0 oder M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Derzeit keine Bedeutung
MCP1StatRec MCP2StatRec	E	POINTER	A0.0 bis A124.0 oder M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Derzeit keine Bedeutung
MCP1BusAdr MCP2BusAdr	E	INT	1 ... 126 192 .. 223	DP-Slave: PROFIBUS-Adresse Ethernet MCP: DIP-Einstellung

Signal	Art	Typ	Wertebereich	Bemerkung
MCP1Timeout MCP2Timeout	E	S5time	Empfehlung: 700 ms	Zyklische Lebenszeichen-Überwachung für die Maschinensteuertafel
MCP1Cycl MCP2Cycl	E	S5time	Empfehlung: 200 ms	Nur bei PROFIBUS relevant
MCPMPI	E	BOOL	false	Wegen Kompatibilität vorhanden
MCP1Stop MCP2Stop	E	BOOL		0: Übertragung der Maschinensteuertafel-Signale starten 1: Übertragung der Maschinensteuertafel-Signale anhalten DP-Slave: Slave wird deaktiviert
MCP1NotSend MCP2NotSend	E	BOOL		0: Sende- und Empfangsbetrieb aktiviert 1: Nur Empfang der Maschinensteuertafel-Signale
MCPsDB210	E	BOOL	false	Wegen Kompatibilität vorhanden
MCPCopyDB77	E	BOOL	false	Wegen Kompatibilität vorhanden
MCPBusType	E	BYTE		rechtes Halbbyte (Bit 0...3) für MCP1 linkes Halbbyte (Bit 4...7) für MCP2 b#16#33: PROFIBUS b#16#44: PROFIBUS am MPI/DP-Anschluss b#16#55: Ethernet B#16#66: PROFINET  Mischbetrieb möglich, siehe Kapitel "Projektierung von Maschinensteuertafel, Bedienhandgerät, Direktasten (Seite 962)"
BHG	E	INT	0, 5	Bedienhandgerät-Schnittstelle: 0: kein BHG 5: BHG an Ethernet
BHGIn	E	POINTER	E0.0 bis E124.0 oder M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse Empfangsdaten der PLC vom Bedienhandgerät
BHGOut	E	POINTER	A0.0 bis A124.0 oder M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse Sendedaten der PLC zum Bedienhandgerät

12.15 Bausteinbeschreibungen

Signal	Art	Typ	Wertebereich	Bemerkung
BHGStatSend	E	POINTER	A0.0 bis A124.0 oder M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Wegen Kompatibilität vorhanden
BHGStatRec	E	POINTER	A0.0 bis A124.0 oder M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Wegen Kompatibilität vorhanden
BHGInLen	E	BYTE	BHG default: B#16#6 (6 Byte)	Wegen Kompatibilität vorhanden
BHGOutLen	E	BYTE	BHG default: B#16#14 (20 Byte)	Wegen Kompatibilität vorhanden
BHGTimeout	E	S5time	Empfehlung: 700 ms	Wegen Kompatibilität vorhanden
BHGCycl	E	S5time	Empfehlung: 100 ms	Wegen Kompatibilität vorhanden
BHGRecGDNo	E	INT	BHG default: 2	Ethernet DIP-Schalter
BHGRecGBZNo	E	INT	BHG default: 2	Wegen Kompatibilität vorhanden
BHGRecObjNo	E	INT	BHG default: 1	Wegen Kompatibilität vorhanden
BHGSendGDNo	E	INT	BHG default: 2	Wegen Kompatibilität vorhanden
BHGSendGBZNo	E	Int	BHG default: 1	Wegen Kompatibilität vorhanden
BHGSendObjNo	E	INT	BHG default: 1	Wegen Kompatibilität vorhanden
BHGMPI	E	BOOL	false	Wegen Kompatibilität vorhanden
BHGStop	E	BOOL		0: Übertragung der Bedienhandgerät-Signale starten
				1: Übertragung der Bedienhandgerät-Signale anhalten
BHGNotSend	E	BOOL		0: Sende und Empfangsbetrieb aktiviert
				1: Nur Empfang der Bedienhandgerät-Signale
NCCyclTimeout	E	S5time	Empfehlung: 200 ms	Zyklische Lebenszeichen-Überwachung NCK
NCRunupTimeout	E	S5time	Empfehlung: 50 s	Hochlauf-Überwachung NCK
ListMDecGrp	E	INT	0 ... 16	Aktivierung der erweiterten M-Gruppen-Decodierung
				0: nicht aktiv
				1...16: Anzahl der M-Gruppen
NCKomm	E	BOOL		PLC-NC-Kommunikationsdienste (FB 2/3/4/5/7: Put/Get/PI_SERV/GETGUD)
				TRUE: aktiv

Signal	Art	Typ	Wertebereich	Bemerkung
MMCToIF	E	BOOL		Übertragung der HMI-Signale an die Nahtstelle (Betriebsarten, Programmbeeinflussung, usw.)
				TRUE: aktiv
HWheelMMC	E	BOOL		TRUE: Handradanwahl über HMI
				FALSE: Handradanwahl durch Anwenderprogramm
ExtendAIMsg	E	BOOL		Aktivierung Erweiterung des FC 10 (siehe Kapitel "Struktur und Funktionen des Grundprogramms (Seite 941)")
MsgUser	E	INT	0 ... 32	Anzahl Userbereiche für Meldungen (DB 2)
UserIR	E	BOOL		Lokaldatenerweiterung OB 40 notwendig zur Verarbeitung der dortigen Signale vom Anwender
IRAuxfuT	E	BOOL		T-Funktion in OB 40 auswerten
IRAuxfuH	E	BOOL		H-Funktion in OB 40 auswerten
IRAuxfuE	E	BOOL		DL-Funktion in OB 40 auswerten
UserVersion	E	POINTER	DBxx	Zeiger auf String-Variable. Die zugehörige Stringvariable wird im Versionsbild angezeigt (max. 41 Zeichen).
OpKeyNum	E	INT	0 ... 2	Anzahl der aktiven Direkttasten-Module
				0: keine Ethernet-Direkttasten vorhanden
Op1KeyIn Op2KeyIn	E	POINTER	P#Ex.0 oder P#Mx.0 oder P#DBn.DBXx.0.	Anfangsadresse für die Eingangssignale der betreffenden Direkttasten-Module
Op1KeyOut Op2KeyOut	E	POINTER	P#Ax.0 oder P#Mx.0 oder P#DBn.DBXx.0.	Anfangsadresse für die Ausgangssignale der betreffenden Direkttasten-Module
Op1KeyBusAdr Op2KeyBusAdr	E	INT	1 ... 191	Direkttasten über Ethernet: TCU-Index
Op1KeyStop Op2KeyStop	E	BOOL		0: Übertragung der Direkttasten-Signale starten
				1: Übertragung der Direkttasten-Signale anhalten
Op1KeyNotSend Op2KeyNotSend	E	BOOL		0: Sende und Empfangsbetrieb aktiviert
				1: Nur Empfang der Direkttasten-Signale
OpKeyBusType	E	BYTE	b#16#55	b#16#55: Ethernet
IdentMcpBusAdr	E	INT	1 ... 254	nur IE-Geräte
IdentMcpProfilNo	E	BYTE	0, 1	Profil eines Geräts

Signal	Art	Typ	Wertebereich	Bemerkung
				0: vollständiges Gerät 1: nur Direkttasten
IdentMcpBusType	E	BYTE	b#16#5	nur IE-Geräte
IdentMcpStrobe	E	BOOL		Abfrage aktivieren
MaxBAG	A	INT	1 ... 10	Anzahl BAGs
MaxChan	A	INT	1 ... 10	Anzahl Kanäle
MaxAxis	A	INT	1 ... 31	Anzahl Achsen
ActivChan	A	ARRAY [1..10] OF BOOL		Bitleiste aktive Kanäle
ActivAxis	A	ARRAY [1..31] OF BOOL		Bitleiste aktive Achsen
UDInt	A	INT		Anzahl INTEGER-Maschinendaten im DB 20
UDHex	A	INT		Anzahl Hexadezimal-Maschinendaten im DB 20
UDReal	A	INT		Anzahl REAL-Maschinendaten im DB 20
IdentMcpType	A	BYTE		Type (HT2, HT8, ...)
IdentMcpLengthIn	A	BYTE		Längen-Info Eingangsdaten in PLC
IdentMcpLengthOut	A	BYTE		Längen-Info Ausgangsdaten in PLC

### Überwachung MCP / BHG (840D sl)

Für die Kommunikation mit den Maschinensteuertafeln (MCP) werden im Fehlerfalle folgende Alarme am HMI angezeigt:

- 400260: MSTT 1 ausgefallen oder
- 400261: MSTT 2 ausgefallen.
- 400262: BHG ausgefallen.

In diesem Falle werden die Eingangssignale von der MCP bzw. vom Bedienhandgerät (MCP1In / MCP2In bzw. BHGIn) mit 0 initialisiert. Sollte eine Neusynchronisation zwischen PLC und MCP/BHG möglich sein, wird die Kommunikation automatisch wieder aufgenommen und die Fehlermeldung vom GP gelöscht.

## Aufrufbeispiel für 840D sl

Im Folgenden ist ein Aufrufbeispiel für den FB 1 im OB 100 aufgeführt. Dieses Beispiel ist Bestandteil der Diskette mit dem Grundprogramm für 840D sl.

```
ORGANIZATION_BLOCK OB 100
VAR_TEMP
    OB100_EV_CLASS :          BYTE;
    OB100_STRTUP   :          BYTE;
    OB100_PRIORITY :          BYTE;
    OB100_OB_NUMBR :          BYTE;
    OB100_RESERVED_1 :       BYTE;
    OB100_RESERVED_2 :       BYTE;
    OB100_STOP     :          WORD;
    OB100_RESERVED_3 :       WORD;
    OB100_RESERVED_4 :       WORD;
    OB100_DATE_TIME :        DATE_AND_TIME;
END_VAR
BEGIN
    CALL FB 1, DB 7(
        MCPNum :=          1,
        MCP1In :=          P#E0.0,
        MCP1Out :=         P#A0.0,
        MCP1StatSend :=    P#A8.0,
        MCP1StatRec :=     P#A12.0,
        MCP1BusAdr :=      6,
        MCP1Timeout :=     S5T#700MS,
        MCP1Cycl :=        S5T#200MS,
        NC-CyclTimeout :=  S5T#200MS,
        NC-RunupTimeout := S5T#50S);
//HIER ANWENDERPROGRAMM EINFÜGEN
END_ORGANIZATION_BLOCK
```

## 12.15.2 FB 2: GET NC-Variable lesen

### Funktion

Mit dem Funktionsbaustein FB 2 "GET" können vom PLC- Anwenderprogramm aus Variablen aus dem Bereich der NC gelesen werden. Der FB ist Multi-Instanzfähig.

Zum FB 2 gehört ein Instanz-DB aus dem Anwenderbereich.

Durch Aufruf des FB 2 mit positivem Flankenwechsel am Steuereingang "Req" wird ein Auftrag gestartet, die durch "Addr1" bis "Addr8" referenzierten NC-Variablen zu lesen und in die durch "RD1" bis "RD8" referenzierten PLC-Operandenbereiche zu kopieren. Der erfolgreiche Abschluss des Lesevorganges wird am Zustandsparameter "NDR" mit logisch "1" angezeigt.

Der **Lesevorgang** erstreckt sich über mehrere PLC-Zyklen (Normalfall: 1 - 2 PLC-Zyklen). Der Baustein kann nur im zyklischen Betrieb aufgerufen werden.

Aufgetretene Fehler werden über "Error" und "State" angezeigt.

Um die NC-Variablen zu referenzieren, werden diese zunächst mit dem Tool "NC-VAR-Selector" ausgewählt und in einen Datenbaustein als AWL-Quelle generiert. Für diesen Datenbaustein muss anschließend in der Symbolliste ein Name vergeben werden. Beim Aufruf des FB 2 werden dann für die Parameter "Addr1" bis "Addr8" die Variablenadresse in folgender Form übergeben:

"<DB-Name>.<S7-Name>".

### Variable Adressierung

Für einige NC-Variable ist es notwendig, im NC-VAR-Selector "Bereichs-Nr." und/oder "Zeile" bzw. "Spalte" auszuwählen. Für diese Variablen ist es möglich, einen Basistyp auszuwählen, d. h. "Bereichs-Nr", "Spalte" und "Zeile" werden mit "0" vorbelegt.

Im FB 2 wird der Inhalt der vom NC-VAR-Selector vorgegebenen "Bereichs-Nr.", "Zeile" und "Spalte" auf "0" geprüft. Liegt "0" vor, wird der Wert vom Eingangsparameter übernommen.

Vor dem Aufruf des FB 2 muss der Anwender die entsprechenden Parameter versorgen:

Paramter: FB 2	Parameter: NC-VAR-Selector
"Unit<x>"	"Bereichs-Nr"
"Column<x>"	"Spalte"
"Line<x>"	"Zeile"
mit <x> = 1 - 8	



<p><b>ACHTUNG</b></p> <p>Der FB 2 kann NC-Variablen nur lesen, wenn der Grundprogramm-Parameter "NCKomm" auf "1" gesetzt wurde (in OB 100: FB 1, DB 7). Der Aufruf ist nur im zyklischen Programm OB 1 zulässig. Eine Zuweisung für alle Parameter mit "Req" = 0 ist auch im OB 100 zulässig.</p> <p><b>Kanalspezifische Variablen</b>                  Beim Lesen von kanalspezifischen Variablen dürfen bei einem Aufruf des FB 2 über "Addr1" bis "Addr8" nur Variablen von genau <b>einem</b> Kanal adressiert werden.</p> <p><b>Antriebsspezifischen Variablen</b>                  Beim Lesen von antriebsspezifischen Variablen dürfen in einem Aufruf des FB 2 über "Addr1" bis "Addr8" nur Variablen von genau <b>einem</b> Antriebsobjekt SERVO adressiert werden. Das Antriebsobjekt SERVO muss einer Maschinenachse der NC zugeordnet sein. Der Zeilenindex entspricht der logischen Antriebsnummer.</p> <p><b>Fehlerfall</b>                  Im Fehlerfall, Lesen von Variablen aus unterschiedlichen Kanälen oder Antriebsobjekten, oder gleichzeitig von einem Kanal und einem Antriebsobjekt, erfolgt folgende Rückmeldung:</p> <ul style="list-style-type: none"> <li>• "Error" == TRUE</li> <li>• "State" == W#16#02</li> </ul>
--

In einem Auftrag können Variablen innerhalb **einer** Gruppe kombiniert werden:

	Bereich				
Gruppe 1	C[1]	N	B	A	T
Gruppe 2	C[2]	N	B	A	T
Gruppe 3	V[.]	H[.]			
Für Kanal 3 bis 10 gelten die gleichen Regeln, wie für Gruppe 1 und Gruppe 2 beispielhaft dargestellt.					

**Hinweis**

Beim gleichzeitigen Lesen mehrerer Variablen vom Typ "String" kann die Anzahl der nutzbaren Variablen kleiner 8 sein.

### Deklaration der Funktion

```
FUNCTION_BLOCK FB 2
VAR_INPUT
    Req :                BOOL;
    NumVar :             INT;
    Addr1 :              ANY;
    Unit1 :              BYTE;
    Column1 :            WORD;
    Line1 :              WORD;
    Addr2 :              ANY;
    Unit2 :              BYTE;
    Column2 :            WORD;
    Line2 :              WORD;
    Addr3 :              ANY;
    Unit3 :              BYTE;
    Column3 :            WORD;
    Line3 :              WORD;
    Addr4 :              ANY;
    Unit4 :              BYTE;
    Column4 :            WORD;
    Line4 :              WORD;
    Addr5 :              ANY;
    Unit5 :              BYTE;
    Column5 :            WORD;
    Line5 :              WORD;
    Addr6 :              ANY;
    Unit6 :              BYTE;
    Column6 :            WORD;
    Line6 :              WORD;
    Addr7 :              ANY;
    Unit7 :              BYTE;
    Column7 :            WORD;
    Line7 :              WORD;
    Addr8 :              ANY;
    Unit8 :              BYTE;
    Column8 :            WORD;
    Line8 :              WORD;
END_VAR
VAR_OUTPUT
    Error :             BOOL;
    NDR :               BOOL;
    State :             WORD;
END_VAR
```

```

VAR_IN_OUT
  RD1 : ANY;
  RD2 : ANY;
  RD3 : ANY;
  RD4 : ANY;
  RD5 : ANY;
  RD6 : ANY;
  RD7 : ANY;
  RD8 : ANY;
END_VAR

```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter des FB 2.

Parameter	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL	-	Auftragsstart mit positiver Flanke
NumVar	E	INT	1 ... 8	Anzahl zu lesender Variablen: Addr1 - Addr8
Addr1 - Addr8	E	ANY	[DBName].[VarName]	Variablenbezeichner aus <b>NC-VAR-Selector</b>
Unit1 - Unit8	E	BYTE	-	Bereichsadresse, optional für variable Adressierung
Column1 - Column8	E	WORD	-	Spaltenadresse, optional für variable Adressierung
Line1 - Line8	E	WORD	-	Zeilenadresse, optional für variable Adressierung
Error	A	BOOL	-	Auftrag wurde negativ quittiert bzw. konnte nicht ausgeführt werden
NDR	A	BOOL	-	Auftrag wurde erfolgreich ausgeführt. Daten stehen zur Verfügung
State	A	WORD	-	siehe Fehlerkennungen
RD1 - RD8	E/A	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	Zielbereich für gelesene Daten

### Fehlerkennungen

Konnte ein Auftrag nicht ausgeführt werden, wird dies am Zustandsparameter "Error" mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang State kodiert:

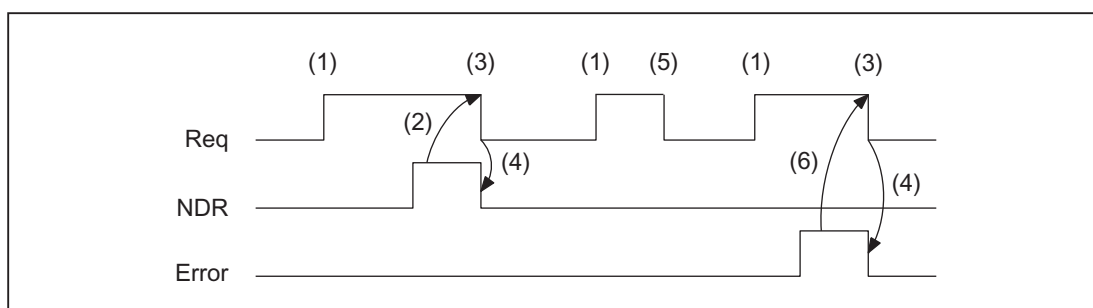
State		Bedeutung	Hinweis
WORT-H	WORT-L		
1 - 8	1	Zugriffsfehler	WORT-H: Nummer der Variablen, bei der der Fehler auftrat
0	2	Fehler im Auftrag	falsche Zusammenstellung von Variablen in einem Auftrag
0	3	negative Quittung, Auftrag nicht ausführbar	interner Fehler, evtl. Abhilfe: <ul style="list-style-type: none"> <li>• Auftrag überprüfen</li> <li>• NC-Reset</li> </ul>
1 - 8	4	nicht ausreichend lokaler Anwenderspeicher zur Verfügung	gelesene Variable ist länger als in "RD1" - "RD8" angegeben; WORT-H: Nummer der Variablen, bei der der Fehler auftrat
0	5	Formatwandlungsfehler	Fehler bei Wandlung vom Var.-Typ double: Variable liegt nicht im Bereich von S7-REAL
0	6	FIFO voll	Auftrag muss wiederholt werden, da die Warteschlange voll ist
0	7	Option nicht gesetzt	GP-Parameter "NCKomm" ist nicht gesetzt
1 - 8	8	falscher Zielbereich (RD)	RD1 bis RD8 dürfen keine Lokaldaten sein
0	9	Übertragung belegt	Auftrag muss wiederholt werden
1 - 8	10	Fehler bei variabler Adressierung	"Unit" oder "Column"/"Line" enthält den Wert 0
0	11	Variablenadresse ungültig	"Addr" (bzw. Variablenname), "Area", "Unit" überprüfen
0	12	NumVar = 0	Parameter NumVar prüfen
1 - 8	13 (0x0d)	Referenz des ANY Datums falsch	Gefordertes NcVar-Datum ist nicht parametrierbar

## Projektierungsschritte

Für das Lesen von NC-Variablen sind folgende Projektierungsschritte nötig:

- Auswahl der Variablen mit dem NC-VAR-Selector
- Speichern der ausgewählten Variablen in einer Datei \*.VAR
- Erzeugen einer STEP 7-Quelldatei \*.AWL
- Erzeugen eines DBs mit den zugehörigen Adressangaben
- Eintrag des Symbols für den generierten DB in die Symboltabelle, damit im Anwenderprogramm symbolisch auf die Adressparameter zugegriffen werden kann
- Parametrierung des FB 2

## Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positive Quittung: Neue Daten empfangen
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) nicht zulässig
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter State

## Aufrufbeispiel

Lesen von drei kanalspezifischen Maschinendaten von Kanal 1, deren Adressangaben im DB 120 hinterlegt werden.

**Auswahl der Daten mit NC-VAR-Selector** und Speicherung in der Datei DB120.VAR; anschließend erzeugen der Datei DB120.AWL:

Bereich	Baustein	Name	Typ	Nr.	Byte	S7-Name
C[1]	M	MD20070 \$MC_AXCONF_MACHAX_USED[1]	CHAR	20070	1	C1AxConfMachAx Used1
C[1]	M	MD20070 \$MC_AXCONF_MACHAX_USED[2]	CHAR	20070	1	C1AxConfMachAx Used2
C[1]	M	MD20090 \$MC_SPIND_DEF_MASTER_SPIND	INT	20090	1	C1SpindDefMaster Spind

Es wurden S7-(ALIAS-)Namen gewählt, um:

- die Kanal-Bezeichnung in den Namen aufzunehmen.  
und
- die Zeichen [ ] zu entfernen, die in einem STEP 7-Symbol nicht zulässig sind.

Eintrag des Namens in die S7-SYMBOL-Tabelle (z. B. NCVAR für DB 120):

Symbol	Operand	Datentyp
NCVAR	DB 120	DB 120

Die Datei DB120.AWL muss kompiliert und in die PLC übertragen werden.

Parametrierung des FB 2 mit Instanz-DB 110:

```

DATA_BLOCK DB 110           //freier Anwender-DB, als Instanz für FB 2
FB 2
BEGIN
END_DATA_BLOCK
Function FC "VariablenCall" : VOID
    U      E 7.7;           //freie Taste Maschinensteuertafel
    S      M 100.0;        //Req aktivieren
    U      M 100.1;        //Fertigmeldung NDR
    R      M 100.0;        //Auftrag beenden
    U      E 7.6;           //Fehlerquittierung von Hand
    U      M 102.0;        //Fehler steht an
    R      M 100.0;        //Auftrag beenden
    CALL FB 2, DB 110(
        Req :=             M 100.0,
        NumVar :=          3,           //3 Variablen lesen
        Addr1 :=           NCVAR.C1AxConfMachAxUsed1,
        Addr2 :=           NCVAR.C1AxConfMachAxUsed2,
        Addr3 :=           NCVAR.C1SpindDefMasterSpind,
        Error :=           M102.0,
        NDR :=             M100.1,
        State :=           MW104,
        RD1 :=             P#DB99.DBX0.0 BYTE 1,
        RD2 :=             P#DB99.DBX1.0 BYTE 1,
        RD3 :=             P#M110.0 INT 1);
    
```

### Beispiel: Variable Adressierung

Lesen von zwei R-Parametern aus Kanal 1, deren Adressangaben im DB 120 als Basistyp hinterlegt werden. Die R-Parameter-Nummer wird über den Parameter Line<x> parametrieret.

```
DATA_BLOCK DB 120
VERSION : 0.0
STRUCT
  C1_RP_rpa0_0 :
  STRUCT
  SYNTAX_ID :          BYTE := B#16#82;
  bereich_u_einheit :  BYTE := B#16#41;
  spalte :             WORD := W#16#1;
  zeile :              WORD := W#16#0;
  bausteintyp :        BYTE := B#16#15;
  ZEILENZAHL :         BYTE := B#16#1;
  typ :                BYTE := B#16#F;
  laenge :             BYTE := B#16#8;
  END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
CALL FB 2 , DB 110 (
  Req :=                M 0.0,
  NumVar :=            2,
  Addr1 :=             "NCVAR".C1_RP_rpa0_0,
  Line1 :=             W#16#1,
  Addr2 :=             "NCVAR".C1_RP_rpa0_0,
  Line2 :=             W#16#2,
  Error :=             M 1.0,
  NDR :=              M 1.1,
  State :=            MW 2,
  RD1 :=              P#M 4.0 REAL 1,
  RD2 :=              P#M 24.0 REAL 1);
```

## Datentypen

Im NC-VAR-Selector werden die Datentypen der NCK bei den Variablen aufgeführt. In der folgenden Tabelle sind die Zuordnungen zu S7-Datentypen angegeben.

Zuordnung der Datentypen	
NCK-Datentyp	S7-Datentyp
double	REAL
double	REAL2
float	REAL
long	DINT
integer	DINT
uint_32	DWORD
int_16	INT
uint_16	WORD
unsigned	WORD
char	CHAR oder BYTE
string	STRING
bool	BOOL
datetime	DATE_AND_TIME

Um eine Variable vom Typ "double" vom NCK ohne Formatanpassung zu lesen, muss im Zielbereich für gelesene Daten ein ANY-Pointer mit Typ REAL Anzahl 2 angegeben werden (z. B.: P#M100.0 REAL 2). Erkennt das Grundprogramm beim Lesen einer Variable vom Typ "double" als Zieltyp REAL 2, werden die Daten als 64-Bit Gleitkommazahl in den PLC-Datenbereich übernommen.

### 12.15.3 FB 3: PUT NC-Variable schreiben

#### Funktion

Mit dem Funktionsbaustein FB 3 "PUT" können vom PLC-Anwenderprogramm aus Variablen im Bereich der NC geschrieben werden. Der FB ist Multi-Instanzfähig

Zu jedem FB 3-Aufruf muss ein separater Instanz-DB aus dem Anwenderbereich zugeordnet werden.

Durch Aufruf des FB 3 mit positivem Flankenwechsel am Steuereingang "Req" wird ein Auftrag gestartet, um die durch "Addr1" bis "Addr8" referenzierten NC-Variablen mit den Daten der lokal durch "SD1" bis "SD8" referenzierten PLC-Operandenbereiche zu überschreiben. Der erfolgreiche Abschluss des Schreibvorgangs wird am Zustandsparameter "Done" mit logisch "1" angezeigt.



Der **Schreibvorgang** erstreckt sich über mehrere PLC-Zyklen (Normalfall: 1 - 2 PLC-Zyklen). Der Baustein kann nur im zyklischen Betrieb aufgerufen werden.

Aufgetretene Fehler werden über "Error" und "State" angezeigt.

Um die NC-Variablen zu referenzieren, werden zunächst alle benötigten Variablen mit dem Tool "NC-VAR-Selector" ausgewählt und in einem Datenbaustein als AWL-Quelle generiert. Für diesen DB muss dann in der Symbolliste ein Name vergeben werden. Beim Aufruf des FB 3 werden dann für die Parameter "Addr1" bis "Addr8" die Variablenadresse in folgender Form übergeben:

"<DB-Name>.<S7-Name>".

### Variable Adressierung

Für einige NC-Variable ist es notwendig, im NC-VAR-Selector "Bereichs-Nr" und/oder "Zeile" bzw. "Spalte" auszuwählen. Für diese Variablen ist es möglich, einen Basistyp auszuwählen, d. h. "Bereichs-Nr", "Spalte" und "Zeile" werden mit "0" vorbelegt.

Im FB 3 wird der Inhalt der vom NC-VAR-Selector vorgegebenen "Bereichs-Nr", "Zeile" und "Spalte" auf "0" geprüft. Liegt "0" vor, wird der Wert vom Eingangsparameter übernommen.

Vor dem Aufruf des FB 3 muss der Anwender die entsprechenden Parameter versorgen:

Parameter: FB 3	Parameter: NC-VAR-Selector
"Unit<x>"	"Bereichs-Nr"
"Column<x>"	"Spalte"
"Line<x>"	"Zeile"
mit <x> = 1 - 8	

### Maschinendaten, GUD

Um Maschinendaten und GUD ohne Passwort beschreiben zu können, muss die Schutzstufe des jeweiligen Datums auf die niedrigste Stufe umdefiniert werden.

**Literatur:**

- Inbetriebnahmehandbuch; Kapitel: "Schutzstufenkonzept"
- Programmierhandbuch Arbeitsvorbereitung; Kapitel: "Schutzstufen für Anwenderdaten definieren"

**ACHTUNG**

Der FB 3 kann NC-Variablen nur schreiben, wenn der Grundprogramm-Parameter "NCKomm" auf "1" gesetzt wurde (in OB 100: FB 1, DB 7). Der Aufruf ist nur im zyklischen Programm OB 1 zulässig. Eine Zuweisung für alle Parameter mit "Req" = 0 ist auch im OB 100 zulässig.

**Kanalspezifische Variablen**  
 Beim Schreiben von kanalspezifischen Variablen dürfen bei einem Aufruf des FB 3 über "Addr1" bis "Addr8" nur Variablen von genau **einem** Kanal adressiert werden.

**Antriebsspezifischen Variablen**  
 Beim Schreiben von antriebsspezifischen Variablen dürfen in einem Aufruf des FB 3 über "Addr1" bis "Addr8" nur Variablen von genau **einem** Antriebsobjekt SERVO adressiert werden. Das Antriebsobjekt SERVO muss einer Maschinenachse der NC zugeordnet sein. Der Zeilenindex entspricht der logischen Antriebsnummer.

**Fehlerfall**  
 Im Fehlerfall, Schreiben von Variablen aus unterschiedlichen Kanälen oder Antriebsobjekten, oder gleichzeitig von einem Kanal und einem Antriebsobjekt, erfolgt folgende Rückmeldung:

- "Error" == TRUE
- "State" == W#16#02

In einem Auftrag können NC-Variablen innerhalb **einer** Gruppe kombiniert werden:

	Bereich				
Gruppe 1	C[1]	N	B	A	T
Gruppe 2	C[2]	N	B	A	T
Gruppe 3	V[.]	H[.]			

Für Kanal 3 bis 10 gelten die gleichen Regeln, wie für Gruppe 1 und Gruppe 2 beispielhaft dargestellt.

**Hinweis**

Beim gleichzeitigen Schreiben mehrerer Variablen vom Typ "String" kann die Anzahl der nutzbaren Variablen kleiner 8 sein.

## Deklaration der Funktion

```
FUNCTION_BLOCK FB 3
VAR_INPUT
    Req :                BOOL;
    NumVar :              INT;
    Addr1 :               ANY;
    Unit1 :               BYTE;
    Column1 :             WORD;
    Line1 :               WORD;
    Addr2 :               ANY;
    Unit2 :               BYTE;
    Column2 :             WORD;
    Line2 :               WORD;
    Addr3 :               ANY;
    Unit3 :               BYTE;
    Column3 :             WORD;
    Line3 :               WORD ;
    Addr4 :               ANY;
    Unit4 :               BYTE;
    Column4 :             WORD;
    Line4 :               WORD;
    Addr5 :               ANY;
    Unit5 :               BYTE;
    Column5 :             WORD;
    Line5 :               WORD;
    Addr6 :               ANY;
    Unit6 :               BYTE;
    Column6 :             WORD;
    Line6 :               WORD;
    Addr7 :               ANY;
    Unit7 :               BYTE;
    Column7 :             WORD;
    Line7 :               WORD;
    Addr8 :               ANY;
    Unit8 :               BYTE;
    Column8 :             WORD;
    Line8 :               WORD;
END_VAR
VAR_OUTPUT
    Error :              BOOL;
    Done :                BOOL;
    State :               WORD;
END_VAR
```

```

VAR_IN_OUT
  SD1 : ANY ;
  SD2 : ANY ;
  SD3 : ANY ;
  SD4 : ANY ;
  SD5 : ANY ;
  SD6 : ANY ;
  SD7 : ANY ;
  SD8 : ANY ;
END_VAR
    
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion PUT.

Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL	-	Auftragsstart mit positiver Flanke
NumVar	E	INT	1 ... 8	Anzahl zu schreibender Variablen: Addr1 - Addr8
Addr1 - Addr8	E	ANY	[DBName].[VarName]	Variablenbezeichner aus <b>NC-VAR-Selector</b>
Unit 1 - Unit 8	E	BYTE	-	Bereichsadresse, optional für variable Adressierung
Column 1 - Column 8	E	WORD	-	Spaltenadresse, optional für variable Adressierung
Line 1 - Line 8	E	WORD	-	Zeilenadresse, optional für variable Adressierung
Error	A	BOOL	-	Auftrag wurde negativ quittiert, bzw. konnte nicht ausgeführt werden
Done	A	BOOL	-	Auftrag wurde erfolgreich ausgeführt.
State	A	WORD	-	siehe Fehlerkennungen
SD1 - SD8	E/A	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	zu schreibende Daten

## Fehlerkennungen

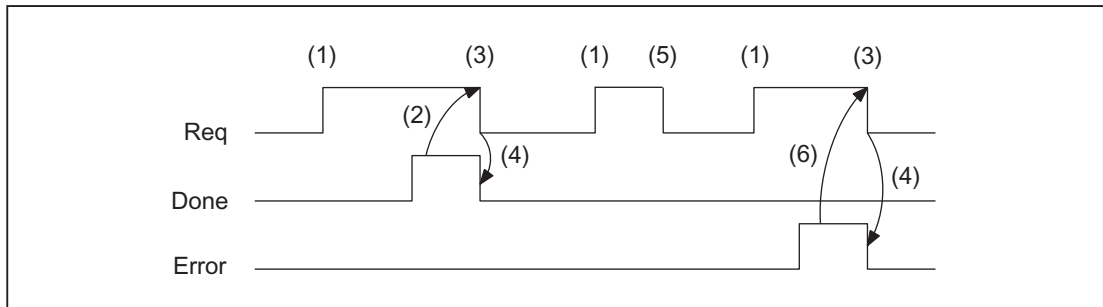
Konnte ein Auftrag nicht ausgeführt werden, wird dies am Zustandsparameter Error mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang State kodiert:

State		Bedeutung	Hinweis
WORT-H	WORT-L		
1 - 8	1	Zugriffsfehler	WORT-H: Nummer der Variablen, bei der der Fehler auftrat
0	2	Fehler im Auftrag	falsche Zusammenstellung von Variablen in einem Auftrag
0	3	negative Quittung, Auftrag nicht ausführbar	interner Fehler, evtl. Abhilfe: <ul style="list-style-type: none"> <li>• Auftrag überprüfen</li> <li>• NC-Reset</li> </ul>
1 - 8	4	Datenbereiche oder Datentypen stimmen nicht überein oder String ist leer	zu schreibende Daten in "SD1" - "SD8" überprüfen; WORT-H: Nummer der Variablen, bei der der Fehler auftrat
0	6	FIFO voll	Auftrag muss wiederholt werden, da die Warteschlange voll ist
0	7	Option nicht gesetzt	GP-Parameter "NCKomm" ist nicht gesetzt
1 - 8	8	falscher Zielbereich (SD)	"SD1" - "SD8" dürfen keine Lokaldaten sein
0	9	Übertragung belegt	Auftrag muss wiederholt werden
1 - 8	10	Fehler bei variabler Adressierung	"Unit" oder "Column"/ "Line" enthält den Wert 0
0	11	Variablenadresse ungültig oder Variable nur lesbar	"Addr" (bzw. Variablenname), "Area", "Unit" überprüfen
0	12	NumVar = 0	Parameter NumVar prüfen
1 - 8	13 (0x0d)	Referenz des ANY-Datums falsch	Gefordertes NcVar-Datum ist nicht parametrisiert
1 - 8	15 (0x0f)	Nutzdaten zu lang	Abhilfe: Weniger Variable pro Auftrag schreiben oder kürzere Stringvariablen verwenden

## Projektierungsschritte

Für das Schreiben von NC-Variablen sind die gleichen Projektierungsschritte wie für das Lesen von NC-Variablen notwendig. Es ist zweckmäßig, die Adressangaben aller NC-Variablen, die gelesen oder geschrieben werden sollen, in einem DB zu hinterlegen.

### Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positive Quittung: Variablen wurden geschrieben
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) nicht zulässig
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter State

### Aufrufbeispiel

Schreiben von drei kanalspezifischen Maschinendaten von Kanal 1:

**Auswahl der drei Daten mit NC-VAR-Selector und Speicherung in der Datei DB120.VAR:**

Bereich	Baustein	Name	Typ	Byte	S7-Name
C[1]	RP	rpa[5]	DOUBLE	4	rpa_5C1RP
C[1]	RP	rpa[11]	DOUBLE	4	rpa_11C1RP
C[1]	RP	rpa[14]	DOUBLE	4	rpa_14C1RP

**Eintrag NCVAR für DB 120 mit dem S7-SYMBOL-Editor:**

Symbol	Operand	Datentyp
NCVAR	DB 120	DB 120

Die Datei DB120.AWL muss kompiliert und in die PLC übertragen werden.

### Aufruf und Parametrierung des FB 3 mit dem Instanz-DB 111:

```
DATA_BLOCK DB 111          //freier Anwender DB, als Instanz für FB 3
FB 3
BEGIN
Function FC "VariablenCall" : VOID
END_DATA_BLOCK
    U      E 7.7;           //freie Taste Maschinensteuertafel
    S      M 100.0;        //Req aktivieren
    U      M 100.1;        //Fertigmeldung Done
    R      M 100.0;        //Auftrag beenden
    U      E 7.6;           //Fehlerquittierung von Hand
    U      M 102.0;        //Fehler steht an
    R      M 100.0;        //Auftrag beenden
CALL FB 3, DB 111(
    Req :=      M 100.0,
    NumVar :=   3,                //3 Variablen schreiben
    Addr1 :=   NCVAR.rpa_5C1RP,
    Addr2 :=   NCVAR.rpa_11C1RP,
    Addr3 :=   NCVAR.rpa_14C1RP,
    Error :=   M102.0,
    Done :=    M100.1,
    State :=   MW104,
    SD1 :=     P#DB99.DBX0.0 REAL 1,
    SD2 :=     P#DB99.DBX4.0 REAL 1,
    SD3 :=     P#M110.0 REAL 1);
```

### Beispiel: Variable Adressierung

Schreiben von zwei R-Parametern von Kanal 1, deren Adressangaben im DB 120 als Basistyp hinterlegt werden. Die R-Parameter-Nummer wird über den Parameter LineX parametrieret.

```
DATA_BLOCK DB 120
VERSION : 0.0
STRUCT
  C1_RP_rpa0_0:
  STRUCT
  SYNTAX_ID :          BYTE := B#16#82;
  bereich_u_einheit :  BYTE := B#16#41;
  spalte :            WORD := W#16#1;
  zeile :             WORD := W#16#0;
  bausteintyp :       BYTE := B#16#15;
  ZEILENANZAHL :      BYTE := B#16#1;
  typ :               BYTE := B#16#F;
  laenge :            BYTE := B#16#8;
  END_STRUCT ;
END_STRUCT ;
BEGIN
END_DATA_BLOCK
CALL FB 3 , DB 122 (
  Req :=                M 10.0,
  NumVar :=             2,
  Addr1 :=              "NCVAR".C1_RP_rpa0_0,
  Line1 :=              W#16#1,
  Addr2 :=              "NCVAR".C1_RP_rpa0_0,
  Line3 :=              W#16#2,
  Error :=              M 11.0,
  Done :=               M 11.1,
  State :=              MW 12,
  SD1 :=                P#M 4.0 REAL 1,
  SD2 :=                P#M 24.0 REAL 1);
```



## 12.15.4 PI-Dienste

### 12.15.4.1 FB 4: PI\_SERV PI-Dienst anfordern

#### Funktion

Mit dem Funktionsbaustein FB 4 "PI\_SERV" können Programminstanz-Dienste (PI-Dienste) in der NC gestartet werden. Jedem FB 4-Aufruf muss dabei ein Instanz-DB aus dem Anwender-Bereich zugeordnet werden.

---

#### Hinweis

Es wird empfohlen, anstelle des FB 4 den erweiterten Funktionsbaustein FB 7 zu verwenden. Siehe Kapitel "FB 7: PI\_SERV2 (PI-Dienst anfordern) (Seite 1065)".

---

Über den Parameter "PIService" ist der gewünschte PI-Dienst zu referenzieren. Über die Parameter Addr und WVar erfolgt die Parametrierung des angewählten PI-Dienstes durchgeführt.

Durch Aufruf des FB 4 mit positivem Flankenwechsel am Steuereingang "Req" wird der Auftrag gestartet. Die erfolgreiche Durchführung wird am Ausgang "Done" mit logisch 1 angezeigt. Eventuell aufgetretene Fehler werden über die Ausgänge "Error" und "State" angezeigt.

Der Datenbaustein DB 16 "PI" enthält interne Beschreibungen der möglichen PI-Dienste. Für diesen Datenbaustein muss in der Symbolliste ein Name vergeben werden. Als Aktualparameter für "PIService" wird "DB-Name.PI-Name" beim Aufruf des FB 4 übergeben.

Die Ausführung eines PI-Dienstes erstreckt sich in der Regel über mehrere PLC-Zyklen.

#### Voraussetzungen

Der FB 4 kann PI-Dienste nur starten, wenn der Grundprogramm-Parameter NCKomm = 1 gesetzt wurde (OB 100: FB 1, DB 7).

#### Randbedingungen

Der Aufruf ist nur im zyklischen Programm OB 1 zulässig. Ohne Start der Anforderung (Req = 0) ist das Schreiben der Parameter auch im OB 100 zulässig.

## Deklaration der Funktion

```
FUNCTION_BLOCK FB 4
VAR_INPUT
w   Req :          BOOL;
    PIService :    ANY;
    Unit :         INT;
    Addr1 :       ANY;
    Addr2 :       ANY;
    Addr3 :       ANY;
    Addr4 :       ANY;
    WVar1 :       WORD;
    WVar2 :       WORD;
    WVar3 :       WORD;
    WVar4 :       WORD;
    WVar5 :       WORD;
    WVar6 :       WORD;
    WVar7 :       WORD;
    WVar8 :       WORD;
    WVar9 :       WORD;
    WVar10 :      WORD;
END_VAR
VAR_OUTPUT
    Error :        BOOL;
    Done :         BOOL;
    State :        WORD;
END_VAR
```

## Erläuterung der Formalparameter

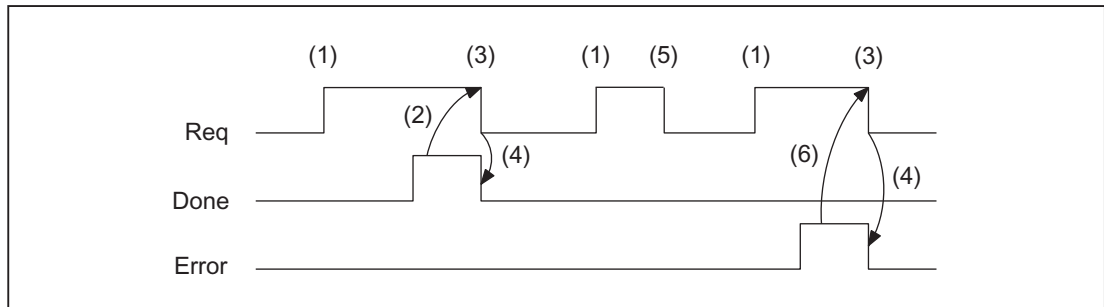
Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL		Auftragsanforderung
PIService	E	ANY	[DBName].[VarName] Standard: "PI".[VarName]	Bezeichnung des PI-Dienstes, siehe Kapitel "Liste verfügbarer PI-Dienste (Seite 1028)"
Unit	E	INT	1...	Bereichsnummer
Addr1 bis Addr4	E	ANY	[DBName].[VarName]	Referenz auf Strings Spezifikation gemäß angewähltem PI-Dienst
WVar1 bis WVar10	E	WORD	1...	INTEGER- oder WORD-Variablen. Spezifikation gemäß angewähltem PI-Dienst
Error	A	BOOL	TRUE/FALSE	Auftrag wurde negativ quittiert bzw. konnte nicht ausgeführt werden
Done	A	BOOL	TRUE/FALSE	Auftrag wurde erfolgreich ausgeführt
State	A	WORD	siehe Fehlerkennungen	-
E: Eingang A: Ausgang				

## Fehlerkennungen

Konnte ein Auftrag nicht ausgeführt werden, wird dies am Ausgang "Error" mit "logisch 1" angezeigt. Die Fehlerursache wird am Ausgang "State" angezeigt:

State	Bedeutung	Hinweis
3	negative Quittung, Auftrag nicht ausführbar	Interner Fehler, mögliche Abhilfe durch NC-RESET
6	FIFO voll	Auftrag wiederholen, Warteschlange ist voll
7	Option nicht gesetzt	FB 1, Parameter "NCKomm" ist nicht gesetzt
9	Übertragung belegt	Auftrag wiederholen
13 (0x0d)	Addr1.. Addr4: Referenz ungültig	Fehlenden String angeben
14 (0x0e)	"PIService": Referenz unbekannt	Keine gültige PI-Bezeichnung
15 (0x0f)	Addr1.. Addr4: String zu lang	Stringlängen überprüfen

**Ablaufdiagramm**



- (1) Funktionsanstoß
- (2) Positive Quittung: PI-Dienst wurde ausgeführt
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) nicht zulässig
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter State

**12.15.4.2 Liste verfügbarer PI-Dienste**

Die folgenden PI-Dienste sind von der PLC aus startbar.

**Allgemeine PI-Dienste**

PI-Dienst	Funktion
ASUP (Seite 1029)	Interrupt zuordnen
CANCEL (Seite 1030)	Cancel durchführen
CONFIG (Seite 1031)	Umkonfiguration von gekennzeichneten Maschinendaten
DIGION (Seite 1031)	Digitalisieren ein
DIGIOF (Seite 1031)	Digitalisieren aus
FINDBL (Seite 1032)	Suchlauf aktivieren
LOGIN (Seite 1032)	Kennwort aktivieren
LOGOUT (Seite 1033)	Kennwort rücksetzen
NCRES (Seite 1033)	NC-RESET auslösen
SELECT (Seite 1033)	Programm zur Abarbeitung für einen Kanal anwählen
SETUDT (Seite 1034)	Setzt die aktuellen User Daten aktiv
SETUFR (Seite 1034)	Anwender Frame aktivieren
RETRAC (Seite 1035)	Freifahren des Werkzeugs in Werkzeugrichtung

**PI-Dienste der Werkzeugverwaltung**

PI-Dienst	Funktion
CRCEDN (Seite 1035)	Lege neue Schneide an
CREACE (Seite 1036)	Werkzeugschneide anlegen
CREATO (Seite 1037)	Werkzeug erzeugen
DELECE (Seite 1037)	Löschen einer Werkzeugschneide
DELETO (Seite 1038)	Werkzeug löschen
MMCSEM (Seite 1038)	Semaphoren für verschiedene PI-Dienste
TMCRT0 (Seite 1039)	Werkzeug anlegen
TMFDPL (Seite 1040)	Leerplatzsuche zum Beladen
TMFPBP (Seite 1042)	Leerplatzsuche
TMGETT (Seite 1043)	T-Nummer zum vorgegebenen Werkzeugbezeichner mit Duplonummer
TMMVTL (Seite 1044)	Magazinplatz zum Beladen bereitstellen, Werkzeug entladen
TMPOSM (Seite 1046)	Magazinplatz oder Werkzeug positionieren
TMPCIT (Seite 1047)	Inkrementwert für Stückzahlzähler setzen
TMRASS (Seite 1047)	Rücksetzen des Aktiv-Status
TRESMO (Seite 1048)	Rücksetzen der Überwachungswerte
TSEARC (Seite 1049)	Komplexes Suchen über Suchmasken
TMCRMT (Seite 1052)	Multitool erzeugen
TMDLMT (Seite 1053)	Multitool löschen
POSMT (Seite 1053)	Multitool positionieren
FDPLMT (Seite 1054)	Leerplatz innerhalb des Multitools suchen/prüfen

**12.15.4.3 PI-Dienst: ASUP****Funktion: Interrupt zuordnen**

Ein auf der NC abgelegtes Programm wird einem Interrupt-Signal eines Kanals zugeordnet. Das Programm muss ausführbar sein und der Pfad- und Programmname muss vollständig und korrekter angegeben sein. Ausführliche Informationen dazu finden sich in:

**Literatur**

Programmierhandbuch Arbeitsvorbereitung; Kapitel: "Datei- und Programmverwaltung" > "Programmspeicher"

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.ASUP	Interrupt zuordnen <sup>1)</sup>
Unit	INT	1 ... 10	Kanal
Addr1	STRING		Pfadname
Addr2	STRING		Programmname
WVar1	WORD	1 ... 8	Interrupt-Nummer
WVar2	WORD	1 ... 8	Priorität
WVar3	WORD	0/1	LIFTFAST <sup>2)</sup>
WVar4	WORD	0/1	BLSYNC <sup>3)</sup>
1) Alternativ kann für die Interrupt-Zuordnung der Befehl SETINT verwendet werden. Siehe 2) 2) <b>Literatur</b> Programmierhandbuch Arbeitsvorbereitung, Kapitel: "Flexible NC-Programmierung" > "Interruptroutine (ASUP)" > "Schnellabheben von der Kontur (SETINT, LIFTFAST, ALF)" 3) <b>Literatur</b> Programmierhandbuch Arbeitsvorbereitung, Kapitel: "Flexible NC-Programmierung" > "Interruptroutine (ASUP)" > "Interruptroutine zuordnen und starten (SETINT, PRIO, BLSYNC)"			

**Hinweis**

Der PI-Dienst ASUP darf nur im RESET-Zustand des angegebenen Kanals ausgeführt werden.

**Literatur**

Programmierhandbuch Arbeitsvorbereitung, Kapitel: "Flexible NC-Programmierung" > "Interruptroutine (ASUP)".

**12.15.4.4 PI-Dienst: CANCEL**

**Funktion: Cancel durchführen**

Löst die Funktion "Cancel" gleichwertig der entsprechenden "Alarm Cancel"-Taste der Bedienoberfläche (Bedientafelfront) aus.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CANCEL	Cancel

**12.15.4.5 PI-Dienst: CONFIG****Funktion: Umkonfiguration**

Bewirkt, dass Maschinendaten, die sequentiell vom Bediener oder auch PLC eingegeben wurden, quasi parallel aktiviert werden.

Das Kommando kann nur im RESET-Zustand der Steuerung bzw. bei Programmunterbrechung (NC-Stop an Satzgrenze) aktiviert werden. Bei Nichteinhaltung dieser Bedingung erfolgt eine Fehlerrückmeldung des FB 4 (State = 3).

**Parametrierung**

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CONFIG	Umkonfiguration
Unit	INT	1	
WVar1	INT	1	Klassifizierung

**12.15.4.6 PI-Dienst: DIGION****Funktion: Digitalisieren ein**

Digitalisieren im parametrierten Kanal anwählen.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DIGION	Digitalisieren ein
Unit	INT	1 bis 10	Kanal

**12.15.4.7 PI-Dienst: DIGIOF****Funktion: Digitalisieren aus**

Digitalisieren im parametrierten Kanal ausschalten.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DIGIOF	Digitalisieren aus
Unit	INT	1 bis 10	Kanal

**12.15.4.8 PI-Dienst: FINDBL**

**Funktion: Suchlauf aktivieren**

Ein Kanal wird auf Suchlaufmodus geschaltet und daraufhin die Quittung gesendet. Der Suchlauf wird danach von der NC sofort ausgeführt. Der Suchlaufzeiger muss sich zu diesem Zeitpunkt bereits in der NC befinden. Der Suchlauf kann jederzeit durch NC-RESET abgebrochen werden. Ist der Suchlauf abgeschlossen, wird automatisch wieder der normale Abarbeitungsmodus aktiviert. NC-Start wirkt dann ab dem gefundenen Suchziel.

Es liegt ausschließlich in der Verantwortung des Bedieners, einen kollisionsfreien Anfahrweg sicherzustellen.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung	
PIService	ANY	PI.FINDBL	Suchlauf	
Unit	INT	1 bis 10	Kanal	
WVar1	WORD	1, 2, 3	Vorlaufmodus:	
			1	ohne Berechnung
			2	mit Berechnung
			3	mit Hauptsatzbetrachtung

**12.15.4.9 PI-Dienst: LOGIN**

**Funktion: Kennwort anlegen**

Übergibt das parametrierte Kennwort an die NC. Das Kennwort besteht generell aus 8 Zeichen. Gegebenenfalls müssen in dem String des Kennworts Leerzeichen ergänzt werden.

**Beispiel**

Kennwort : STRING[8] := 'SUNRISE';

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.LOGIN	Kennwort anlegen
Unit	INT	1	NC
Addr1	STRING	8 Zeichen	Kennwort



#### 12.15.4.10 PI-Dienst: LOGOUT

**Funktion: Kennwort rücksetzen**

Das zuletzt an die NC übergebene Kennwort wird zurückgesetzt.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.LOGOUT	Kennwort rücksetzen
Unit	INT	1	NC

#### 12.15.4.11 PI-Dienst: NCRES

**Funktion: NC-RESET auslösen**

Löst NC-RESET aus. Parameter "Unit" und "WVar1" sind immer mit 0 zu versorgen.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.NCRES	NC-RESET auslösen
Unit	INT	0	-
WVar1	WORD	0	-

#### 12.15.4.12 PI-Dienst: SELECT

**Funktion: Abarbeitung für einen Kanal anwählen**

Ein auf der NC abgelegtes Programm wird für einen Kanal zur Abarbeitung angewählt. Dies ist nur möglich, wenn das File ausgeführt werden darf. Die Pfadnamen und Programmnamen sind in korrekter Schreibweise einzugeben. Ausführliche Informationen dazu finden sich in:

**Literatur**

Programmierhandbuch Arbeitsvorbereitung; Kapitel: "Datei- und Programmverwaltung" > "Programmspeicher"

**Mögliche Bausteintypen**

Bausteintypen	
Werkstückverzeichnis	WPD
Hauptprogramm	MPF
Unterprogramm	SPF
Zyklen	CYC
Asynchrone Unterprogramme	ASP
Binärfiles	BIN

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.SELECT	Programm-Anwahl
Unit	INT	1 ... 10	Kanal
Addr1	STRING		Pfad-Name
Addr2	STRING		Programm-Name

**12.15.4.13 PI-Dienst: SETUdT**

**Funktion: Aktuelle User-Daten aktiv setzen**

Die aktuellen User-Daten wie Werkzeugkorrekturen, Basisframes und einstellbare Frames werden nur im Stop-Zustand zum nächsten NC-Satz gesetzt.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung	
PIService	ANY	PI.SETUdT	User Daten aktivieren	
Unit	INT	1 bis 10	Kanal	
WVar1	WORD	1 bis 5	User Data Type:	
			1	aktive Werkzeug-Korrektur
			2	aktiver Basis-Frame
			3	aktiver einstellbarer Frame
			4	aktiver globaler Basis-Frame
5	aktiver globaler einstellbarer Frame			
WVar2	WORD	0	Reserve	
Wvar3	WORD	0	Reserve	

**12.15.4.14 PI-Dienst: SETUFR**

**Funktion: Anwender-Frames aktivieren**

User-Frames werden in die NC geladen. Alle notwendigen Werte der Frames müssen vorher mit FB 3 "Variablen schreiben" an die NC übermittelt worden sein.

**Parametrierung**

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.SETUFR	User-Frames aktivieren
Unit	INT	1bis 10	Kanal

### 12.15.4.15 PI-Dienst: RETRAC

#### Funktion: JOG-Retract anwählen

Wählt die Betriebsart JOG-Retract an. Die Rückzugsachse bzw. die Geometrieachse des WKS mit welcher der Rückzug ausgeführt wird, kann von der NC bestimmt oder explizit angegeben werden.

#### Hinweis

Der PI-Dienst kann nur im Zustand "Reset" in der Betriebsart JOG aktiviert werden.

#### Parametrierung

Parametrierung				
Signal	Typ	Wertebereich	Bedeutung	
PIService	ANY	PI.RETRAC	Betriebsart JOG-Retract anwählen	
Unit	INT	1bis 10	Kanal	
WVar1	INT	0 bis 3	Rückzugachse:	
			0	Ermittlung der Rückzugachse durch die NC. Fall 1: JOG-Retract war schon einmal angewählt, aber noch nicht abgeschlossen ⇒ Die zuletzt angewählte Rückzugsachse wird wieder angewählt. Fall 2: JOG-Retract wird erstmalig angewählt ⇒ Die Rückzugsachse wird von der NC ermittelt: - Standard: 3. Geometrieachse - Schleif- und Drehwerkzeuge: 1. Geometrieachse <b>Hinweis:</b> Die aktive Rückzugsachse ist über die BTSS-Variable retractState.Bit 2/3 lesbar
			1	Rückzugachse ist 1. Geometrieachse des WKS
			2	Rückzugachse ist 2. Geometrieachse des WKS
			3	Rückzugachse ist 3. Geometrieachse des WKS
WVar2	INT	0	Reserviert. Der Wert ist mit 0 vorzubelegen.	

### 12.15.4.16 PI-Dienst: CRCEDN

#### Funktion: Legt neue Schneide an

Wird in dem PI-Dienst unter dem Parameter "T-Nummer" die T-Nummer eines existierenden Werkzeugs angegeben, wird die Schneide zu diesem Werkzeug angelegt (in diesem Fall hat der Parameter "D-Nummer" - die Nummer der anzulegenden Schneide - einen Wertebereich von 00001 - 00009.

Ist eine positive T-Nummer als Parameter angegeben und das Werkzeug zu der angegebenen T-Nummer existiert nicht, schlägt der PI-Dienst fehl.

Wird für die T-Nummer der Wert 00000 angegeben (Modell der absoluten D-Nummern), kann sich der Wertebereich der D-Nummer von 00001 - 31999 erstrecken. Die neue Schneide wird mit der vorgegebenen D-Nummer erzeugt.

Existiert die angegebene Schneide schon, schlägt der PI-Dienst in beiden Fällen fehl.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CRCEDN	Lege neue Schneide an
Unit	INT	1 ... 10	TOA
WVar1	INT		T-Nummer des Werkzeuges, zu dem die Werkzeugschneide angelegt werden soll. Ein Wert 00000 besagt, dass kein Bezug zu einem Werkzeug existieren soll (absolute D-Nummer)
WVar2	INT	1 ... 9 bzw. 01 - 31999	Schneiden-Nummer der Werkzeugschneide

**12.15.4.17 PI-Dienst: CREACE**

**Funktion: Werkzeugschneide anlegen**

Erzeugen der Schneide mit der nächst höheren/freien D-Nummer zu dem Werkzeug mit der übergebenen T-Nummer in TO, TS (sofern vorhanden), im TUE-Baustein (sofern vorhanden) wird die Schneide für die OEM-Schneidendaten mit angelegt.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CREACE	Werkzeugschneide anlegen
Unit	INT	1 ... 10	TOA
WVar1	INT		T-Nummer

**12.15.4.18 PI-Dienst: CREATO****Funktion: Werkzeug anlegen**

Anlegen eines Werkzeuges unter Vorgabe einer T-Nummer. Das Werkzeug wird im Bereich TV (Werkzeug-Directory) als vorhanden eingetragen, im TO-Baustein wird die erste Schneide "D1" (mit Null-Inhalt) für die Korrekturen angelegt, im TUE-Baustein - sofern vorhanden - wird die erste Schneide "D1" (mit Null-Inhalt) für die OEM-Schneidendaten angelegt. Falls ein TU-Baustein vorhanden ist, ist dort der Datensatz für das Werkzeug bereitgestellt.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CREATO	Werkzeug anlegen
Unit	INT	1 ... 10	TOA
WVar1	INT		T-Nummer

**12.15.4.19 PI-Dienst: DELECE****Funktion: Löschen einer Werkzeugschneide**

Wird in dem PI-Dienst unter dem Parameter T-Nummer die T-Nummer eines existierenden Werkzeugs angegeben, so wird die Schneide zu diesem Werkzeug gelöscht (in diesem Fall hat der Parameter D-Nummer - die Nummer der anzulegenden Schneide - einen Wertebereich von 00001 - 00009. Ist eine positive T-Nummer als Parameter angegeben und das Werkzeug zu der angegebenen T-Nummer existiert nicht, so schlägt der PI-Dienst fehl. Wird für die T-Nummer der Wert 00000 angegeben (Modell der absoluten D-Nummern), so kann sich der Wertebereich der D-Nummer von 00001 - 31999 erstrecken. Existiert die angegebene Schneide nicht, so schlägt der PI-Dienst in beiden Fällen fehl.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DELECE	Lösche Schneide
Unit	INT	1 ... 10	TOA
WVar1	INT		T-Nummer des Werkzeuges, zu dem die Werkzeugschneide angelegt werden soll. Ein Wert 00000 besagt, dass kein Bezug zu einem Werkzeug existieren soll (absolute D-Nummer)
WVar2	INT	1 ... 9 bzw. 01 ... 31999	Schneiden-Nummer der Werkzeugschneide, welche gelöscht werden soll

12.15.4.20 PI-Dienst: DELETO

**Funktion: Werkzeug löschen**

Löscht das Werkzeug zu der übergebenen T-Nummer mit allen Schneiden (in TO, ggf. TU, TUE und TG (Typ 4xx), TD und TS Bausteinen).

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DELETO	Werkzeug löschen
Unit	INT	1 ... 10	TOA
WVar1	INT		T-Nummer

12.15.4.21 PI-Dienst: MMCSEM

**Semaphoren für verschiedene PI-Dienste, Anwendbar durch HMI und PLC**

Kanalspezifisch stehen jeweils 10 Semaphoren zum Schützen von kritischen Funktionen für HMI/PLC zur Verfügung. Hat eine Funktion einen kritischen Abschnitt bezüglich von NC zu lesender Daten, können sich mehrere HMI/PLC-Einheiten durch Setzen der Semaphore mit der entsprechenden Funktionsnummer synchronisieren.

HMI/PLC verwaltet die Semaphore. Ein Semaphor-Wert von 1 spezifiziert eine Teste & Setze-Operation für die Semaphore zu der angegebenen Funktionsnummer. Der Rückgabewert des PI-Dienstes gibt das Ergebnis dieser Operation an:

- Rückgabewert Done := TRUE: Semaphore konnte gesetzt werden, kritische Funktion kann aufgerufen werden
- Rückgabewert Error := TRUE mit State = 3: Semaphore war bereits gesetzt, kritische Funktion kann zur Zeit nicht aufgerufen werden. Dies muss zu einem späteren Zeitpunkt wiederholt werden

**ACHTUNG**

Nach Beenden der Operation (Lesen der Daten dieses PI-Dienstes) **muss** die **Semaphore** unbedingt wieder **freigegeben werden**.

**Parameter**

- WVar1 = <Funktionsnummer>

Funktionsnummer	PI-Dienst
1	TMCRTO (Werkzeug anlegen)
2	TMFDPL (Leerplatzsuche zum Beladen)
3	TMMVTL (Magazinplatz zum Beladen bereitstellen, Werkzeug entladen)
4	TMFPBP (Suche Platz)
5	TMGETT (Suche Werkzeugnummer)
6	TSEARC (Werkzeug suchen)
7 ... 10	Reserviert

- WVar2 = <Wert>

Wert	Bedeutung
0	Semaphore zurücksetzen
1	Semaphore testen und setzen

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.MMCSEM	Semaphore setzen
Unit	INT	1, 2 bis 10	Kanal
WVar1	INT	1 ... 10	FunctionNumber
WVar2	WORD	0, 1	SemaphoreValue

**12.15.4.22 PI-Dienst: TMCRTO****Funktion Werkzeug anlegen:**

Anlegen eines Werkzeuges unter Angabe eines Bezeichners, einer Duplonummer, z. B. mit:

- \$TC\_TP1[y] = Duplonummer;
- \$TC\_TP2[y] = "Werkzeugbezeichner"

Oder optional durch einer T-Nummer, z. B. mit y = T-Nummer

Das Werkzeug wird im Bereich TV (Werkzeug-Directory) als vorhanden eingetragen, im TO-Baustein wird die erste Schneide "D1" (mit Null-Inhalt) für die Korrekturen angelegt, im TS-Baustein wird die erste Schneide "D1" (mit Null-Inhalt) für die Überwachungsdaten angelegt, im TUE-Baustein - sofern vorhanden - wird die Schneide "D1" für die OEM-Schneidendaten mit angelegt, im TD-Baustein stehen Bezeichner, Duplonummer und Anzahl der Schneiden (=1) zu der optional vorgegebenen oder von der NC vergebenen T-Nummer.

Ist ein TU-Baustein vorhanden, ist dort der Datensatz für das Werkzeug bereitgestellt. Nach Ausführung des PI-Dienstes steht im TV-Baustein unter **TnumWZV** die T-Nummer zu dem angelegten Werkzeug.

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMCRT0	Werkzeug anlegen
Unit	INT	1 - 10	TOA
WVar1	INT		T-Nummer
WVar2	INT		Duplonummer
Addr1	STRING	max. 32 Zeichen	Werkzeugbezeichner
T-Nummer > 0 bedeutet, eine T-Nummer vorgeben T-Nummer = -1 bedeutet, NCK soll T-Nummer vergeben Beispiel zeigt T-Nummer = -1 ⇒ T-Nummer von NCK vergeben			

**12.15.4.23 PI-Dienst: TMFDPL**

**Funktion: Leerplatzsuche zum Beladen, in Abhängigkeit der Parameterbelegung**

- Platz-Nummer\_Wohin = -1, Magazin-Nummer\_Wohin = -1

Sucht unter allen Magazinen zum angegebenen Bereich (= Kanal) einen Leerplatz für das mit T-Nummer spezifizierte Werkzeug. Die gefundene Magazin- und Platznummer steht nach Ausführung des PI-Dienstes im Konfigurationsblock des Kanals (Komponente **magCMCcmdPar1** (Magazinnummer) und **magCMCcmdPar2** (Platznummer)). Platz-Nummer\_Bez und Magazin- Nummer\_Bez können als Suchkriterium besetzt sein oder nicht (= -1). Der PI-Dienst wird in Abhängigkeit vom Suchergebnis positiv oder negativ quittiert.
- Platz-Nummer\_Wohin = -1, Magazin-Nummer\_Wohin = Magazin-Nummer

Es wird in dem angegebenen Magazin ein Leerplatz für das mit T-Nummer spezifizierte Werkzeug gesucht. Platz-Nummer\_Bez und Magazin- Nummer\_Bez können als Suchkriterium besetzt sein oder nicht (= -1). Der PI wird in Abhängigkeit vom Suchergebnis positiv oder negativ quittiert.
- Platz-Nummer\_Wohin = Platz-Nummer, Magazin-Nummer\_Wohin = Magazin-Nummer

Der angegebene Platz wird geprüft, ob er zum Beladen mit dem angegebenen Werkzeug frei ist. Platz-Nummer\_Bez und Magazin-Nummer\_Bez können als Suchkriterium besetzt sein oder nicht (= -1). Der PI-Dienst wird in Abhängigkeit vom Prüfergebnis positiv oder negativ quittiert.



Die Kommando-Parameter 1 und 2 stehen bei der Quelle.

Laden: Wenn Quelle ein internes Belademagazin ist, dann stehen die Kommando-Parameter beim Ziel (einem realen Magazin).

Entladen: Quelle ist immer reales Magazin.

---

### Hinweis

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

---

### Parametrierung

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMFDPL	Leerplatz zum Beladen
Unit	INT	1 - 10	TOA
WVar1	INT		T-Nummer
WVar2	INT		Platz-Nummer _Wohin
WVar3	INT		Magazin-Nummer _Wohin
WVar4	INT		Platz-Nummer _Bez
WVar5	INT		Magazin-Nummer _Bez

12.15.4.24 PI-Dienst: TMFPBP

**Funktion: Leerplatzsuche**

Der Dienst sucht in den angegebenen Magazin(en) einen Leerplatz, welcher den angegebenen Kriterien wie Werkzeuggröße und Platztyp genügt.

Das Ergebnis kann bei erfolgreicher Suche aus folgenden BTSS-Variablen gelesen werden:

- magCMCmdPar1 (Magazin-Nummer)
- magCMCmdPar2 (Platznummer)

<b>ACHTUNG</b>
Der PI-Dienst kann nur mit FB 7 angefordert werden. Siehe Kapitel "FB 7: PI_SERV2 (PI-Dienst anfordern) (Seite 1065)".

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMFPBP	Leerplatzsuche
Unit	INT	1 bis 10	TOA
WVar1	INT		<b>Magazinnummer</b> des Magazins, <b>ab</b> dem gesucht werden soll
WVar2	INT		<b>Platznummer</b> des Platzes im Magazin aus WVar1
WVar3	INT		<b>Magazinnummer</b> des Magazins, bis zu dem gesucht werden soll
WVar4	INT		<b>Platznummer</b> des Platzes im Magazin aus WVar3
WVar5	INT		Magazinnummer Bezug
WVar6	INT		Platznummer Bezug
WVar7	INT	0 bis 7	Anzahl der benötigten Halbplätze nach links
WVar8	INT	0 bis 7	Anzahl der benötigten Halbplätze nach rechts
WVar9	INT	0 bis 7	Anzahl der benötigten Halbplätze nach oben
WVar10	INT	0 bis 7	Anzahl der benötigten Halbplätze nach unten
WVar11	INT		Nummer des gewünschten Platztyps
WVar12	INT	0 bis 4	spezifiziert die gewünschte Suchrichtung
			0 Suchstrategie wie in \$TC_MAMP2 eingestellt
			1 vorwärts
			2 rückwärts
			3 symmetrisch

## Beispiele zum Einstellen des Suchbereichs

WVar1: (ab)	WVar2: (Platz)	WVar3: (bis)	WVar4: (Platz)	Suchbereich
#M1	#P1	#M1	#P1	Es wird nur der Platz #P1 im Magazin #M1 überprüft
#M1	#P1	#M2	#P2	Es werden die Plätze beginnend von Magazin #M1, Platz #P1 bis Magazin #M2, Platz #P2 durchsucht
#M1	-1	#M1	-1	Es werden alle Plätze von Magazin #M1 - und nur diese - durchsucht
#M1	-1	-1	-1	Es werden alle Plätze beginnend ab Magazin #M1 durchsucht
#M1	#P1	-1	-1	Es werden alle Plätze beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M1	-1	Es werden die Plätze in Magazin #M1 beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M2	-1	Es werden die Plätze beginnend ab Magazin #M1 und darin Platz #P1 bis einschließlich Magazin #M2 durchsucht
#M1	-1	#M2	#P2	Es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 und darin Platz #P2 durchsucht
#M1	-1	#M2	-1	Es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 durchsucht
-1	-1	-1	-1	Es werden alle Magazinplätze durchsucht

## 12.15.4.25 PI-Dienst: TMGETT

**Funktion: T-Nummer zum vorgegebenen Werkzeugbezeichner mit Duplonummer bestimmen**

Bestimmen der T-Nummer zu einem vorgegebenen Werkzeugbezeichner mit Duplonummer. Ob eine T-Nummer vom PI-Dienst gefunden wurde, wird in der Variablen resultNrOfTools im Baustein TF angezeigt. Existiert das angegebene Werkzeug nicht, so wird die Anzahl 0 zurückgeliefert. Wird die Anzahl 1 geliefert, wird die T-Nummer in der Variablen resultToolNr im Baustein TF angezeigt. Da der PI-Dienst ein Ergebnis in der Variablen resultToolNr zurückliefert, ist der Dienst mit dem Semaphore-Mechanismus (PI-Dienst \_N\_MMCSEM) mit der Funktionsnummer für \_N\_TMGETT abzusichern.

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechenden Parameter "WVar1" für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMGETT	T-Nummer ermitteln
Unit	INT	1 ... 10	TOA
Addr1	STRING	max. 32 Zeichen	Name des Werkzeugs, zu dem T-Nummer gesucht wird
WVar1	INT		Duplonummer des Werkzeugs, zu dem T-Nummer gesucht wird

**12.15.4.26 PI-Dienst: TMMVTL**

**Funktion: Magazinplatz zum Beladen bereitstellen, Werkzeug entladen**

Der PI-Dienst wird sowohl zum Beladen als auch zum Entladen verwendet. Welche Operation mit dem PI-Dienst angestoßen wird, ist in der Zuordnung der realen Plätze zu den "Von"-Parametern und den "Zu"-Parametern abhängig: Beladen ⇒ "Von" = Beladestelle/-station, Entladen ⇒ "Zu" = Beladestelle/-station.

Der PI-Dienst TMMVTL wird für alle Bewegungen verwendet:

1. Be- und Entladen (Beladestelle ↔ Magazin)
2. Be- und Entladen (Beladestelle ↔ Zwischenspeicher, z. B. Spindel)
3. Umsetzen innerhalb eines Magazins
4. Umsetzen zwischen verschiedenen Magazinen
5. Umsetzen zwischen Magazin und Zwischenspeicher
6. Umsetzen innerhalb des Zwischenspeichers

Für die Überwachung von Fall 1, 3, 4, 5 werden folgende Variablen aus Baustein TM verwendet:

- magCmd (BereichsNr = TO-Einheit, Zeile = Magazinnummer)
- magCmdState <- "Quittung"

Für die Überwachung von Fall 2, 6 werden folgende Variablen aus Baustein TMC verwendet:

- magCBCmd (BereichsNr = TO-Einheit)
- magCBCmdState <- "Quittung"

### Beladen

Bereitet das angegebene reale Magazin des spezifizierten Kanals zum Beladen vor, d. h. das Magazin verfährt auf den gewünschten Platz zum Beladen an der angegebenen Beladestation/-platz (Platz-Nummer\_Von, Magazin-Nummer\_Von) und wechselt des Werkzeug ein.

Bei Platznummer\_Zu = -1 wird in dem angegebenen Magazin zuerst ein Leerplatz für das mit T-Nummer spezifizierte Werkzeug gesucht und dann das Magazin verfahren. Die gefundene Platznummer steht nach Ausführung des PI-Dienstes im TM-Bereich in der Komponente **magCMCmdPar2** zu dem **realen** Magazin des Kanals.

Bei Platznummer\_Zu = -2 und gültiger Magazinnummer wird in die aktuell anstehende Magazinposition des angegebenen Magazins beladen. Die Platznummer des zu beladenden Platzes steht nach Ausführung des PI-Dienstes im TM-Bereich in der Komponente **magCMCmdPar2** zu dem realen Magazin des Kanals.

### Entladen

Das mit T-Nummer spezifizierte Werkzeug wird an der angegebenen Beladestelle/-station (Platz-Nummer\_Zu, Magazin-Nummer\_Zu) entladen, d. h. das Magazin wird zum Entladen verfahren, das Werkzeug ausgewechselt. Im TP-Baustein wird der Magazinplatz zu dem Werkzeug als frei gekennzeichnet. Das Werkzeug kann wahlweise über T-Nummer oder über Platz- und Magazinnummer angegeben werden. Eine nicht benutzte Spezifikation trägt den Wert -1.

### Hinweis

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

### Parametrierung

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMMVTL	Magazinplatz zum Beladen bereitstellen, WZ entladen
Unit	INT	1 ... 10	TOA
WVar1	INT		T-Nummer
WVar2	INT		Platz-Nummer_Von
WVar3	INT		Magazin-Nummer_Von
WVar4	INT		Platz-Nummer_Zu
WVar5	INT		Magazin-Nummer_Zu

12.15.4.27 PI-Dienst: TMPOSM

**Funktion: Magazinplatz oder Werkzeug positionieren, in Abhängigkeit der Parameterbelegung**

Über den PI-Dienst wird ein Magazinplatz, der entweder direkt angegeben ist oder über ein darauf befindliches Werkzeug qualifiziert wurde, an eine vorgegebene Position (z B. vor einen Beladeplatz) gefahren.

Der PI-Dienst lässt einen auf unterschiedliche Arten qualifizierbaren Magazinplatz vor einen angegebenen Beladeplatz fahren. Der Beladeplatz wird in den PI-Parametern "Platz-Nummer\_Von" und "Magazin-Nummer\_Von" angegeben (muss!).

Die Qualifikation des zu fahrenden Magazinplatzes kann erfolgen durch:

- T-Nummer des Werkzeugs  
 Der Platz, auf dem sich das Werkzeug befindet, fährt; die Parameter "WZ-Bezeichner", "Duplonummer", "Platz-Nummer\_Von" und "Magazin-Nummer\_Von" sind irrelevant (d. h. Werte "", "-0001", "-0001", "-0001").
- WZ-Bezeichner und Duplonummer  
 Der Platz, auf dem sich das Werkzeug befindet, fährt; die Parameter "T-Nummer", "Platz-Nummer\_Von" und "Magazin-Nummer\_Von" sind irrelevant (d. h. jeweils Wert "-0001").
- Direkte Angabe des Platzes in den Parametern "Platz-Nummer\_Von" und "Magazin-Nummer\_Von".  
 Die das Werkzeug qualifizierenden Parameter T-Nummer, "WZ-Bezeichner" und "Duplonummer" sind irrelevant (d. h. Werte "-0001", "", "-0001").

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMPOSM	Magazinplatz oder Werkzeug positionieren
Unit	INT	1 ... 10	TOA
Addr1	STRING	max 32. Zeichen	Werkzeugbezeichner
WVar1	INT		T-Nummer
WVar2	INT		Duplonummer
WVar3	INT		Platz-Nummer_Von
WVar4	INT		Magazin-Nummer_Von
WVar5	INT		Platz-Nummer_Bezug
WVar6	INT		Magazin-Nummer_Bezug

**12.15.4.28 PI-Dienst: TMPCIT****Funktion: Inkrementwert für Stückzahlzähler setzen**

Inkrementieren des Stückzahlzählers des Spindelwerkzeugs.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMPCIT	Inkrementwert für Stückzahlzähler setzen
Unit	INT	1 ... 10	TOA
WVar1	WORD	0 ... max.	Spindelnummer; entspricht dem Artindex bei den Platzdaten mit Platzart Spindel des Zwischenspeichermagazins im Kanal.000 = Hauptspindel
WVar2	WORD	0 ... max.	Inkrementwert; gibt an, nach wie vielen Umdrehungen der Spindel der Stückzahlzähler inkrementiert wird.

**12.15.4.29 PI-Dienst: TMRASS****Funktion: Rücksetzen des Aktiv-Status**

Rücksetzen des Aktiv-Status bei verschlissenen Werkzeugen.

Mit diesem PI-Dienst werden alle Werkzeuge gesucht, die den Werkzeug-Zustand aktiv und gesperrt haben. Diesen Werkzeugen wird dann der Aktiv-Zustand wieder weggenommen. Mögliche sinnvolle Zeitpunkte für diesen PI-Dienst sind die negative Flanke des NC/PLC-Nahtstellensignals "Werkzeug-Sperre unwirksam", ein Programmende, ein Kanal-RESET. Dieser PI-Dienst ist hauptsächlich für die PLC sinnvoll, weil hier auch bekannt ist, wann das gesperrte Werkzeug endgültig nicht mehr verwendet werden soll.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI. TMRASS	Rücksetzen des Aktiv-Status
Unit	INT	1 ... 10	TO-Bereich

12.15.4.30 PI-Dienst: TRESMO

**Funktion: Rücksetzen der Überwachungswerte**

Mit diesem PI-Dienst können die Überwachungswerte der bezeichneten Schneiden der bezeichneten Werkzeuge auf die Soll- (Ausgangs-) werte zurückgesetzt werden. Dies erfolgt nur für Werkzeuge mit aktiver Überwachung.

Siehe dazu auch den Befehl `RESETMON`.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI. TRESMO	Rücksetzen der Überwachungswerte
Unit	INT	1 ... 10	TO-Bereich
WVar1	WORD	- max ... max	Toolnummer
			0: alle Werkzeuge behandelt
			> 0: nur dieses Werkzeug wird behandelt
			< 0: alle Schwesterwerkzeuge zur angegebenen T-Nr. werden behandelt
WVar2	WORD	0 ... max.	D-Nummer
			< 0: Überwachung der angegebenen Schneide der angegebenen Werkzeuge wird zurückgesetzt.
			0: Überwachung aller Schneiden der angegebenen Werkzeuge wird zurückgesetzt.
Wvar3	WORD	0 ...15	Überwachungsarten Art der Überwachung, die zurückgesetzt werden soll. Dieser Parameter ist binär codiert:
			1: Standzeit-Überwachung wird zurückgesetzt.
			2: Stückzahl-Überwachung wird zurückgesetzt.
			4: Verschleiß-Überwachung wird zurückgesetzt.
			8: Summenkorrektur-Überwachung wird zurückgesetzt. Es können durch Addition der obigen Werte auch Kombination von Überwachungen zurückgesetzt werden.
			0: Es werden alle aktiven Überwachungen des Werkzeugs (\$TC_TP9) zurückgesetzt.



### 12.15.4.31 PI-Dienst: TSEARC

#### **Funktion: Komplexes Suchen über Suchmaske, in Abhängigkeit der Parameterbelegung**

Mit dem PI-Dienst können innerhalb eines Suchbereichs (in einem oder über mehrere Magazine, beginnend ab einem bestimmten Platz bis zu einem bestimmten Platz) Werkzeuge mit vorgegebenen Eigenschaften gesucht werden. Die vorgebbaren Eigenschaften beziehen sich nur auf Daten der Werkzeuge und deren Schneiden.

Der PI-Dienst ist nur bei aktivierter Werkzeugverwaltung verfügbar.

Für die Ausführung des Dienstes können eine Suchrichtung vorgegeben werden sowie die Ergebnismenge (ein Werkzeug (entspricht nächstes Werkzeug mit dieser Eigenschaft) oder alle Werkzeuge mit der vorgegebenen Eigenschaft) spezifiziert werden.

Als Ergebnis des Dienstes erhält der Aufrufer eine Liste mit den internen T-Nummern der gefundenen Werkzeuge zurück.

Die Suchkriterien können nur als UND-Verknüpfung vorgegeben werden. Will eine Applikation eine ODER-Verknüpfung als Suchkriterium vorgeben, so muss sie dies über mehrere nacheinander ausgeführte Anfragen mit UND-Verknüpfungen realisieren und die jeweiligen Ergebnisse der einzelnen Anfragen entsprechend verknüpfen/auswerten.

Zur Parametrierung des PI-Dienstes werden zunächst über Variablendienst im Baustein TF die Eigenschaften der gesuchten Werkzeuge vorgegeben. Dazu werden im Baustein TF in den Operanden-Masken (parMaskT..) die relevanten Vergleichskriterien (welche Werkzeugdaten sollen verglichen werden?) markiert, die Vergleichsoperatoren-Daten (parDataT..) mit den entsprechend auszuführenden Vergleichsarten (==, <, >, <=, >=, &&) belegt und in den Operandendaten die Vergleichswerte eingetragen.

Anschließend wird der PI-Dienst angestoßen und nach dessen erfolgreicher Rückkehr über den Variablendienst aus dem Baustein TF unter der Variablen resultNrOfTools die Anzahl der Treffer und unter der Variablen resultToolNr die Ergebnisliste (interne T-Nummern der Werkzeuge, die bei der Suche gefunden wurden (resultNrOfTools Stück), ausgelesen.

Der PI-Dienst ist von seiner Vorbereitung bis zum erfolgten Abholen des Ergebnisses mit einer Semaphore zu kapseln. Nur so kann der exklusive Zugriff und die exklusive Verwendung des Bausteins TF zusammen mit dem PI-Dienst TSEARC sichergestellt werden (siehe PI-Dienst MMCSEM).

Ist der Dienst fehlerhaft parametrierung, schlägt er fehl. Ansonsten liefert er immer ein Ergebnis, auch wenn kein Werkzeug gefunden wird (resultNrOfTools = 0).

Der Suchbereich kann über die Belegung der Parameter "MagNrFrom", "PlaceNrFrom", "MagNrTo", "PlaceNrTo" wie folgt vorgegeben werden:

MagNr From	PlaceNr From	MagNr To	PlaceNr To	Suchbereich
WVar1	WVar2	WVar3	WVar4	
#M1	#P1	#M2	#P2	es werden die Plätze beginnend von Magazin #M1, Platz #P1 bis Magazin #M2, Platz #P2 durchsucht
#M1	-1	#M1	-1	es werden alle Plätze von Magazin #M1 - und nur diese - durchsucht
#M1	-1	-1	-1	es werden alle Plätze beginnend ab Magazin #M1 durchsucht
#M1	#P1	-1	-1	es werden alle Plätze beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M1	-1	es werden die Plätze in Magazin #M1 beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M2	-1	es werden die Plätze beginnend ab Magazin #M1 und darin Platz #P1 bis einschließlich Magazin #M2 durchsucht
#M1	-1	#M2	#P2	es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 und darin Platz #P2 durchsucht
#M1	-1	#M2	-1	es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 durchsucht
-1	-1	-1	-1	es werden alle Magazinplätze durchsucht

Für eine symmetrische Suche (vgl. Parameter "SearchDirection"), darf sich der Suchbereich nur über ein einziges Magazin erstrecken (Fälle 2 und 5 aus obiger Tabelle). Ist ein anderer Suchbereich angegeben, schlägt der Dienst fehl. Und es muss in den Parametern "MagNrRef" und "PlaceNrRef" ein Referenzplatz angegeben werden, bezüglich dem symmetrische Suche erfolgt.

Der Referenzplatz ist ein Zwischenspeicher-Platz (ein Platz aus dem Magazin Zwischenspeicher, d. h. Wechselstelle, Greifer ...) oder ein Beladeplatz/-stelle (ein Platz aus dem internen Belademagazin). Die symmetrische Suche erfolgt bezüglich dem Magazinplatz, vor dem angegebenen Referenzplatz. Für den angegebenen Referenzplatz muss eine Mehrfachzuordnung zu dem zu durchsuchenden Magazin im Baustein TPM konfiguriert sein. Ist dies nicht der Fall, so schlägt der Dienst fehl. Liegt der Magazinplatz vor dem Referenzplatz außerhalb des Suchbereichs, verhält sich der Dienst so, als ob er keinen passenden Platz gefunden hat.

#### Hinweis

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

## Parametrierung

Signal	Typ	Wertebereich	Bedeutung	
PIService	ANY	PI.TSEARC	Komplexes Suchen über Suchmasken	
Unit	INT	1 ... 10	TOA	
WVar1	INT		MagNrFrom Magazin-Nummer des Magazins, ab dem der Suchbereich beginnen soll.	
WVar2	INT		PlaceNrFrom Platz-Nummer des Platzes in dem Magazin MagNrFrom, ab dem der Suchbereich beginnen soll.	
WVar3	INT		MagNrTo Magazin-Nummer des Magazins, bei dem der Suchbereich enden soll.	
WVar4	INT		PlaceNrTo Platz-Nummer des Platzes in dem Magazin MagNrTo, bei dem der Suchbereich enden soll.	
WVar5	INT		MagNrRef Magazin-Nummer des (internen) Magazins, bezüglich dem symmetrisch gesucht werden soll. (dieser Parameter ist nur relevant bei einer Suchrichtung "symmetrisch")	
WVar6	INT		PlaceNrRef Platz-Nummer des Platzes in dem Magazin MagNrRef, bezüglich dem symmetrisch gesucht werden soll. Dieser Parameter ist nur relevant bei einer Suchrichtung "symmetrisch".	
WVar7	INT	1, 2, 3	SearchDirection spezifiziert die gewünschte Suchrichtung:	
			1:	vorwärts vom ersten Platz des Suchbereichs
			2:	rückwärts vom letzten Platz des Suchbereichs
			3:	symmetrisch zu realen Magazinplatz, der vor dem mit Magazin-Nummer_Bezug und Platz-Nummer_Bezug angegebenen Platz steht
WVar8	INT	0, 1, 2, 3	KindofSearch	
			0:	schneidenspezifisch alle Werkzeuge mit dieser Eigenschaft suchen
			1:	das erste gefundene Werkzeug mit dieser Eigenschaft suchen (schneidenspezifisch)
			2:	alle Werkzeuge über alle Schneiden mit dieser Eigenschaft suchen
			3:	das erste gefundene Werkzeug mit dieser Eigenschaft suchen (über alle Schneiden)

12.15.4.32 PI-Dienst: TMCRMT

**Funktion: Multitool erzeugen**

Anlegen eines neuen Multitools unter Vorgabe eines Bezeichners, optional einer Multitool-Nummer, Anzahl der Plätze und Art der Abstandskodierung. WZ-T-Nummern, Magazinnummern und Multitool-Nummern sind im erlaubten Nummernband 1 ... 32000 eindeutig. Mit dem Erzeugen des Multitools werden auch alle zugehörigen Plätze erzeugt.

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechenden Parameter WVar1 für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung	
PIService	ANY	PI.TMCRMT	Multitool anlegen	
Unit	INT	1 ... 10	TOA	
Addr1	STRING	max. 32 Zeichen	Multitool-Bezeichner	
WVar1	INT	0	reserviert	
WVar2	INT	-1 ... 32000	Multitool-Nummer	
			1 bis 32000:	vom Anwender vorgegebene Multitool-Nummer
			-1:	NCK vergibt Multitool-Nummer selbst
WVar3	INT	2 ... MD17504 \$MN_MAX_TOOLS_PER_MULTITool	Anzahl der Plätze im Multitool	
WVar4	INT	1 ... 3	Art der Abstandskodierung	

**12.15.4.33 PI-Dienst: TMDLMT****Funktion: Multitool löschen**

Löscht das Multitool in allen Datenbausteinen, in denen es gespeichert ist. Im Multitool bestückte Werkzeuge sind anschließend nicht mehr bestückt und nicht mehr beladen, aber sie sind weiterhin definiert.

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMDLMT	Multitool löschen
Unit	INT	1 ... 10	TOA
Addr1	STRING	max. 32 Zeichen	Multitool-Bezeichner
WVar1	INT	0	reserviert
WVar2	INT	-1 ... 32000	Multitool-Nummer
			1 bis 32000: lösche die angegebene Multitool-Nummer (Addr1 wird nicht ausgewertet)
			-1: lösche das Multitool mit dem in Addr1 angegebenen Namen
WVar3	INT	0, 1	lösche enthaltene Werkzeuge
			0: nicht löschen
			1: löschen

**12.15.4.34 PI-Dienst: POSMT****Funktion: Multitool positionieren**

Positioniert das Multitool auf den programmierten Platz oder alternativ auf das programmierte Werkzeug, das sich auf einem der Plätze des Multitools befindet. Das Werkzeug selbst kann entweder mit seiner T-Nummer angegeben werden oder mit seinem Namen plus Duplonummer. Ein Multitool kann nur positioniert werden, wenn es sich auf einem WZ-Halterplatz befindet und wenn keine WZ-Korrektur bezüglich dieses WZ-Halterplatzes aktiv ist.

Die Positionierung wird programmiert durch Programmierung von WVar1 (= des WZ-Halters, dessen Multitool positioniert werden soll) und entweder:

- mit WVar2 (Werkzeugnummer), oder
- mit Addr1 und WVar3 (Werkzeugbezeichner / Duplonummer), oder aber
- mit WVar4 (Multitool-Platznummer).

**Parametrierung**

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.POSMT	Multitool positionieren
Unit	INT	1 ... 10	TOA
Addr1	STRING	max. 32 Zeichen	Werkzeugbezeichner des im Multitool zu positionierenden Werkzeugs <b>Hinweis:</b> Wird kein Werkzeugbezeichner vorgegeben, dann muss hier ein Leer-String eingegeben werden (dann muss WVar2 programmiert sein).
WVar1	INT	1 ... 999	Nummer des WZ-Halters
WVar2	INT	-1 ... 32000	Werkzeugnummer (T-Nummer) des im Multitool zu positionierenden Werkzeugs -1: Werkzeugnummer ist irrelevant (dann müssen Addr1 und WVar3 programmiert sein)
WVar3	INT	-1 ... 32000	Duplonummer des im Multitool zu positionierenden Werkzeugs -1: Duplonummer ist irrelevant (dann muss WVar2 programmiert sein)
WVar4	INT	1 ... 999	Multitool-Platznummer des Platzes, auf den positioniert werden soll

**12.15.4.35 PI-Dienst: FDPLMT**

**Funktion: Leerplatz innerhalb des Multitools suchen/prüfen, in Abhängigkeit der Parameterbelegung**

Sucht im Multitool einen freien Platz zur Aufnahme des genannten Werkzeugs bzw. prüft den angegebenen Platz im Multitool, ob er frei ist zur Aufnahme des angegebenen Werkzeugs. Das Werkzeug kann entweder über seine T-Nummer angegeben werden oder alternativ dazu mit seinem Bezeichner und seiner Duplonummer.

Der Aufruf des PI-Dienstes wird in Abhängigkeit vom Suchergebnis positiv oder negativ quittiert.

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechenden Parameter WVar1 für diesen PI-Dienst aufzurufen. Siehe Kapitel "PI-Dienst: MMCSEM (Seite 1038)".

### Parametrierung

Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.FDPLMT	Leerplatz innerhalb des Multitools suchen/prüfen
Unit	INT	1 ... 10	TOA
Addr1	STRING	max. 32 Zeichen	Werkzeugbezeichner des im Multitool zu positionierenden Werkzeugs <b>Hinweis:</b> Wird kein Werkzeugbezeichner vorgegeben, dann muss hier ein Leer-String eingegeben werden (dann muss WVar2 programmiert sein).
WVar1	INT	-1 ... 32000	Werkzeugnummer (T-Nummer) des im Multitool zu positionierenden Werkzeugs -1: Werkzeugnummer ist irrelevant (dann müssen alternativ Addr1 und WVar2 programmiert sein)
WVar2	INT	-1 ... 32000	Duplonummer des im Multitool zu positionierenden Werkzeugs -1: Duplonummer ist irrelevant (dann muss alternativ WVar1 programmiert sein)
WVar3	INT	-1 ... 32000	Nummer des Multitools, in dem die Leerplatzsuche/-prüfung stattfinden soll -1: suche über alle Multitools hinweg nach einem Leerplatz bzw. prüfe über alle Multitools hinweg, ob in einem davon der programmierte Platz frei ist zur Aufnahme des programmierten Werkzeugs
WVar4	INT	-1 ... 999	Multitool-Platznummer des Platzes, auf den positioniert werden soll -1: mache keine Platzprüfung, sondern suche einen leeren Platz innerhalb des Multitools

### Aufrufbeispiel

#### Programmanwahl im Kanal 1 (Hauptprogramm und Werkstückprogramm)

Eintrag PI-Dienst für DB 16 und STR für DB 124 mit dem S7-SYMBOL-Editor:

Parametrierung		
Symbol	Operand	Datentyp
PI	DB 16	DB 16
STR	DB 124	DB 124

```
DATA_BLOCK DB 126          //freier Anwender-DB, als Instanz für FB 4
FB 4
BEGIN
END_DATA_BLOCK

DATA_BLOCK DB 124
  struct
    PName:      string[32] := '_N_TEST_MPF';
    Path:       string[32] := '/_N_MPF_DIR/';    //Hauptprogramm
    PName_WST:  string[32] := '_N_ABC_MPF';
    Path_WST:   string[32] :=                //Werkstückprogramm
                '/_N_WKS_DIR/_N_ZYL_WPD';
  end_struct
BEGIN
END_DATA_BLOCK

Function FC "PICall" : VOID
  U  E 7.7;      //freie Taste Maschinensteuertafel
  S  M 0.0;      //Req aktivieren
  U  M 1.1;      //Fertigmeldung Done
  R  M 0.0;      //Auftrag beenden
  U  E 7.6;      //Fehlerquittierung von Hand
  U  M 1.0;      //Fehler steht an
  R  M 0.0;      //Auftrag beenden

  CALL FB 4, DB 126 (
    Req:=        M0.0,
    PIService:=  PI.SELECT,
    Unit:=       1,                                //CHAN 1
    Addr1:=      STR.Path,
    Addr2:=      STR.PName,                        //Hauptprogrammanwahl
                //Addr1:=STR.Path_WST,
                //Addr2:=STR.PName_WST,          //Werkstückprogrammanwahl
    Error:=     M1.0,
    Done:=      M1.1,
    State:=     MW2);
```



## 12.15.5 FB 5: GETGUD GUD-Variable lesen

### Funktion

Mit dem FB GETGUD kann das PLC-Anwenderprogramm eine GUD Variable (GUD = Global User Data, Globales anwenderdefiniertes Datum) im NCK- oder Kanal-Bereich lesen.

Der FB ist multiinstanzfähig. Der Aufruf ist nur im zyklischen Programm OB 1 zulässig. Eine Zuweisung für alle Parameter mit Req = 0 ist auch im OB 100 zulässig. Bei den Namen der GUD-Variablen sind nur Großbuchstaben zulässig.

Zu jedem FB 5-Aufruf muss ein separater Instanz-DB aus dem Anwender-Bereich zugeordnet werden.

Durch **Aufruf des FB 5** mit positivem Flankenwechsel am Steuereingang Req wird ein Auftrag gestartet. Zu diesem Auftrag gehört der Name der zu lesenden GUD-Variable in dem Parameter "Addr" mit dem Datentyp "STRING". Der Zeiger auf den Namen der GUD-Variablen wird dem Parameter "Addr" symbolisch mit <DatenbausteinName>.<VariablenName> zugewiesen. In den weiteren Parametern "Area" "Unit" "Index1" und "Index2" werden Zusatz-Informationen zu dieser Variablen angegeben (siehe Tabelle der Bausteinparameter).

Mit Aktivierung des Parameters "CnvtToken" kann optional ein Variablenzeiger (Token) für diese GUD-Variable erstellt werden. Dieser Variablenzeiger wird bei Systemvariablen der NC über den NC-VAR-Selector erstellt. Für die GUD-Variablen besteht nur diese Variante der Zeigergenerierung. Nachdem dieser GUD-Variablen-Zeiger gebildet wurde, kann über den FB2 und FB 3 (GET, PUT) mit Referenzierung auf diesen Variablenzeiger gelesen bzw. auch geschrieben werden. Dies ist der einzige Weg GUD-Variablen zu schreiben. Hierbei muss bei der Parametrierung des FB 2 bzw. FB 3 nur der Parameter Addr1 ... Addr8 auf diesen GUD-Variablenzeiger parametrieren. Eine Ausnahme bilden GUD-Variablenfelder. Bei diesen Feldern ist zusätzlich Line1 ... Line8 mit dem Feldindex dieser Variablen zu parametrieren. Der erfolgreiche Abschluss des Lesevorgangs wird am Zustandsparameter "Done"=TRUE angezeigt.

Der Lesevorgang erstreckt sich über mehrere (in der Regel 1 ... 2) PLC-Zyklen.

Eventuell aufgetretene Fehler werden über die Ausgangsparameter "Error" und "State" angezeigt.

---

### Hinweis

Um eine double-Variable vom NCK ohne Formatanpassung zu lesen, muss im Zielbereich für gelesene Daten ein ANY-Pointer mit Typ REAL Anzahl 2 angegeben werden (z. B.: P#M100.0 REAL 2). Erkennt das Grundprogramm beim Lesen einer "double"-Variablen als Zieltyp REAL 2, werden die Daten als 64-Bit Gleitkommazahl in den PLC-Datenbereich übernommen.

Der FB 5 kann GUD-Variablen nur dann lesen, wenn der Grundprogramm-Parameter "NCKomm" auf "TRUE" gesetzt wurde (im OB 100: FB 1, DB 7; siehe "FB 1: RUN\_UP Grundprogramm, Anlaufteil (Seite 999)").

---

## Deklaration der Funktion

```
FUNCTION_BLOCK FB 5           //Name-Server
    KNOW_HOW_PROTECT
    VERSION: 3.0
VAR_INPUT
    Req :          BOOL;
    Addr :         ANY;           //NamenString der Variablen
    Area  :        BYTE;         //Bereich: NCK = 0, Kanal = 2
    Unit  :        BYTE;
    Index1 :       INT;          //Feldindex 1
    Index2 :       INT;          //Feldindex 2
    CnvtToken :    BOOL;         //Umsetzung in 10 Byte Token
    VarToken :     ANY;          //Struct mit 10 Byte für den variablen Token
END_VAR
VAR_OUTPUT
    Error :        BOOL;
    Done  :        BBOL;
    State :        WORD;
END_VAR
VAR_IN_OUT
    RD :          ANY;
END_VAR
BEGIN
END_FUNCTION_BLOCK
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion GETGUD.

Signal	Art	Typ	Wertebereich	Bemerkung	
Req	E	BOOL		Auftragsstart mit positiver Flanke	
Addr	E	ANY	[DBName].[VarName]	GUD Variablenname in einer Variable vom Datentyp String	
Area	E	BYTE		Bereichsadresse:	
				0:	NCK-Variable
				2:	Kanal-Variable
Unit	E	BYTE		Bereich NCK: Unit:= 1 Bereich Kanal: Kanalnr.	
Index1	E	INT		Feldindex 1 der Variable Der Wert der Variable ist 0, wenn der Feldindex nicht benutzt wird.	
Index2	E	INT		Feldindex 2 der Variable Der Wert der Variable ist 0, wenn der Feldindex nicht benutzt wird.	
CnvtToken	E	BOOL		Aktivierung der Generierung eines 10 Byte Variablen-Token	
VarToken	E	ANY	[DBName].[VarName]	Adresse auf einen 10 Byte Token (siehe Beispiel)	
Error	A	BOOL		Auftrag wurde negativ quittiert, bzw. konnte nicht ausgeführt werden	
Done	A	BOOL		Auftrag wurde erfolgreich ausgeführt.	
State	A	WORD		siehe Fehlerkennungen	
RD	E/A	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	zu lesende Daten	

### Fehlerkennungen

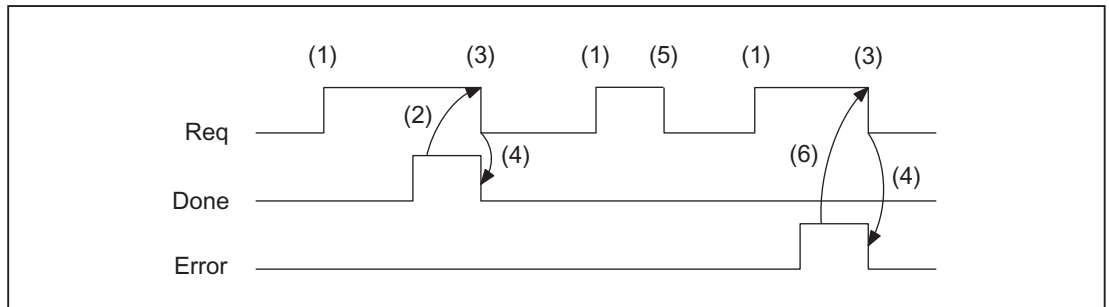
Konnte ein Auftrag nicht ausgeführt werden, wird dies am Zustandsparameter Error mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang State kodiert:

State		Bedeutung	Hinweis
WORT-H	WORT-L		
0	1	Zugriffsfehler	
0	2	Fehler im Auftrag	falsche Zusammenstellung von Var. in einem Auftrag
0	3	negative Quittung, Auftrag nicht ausführbar	interner Fehler, evtl. Abhilfe: NC-RESET
0	4	Datenbereiche oder Datentypen stimmen nicht überein	zu lesendes Datum in RD überprüfen;
1	4	nicht ausreichend lokaler Anwenderspeicher zur Verfügung	gelesene Var. ist länger als in RD angegeben
0	6	FIFO voll	Auftrag muss wiederholt werden, da die Warteschlange voll ist
0	7	Option nicht gesetzt	GP-Parameter "NCKomm" ist nicht gesetzt
0	8	falscher Zielbereich (SD)	RD dürfen keine Lokaldaten sein
0	9	Übertragung belegt	Auftrag muss wiederholt werden
0	10	Fehler bei Adressierung	Unit enthält Wert 0
0	11	Variablenadresse ungültig	Addr. (bzw. Variablenname), Area, Unit überprüfen
1 bis 8	13 (0x0d)	Referenz des ANY Datums falsch	Gefordertes String bzw. NcVar-Datum ist nicht parametrierbar
0	15 (0x0f)	String mehr als 32 Zeichen	GUD Variablenname zu lang

### Projektierungsschritte

Für das Lesen einer GUD-Variablen ist der Name der GUD-Variablen in einer String-Variablen zu hinterlegen. Der Datenbaustein mit dieser String-Variablen ist in der Symbolliste zu definieren, so dass eine symbolische Zuweisung des Parameters "Addr" für den FB GETGUD erfolgen kann. Optional ist eine Strukturvariable in einem beliebigen Datenbereich der PLC für den Empfang des Variablenzeigers zu definieren, siehe Vorgabe im folgenden Beispiel.

## Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positive Quittung: Variablen wurden geschrieben
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) nicht zulässig
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter "State"

## Aufrufbeispiel 1

Lesen einer GUD-Variablen aus Kanal 1 mit dem Namen "GUDVAR1" (Typ-Definition der Variablen: INTEGER). Die anwenderdefinierte Variable soll in einen 10 Byte Variablen-Zeiger konvertiert werden für nachfolgendes Schreiben mit dem FB 3 (siehe auch Tabelle "Zuordnung der Datentypen" in "FB 2: GET NC-Variablen lesen (Seite 1008)").

### Aufruf und Parametrierung des FB 5 mit dem Instanz-DB 111:

```
// Datenbaustein für GUD-Variablen
DATA_BLOCK DB_GUDVAR          //Zuordnung in Symbolliste vornehmen

STRUCT
  GUDVar1 : STRING[32] := 'GUDVAR1';          //Name ist vom Anwender def.
  GUDVar1Token :
    STRUCT
      SYNTAX_ID : BYTE;
      bereich_u_einheit : BYTE;
      spalte : WORD;
      zeile : WORD;
      bausteintyp : BYTE;
      ZEILENANZAHL : BYTE;
      typ : BYTE;
      laenge : BYTE;
    END_STRUCT ;
END_STRUCT;

BEGIN
```

```

END_DATA_BLOCK

// freier Anwender DB, als Instanz für FB 5
DATA_BLOCK DB 111

        FB 5
BEGIN
END_DATA_BLOCK

// freier Anwender DB, als Instanz für FB 3
DATE_BLOCK DB 112

        FB 3
BEGIN
END_DATA_BLOCK

//Eine anwenderdefinierte Kanalvariable aus Kanal 1 soll gelesen werden
//mit einer Konvertierung in einen Variablen-Zeiger für ein
//nachfolgendes Schreiben dieser Variablen.

Function FC "VariablenCall" : VOID
U      E 7.7;           //freie Taste Maschinensteuertafel
S      M 100.0;        //Req aktivieren
U      M 100.1;        //Fertigmeldung Done
R      M 100.0;        //Auftrag beenden
U      E 7.6;          //Fehlerquittierung von Hand
U      M 102.0;        //Fehler steht an
R      M 100.0;        //Auftrag beenden

CALL FB 5, DB 111(
        Req      := M 100.0,      //Startflanke für Lesen
        Area     := B#16#2,      //Kanal-Variable
        Unit     := B#16#1,      //Kanal 1
        Index1   := 0,           //kein Feldindex
        Index2   := 0,           //kein Feldindex
        CnvtToken := TRUE,       //Umsetzung in 10 Byte Token
        VarToken := DB_GUDVAR.GUDVar1Token,
        Error    := M 102.0,
        Done     := M 100.1,
        State    := MW 104,
        RD      := P#DB99.DBX0.0 DINT 1 // freier Speicherplatz
);

```

Nach erfolgreichem FB 5-Aufruf kann über die zurückgegebene Adresse des FB 5-Parameter ("VarToken") mittels FB 3 geschrieben werden.

```
CALL FB 3, DB 112(  
    Req           := M 200.0,  
    NumVar        := 1,           //1 GUD-Variable schreiben  
    Addr1         := DB_GUDVAR.GUDVar1Token,  
    Error         := M 102.0,  
    Done          := M 100.1,  
    State         := MW 104,  
    SD1           := P#DB99.DBX0.0 DINT 1);
```

## Aufrufbeispiel 2

Lesen einer GUD-Variablen aus Kanal 1 mit dem Namen "GUD\_STRING" (Typ-Definition der Variablen: STRING mit Länge 30 Byte). Die anwenderdefinierte Variable soll in einen 10 Byte Variablen-Zeiger konvertiert werden für nachfolgendes Schreiben mit dem FB 3.

### Aufruf und Parametrierung des FB 5 mit dem Instanz-DB 111:

```
// Datenbaustein für GUD-Variable  
DATA_BLOCK DB_GUDVAR           //Zuordnung in Symbolliste vornehmen  
  
STRUCT  
    GUDVarS : STRING[32] := 'GUD_STRING';           //Name ist vom Anwender def.  
    GUDVarSToken :  
        STRUCT  
            SYNTAX_ID : BYTE;  
            bereich_u_einheit : BYTE;  
            spalte : WORD;  
            zeile : WORD;  
            bausteintyp : BYTE;  
            ZEILENANZAHL : BYTE;  
            typ : BYTE;  
            laenge : BYTE;  
        END_STRUCT ;  
  
    string_of_GUD : STRING[30];           // muss mindestens so lang sein wie  
                                           // die Definition von 'GUD_STRING'!  
    new_name : STRING[30] := 'GUD_123';  
END_STRUCT;  
  
BEGIN  
END_DATA_BLOCK
```

```

// freier Anwender DB, als Instanz für FB 5
DATA_BLOCK DB 111

        FB 5
BEGIN
END_DATA_BLOCK

// freier Anwender DB, als Instanz für FB 3
DATE_BLOCK DB 112

        FB 3
BEGIN
END_DATA_BLOCK

//Eine anwenderdefinierte Kanalvariable aus Kanal 1 soll gelesen werden
//mit einer Konvertierung in einen Variablen-Zeiger für ein
//nachfolgendes Schreiben dieser Variablen.

Function FC "VariablenCall" : VOID
U      E 7.7;           //freie Taste Maschinensteuertafel
S      M 100.0;        //Req aktivieren
U      M 100.1;        //Fertigmeldung Done
R      M 100.0;        //Auftrag beenden
U      E 7.6;           //Fehlerquittierung von Hand
U      M 102.0;        //Fehler steht an
R      M 100.0;        //Auftrag beenden

CALL FB 5, DB 111(
        Req      := M 100.0,      //Startflanke für Lesen
        Addr     := DB_GUDVAR.GUDVarS,
        Area     := B#16#2,      //Kanal-Variable
        Unit     := B#16#1,      //Kanal 1
        Index1   := 0,           //kein Feldindex
        Index2   := 0,           //kein Feldindex
        CnvtToken := TRUE,       //Umsetzung in 10 Byte Token
        VarToken := DB_GUDVAR.GUDVarSToken,
        Error    := M 102.0,
        Done     := M 100.1,
        State    := MW 104,
        RD       := DB_GUDVAR.string_of_GUD);

```



Nach erfolgreichem FB 5-Aufruf kann über die zurückgegebene Adresse des FB 5-Parameter ("VarToken") mittels FB 3 geschrieben werden.

```
CALL FB 3, DB 112(  
    Req           := M 200.0,  
    NumVar        := 1,           //1 GUD-Variable schreiben  
    Addr1         := DB_GUDVAR.GUDVarSToken,  
    Error         := M 102.0,  
    Done          := M 100.1  
    State         := MW 104,  
    SD1           := DB_GUDVAR.new_name);
```

## 12.15.6 FB 7: PI\_SERV2 (PI-Dienst anfordern)

### Funktion

Der Funktionsbaustein FB 7 hat bis auf die größere Anzahl an WVar-Parametern die gleiche Funktionalität wie der FB 4. Es wird daher empfohlen, den Funktionsbaustein FB 7 anstelle des FB 4 zu verwenden.

Zur ausführlichen Beschreibung zum Anfordern eines PI-Dienstes siehe Kapitel "FB 4: PI\_SERV PI-Dienst anfordern (Seite 1025)".

### Deklaration der Funktion

```
FUNCTION_BLOCK FB 7  
Var_INPUT  
    Req:           BOOL;  
    PIService:    ANY;  
    Unit:         INT;  
    Addr1:        ANY;  
    Addr2:        ANY;  
    Addr3:        ANY;  
    Addr4:        ANY;  
    WVar1:        WORD;  
    WVar2:        WORD;  
    WVar3:        WORD;  
    WVar4:        WORD;  
    WVar5:        WORD;  
    WVar6:        WORD;  
    WVar7:        WORD;  
    WVar8:        WORD;  
    WVar9:        WORD;
```

```

WVar10:      WORD;
WVar11:      WORD;
WVar12:      WORD;
WVar13:      WORD;
WVar14:      WORD;
WVar15:      WORD;
WVar16:      WORD;
END_VAR
VAR_OUTPUT
  Error :      BOOL;
  Done :      BOOL;
  State :      WORD;
END_VAR
    
```

**Erläuterung der Formalparameter**

Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL		Auftragsanforderung
PIService	E	ANY	[DBName].[VarName] Standard ist: "PI".[VarName]	PI Dienst Beschreibung
Unit	E	INT	1...	Bereichsnummer
Addr1 bis Addr4	E	ANY	[DBName].[VarName]	Referenz auf Strings Spezifikation, gemäß angewähltem PI-Dienst
WVar1 bis WVar16	E	WORD	1...	Integer oder Word-Variablen. Spezifikation gemäß angewähltem PI- Dienst,
Error	A	BOOL		Auftrag wurde negativ quittiert, bzw. konnte nicht ausgeführt werden
Done	A	BOOL		Auftrag wurde erfolgreich ausgeführt.
State	A	WORD		siehe Fehlerkennungen

**12.15.7 FB 9: MzuN Bedieneinheitenumschaltung**

**Funktion**

Dieser Baustein ermöglicht die Umschaltung mehrerer **Bedieneinheiten** (Bedientafeln HMI und / oder Maschinensteuertafeln MCP), die über ein Bussystem mit einer / mehreren Steuerungsbaugruppen NCU verbunden sind.

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafelfronten an mehreren NCUs, Dezentrale Systeme (B3)

Die **Schnittstelle** zwischen den einzelnen Bedieneinheiten und der NCU (PLC) ist die M : N-Schnittstelle im Datenbaustein DB 19. Der FB 9 arbeitet mit den Signalen dieser Schnittstellen.

Folgende **Grundfunktionen** werden neben der Initialisierung, der Lebenszeichenüberwachung und der Fehlerrouinen vom Baustein für die Bedieneinheitenumschaltung ausgeführt:

Tabellarische Übersicht der Funktionen:	
Grundfunktion	Bedeutung
HMI klopft an	HMI hat den Wunsch an eine NCU online zu gehen
HMI kommt	HMI verbindet sich mit einer NCU
HMI geht	HMI trennt sich von einer NCU
Verdrängung	HMI muss Verbindung zu einer NCU lösen
Bedienfocuswechsel im Serverbetrieb	Bedienfocus von einer auf die andere NCU umschalten
Aktiver / Passiver Bedienmodus	Bedienen und beobachten / nur beobachten
MCP-Umschaltung	MCP kann optional mit dem HMI umgeschaltet werden

### Kurzbeschreibung einiger wichtiger Funktionen

Aktiver-/ Passiver Bedienmodus:

Ein Online-HMI kann zwei verschiedene Bedienmodi haben:

Aktiv-Modus: Bediener kann bedienen und beobachten

Passiv-Modus: Bediener kann beobachten (nur Kopfzeile HMI)

Nach Umschaltung auf eine NCU fordert sie in der PLC der Online-NCU zuerst den aktiven Bedienmodus an. Falls zwei HMIs zu einer Zeit an einer NCU online sind, ist eine der beiden immer im aktiven, die andere im passiven Bedienmodus. Der Bediener kann den aktiven Bedienmodus an der passiven HMI per Tastendruck anfordern.

### MCP-Umschaltung

Zusammen mit dem HMI kann optional eine ihm zugeordnete MCP mit umgeschaltet werden. Voraussetzung ist, dass die MCP-Adresse im Parameter "mstt\_adress" der Konfigurationsdatei NETNAMES.INI der HMI 's eingetragen und "MCPEnable" = TRUE gesetzt ist. Die MCP des passiven HMI 's ist deaktiviert. Dadurch gibt es an einer NCU immer nur eine aktive MCP.

### Hochlaufbedingung

Um zu verhindern, dass bei einem Neustart der NCU, die davor zuletzt angewählte MCP aktiviert wird, muss beim Aufruf von FB 1 in OB 100 der Eingangsparameter "MCP1BusAdr" = **255** (Adresse 1. MCP) und "MCP1Stop" = **TRUE** (1. MCP ausschalten) gesetzt werden.

## Freigaben

Wenn von einer MCP auf eine andere umgeschaltet wird, dann bleiben eventuell eingeschaltete Vorschub- sowie Achsfreigaben erhalten.

---

### Hinweis

Die zur Zeit der Umschaltung betätigten Tasten wirken bis zur Aktivierung der neuen MCP (vom HMI, der anschließend aktiviert wird) weiter. Auch die Override-Stellungen für Vorschub, Spindel bleiben erhalten. Um die betätigten Tasten zu deaktivieren, ist bei fallender Flanke des Signals DB10.DBX104.0 (MSTT 1 bereit) das Eingangsabbild der Maschinensteuer-Signale auf nicht betätigte Signalpegel zu legen. Die Override-Stellungen sollten unverändert bleiben. Maßnahmen zur Deaktivierung der Tasten sind im PLC-Anwenderprogramm zu realisieren (siehe Beispiel "Override-Umschaltung").

Der Aufruf ist nur im zyklischen Programm OB 1 zulässig.

---

## Deklaration der Funktion

```

FUNKTION_BLOCK FB 9
VAR_INPUT
    Quit:          BOOL;          //Quittierung Alarme
    OPMixedMode:   BOOL:=FALSE;   //Mischbetrieb mit nicht M zu N fähigen OP
                                //deaktiviert
    ActivEnable:   BOOL:=TRUE;    //wird nicht unterstützt
    MCPEnable:     BOOL:=TRUE;    //MCP Umschaltung aktivieren
END_VAR
VAR_OUTPUT
    Alarm1:        BOOL;          //Alarm: Fehler HMI-Busadresse, Bustyp!
    Alarm2:        BOOL;          //Alarm: Keine Bestätigung HMI 1 offline!
    Alarm3:        BOOL;          //Alarm: HMI 1 geht nicht offline!
    Alarm4:        BOOL;          //Alarm: Keine Bestätigung HMI 2 offline!
    Alarm5:        BOOL;          //Alarm: HMI 2 geht nicht offline!
    Alarm6:        BOOL;          //Alarm: Anklopf- HMI geht nicht online!
    Report:        BOOL;          //Meldung: Lebenszeichenüberwachung
    ErrorMMC:      BOOL;          //Fehlererkennung HMI
END_VAR
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion MzuN:

Formalparameter der Funktion MzuN				
Signal	Art	Typ	Bemerkung	
Quit	E	BOOL	Quittierung Alarme	
OPMixedMode	E	BOOL	Mischbetrieb mit einem nicht M zu N fähigen OP	
ActivEnable	E	BOOL	Funktion wird nicht unterstützt. Bedienfeldumschaltung Verriegelung über MMCx_SHIFT_LOCK im DB 19	
MCPEnable	E	BOOL	MCP Umschaltung aktivieren:	
			<b>TRUE:</b>	MCP wird mit Bedienfeld umgeschaltet.
			<b>FALSE:</b>	MCP wird nicht mit Bedienfeld umgeschaltet. Hiermit kann eine MCP fest verbunden werden. Siehe auch MMCx_MSTT_SHIFT_LOCK im DB 19.
Alarm1	A	BOOL	Alarm: Fehler HMI -Busadresse, Bustyp!	
Alarm2	A	BOOL	Alarm: Keine Bestätigung HMI 1 offline!	
Alarm3	A	BOOL	Alarm: HMI 1 geht nicht offline!	
Alarm4	A	BOOL	Alarm: Keine Bestätigung HMI 2 offline!	
Alarm5	A	BOOL	Alarm: HMI 2 geht nicht offline!	
Alarm6	A	BOOL	Alarm: Anklopf- HMI geht nicht online!	
Report	A	BOOL	Meldung: Lebenszeichenüberwachung HMI	
ErrorMMC	A	BOOL	Fehlererkennung HMI	

### Hinweis

Der Baustein ist vom Anwenderprogramm aufzurufen. Hierbei ist ein Instanz-DB mit beliebiger Nummer vom Anwender beizustellen. Der Aufruf ist nicht Multi-Instanzfähig.

### Aufrufbeispiel für FB 9

```
CALL FB 9, DB 109 (  
Quit           := Fehler_Quitt,      //z.B. MCP-RESET  
OPMixedMode   := FALSE,  
ActivEnable   := TRUE,  
MCPEnable     := TRUE);           //Freigabe MCP Umschaltung
```

---

#### Hinweis

Der Eingangsparameter "MCPEnable" muss TRUE sein, um die MCP Umschaltung zu ermöglichen. Der Defaultwert dieser Parameter ist so geschaltet und muss beim Aufruf der Funktion nicht zusätzlich beschaltet werden.

---

### Alarmer, Fehler

Die Ausgangsparameter "Alarm1" bis "Alarm6" und "Report" existieren als Information in der PLC, dass bei M zu N Fehler aufgetreten sind, die durch die Alarmer 410900 - 410906 am HMI sichtbar werden.

Konnte eine HMI-Funktion nicht ausgeführt werden (bei der die Fehlermeldung nicht angezeigt werden kann), wird dies am Zustandsparameter "ErrorMMC" mit logisch 1 angezeigt (z. B. Fehler im Hochlauf, wenn keine Verbindung aufgebaut wird).

### Aufrufbeispiel für FB 1 (Aufruf im OB 100)

```
CALL "RUN_UP" , "gp_par" (  
    MCPNum           := 1,  
    MCP1In           := P#E 0.0,  
    MCP1Out          := P#A 0.0,  
    MCP1StatSend     := P#A 8.0,  
    MCP1StatRec      := P#A 12.0,  
    MCP1BusAdr       := 255,           //Adresse 1. MCP  
    MCP1Timeout      := S5T#700MS,  
    MCP1Cycl         := S5T#200MS,  
    MCP1Stop         := TRUE,         //MCP ausgeschaltet  
    NCCyclTimeout    := S5T#200MS,  
    NCRunupTimeout   := S5T#50S);
```

### Beispiel: Override-Umschaltung

```
//verwendete Hilfsmerker M100.0, M100.1, M100.2, M100.3  
//Flanke positiv von MCP1Ready muss Override prüfen  
//und Maßnahmen für Aktivierung  
//MCP Baustein einleiten  
//Dieses Beispiel gilt für Vorschub-Override;  
//für Spindel-Override sind Nahtstellen- und Eingangsbyte auszutauschen.  
U    DB10.DBX 104.0;    //MCP1Ready  
FN   M    100.0;       //Flankenmerker 1  
SPBN weil;  
S    M    100.2;       //Hilfsmerker 1 setzen  
R    M    100.3;       //Hilfsmerker 2 rücksetzen
```

```
//Override speichern  
    L DB21.DBB 4;       //Nahtstelle Vorschub-Override  
    T EB 28;           //Zwischenspeicher (freies Eingangs-  
                       //oder Merkerbyte)  
  
weil:  
U    M    100.2;       //Umschaltung erfolgt  
O    DB10.DBX 104.0;   //MCP1Ready  
SPBN weil2;  
U    DB10.DBX 104.0;   //MCP1Ready  
FP   M    100.1;       //Flankenmerker 2  
SPB  weil2;  
U    M    100.2;       //Umschaltung erfolgt
```

```

R      M      100.2;           //Hilfsmerker 1 rücksetzen
SPB wei2;
U      M      100.3;           //Vergleich ist erfolgt
SPB MCP;                       //MCP-Programm aufrufen
//gespeicherten Override auf Nahtstelle der umgeschalteten MCP lenken
//bis die Override-Werte übereinstimmen
      L EB 28;                 //Zwischenspeicher auf
      T DB21.DBB 4;           //Nahtstelle Override lenken
      L EB 3;                 //Override-Eingangsbyte für Vorschub
      <>i;                       //Übereinstimmung?
SPB wei2;                       //nein, Absprung
S      M      100.3;           //ja, Hilfsmerker 2 setzen
//nach Übereinstimmung der Override-Werte MCP-Programm wieder aufrufen
MCP: CALL "MCP_IFM" (          //FC 19
      BAGNo      := B#16#1,
      ChanNo     := B#16#1,
      SpindleIFNo := B#16#0,
      FeedHold   := M 101.0,
      SpindleHold := M 101.1);
wei2: NOP      0;

```

## 12.15.8 FB 10: Sicherheits-Relais (SI-Relais)

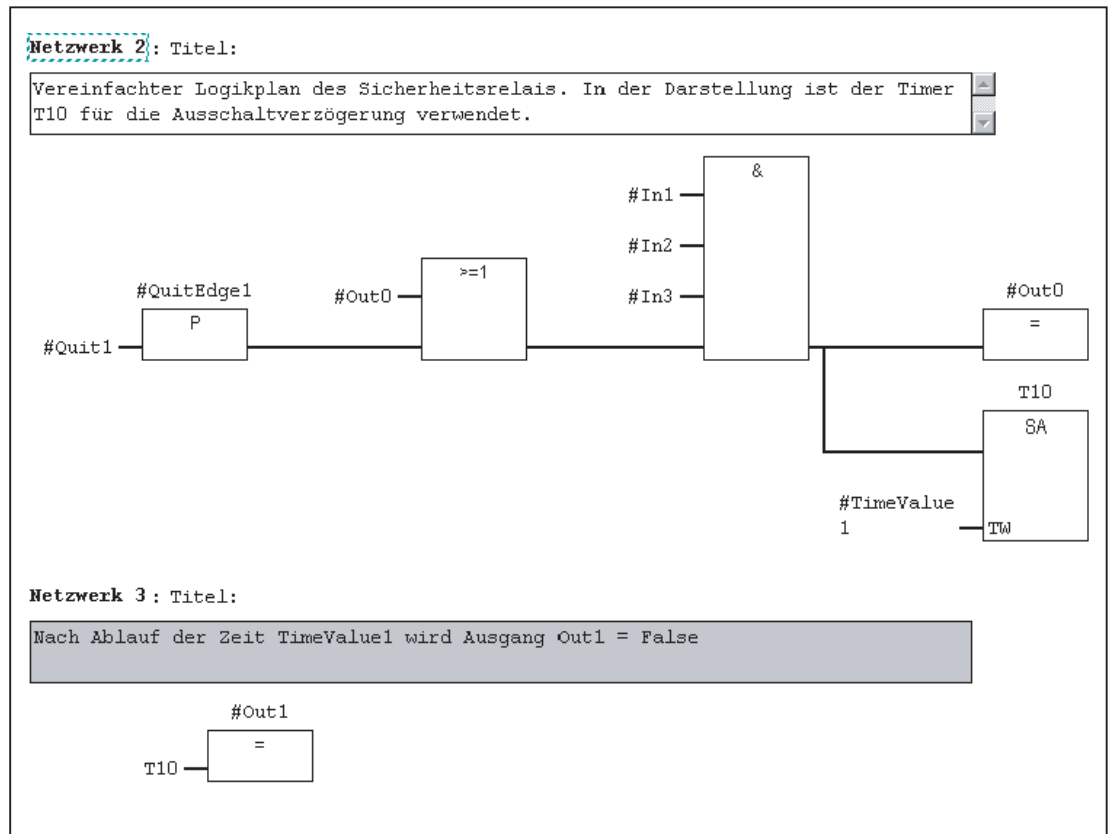
### Funktion

Der SPL-Baustein "Sicherheitsrelais" für "Safety Integrated" ist das Äquivalent auf der PLC zu der gleichnamigen NC-Funktion. Der Standard-SPL-Baustein "Sicherheitsrelais" ist für die Anforderungen einer Not-Halt-Realisierung mit sicherer programmierbarer Logik ausgelegt. Er kann aber auch für weitere ähnliche Anforderungen, z. B. einer Schutztür-Ansteuerung, genutzt werden. Die Funktion enthält 3 Eingangsparameter (In1, In2, In3). Bei Schalten eines dieser Parameter auf den Wert 0, wird der Ausgang Out0 unverzüglich abgeschaltet und die Ausgänge Out1, Out2 und Out3 über die parametrisierten Zeitwerte (Parameter TimeValue1, TimeValue2, TimeValue3) verzögert abgeschaltet. Die Ausgänge werden unverzüglich wieder eingeschaltet, wenn die Eingänge In1 bis In3 den Wert 1 annehmen und an einem der Quittierungseingänge Quit1, Quit2 ein positiver Flankenwechsel erkannt wird. Um die Ausgänge in Grundstellung (Werte = 0) nach Neueinschalten zu bringen, ist der Parameter "FirstRun" wie folgt zu beschalten. Der Parameter "FirstRun" muss beim 1. Durchlauf nach Hochlauf der Steuerung über ein remanentes Datum (Merkerbit, Bit im Datenbaustein) auf den Wert TRUE geschaltet sein. Das Datum kann z. B. im OB 100 vorbesetzt werden. Der Parameter wird nach dem erstmaligen Durchlauf des FB 10 auf den Wert FALSE zurückgesetzt. Für jeden Aufruf mit eigener Instanz ist für den Parameter "FirstRun" ein separates Datum zu verwenden.



## Vereinfachtes Prinzipschaltbild im Funktionsplan

In der folgenden Darstellung sind nur ein Quittiereingang Quit1 und ein ausschaltverzögerter Ausgang Out1 dargestellt. Schaltung für Quit2 und die weiteren ausschaltverzögerten Ausgänge sind identisch realisiert. In dem Funktionsplan fehlt auch der Parameter FirstRun. Die Wirkungsweise ist oben beschrieben.



## Deklaration der Funktion

```

FUNCTION_BLOCK FB 10
VAR_INPUT
    In1 : BOOL      := TRUE;           //Input 1
    In2 : BOOL      := TRUE;           //Input 2
    In3 : BOOL      := TRUE;           //Input 3
    Quit1:         : BOOL;             //Quit 1 Signal
    Quit2:         : BOOL;             //Quit 2 Signal
    TimeValue1:    TIME := T#0ms;      //TimeValue for Output 1
    TimeValue2:    TIME := T#0ms;      //TimeValue for Output 2
    TimeValue3:    TIME := T#0ms;      //TimeValue for Output 3
END_VAR

```

```

VAR_OUTPUT
    Out0          : BOOL;           //Output without Delay
    Out1          : BOOL;           //Delayed Output to False by Timer 1
    Out2          : BOOL;           //Delayed Output to False by Timer 2
    Out3          : BOOL;           //Delayed Output to False by Timer 3
END_VAR
VAR_INOUT
    FirstRun      : BOOL;           //TRUE by User after 1. Start of SPL
END_VAR
    
```

**Erläuterung der Formalparameter**

Die folgende Tabelle zeigt alle Formalparameter der Funktion SI-Relais:

Formalparameter der Funktion SI-Relais			
Signal	Art	Typ	Bemerkung
In1	E	BOOL	Eingang 1
In2	E	BOOL	Eingang 2
In3	E	BOOL	Eingang 3
Quit1	E	BOOL	QuittierEingang 1
Quit2	E	BOOL	QuittierEingang 2
TimeValue1	E	TIME	Zeit-Wert 1 für Ausschaltverzögerung
TimeValue2	E	TIME	Zeit-Wert 2 für Ausschaltverzögerung
TimeValue3	E	TIME	Zeit-Wert 3 für Ausschaltverzögerung
Out0	A	BOOL	Ausgang unverzögert
Out1	A	BOOL	Ausgang verzögert mit TimeValue1
Out2	A	BOOL	Ausgang verzögert mit TimeValue2
Out3	A	BOOL	Ausgang verzögert mit TimeValue3
FirstRun	E/A	BOOL	Aktivierung der Grundstellung

**Hinweis**

Der Baustein ist vom Anwenderprogramm einmalig (pro SI-Relais) im OB1-Zyklus aufzurufen ab dem Starten des SPL-Programms. Hierbei ist ein Instanz-DB mit beliebiger Nummer vom Anwender beizustellen. Der Aufruf ist Multi-Instanzfähig.

## 12.15.9 FB 11: Bremsentest

### Funktion

Die Funktionsprüfung der Bremsenmechanik sollte für alle Achsen verwendet werden, die durch eine Haltebremse gegen Bewegungen im ungeregelten Betrieb gesichert werden müssen. Der Hauptanwendungsfall sind dabei die sogenannten "hängenden Achsen".

In seinem PLC-Anwenderprogramm kann der Maschinenhersteller zu einem geeigneten Zeitpunkt (Richtwert alle 8 h, wie auch beim SI-Teststopp) die Bremse schließen und vom Antrieb ein zusätzliches Moment / eine zusätzliche Kraft zu der Gewichtskraft der Achse aufbringen lassen. Im fehlerfreien Zustand kann die Bremse das notwendige Bremsmoment / die Bremskraft aufbringen, die Achse wird sich kaum bewegen. Im Fehlerfall wird ein Verlassen des parametrierten Überwachungsfensters für den Positionswert erkannt. In diesem Fall wird durch den Lageregler ein Absacken der Achse verhindert und die Funktionsprüfung der Bremsenmechanik negativ quittiert.

Die notwendige Parametrierung von NC und Antrieb ist beschrieben in:

#### Literatur:

Funktionshandbuch Safety Integrated

Der Start des Bremsentests muss immer im Stillstand der Achse erfolgen. Für den gesamten Zeitraum des Bremsentests sind die Freigabesignale der parametrierten Achse auf Freigabe zu setzen (z.B. die Signale Reglersperre, Vorschubfreigabe). Weiterhin ist das Signal an Achse/Spindel DB31, ... .DBX28.7 (PLC kontrolliert Achse) während der gesamten Dauer des Tests auf Zustand 1 vom Anwenderprogramm zu setzen.

Vor Aktivierung des Signals DB31, ... .DBX28.7 (PLC kontrolliert Achse) ist die Achse als "neutrale Achse" zu schalten, z. B. ist DB31, ... .DBX8.0 - 8.3 (NC-Achse Kanal zuordnen) auf Kanal 0 zu setzen sowie DB31, ... .DBX8.4 (Aktiviersignal bei Veränderung dieses Bytes) zu setzen.

Die Rückmeldung:

- über den aktuellen Zustand kann abgefragt werden in DB31, ... .DBB68.
- der NC über das Signal DB31, ... .DBX63.1 (PLC kontrolliert Achse) ist abzuwarten bevor der Baustein gestartet wird. Die Richtung, in die der Antrieb sein Moment / seine Kraft aufbringt, wird von der PLC durch eine "Verfahrbewegung", z. B. über den FC 18 vorgegeben.

Der Zielpunkt dieser Verfahrbewegung muss gefahrlos erreicht werden können (keine Kollision), falls die Bremse das notwendige Moment / die Kraft nicht aufbringen kann.

---

#### Hinweis

##### Hinweis zu FC 18

Falls FC 18 für die gleiche Achse im weiteren Anwenderprogramm aufgerufen werden, müssen die Aufrufe gegeneinander verriegelt werden. Dies kann z. B. über einen gemeinsamen Aufruf dieser Funktion mit einer verriegelten gemeinsamen Datenschnittstelle für die Parameter des FC 18 erfolgen. Eine 2. Möglichkeit ist der Mehrfachaufruf des FC 18, wobei der inaktive FC 18 programmäßig nicht durchlaufen wird. Eine Verriegelung für Mehrfachnutzung ist vorzusehen.

---

Der Bremsentest gliedert sich in folgende Schritte:

Bremsentest-Gliederung		
Schritt	Erwartete Rückmeldung	Überwachungs-Zeitwert
Bremsentest starten	DBX 71.0 = 1	TV_BTactiv
Bremse schließen	Bclosed = 1	TV_Bclose
Fahrbefehl ausgeben	DBX 64.6 Or DBX 64.7	TV_FeedCommand
Prüfung Fahrbefehl ausgegeben	DBX 62.5 = 1	TV_FXSreached
Haltezeit abwarten	DBX 62.5 = 1	TV_FXShold
Abwahl Bremsen Test / Bremsen öffnen	DBX 71.0 = 0	TV_BTactiv
Test ok ausgeben		

### Deklaration der Funktion

```

Function_BLOCK FB 11
VAR_INPUT
    Start          : BOOL;           //Start for Braketest
    Quit           : BOOL;           //Quit Error
    Bclosed        : BOOL;           //brake closed input (single channel - PLC)
    Axis           : INT;            //testing axisno.
    TimerNo        : TIMER;         //Timer from User
    TV_BTactiv     : S5TIME;        //TimeValue -> braketest activ
    TV_Bclose      : S5TIM;         //TimeValue -> close Brake
    TV_FeedCommand : S5TIME;        //TimeValue -> force FeedCommand
    TV_FXSreached  : S5TIME;        //TimeValue -> Fixed stop reached
    TV_FXShold     : S5TIME;        //TimeValue -> test brake
END_VAR
VAR_OUTPUT
    CloseBrake     : BOOL;           //Signal Close brake
    MoveAxis       : BOOL;           //do move axis
    Done           : BOOL;
    Error          : BOOL;
    State          : BYTE;           //Errorbyte
END_VAR
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion Bremsentest:

Formalparameter der Funktion Bremsentest			
Signal	Art	Typ	Bemerkung
Start	E	BOOL	Start des Bremsentests
Quit	E	BOOL	Fehler Quittierung
Bclosed	E	BOOL	Rückmelde-Eingang ob Bremse schließen angesteuert ist (einkanalig - PLC)
Axis	E	INT	<b>Achsnummer der zu testenden Achse</b>
TimerNo	E	TIMER	Timer aus Anwenderprogramm
TV_BTactiv	E	S5TIME	Überwachungs-Zeitwert → Bremsentest aktiv, Prüfung des Achs-Signals DBX 71.0
TV_Bclose	E	S5TIME	Überwachungs-Zeitwert → Bremse schließen. Prüfung des Eingangssignals Bclosed nachdem Ausgang CloseBrake gesetzt wurde
TV_FeedCommand	E	S5TIME	Überwachungs-Zeitwert → Fahrbefehl ausgegeben Prüfung Fahrbefehl nachdem MoveAxis gesetzt wurde
TV_FXSreched	E	S5TIME	Überwachungs-Zeitwert → Festanschlag erreicht
TV_FXShold	E	S5TIME	Überwachungs-Zeitwert → Bremse testen
CloseBrake	A	BOOL	Anforderung für Bremse schließen
MoveAxis	A	BOOL	Anforderung Verfahrbewegung anstoßen
Done	A	BOOL	Test erfolgreich beendet
Error	A	BOOL	Fehler aufgetreten
State	A	BYTE	Fehler Status

## Fehlererkennungen

State	Bedeutung
0	Kein Fehler
1	Startbedingungen nicht gegeben, z. B. Achse nicht in Regelung / Bremse zu / Achsensperre steht an
2	Bei Anwahl Bremsentest erfolgt keine Rückmeldung der NC im Signal "Bremsentest aktiv"
3	Keine Rückmeldung "Bremse eingefallen" durch Eingangssignal Bclosed
4	Keine Fahrbefehlausgabe (z. B. Bewegung der Achse wurde nicht gestartet)
5	Festanschlag wird nicht erreicht → Achs-RESET wurde ausgelöst.
6	Verfahrsperr/Anfahrt zu langsam → Festanschlag kann nicht erreicht werden. Überwachungszeit TV FXSreached abgelaufen.
7	Bremse hält überhaupt nicht (Endposition wird erreicht) / Anfahrsgeschwindigkeit zu groß
8	Bremse bricht während der Haltezeit auf
9	Fehler bei Bremsentest-Abwahl
10	interner Fehler
11	Signal "PLC kontrolliert Achse" nicht eingeschaltet vom Anwenderprogramm

---

**Hinweis**

Der Baustein ist vom Anwenderprogramm aufzurufen. Hierbei ist ein Instanz-DB mit beliebiger Nummer vom Anwender beizustellen. Der Aufruf ist Multi-Instanzfähig.

---

**Aufrufbeispiel für FB 11:**

```
UN      M      111.1;    //Anforderung Bremse schließen Z-Achse von FB
=       A      85.0;    //Ansteuerung Bremse Z-Achse
AUF     Axis3";    //Bremsentest Z-Achse
O       E      73.0;    //Anstoß für den Bremsentest Z-Achse
O       M      110.7;   //Bremsentest läuft
FP      M      110.0;
UN      M      111.4;   //Fehler aufgetreten
S       M      110.7;   //Bremsentest läuft
S       M      110.6;   //nächster Schritt
SPBN    m001
L       DBB    68;
UW      W#16#F;
T       MB     115;    //Kanalzustand merken
L       B#16#10
T       DBB    8;     //Neutrale Achse anfordern
m001:  U       DBX    68.6; //Rückmeldung Achse ist neutral
        U       M      110.6;
        FP      M      110.1;
        R       M      110.6;
        S       M      110.5; //nächster Schritt
        S       DBX    28.7; //PLC-kontrollierte Achse anfordern

        U       DBX    63.1; //Rückmeldung Achse wird von PLC kontrolliert
        U       M      110.5;
        FP      M      110.2;
        R       M      110.5;
        S       M      111.0; //Start Bremsentest für FB
```

```

CALL FB 11 , DB 211 (//Bremsentestbaustein
    Start           := M 111.0, //Start des Bremsentests
    Quit            := E 3.7, //Fehlerquittierung mit RESET-Taste
    Bclosed         := E 54.0, //Rückmeldung Bremse schließen
                    //angesteuert
    Axis            := 3, //Achsnnummer der zu testenden Achse
                    //Z-Achse
    TimerNo         := T 110, //Timernummer
    TV_BTactiv      := S5T#200MS, //Überwachungs-Zeitwert:
                    //Bremsentest aktiv DBX71.0
    TV_Bclose       := S5T#1S, //Überwachungs-Zeitwert:
                    //Bremse geschlossen
    TV_FeedCommand := S5T#1S, //Überwachungs-Zeitwert:
                    //Fahrbefehl ausgegeben
    TV_FXSreache    := S5T#1S, //Überwachungs-Zeitwert:
                    //Festanschlag erreicht
    TV_FXShold      := S5T#2S, //Überwachungs-Zeitwert:
                    //Testzeit Bremse
    CloseBrake      := M 111.1, //Anforderung Bremse schließen
    MoveAxis        := M 111.2, //Anforderung Verfahrbewegung //anstoßen
    Done            := M 111.3, //Test erfolgreich beendet
    Error           := M 111.4, //Fehler aufgetreten
    State           := MB 112); //Fehler Status

AUF "Axis3"; //Bremsentest Z-Achse

U M 111.2; //Moveaxis
FP M 111.5; //FC18 Start
S M 111.7; //Start FC18

O M 111.3; //Test erfolgreich beendet
O M 111.4; //Fehler aufgetreten
FP M 110.3;
R DBX 28.7; //Anforderung PLC-kontrollierte Achse

UN DBX 63.1; //Rückmeldung Achse wird von PLC kontrolliert
U M 111.0; //Start Bremsentest für FB
U M 110.7; //Bremsentest läuft
FP M 110.4;
R M 111.0; //Start Bremsentest für FB
R M 110.7; //Bremsentest läuft

```

12.15 Bausteinbeschreibungen

```
//optional begin
    SPBN    m002;
    L       MB        115;          //alter Kanalzustand
    OW      W#16#10;
    T       DBB       8;           //Kanal Achse anfordern
m002:     NOP      0;
//optional end

    CALL "SpinCtrl" (//Z-Achse verfahren
        Start      := M        111.2,          //Start des Verfahrwegbewegung
        Stop        := FALSE,
        Funct       := B#16#5,                //Mode: Achsbetrieb
        Mode        := B#16#1,                //Verfahren: Inkrementell
        AxisNo      := 3,                     //Achsnnummer der zu verfahrenenden
                                                //Achse Z-Achse
        Pos         := -5.000000e+000,        //Verfahrweg: minus 5 mm
        FRate       := 1.000000e+003,        //Vorschub: 1000 mm/min
        InPos       := M        113.0,        //Position erreicht
        Error       := M        113.1,        //Fehler aufgetreten
        State       := MB        114);        //Fehler Status

    AUF      "Axis3"; //Bremsentest Z-Achse
    U       M        113.0; //Position erreicht
    O       M        113.1; //Fehler aufgetreten
    FP      M        113.2;
    R       M        111.7; //Start FC18
```



## 12.15.10 FB 29: Diagnose Signalrekorder und Datentrigger

### Funktion

#### Signalrekorder

Mit dem FB Diagnose besteht die Möglichkeit verschiedene Diagnosen am PLC-Anwenderprogramm durchzuführen. Ein Diagnosefall ist das Protokollieren von Signalzuständen und Signaländerungen. Bei diesem Diagnosefall wird die Funktionsnummer 1 dem Parameter "Func" zugeordnet. Bis zu 8 Signale der Parameter "Signal\_1" bis "Signal\_8" werden bei Änderung von einem dieser Signale in einem Ringpuffer aufgezeichnet. Zusätzlich werden noch die aktuellen Informationen der Parameter "Var1" als BYTE-Wert sowie "Var2", "Var3" als INTEGER-Werte im Ringpuffer mit abgelegt.

Weiterhin wird die Anzahl der vergangenen OB 1-Zyklen als Zusatzinformation im Ringpuffer abgelegt. Hiermit ist eine graphische Auswertung von Signalen und Werten im OB 1-Zyklusraster möglich. Beim ersten Aufruf des FB Diagnose innerhalb eines neuen PLC-Zyklus muss der Parameter "NewCycle" auf TRUE parametrieren sein. Bei mehrfachen Aufrufen des FB Diagnose in einem OB 1-Zyklus muss beim zweiten und folgenden Aufruf der Parameter "NewCycle" FALSE sein. Hiermit wird verhindert, dass eine neue Anzahl von OB 1-Zyklen berechnet wird.

Der Ringpuffer wird vom Anwender zur Verfügung gestellt muss eine ARRAY-Struktur wie im Quellcode angegeben haben. Die Anzahl der ARRAY-Elemente ist beliebig. Empfohlen wird eine Größe von 250 Elementen. Über den Parameter "ClearBuf" wird der Ringpuffer gelöscht und der Zeiger "BufAddr" (Ein-/Ausgangsparameter) auf den Anfang gesetzt. Der zugehörige Instanz-DB zum FB ist ein DB aus dem Anwenderbereich und ist an dem FB Diagnose mit dem Parameter "BufDB" zu übergeben.

#### Datentrigger

Die Funktion "Datentrigger" soll eine Triggerung auf bestimmte Werte (auch Bits) an einer beliebigen zulässigen Speicherzelle ermöglichen. Hierbei wird die zu triggernde Zelle mit einer Bitmaske (Parameter "AndMask") vor dem Vergleich des Parameters "TestVal" in dem Diagnose-Baustein "verundet".

---

#### Hinweis

Die Funktion steht als Quelle im Quellcontainer der Grundprogramm-Bibliothek mit Namen "Diagnose.awl" zur Verfügung. In diesem Quellbaustein sind auch Instanz-DB und Ringpuffer-DB definiert. Weiterhin ist der Aufruf der Funktion dargestellt. Hierbei sind die DB-Nummern und auch der Aufruf anzupassen.

---

## Deklaration der Funktion

```
FUNCTION_BLOCK FB 29
VAR_INPUT
Func          : INT;          //Function number: 0 = No Function,
                               //1 = Signalrecorder, 2 = Datatrigger
    Signal_1   : BOOL;        //Start for Braketest
    Signal_2   : BOOL;
    Signal_3   : BOOL;
    Signal_4   : BOOL;
    Signal_5   : BOOL;
    Signal_6   : BOOL;
    Signal_7   : BOOL;
    Signal_8   : BOOL;
    NewCycle   : BOOL;
    Var1       : BYTE;
    Var2       : INT;
    Var3       : INT;
    BufDB      : INT;
    ClearBuf   : BOOL;
    DataAdr    : POINTER;     //Area pointer to testing word
    TestVal    : WORD;        //Value for triggering
    AndMask    : WORD;        //AND- Mask to the testing word
END_VAR
VAR_OUTPUT
    TestIsTrue : BOOL;
END_VAR
VAR_IN_OUT
    BufAddr    : INT;
END_VAR
```

## Struktur Ringpuffer

```
TITLE =  
  
//Ring buffer-DB for FB 29  
VERSION : 1.0  
  
STRUCT  
    Feld : ARRAY [0 .. 249 ] OF STRUCT    //can be any size of this struct  
  
    Cycle : INT;                          //Delta cycle to last storage in buffer  
    Signal_1 : BOOL;                       //Signal names same as FB 29  
    Signal_2 : BOOL;  
    Signal_3 : BOOL;  
    Signal_4 : BOOL;  
    Signal_5 : BOOL;  
    Signal_6 : BOOL;  
    Signal_7 : BOOL;  
    Signal_8 : BOOL;  
    Var1 : BYTE;  
    Var2 : WORD;  
    Var3 : WORD;  
    END_STRUCT;  
END_STRUCT;  
BEGIN  
END_DATA_BLOCK
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion Diagnose:

Formalparameter der Funktion Diagnose				
Signal	Art	Typ	Wertebereich	Bemerkung
Func	E	INT	0, 1, 2	Funktion
				0: Abschalten
				1: Signalrekorder
				2: Datentrigger
Parameter für Funktion 1				
Signal_1 bis Signal_8	E	BOOL		Bit-Signale, die auf Änderung geprüft werden
NewCycle	E	BOOL		Siehe obige Beschreibung "Signalrekorder"
Var1	E	BYTE		Zusatzwert
Var2	E	INT		Zusatzwert
Var3	E	INT		Zusatzwert
BufDB	E	INT		Ringpuffer DB-Nr.
ClearBuf	E	BOOL		Löschen des Ringpuffer DB mit Rücksetzen des Zeiger BufAddr
BufAddr	E/A	INT		Zielbereich für gelesene Daten
Parameter für Funktion 2				
DataAdr	E	POINTER		Zeiger auf zu prüfendes Wort
TestVal	E	WORD		Vergleichswert
AndMask	E	WORD		siehe Beschreibung
TestIsTrue	A	BOOL		Ergebnis des Vergleichs

### Projektierungsschritte

- Funktion des Diagnosebausteins auswählen.
- Festlegung der geeigneten Daten für die Aufzeichnung als Signalrekorder bzw. der Datentriggerung.
- Geeignete Stelle(n) im Anwenderprogramm suchen für den Aufruf des FB Diagnose.
- Erzeugung eines Datenbausteins für den Ringpuffer, siehe Aufrufbeispiel.
- Aufruf des FB Diagnose im Anwenderprogramm mit Parametern.

Bei der Funktion 1 ist ein Löschen des Ringpuffers mit dem Parameter "ClearBuf" sinnvoll. Nach Beendigung der Aufzeichnung mit Funktion 1 wird der Ringpuffer-DB über STEP 7 mit der Funktion "Datenbaustein öffnen in Datenansicht" ausgelesen. Die Inhalte des Ringpuffer-DB können nun analysiert werden.

## Aufrufbeispiel

```
FUNCTION FC 99: VOID
TITLE =
VERSION : 0.0

BEGIN
NETWORK
TITLE = NETWORK

CALL FB 29 , DB 80 (
Func                := 1,
    Signal_1        := M      100.0,
    Signal_2        := M      100.1,
    Signal_3        := M      100.2,
    Signal_4        := M      100.3,
    Signal_5        := M      10.4,
    Signal_6        := M      100.5,
    Signal_7        := M      100.6,
    Signal_8        := M      100.7,
    NewCycle        := TRUE,
    Var1            := MB      100,
    BufDB           := 81,
    ClearBuf        := M      50.0);
END_FUNCTION
```

### 12.15.11 FC 2: GP\_HP Grundprogramm, zyklischer Teil

#### Funktion

Die komplette Bearbeitung der NCK-PLC-Nahtstelle erfolgt im zyklischen Betrieb. Um die Grundprogrammlaufzeit gering zu halten, werden nur die Steuer-/Statussignale zyklisch übertragen, die Hilfs- und G-Funktionsübergabe wird nur auf Anforderung vom NCK bearbeitet.

## Deklaration

```
FUNCTION FC 2: VOID
//keine Parameter
```

## Aufrufbeispiel

Das Grundprogramm muss - zeitlich gesehen - **vor** der Bearbeitung des Anwenderprogramms durchlaufen werden. Deshalb ist es als erstes im OB 1 aufzurufen.

Das vorliegende Beispiel enthält die Standard-Deklarationen für den OB 1 und die Aufrufe für das Grundprogramm (FC 2), den Transfer der MCP-Signale (FC 19) sowie die Erfassung von Fehler- und Betriebsmeldungen (FC 10).

```
ORGANIZATION_BLOCK OB 1
VAR_TEMP
    OB1_EV_CLASS :      BYTE;
    OB1_SCAN_1 :      BYTE;
    OB1_PRIORITY :     BYTE;
    OB1_OB_NUMBR :     BYTE;
    OB1_RESERVED_1 :   BYTE;
    OB1_RESERVED_2 :   BYTE;
    OB1_PREV_CYCLE :   INT;
    OB1_MIN_CYCLE :    INT;
    OB1_MAX_CYCLE :    INT;
    OB1_DATE_TIME :    DATE_AND_TIME;
END_VAR
BEGIN
CALL FC 2;                //Aufruf Grundprogramm als 1. FC
//HIER ANWENDERPROGRAMM EINFUEGEN
CALL FC 19(                //MCP-Signale an Nahtstelle
    BAGNo := B#16#1,        //BAG Nr. 1
    ChanNo := B#16#1,       //Kanal Nr. 1
    SpindleIFNo := B#16#4,   //Spindel Interface Nummer = 4
    FeedHold := m22.0,       //Vorschub Halt Signal
                                //selbsthaltend
    SpindleHold := db2.dbx151.0); //Spindel Halt selbsthaltend
                                //in Meldungs-DB
CALL FC 10(                //Fehler- und Betriebsmeldungen
    ToUserIF := TRUE,       //Signale aus DB 2 an Nahtstelle
                                //übertragen
    Quit := E6.1);         //Quittierung der Fehlermeldungen
                                //über E 6.1
END_ORGANIZATION_BLOCK
```

## 12.15.12 FC 3: GP\_PRAL Grundprogramm, alarmgesteuerter Teil

### Funktion

Im alarmgesteuerten Teil des Grundprogramms erfolgt die Bearbeitung der satzsynchronen Übergaben von NCK an PLC mittels Hilfs- und G-Funktionen. Bei den **Hilfsfunktionen** wird zwischen normalen und schnellen Hilfsfunktionen unterschieden.

Schnelle Hilfsfunktionen eines NC-Satzes werden zwischengespeichert und die Übernahme an die NC quittiert. Zu Beginn des nächsten OB 1-Zyklus werden diese an die Anwendernahtstelle übergeben.

Schnelle Hilfsfunktionen, die sofort hintereinander programmiert sind, gehen nicht für das Anwenderprogramm verloren. Hierfür sorgt ein Mechanismus im Grundprogramm.

Normale Hilfsfunktionen werden erst dann an die NC quittiert, wenn an diesen eine Zyklusdauer angestanden hat. Damit hat der Anwender bei Bedarf die Möglichkeit, eine Einlesesperre an die NC zu geben.

Die **G-Funktionen** werden sofort ausgewertet und an die Anwendernahtstelle übergeben.

### NC-Prozessalarme

Bei Auslösung des Interrupts durch die NC (ist in jedem IPO-Takt möglich), wird ein Bit im Lokaldatum des OB 40 ("GP\_IRFromNCK") vom Grundprogramm nur dann gesetzt, wenn FB 1 Parameter "UserIR": = TRUE ist. Bei anderen Ereignissen (Prozessalarme durch die Peripherie), ist dieses Datum nicht gesetzt. Mit dieser Information kann in die zugehörige Interruptroutine im Anwenderprogramm verzweigt werden, um die notwendigen Aktionen einzuleiten.

Einer schnellen auftragsgesteuerten Bearbeitung vom Anwenderprogramm für die Maschine, stehen folgende NC-Funktionen in der Interruptbearbeitung (OB 40-Programmteil) für das PLC-Anwenderprogramm zur Verfügung:

- Ausgewählte **Hilfsfunktionen**
- **Werkzeugwechsel-Funktion** bei der Option Werkzeugverwaltung
- **Position erreicht** bei den Positionierachsen, Teilungsachsen und Spindeln bei Aktivierung über PLC

Diese oben aufgeführten Funktionen müssen durch das Anwenderprogramm im OB 40 ausgewertet werden, um schnelle Reaktionen an der Maschine einzuleiten. Hier kann z. B. bei der Programmierung eines T-Befehls an einer Drehmaschine das Revolver-Schaltwerk aktiviert werden.

Weitere Einzelheiten zur Programmierung von Prozessalarmen, Zeitverzögerung, Unterbrechbarkeit, usw. sind in der entsprechenden SIMATIC-Dokumentation nachzulesen.

## Hilfsfunktionen

Generell gilt, dass schnelle oder quittierende Hilfsfunktionen unabhängig von einer Zuordnung mit oder ohne Interruptsteuerung bearbeitet werden.

Durch Grundprogrammparameter im FB 1 kann ausgewählt werden, welche Hilfsfunktionen (T, H, DL) vom Anwenderprogramm nur interruptgesteuert bearbeitet werden sollen.

Die nicht über Interrupt zugeordneten Funktionen werden wie bisher erst durch das zyklische Grundprogramm zur Verfügung gestellt. Hierbei stehen die Änderungssignale der Funktionen einen PLC-Zyklus an.

Wenn die Auswahl für die Hilfsfunktionsgruppen (T, H, DL) mit Interruptsteuerung getroffen ist, dann kann für die ausgewählten Funktionen nur noch eine Interruptbearbeitung durch das Anwenderprogramm erfolgen.

Für das Anwenderprogramm wird kanalorientiert ein Bit in dem Lokaldatum "GP\_AuxFunction" gesetzt (wenn "GP\_AuxFunction[1]" gesetzt ist, dann steht für den 1. Kanal eine Hilfsfunktion bereit).

Im zugehörigen Kanal-DB stehen Änderungssignal und Funktionswert für den Anwender zur Verfügung. Das Änderungssignal dieser interruptgesteuerten Funktion wird im zyklischen Grundprogrammteil nach Ablauf von mindestens einem vollem OB 1-Zyklus (max. ca. zwei OB 1-Zyklen) wieder auf Null zurückgesetzt.

## Werkzeugwechsel

Bei der Option Werkzeugverwaltung werden der Werkzeugwechsel-Befehl für Revolver und der Werkzeugwechsel in die Spindel durch einen Interrupt unterstützt. Hierzu wird das Lokaldatenbit "GP\_TM" im OB 40 gesetzt. Damit kann das PLC-Anwenderprogramm den DB der Werkzeugverwaltung (DB 72 bzw. DB 73) auf die Werkzeugwechselfunktion abprüfen und den Wechselvorgang anstoßen.

## Position erreicht

In der Bitstruktur "GP\_InPosition" der Lokaldaten des OB 40 ist maschinenachsspezifisch (jedes Bit entspricht einer Achse/Spindel, z. B. GP\_InPosition[5] entspricht der 5. Achse) aufgebaut.

Wenn eine Funktion FC 18 (Spindelsteuerung, Positionierachse, Teilungsachse) für eine Achse oder Spindel aktiviert wurde, kann das zugehörige "GP\_InPosition"-Bit eine unverzögerte Auswertung des Signals "InPos" der oben aufgeführten FCs erreicht werden. Hierdurch können z. B. Klemmungen für eine Teilungsachse unverzögert aktiviert werden.

## Deklaration

```
FUNCTION FC 3: VOID  
//keine Parameter
```



## Aufrufbeispiel

Das Grundprogramm muss - zeitlich gesehen - **vor** der Bearbeitung weiterer alarmgesteuerter Anwenderprogramme durchlaufen werden. Deshalb ist es als erstes im OB 40 aufzurufen.

Das vorliegende Beispiel enthält die Standard-Deklarationen für den OB 40 und den Aufruf für das Grundprogramm.

```
ORGANIZATION_BLOCK OB 40
VAR_TEMP
    OB40_EV_CLASS :          BYTE;
    OB40_STRT_INF :          BYTE;
    OB40_PRIORITY :          BYTE;
    OB40_OB_NUMBR :          BYTE;
    OB40_RESERVED_1 :        BYTE;
    OB40_MDL_ID :            BYTE;
    OB40_MDL_ADDR :          INT;
    OB40_POINT_ADDR :        DWORD;
    OB40_DATE_TIME :          DATE_AND_TIME;

//Belegung durch das Grundprogramm
GP_IRFromNCK : BOOL;          //Interrupt durch NCK für Anwender
GP_TM : BOOL;                 //Werkzeugverwaltung
GP_InPosition : ARRAY [1..3] OF BOOL; //Achsenorientiert für Positionier.-,
//Teilungsachsen, Spindeln
GP_AuxFunction : ARRAY [1..10] OF BOOL; //Kanalorientiert für Hilfsfunktionen
GP_FMBlock : ARRAY [1..10] OF BOOL; //derzeit nicht genutzt
//ab hier dürfen weitere Lokaldaten des Anwenders definiert werden
END_VAR
BEGIN
    CALL FC 3;
    //HIER ANWENDERPROGRAMM EINFUEGEN
END_ORGANIZATION_BLOCK
```

### 12.15.13 FC 5: GP\_DIAG Grundprogramm, Diagnosealarm und Baugruppenausfall

#### Funktion

In diesem Teil des Grundprogramms erfolgt die Erfassung von Baugruppenstörungen und Baugruppenausfällen.

Über den Parameter des Bausteins FC 5 kann entschieden werden, ob die PLC in den Stopp-Zustand geschickt wird. Hierbei wird die PLC nur bei kommenden Ereignissen in den Stopp-Zustand geschickt. Ausnahme vom Parameter "PLC-Stop" sind die am FB 1 parametrisierten PROFIBUS-MCPs (müssen am DP1-Bus angeschlossen sein).

#### Deklaration

```
FUNCTION FC 5: VOID
  VAR_INPUT
    PlcStop: BOOL :=TRUE;
  END_VAR
```

#### Aufrufbeispiel

Das Grundprogramm kann - zeitlich gesehen - hinter der Bearbeitung weiterer Anwenderprogramme durchlaufen werden. Das ist empfehlenswert, da die PLC durch die Beschaltung des FC 5 in den Stopp-Zustand geschickt wird.

Das vorliegende Beispiel enthält die Standard-Deklarationen für OB 82 und OB 86 und dem Aufruf des Grundprogramm-Bausteins.

```
ORGANIZATION_BLOCK OB 82
VAR_TEMP
  OB82_EV_CLASS : BYTE ;
  OB82_FLT_ID : BYTE ;
  OB82_PRIORITY : BYTE ;
  OB82_OB_NUMBR : BYTE ;
  OB82_RESERVED_1 : BYTE ;
  OB82_IO_FLAG : BYTE ;
  OB82_MDL_ADDR : INT ;
  OB82_MDL_DEFECT : BOOL ;
  OB82_INT_FAULT : BOOL ;
  OB82_EXT_FAULT : BOOL ;
  OB82_PNT_INFO : BOOL ;
  OB82_EXT_VOLTAGE : BOOL ;
  OB82_FLD_CONNCTR : BOOL ;
  OB82_NO_CONFIG : BOOL ;
  OB82_CONFIG_ERR : BOOL ;
  OB82_MDL_TYPE : BYTE ;
  OB82_SUB_NDL_ERR : BOOL ;
  OB82_COMM_FAULT : BOOL ;
  OB82_MDL_STOP : BOOL ;
```

```
OB82_WTCH_DOG_FLT : BOOL ;
OB82_INT_PS_FLT : BOOL ;
OB82_PRIM_BATT_FLT : BOOL ;
OB82_BCKUP_BATT_FLT : BOOL ;
OB82_RESERVED_2 : BOOL ;
OB82_RACK_FLT : BOOL ;
OB82_PROC_FLT : BOOL ;
OB82_EPROM_FLT : BOOL ;
OB82_RAM_FLT : BOOL ;
OB82_ADU_FLT : BOOL ;
OB82_FUSE_FLT : BOOL ;
OB82_HW_INTR_FLT : BOOL ;
OB82_RESERVED_3 : BOOL ;
OB82_DATE_TIME : DATE_AND_TIME ;
END_VAR
BEGIN
    CALL FC 5
        (PlcStop := FALSE) ;
END_ORGANIZATION_BLOCK

ORGANIZATION_BLOCK OB 86
VAR_TEMP
    OB86_EV_CLASS : BYTE ;
    OB86_FLT_ID : BYTE ;
    OB86_PRIORITY : BYTE ;
    OB86_OB_NUMBR : BYTE ;
    OB86_RESERVED_1 : BYTE ;
    OB86_RESERVED_2 : BYTE ;
    OB86_MDL_ADDR : WORD ;
    OB86_RACKS_FLTD : ARRAY [0 .. 31 ] OF BOOL ;
    OB86_DATE_TIME : DATE_AND_TIME ;
END_VAR
BEGIN
    CALL FC 5
        (PlcStop := TRUE) ;
END_ORGANIZATION_BLOCK
```

## 12.15.14 FC 6: TM\_TRANS2 Transfer-Baustein für Werkzeugverwaltung und Multitool

### Funktion

Der Baustein TM\_TRANS2 wird bei Positionsänderungen der Werkzeuge, Statusänderungen und Multitool eingesetzt.

Der Baustein FC 6 beinhaltet die gleiche Funktionalität wie der FC 8, nur dass zusätzlich die Multitool-Funktionalität integriert ist. Deshalb werden in diesem Kapitel nur die Teile der Multitool-Funktionalität erläutert. Die darüber hinaus beinhaltete Funktionalität ist beschrieben in "FC 8: TM\_TRANS Transfer-Baustein für Werkzeugverwaltung (Seite 1097)".

### Deklaration der Funktion

#### AWL-Darstellung

```
FUNCTION FC 6 : VOID
VAR_INPUT
    Start:          BOOL;
    TaskIdent:      BYTE;
    TaskIdentNo:    BYTE;
    NewToolMag:     INT;
    NewToolLoc:     INT;
    OldToolMag:     INT;
    OldToolLoc:     INT;
    Status:         INT;
    MtoolPlaceNum: INT;
END_VAR
VAR_OUTPUT
    Ready:         BOOL;
    Error:         INT;
END_VAR
BEGIN
END_FUNCTION
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion TM\_TRANS2. Hier fett dargestellt der Parameter, der sich zum FC 8 unterscheidet.

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		siehe Bausteinbeschreibung FC 8
TaskIdent	E	BYTE		siehe Bausteinbeschreibung FC 8
TaskIdentNo	E	BYTE		siehe Bausteinbeschreibung FC 8
NewToolMag	E	INT		siehe Bausteinbeschreibung FC 8
NewToolLoc	E	INT		siehe Bausteinbeschreibung FC 8
OldToolMag	E	INT		siehe Bausteinbeschreibung FC 8
OldToolLoc	E	INT		siehe Bausteinbeschreibung FC 8
Status	E	INT		siehe Bausteinbeschreibung FC 8
<b>MtoolPlaceNum</b>	<b>E</b>	<b>INT</b>		<b>Multitoolplatz-Nr.</b>
Ready	A	BOOL		siehe Bausteinbeschreibung FC 8
Error	A	INT		siehe Bausteinbeschreibung FC 8

## 12.15.15 FC 7: TM\_REV Transfer-Baustein für Werkzeugwechsel mit Revolver

### Funktion

Der Anwender ruft nach erfolgtem Wechsel eines Revolvers diesen Baustein FC TM\_REV auf. Hierzu ist in dem Parameter "ChgdRevNo" die Revolvernummer (entsprechend der Schnittstellen-Nummer im DB 73) mitzuteilen. Mit Aufruf dieses Bausteins wird das zugehörige "Schnittstelle aktiv"-Bit im Datenbaustein DB 73, Wort 0 vom FC 7 zurückgesetzt, nachdem der Parameter "Ready" = TRUE zurückgemeldet wird.

Der Baustein FC TM\_REV darf nur mit Parameter "Start" = TRUE gestartet werden, wenn für diesen Transfer eine Aktivierung der zugehörigen Schnittstelle (DB 73, Wort 0) durch die Werkzeugverwaltung vorliegt.

Wenn dieser Auftrag korrekt ausgeführt wurde, enthält der Ausgangsparameter "Ready" den Wert TRUE. Damit muss der Anwender den Parameter "Start" = FALSE setzen bzw. den Baustein nicht mehr aufrufen.

Falls der Parameter "Ready" = FALSE ist, muss der Fehlercode im Parameter "Error" interpretiert werden.

Wenn der Fehlercode = 0 ist, dann ist dieser Auftrag im nächsten PLC-Zyklus zu wiederholen ("Start" bleibt TRUE). Es bedeutet, dass der Transferauftrag noch nicht beendet ist, siehe Beispiel FC 7-Aufruf und Impulsdiagramm.

Der Parameter "Start" benötigt keine Flanke für einen Folgeauftrag.

 **WARNUNG**

Ein Abbruch eines Transfers (z. B. durch ein externes Signal RESET) ist nicht zulässig. Der Parameter "Start" muss in jedem Fall so lange 1-Signal führen, bis die Parameter "Ready" bzw. "Error"  $\neq$  0 sind.

Bei Fehlercode  $\neq$  0 liegt eine Falschparametrierung vor.

**Hinweis**

Weitere Informationen zur Werkzeugverwaltung (auch in Bezug zur PLC) sind im Funktionshandbuch Werkzeugverwaltung enthalten. Weiterhin stehen noch PI-Dienste für die Werkzeugverwaltung über den FB 4, FC 8 und FC 22 zur Verfügung.

## Revolverschalten von Hand

Wird der Revolver über Handbedienung gedreht, so ist damit weder ein Werkzeugwechsel noch eine Korrekturanwahl verbunden. Datentechnisch wandert auch beim Revolver mit einer Werkzeuganwahl, auch wenn das von HMI Operate anders modelliert wird, das programmierte Werkzeug auf den Werkzeughalter und das Alt-Werkzeug zurück auf seinen Platz im Revolver.

Der erste Schritt ist das Auswechseln des Werkzeugs auf dem Werkzeughalter zurück auf seinen Revolverplatz. Dazu wird mit dem FC8 (alternativ dem FC6) ein asynchroner Transfer durchgeführt.

Die Parametrierung sieht folgendermaßen aus:

```
TaskIdent = 4
TaskIdentNo = Kanal-Nr.
NewToolMag = Magazinnummer des Revolvers
NewToolLoc = Ursprünglicher Platz des Werkzeugs
OldToolMag = Magazinnr. Zwischenspeicher (Spindel) = 9998
OldToolLoc = Zwischenspeichernummer der Spindel
Status = 1
```

Wird jetzt der Revolver auf eine beliebige Position gedreht, auf der ein Werkzeug sitzt, muss dieses Werkzeug aktiviert werden. Im einfachsten Fall geschieht das durch die neue T-Programmierung im Teileprogramm. Soll das jedoch z. B. mit Ende des Revolvertaktens von PLC geschehen, muss PLC dazu ein ASUP starten. Dem ASUP muss die aktuelle Revolverposition übergeben werden. Damit wird im ASUP das Werkzeug auf diesem Platz ermittelt und dieses ausgewählt (siehe Jobshop-Beispiel auf der Toolbox).

## Deklaration der Funktion

### AWL-Darstellung

```

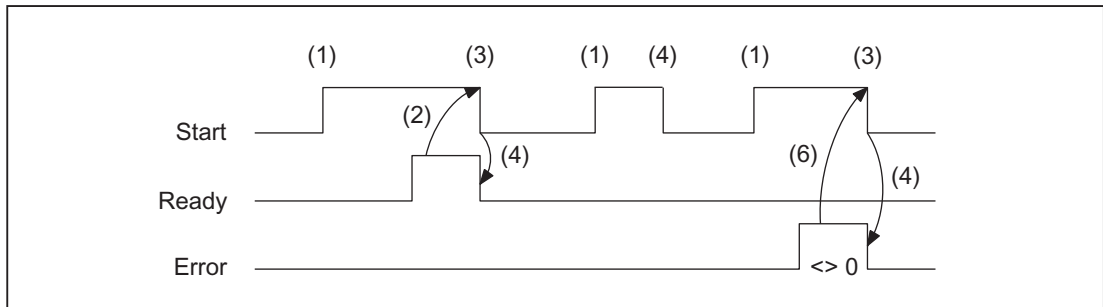
FUNCTION FC 7 :      VOID
//NAME :TM_REV
VAR_INPUT
    Start :          BOOL;
    ChgdRevNo :      BYTE;
END_VAR
VAR_OUTPUT
    Ready :          BOOL;
    Error :          INT;
END_VAR
BEGIN
END_FUNCTION
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion TM\_REV:

Signal	Art	Typ	Wertebereich	Bemerkung	
Start	E	BOOL		1 = Transfer wird gestartet.	
ChgdRevNo	E	BYTE	1...	Nummer der Revolver-Schnittstelle	
Ready	A	BOOL		1 = Transfer abgeschlossen	
Error	A	INT	0 ... 3	Fehler-Rückmeldung	
				0:	kein Fehler aufgetreten
				1:	kein Revolver vorhanden
				2:	unzulässige Revolvernummer in Parameter "ChgdRevNo"
3:	unzulässiger Auftrag ("Signal "Schnittstelle (SS) aktiv" des angewählten Revolvers = "FALSE")				

### Impulsdiagramm



- (1) Funktionsanstoß durch positive Flanke
- (2) Positive Quittung: WZV Transfer wurde ausgeführt
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC
- (5) Dieser Signalverlauf ist unzulässig. Der Auftrag ist generell zu beenden, da die neuen Werkzeug-Positionen der Werkzeugverwaltung im NCK mitzuteilen sind.
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter Error

### Aufrufbeispiel

```
CALL FC 7(                                     //Werkzeugverwaltungs Transfer Baustein
                                                //für Revolver
Start :=          m 20.5,                     //Start := "1 " => Anstoß des Transfers
ChgdRevNo :=     DB61.DBB 1,
Ready :=         m 20.6,
Error :=        DB61.DBW 12);
u m 20.6;                                     //Ready abfragen
r m 20.5;                                     //Start zurücksetzen
spb m001;                                     //Springe, wenn alles in Ordnung
l db61.dbw 12;                               //Fehler-Information
ow w#16#0;                                   //Fehler auswerten
spn fehl;                                     //Springe auf Fehlerbehandlung, wenn <> 0
m001 :                                       //Beginn des weiteren Programms
fehl :
r m 20.5;                                     //Start zurücksetzen, wenn Fehler vorliegt
```



## 12.15.16 FC 8: TM\_TRANS Transfer-Baustein für Werkzeugverwaltung

### Funktion

Der Anwender ruft diesen Baustein FC TM\_TRANS bei Positionsänderungen der Werkzeuge oder bei Statusänderungen des Transfervorgangs auf. Mit dem Parameter "TaskIdent" wird für den Baustein FC 8 der Transferauftrag an der Werkzeugverwaltungs-Schnittstelle angegeben:

- Für Be/-Entladestellen
- Für Spindel-Wechselstellen
- Für Revolver-Wechselstellen als Transferkennung
- Asynchroner Transfer
- Asynchroner Transfer mit Platzreservierung

Die Schnittstellen-Nummer wird im Parameter "TaskIdentNo" mitgeteilt.

Beispiel für Beladestelle 5:

Parameter "TaskIdent":= 1 und "TaskIdentNo":= 5.

Weiterhin werden für diesen Transfer die **aktuellen** Werkzeugpositionen und Statusinformationen, siehe Status-Auflistung des Parameters "Status" im nachfolgenden Text, übermittelt.

---

### Hinweis

Der FC 8 teilt dem NCK die aktuellen Positionen vom alten Werkzeug mit.

Dem NCK ist bekannt, wo sich das alte und das neue Werkzeug bis zur Positionsänderung befunden haben.

---

Bei einem Transfer ohne ein sogenanntes "altes Werkzeug" (z. B. beim Beladen), wird den Parametern "OldToolMag", "OldToolLoc" der Wert 0 zugeteilt.

Der Transfer-Baustein darf nur mit dem Parameter "Start" = TRUE gestartet werden, wenn für diesen Transfer eine Aktivierung der zugehörigen Schnittstelle (DB 71, DB 72, DB 73 im Wort 0) durch die Werkzeugverwaltung vorliegt.

Wenn dieser Auftrag korrekt ausgeführt wurde, enthält der Ausgangsparameter "Ready" den Wert TRUE.

Damit muss der Anwender den Parameter "Start" = FALSE setzen bzw. den Baustein nicht mehr aufrufen.

Falls der Parameter "Ready" = FALSE ist, muss der Fehlercode im Parameter "Error" interpretiert werden, siehe Beispiel FC 8-Aufruf und Impulsdiagramm.

Wenn der Fehlercode = 0 ist, dann ist dieser Auftrag im nächsten PLC-Zyklus zu wiederholen ("Start" bleibt TRUE). Es bedeutet, dass der Transferauftrag noch nicht beendet ist.

Wenn der Parameter "Status" einen Wert kleiner 100 durch den Anwender erhält, wird die zugehörige Schnittstelle im Datenbaustein DB 71 bzw. DB 72 oder DB 73, Wort 0 deaktiviert (Vorgang beendet). Das entsprechende Bit für die Schnittstelle wird auf 0 gesetzt durch den FC 8.

Der Parameter "Start" benötigt keine Flanke für einen Folgeauftrag. Dies bedeutet, dass sofort bei Erhalten von "Ready" = TRUE neue Parameter zugeordnet werden können mit "Start" = TRUE.

## Asynchroner Transfer

Für eine eigenständige Mitteilung der PLC von den Positionsänderungen eines Werkzeugs zur Werkzeugverwaltung (z. B. Spannungsunterbrechung bei einem aktiven Kommando oder selbständige Positionsveränderungen durch PLC) wird dieser FC 8-Transfer-Baustein mit der "TaskIdent": = 4 oder 5 aufgerufen. Zu diesem Aufruf muss keine Schnittstellen-Aktivierung durch die Werkzeugverwaltung vorliegen.

Bei dem Parameter "TaskIdent" = 5 wird zusätzlich zur Positionsveränderung noch eine Platzreservierung von der Werkzeugverwaltung vorgenommen. Diese Platzreservierung erfolgt aber nur, wenn das Werkzeug aus einem realen Magazin in einen Zwischenspeicher transportiert wurde.

Im Parameter "TaskIdentNo" ist ein zugehöriger NC-Kanal zu parametrieren.

In den Parametern "OldToolMag", "OldToolLoc" wird die bisherige Position des Werkzeugs angegeben und in den Parametern "NewToolMag", "NewToolLoc" wird die aktuelle Position dieses Werkzeugs angegeben. Hierbei ist der Status = 1 anzugeben.

Bei Status 5 verbleibt das angegebene Werkzeug auf dem Platz "OldToolMag", "OldToolLoc". Dieser Platz muss ein Zwischenspeicher sein (z. B. Spindel). In den Parametern "NewToolMag", "NewToolLoc" ist das reale Magazin mit Platz anzugeben, wobei sich der Platz an der Position des Zwischenspeichers befindet. Dieses Verfahren ist immer dann anzuwenden, wenn der Werkzeugverwaltung mitgeteilt werden soll, wo sich ein bestimmter Platz des Magazins befindet. Dieses Verfahren dient zum Abgleich bei Suchstrategien.

---

### Hinweis

Ein Abbruch eines Transfers (z. B. durch ein externes Signal RESET) ist nicht zulässig. Der Parameter "Start" muss in jedem Fall so lange 1-Signal führen, bis die Parameter "Ready" bzw. "Error"  $\neq$  0 sind.

---

Bei Fehlercode  $\neq$  0 liegt eine Falschparametrierung vor.

---

### Hinweis

Weitere Informationen zur Werkzeugverwaltung (auch in Bezug zur PLC) sind im Funktionshandbuch Werkzeugverwaltung enthalten. Weiterhin stehen noch PI-Dienste für die Werkzeugverwaltung über den FB 4, FC 7 und FC 22 zur Verfügung.

---

## Deklaration der Funktion

### AWL-Darstellung

```

FUNCTION FC 8 : VOID
//NAME :TM_TRANS
VAR_INPUT
    Start :          BOOL;
    TaskIdent :      BYTE;
    TaskIdentNo :    BYTE;
    NewToolMag :     INT;
    NewToolLoc :     INT;
    OldToolMag :     INT;
    OldToolLoc :     INT;
    Status :        INT;
END_VAR
VAR_OUTPUT
    Ready :         BOOL;
    Error :         INT;
END_VAR
BEGIN
END_FUNCTION
    
```

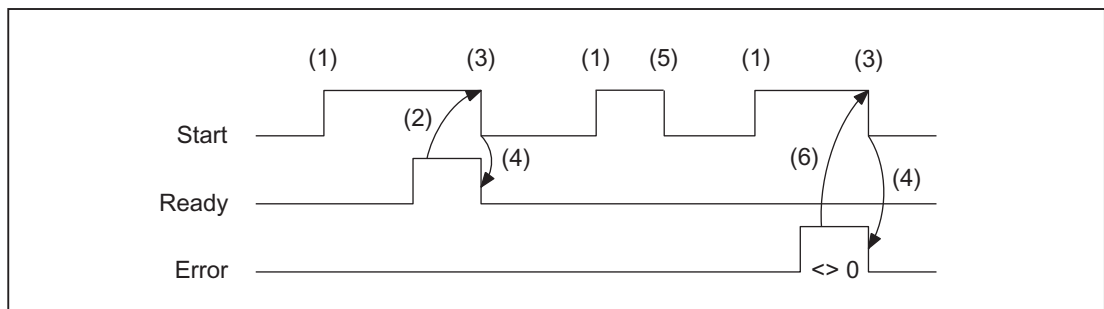
## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion TM\_TRANS:

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		1 = Transfer wird gestartet.
TaskIdent	E	BYTE	1 ... 5	Kennung der Schnittstelle bzw. des Auftrags
				1: Be-/Entladestelle
				2: Spindel-Wechselstelle
				3: Revolver-Wechselstelle
				4: asynchroner Transfer
5: asynchroner Transfer mit Platzreservierung				
TaskIdentNo	E	BYTE	1 ...	Nummer der zugehörigen Schnittstelle bzw. Kanalnummer. Im oberen Nibble kann beim asyn. Transfer die Schnittstellennr. angegeben werden (z.B. B#16#12, 1. Schnittstelle, 2. Kanal).
NewToolMag	E	INT	1, 0 ...	aktuelle Magazinnummer des neuen Werkzeugs
				-1: Werkzeug verbleibt auf seinem bisherigen Platz. NewToolLoc = beliebiger Wert Nur bei TaskIdent = 2 zulässig.

Signal	Art	Typ	Wertebereich	Bemerkung
NewToolLoc	E	INT	0 ... max. Platznummer	aktuelle Platznummer des neuen Werkzeugs
OldToolMag	E	INT	-1, 0 ...	aktuelle Magazinnummer des auszuwechselnden Werkzeugs
				-1: Werkzeug verbleibt auf seinem bisherigen Platz. OLDToolLoc = beliebiger Wert Nur bei TaskIdent = 2 zulässig.
OldToolLoc	E	INT	max. Platznummer	aktuelle Platznummer des auszuwechselnden Werkzeugs
Status	E	INT	1 ... 7, 103 ... 105	Status-Information des Transfervorgangs
Ready	A	BOOL		1= Transfer abgeschlossen
Error	A	INT	0 ... 65535	Fehler-Rückmeldung
				0: kein Fehler aufgetreten
				1: unbekannte "TaskIdent"
				2: unbekannte "TaskIdentNo"
				3: unzulässiger Auftrag ("Signal "Schnittstelle (SS) aktiv" des angewählten Revolvers = "FALSE")
andere Werte:	Die Zahl entspricht der Fehlermeldung der WZV im NCK, die durch diesen Transfer verursacht wurde.			

Impulsdiagramm



- (1) Funktionsanstoß durch positive Flanke
- (2) Positive Quittung: WZV Transfer wurde ausgeführt
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC
- (5) Dieser Signalverlauf ist unzulässig. Der Auftrag ist generell zu beenden, da die neuen Werkzeug-Positionen der Werkzeugverwaltung im NCK mitzuteilen sind.
- (6) Negative Quittung: Fehler aufgetreten, Fehlercode im Ausgangsparameter Error

## Status-Auflistung

### Status = 1:

**Der WZV-Auftrag ist beendet worden** (Be-/Ent-/Umladen, Wechsel vorbereiten, Wechseln).

Die Parameter "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" des FC 8-Bausteins sind auf die tatsächlichen Positionen der beteiligten Werkzeuge zu parametrieren. Außer beim Wechsel vorbereiten sind dies die in der Regel die angegebenen Werkzeug-Zielpositionen der zugehörigen WZV-Schnittstelle, siehe auch "Erläuterungen der Formalparameter".

1. Beim Be-/Ent-/Umladen ist das Werkzeug auf der geforderten Zieladresse angekommen. Wenn das Bit in der Nahtstelle im DB71.DBX(n+0).3 "Positionieren zur Beladestelle" gesetzt ist, dann darf für den Abschluss der Funktion nicht der Status 1 verwendet werden. Für einen korrekten Abschluss ist der Status 5 zu verwenden.
2. Beim "Wechsel vorbereiten" ist die Bereitstellung des neuen Werkzeugs erfolgt. Dieses kann z. B. auf einem Zwischenspeicher (Greifer) abgelegt sein. Gegebenenfalls wurde auch das Ziel (Magazin, Platz) des alten Werkzeugs zur Wechselposition positioniert, nachdem das neue Werkzeug in einem Zwischenspeicher abgelegt wurde. Das alte Werkzeug verbleibt aber noch in der Spindel. Somit ist die Vorbereitung zum Wechseln abgeschlossen. Nach dieser Quittierung kann das Kommando "Wechseln" empfangen werden. Die Positionen in den Parametern "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" entsprechen den aktuellen Positionen der Werkzeuge.
3. Beim "Wechseln" (in Spindel bzw. beim Revolver) sind die in der Schnittstelle angesprochenen Werkzeuge auf den geforderten Zieladressen angekommen. Der Vorgang des Werkzeugwechsels ist damit abgeschlossen.

### Status = 2: Das "neue" Werkzeug kann nicht bereitgestellt werden.

Dieser Status ist nur bei dem Befehl "Wechsel vorbereiten" zulässig. Wenn dieser Status angewendet wird, soll von der PLC ein Wechsel mit dem vorgeschlagenen Werkzeug verhindert werden. Durch die WZV im NCK erfolgt das Sperren dieses vorgeschlagenen (neuen) Werkzeugs. Anschließendes erfolgt ein neues Kommando der Werkzeugverwaltung mit einem Duplo-Werkzeug. Die Positionen in den Parametern "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" entsprechen den ursprünglichen Positionen der Werkzeuge.

### Status = 3: Ein Fehler ist aufgetreten.

Es darf keine Änderung der Werkzeugpositionen erfolgt sein. Wenn sich die Magazinpositionen der Werkzeuge zwischenzeitlich verändert hatten, ist dieses z. B. vorher mit dem Status = 105 über den FC 8-Transfer-Baustein mitzuteilen. Nur dann werden die Positionen von der Werkzeugverwaltung berücksichtigt.

### Status = 4: Das "alte" Werkzeug sollte besser auf der in den Parametern "OldToolMag", "OldToolLoc" angegebenen Magazinposition untergebracht werden.

Dieser Status ist nur bei der Werkzeugwechsel-Vorbereitung (Wechsel in die Spindel) zulässig. Nachdem dieser Status an die Werkzeugverwaltung im NCK gegeben wurde, versucht die Werkzeugverwaltung beim nachfolgenden Kommando die angegebene Magazin-Position zu berücksichtigen. Dieses erfolgt aber nur dann, wenn diese Position frei ist. Die Parameter "NewToolMag" und "NewToolLoc" werden nicht berücksichtigt.

**Status = 5: Der Vorgang ist beendet worden.**

Das "neue" Werkzeug ist an der in den Parametern "NewToolMag", "NewToolLoc" angegebenen Position. Hierbei befindet sich das angegebene Werkzeug nicht wirklich in dieser Position, sondern ist weiterhin im gleichen Magazinplatz. Dieser Magazinplatz wurde jedoch zu dieser Position (z. B. Wechselstelle) hin positioniert. Dieser Status darf nur bei Revolver, Ketten- und Scheibenmagazinen verwendet werden. Der Status dient der Werkzeugverwaltung zum Abgleich der aktuellen Position eines Magazins und zur Verbesserung der Suchstrategie für nachfolgende Befehle. Dieser Status ist nur beim Beladen, Entladen, Umladen und bei der Wechsel-Vorbereitung erlaubt.

Die Parameter "OldToolMag" und "OldToolLoc" müssen mit den Daten eines Zwischenspeichers parametrisiert sein.

- **Beladen, Umladen:**

Beim Beladen oder Umladen ist im NCK schon ein Platz für das Werkzeug reserviert worden. Im Anschluss muss der Maschinen-Bediener das Werkzeug auf dem Zielplatz einsetzen. Achtung: Nach einem erneuten Einschalten der Steuerung ist die Platzreservierung aufgehoben.

- **Wechsel-Vorbereitung:**

Die noch ausstehenden Werkzeugbewegungen erfolgen erst mit dem Werkzeug-Wechsel.

- **Positionieren zur Beladestelle:**

Wenn das Bit in der Nahtstelle im DB71.DBX(n+0).3 "Positionieren zur Beladestelle" gesetzt ist, dann darf für den Abschluss der Funktion nur der Status 5 (nicht Status 1) verwendet werden.

**Status = 6: Der WZV-Auftrag ist beendet worden.**

Dieser Status hat die gleiche Funktion wie der Status 1, aber zusätzlich wird eine Reservierung des Quellplatzes vorgenommen. Dieser Status ist nur beim Umladen erlaubt. Das Kommando wird beendet und der Quellplatz des Werkzeugs wird reserviert, falls der Zielplatz in einem Zwischenspeichermagazin liegt.

**Status = 7: Wiederholung des "Werkzeug Vorbereiten" Befehls anstossen.**

Dieser Status ist nur bei dem "Wechsel vorbereiten"-Befehl zulässig. Der Status soll angewendet werden, wenn das "neue" Werkzeug seine Position verändert hat (z. B. über einen asynchronen Befehl des "neuen" Werkzeugs). Nach "Ready = 1" vom FC 8 erfolgt eine automatische Wiederholung des "Wechsel vorbereiten" mit dem gleichen Werkzeug. Für die automatische Wiederholung wird eine neue Werkzeugsuche durchgeführt. Die Positionen in den Parametern "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" müssen den ursprünglichen Positionen der Werkzeuge entsprechen.

**Status = 103: Das "neue" Werkzeug kann eingesetzt werden.**

Dieser Status ist nur bei der Wechsel-Vorbereitung zulässig, wenn die PLC das neue Werkzeug ablehnen darf (z. B. bei MD20310 \$MC\_TOOL\_MANAGEMENT\_MASK, Bit 4 = 1 für Möglichkeit, geänderte Parameter vom PLC noch einmal anzufordern). Die Positionen der Werkzeuge sind unverändert geblieben. Dieser Status ist damit notwendig, wenn der Vorlauf im NCK ohne einen unnötigen Stopp der Bearbeitung fortgesetzt werden soll.

**Literatur:**

Funktionshandbuch Werkzeugverwaltung

**Status = 104: Das "neue" Werkzeug ist an der in den Parametern "NewToolMag", "NewToolLoc" angegebenen Position.**

Dieser Status ist nur zulässig, wenn das Werkzeug sich noch im Magazin auf dem gleichen Platz befindet. Das "alte" Werkzeug ist auf der in den Parametern "OldToolMag", "OldToolLoc" angegebenen Position (Zwischenspeicher). Hierbei befindet sich das neue Werkzeug nicht wirklich in dieser Position, sondern ist weiterhin im gleichen Magazinplatz. Dieser Magazinplatz wurde jedoch zu dieser Position (z. B. Wechselstelle) hin positioniert. Dieser Status darf nur bei Revolver, Ketten- und Scheibenmagazinen beim "Werkzeug-Wechsel Vorbereiten" verwendet werden. Der Status dient der Werkzeugverwaltung zum Abgleich der aktuellen Position eines Magazins, zur Verbesserung der Suchstrategie für nachfolgende Befehle.

**Status = 105: Der angegebene Zwischenplatz ist erreicht von allen beteiligten Werkzeugen** (Standardfall, wenn der Vorgang noch nicht beendet ist).

Die Werkzeuge sind in den angegebenen Werkzeug-Positionen (Parameter "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc").

## Status-Definition

Generell gilt für den Quittungsstatus, dass die Statusinformationen 1 bis 7 zu einer Beendigung des Kommandos führen. Wenn eine dieser Statusinformationen dem FC 8 mitgeteilt wird, wird das "Schnittstelle Aktiv-Bit" der im FC 8 angegebenen Schnittstelle auf "0" zurückgesetzt (siehe auch Nahtstellen Listen DB 71 bis DB 73). Damit ist der Vorgang abgeschlossen. Anders verhält es sich bei den Statusinformationen 103 bis 105. Bei Übergabe einer dieser Statusinformationen zum FC 8 bleibt das "Schnittstelle Aktiv-Bit" dieser Schnittstelle auf "1". Eine weitere Bearbeitung ist notwendig durch das Anwenderprogramm in der PLC (z. B. Fortführen der Magazinpositionierung). Diese Statusinformation dient in der Regel zum Übermitteln von veränderten Positionen von einem oder beiden Werkzeugen, ohne dass der Vorgang abgeschlossen ist.

## Aufrufbeispiel

```

CALL FC 8(           //Werkzeugverwaltungs Transfer Baustein
  Start :=          m 20.5,           //Start := "1 " => Anstoß des Transfers
  TaskIdent :=      DB61.DBB 0,
  TaskIdentNo :=   DB61.DBB 1,
  NewToolMag :=    DB61.DBW 2,       //aktuelle Position neues Werkzeug
  NewToolLoc :=    DB61.DBW 4,
  OldToolMag :=    DB61.DBW 6,       //aktuelle Position altes Werkzeug
  OldToolLoc :=    DB61.DBW 8,
  Status :=        DB61.DBW 10,      //Status
  Ready :=         m 20.6,
  Error :=         DB61.DBW 12);

u m 20.6;           //Ready abfragen
r m 20.5;           //Start zurücksetzen
spb m001;           //springe wenn alles in Ordnung
l DB61.dbw 12;     //Fehler Information
ow w#16#0;         //Fehler auswerten
spn fehl;          //Springe auf Fehlerbehandlung

m001 :              //normaler Zweig

fehl :              //Fehlerbehandlung
r m 20.5 :          //Start zurücksetzen
    
```

### 12.15.17 FC 9: ASUP Start von asynchronen Unterprogrammen

#### Funktion

Mit dem FC ASUP können beliebige Funktionen in der NC ausgelöst werden. Voraussetzung dafür, dass ein ASUP von der PLC gestartet werden kann, ist dessen Anwahl und Parametrierung durch ein NC-Programm oder durch den FB 4 (PI-Dienst ASUP). Hierbei muss Kanal- und Interruptnummer übereinstimmen zu den Parametern im FC 9.

Ein derart vorbereitetes ASUP kann von der PLC zu einem beliebigen Zeitpunkt gestartet werden. Das in dem betreffenden Kanal laufende NC-Programm wird durch das ASUP unterbrochen. In einem Kanal kann dabei zu einem Zeitpunkt nur ein ASUP gestartet werden. Sollte in **einem** PLC Zyklus für zwei FC 9 der Start-Parameter auf logisch 1 gesetzt werden, werden die ASUP in Aufrufreihenfolge gestartet.

Der Start-Parameter muss vom Anwender auf logisch 0 gesetzt werden, wenn das ASUP beendet (Done) oder ein Fehler aufgetreten ist.



Zur Auftragsbearbeitung benötigt jeder FC ASUP einen eigenen WORD-Parameter "Ref" aus dem globalen Anwender-Speicherbereich. Dieser wird intern verwendet und darf vom Anwender nicht verändert werden. Der Parameter "Ref" wird im ersten OB 1-Zyklus mit dem Wert 0 initialisiert, aus diesem Grund **muss jeder FC 9 absolut aufgerufen werden**. Alternativ kann der Parameter "Ref" mit dem Wert 0 initialisiert werden durch den Anwender im Anlauf. Damit sind auch bedingte Aufrufe möglich. Ein bedingter Aufruf muss bei Aktivierung durch den Parameter "Start" = 1 solange erfolgen, bis der Parameter "Done" einen Zustandswechsel von 1 nach 0 durchführt.

---

### Hinweis

Der FB 4-Aufruf muss beendet sein, bevor der FC 9 gestartet werden darf. Der FC 9 kann nicht gestartet werden, wenn "Not-Halt" gesetzt ist. Der Start des FC 9 darf auch nicht erfolgen, wenn der Kanal-RESET aktiv ist.

---

### Deklaration der Funktion

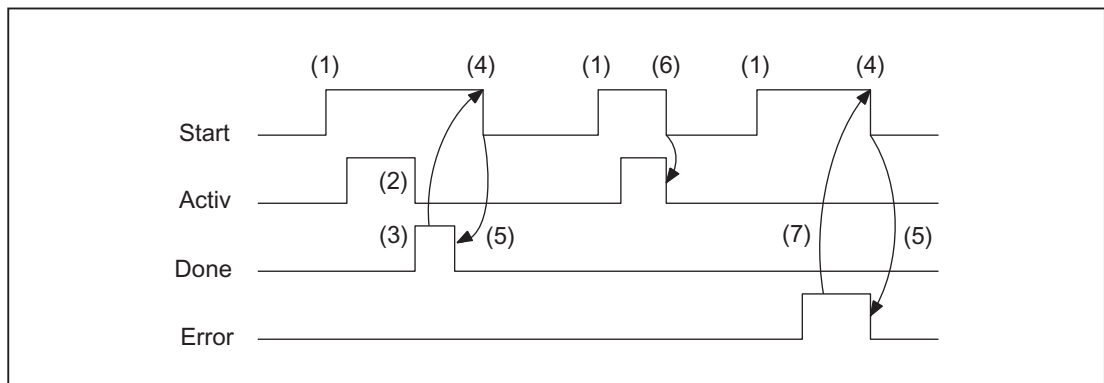
```
FUNCTION FC 9 : VOID
//NAME :ASUP
VAR_INPUT
    Start :      BOOL;
    ChanNo :     INT;
    IntNo  :     INT;
END_VAR
VAR_OUTPUT
    Activ  :     BOOL;
    Done   :     BOOL;
    Error  :     BOOL;
    StartErr :   BOOL;
END_VAR
VAR_IN_OUT
    Ref   :      WORD;
END_VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion ASUP:

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		
ChanNo	E	INT	1 ... 10	Nr. des NC-Kanals
IntNo	E	INT	1 ... 8	Interrupt-Nr.
Activ	A	BOOL		1 = aktiv
Done	A	BOOL		1 = ASUP beendet
Error	A	BOOL		1 = Interrupt ist ausgeschaltet
StartErr	A	BOOL		1 = Interruptnummer nicht vergeben oder gelöscht
Ref	E/A	WORD	globale Variable (MW, DBW, ...)	1 Wort je FC 9 (für interne Verwendung)

### Impulsdiagramm



- (1) Funktionsanstoß
- (2) ASUP aktiv
- (3) Positive Quittung: ASUP beendet
- (4) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (5) Signalwechsel durch FC
- (6) Nicht zulässig. Wird Funktionsanstoß vor Erhalt der Quittung rückgesetzt, werden die Ausgangssignale nicht aktualisiert, ohne Einfluss auf Ablauf der angestoßenen Funktion.
- (7) Negative Quittung: Fehler aufgetreten

## Aufrufbeispiel

```
CALL FC 9(                                     //Start eines asynchronen Unterprogramms
                                                //im Kanal 1 Interruptnummer 1
    Start :=      E 45.7,
    ChanNo :=     1,
    IntNo  :=     1,
    Activ  :=     M 204.0,
    Done   :=     M204.1,
    Error  :=     M 204.4,
    StartErr :=   M 204.5,
    Ref    :=     MW 200);
```

## 12.15.18 FC 10: AL\_MSG Fehler- und Betriebsmeldungen

### Funktion

Mit dem FC AL\_MSG werden die in DB 2 eingetragenen Signale ausgewertet und als kommende und gehende Fehler- und Betriebsmeldungen auf dem HMI zur Anzeige gebracht.

Die kommenden Signale (positive Flanke) werden sowohl bei Fehler- als auch bei Betriebsmeldungen sofort zur Anzeige gebracht.

Gehende Signale (negative Flanke) werden nur bei Betriebsmeldungen sofort gelöscht; bei Fehlermeldungen werden erst mit dem Parameter "Quit" die nicht mehr anstehenden Meldungen gelöscht, d.h. Fehleranzeigen bleiben, auch wenn die Signale nicht mehr anstehen auf dem HMI solange erhalten, bis sie vom Anwender quittiert wurden.

Über den Parameter "ToUserIF" kann die Übertragung der Sammelsignale für Vorschub-, Einlese- und NC-Start-Sperre sowie Vorschub-Halt an vorhandene Achs-, Spindel- und Kanal- Nahtstellen veranlasst werden. Die Sammelsignale werden unabhängig von der Quittierung eines Alarms direkt aus den Zustandsinformationen des DB 2 in die Anwendernahtstelle transferiert.

1. Wenn der Parameter "ToUserIF" = FALSE ist, erfolgt keine Übertragung der Signale an die Anwendernahtstelle. Der Anwender muss in diesem Fall durch sein PLC-Programm dafür sorgen, dass diese Signale in der Nahtstelle beeinflusst werden.
2. Wenn der Parameter "ToUserIF" = TRUE ist, erfolgt eine Übertragung aller oben aufgeführten Signale an die Anwendernahtstelle als jeweils ein Sammelsignal. Das PLC-Programm des Anwenders kann folglich die oben aufgeführten Signale nur über den DB 2 beeinflussen in Verbindung mit einer Meldungs- oder Alarm-Ausgabe. Es erfolgt ein Überschreiben der jeweiligen Information in der Anwendernahtstelle.

Alternativ zu dem unter Punkt 2 beschriebenen Verhalten kann der Anwender eine Beeinflussung des Sperr- und Halt-Signals ohne Meldungsausgabe vornehmen, indem er nach dem Aufruf des FC AL\_MSG die Nahtstellensignale mit einem Sperr- oder Halt-Signalzustand beeinflusst.

Folgende Programmsequenz soll das verdeutlichen:

```
CALL FC 10(  
    ToUserIF := TRUE,  
    Quit := e 6.1);  
  
u m 50.0; //Vorschub Sperre für Kanal 1  
auf db 21;  
s dbx 6.0; //Setzen der Sperrbedingung,  
//Rücksetzen erfolgt über den FC AL_MSG,  
//wenn M 50.0 das Signal "0" führt.
```

### FB 1-Parameter "ExtendAlMsg"

Mit dem Aktivieren des Parameters wird eine neue Strukturierung des DB 2 wirksam (siehe "Nahtstelle PLC/HMI (Seite 933)"). Bei Aktivierung stehen dem Anwender Bitfelder für die Sperr- und Haltsignale zur Verfügung, die keine Alarmer, Meldungen absetzen. Hierdurch braucht der Anwender die vorstehenden Maßnahmen nicht realisieren. Durch einfaches Setzen, Rücksetzen von Signalen in den neuen DB 2-Bereichen ist die gewünschte Funktionalität automatisch gegeben.

Die Fehler- und Betriebsmeldungen werden vom Anwender im Datenbaustein DB 2 versorgt (siehe Beschreibung DB2 in Listen Nahtstellensignale).

---

#### Hinweis

Im DB 2 muss ein "1-Signal" für mehrere OB 1-Zyklen anstehen, damit eine Meldung auch am HMI angezeigt werden kann. Es gibt eine Obergrenze von gleichzeitig anstehenden Alarmen und Meldungen. Diese Obergrenze ist abhängig von der PLC-CPU. Bei der PLC317-2DP beträgt die Obergrenze 60 gleichzeitig anstehende Meldungen.

Siehe auch Listenhandbuch (Listen, Buch 2), Kapitel PLC- Alarme / Meldungen

---

## Deklaration der Funktion

### AWL-Darstellung

```

FUNCTION FC 10:          VOID
    //NAME:              AL_MSG
VAR_INPUT
    ToUserIF :           BOOL;
    Quit :             BOOL;
END_VAR
END_FUNCTION
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion AL\_MSG:

Signal	Art	Typ	Wertebereich	Bemerkung
ToUserIF	E	BOOL		1 = Übertragung der Signale an Anwenderschnittstelle je Zyklus
Quit	E	BOOL		1 = Quittierung von Fehlermeldungen

## Aufrufbeispiel

```

CALL FC 10 (                //Fehler- und Betriebsmeldungen
    ToUserIF := TRUE,       //Signale aus DB 2 werden an
                             //Nahtstelle übertragen.
    Quit := E6.1);         //Quittierung der Fehlermeldung wird über
                             //Eingang E6.1 durchgeführt.
    
```

## 12.15.19 FC 12: AUXFU Aufrufchnittstelle für Anwender bei Hilfsfunktionen

### Funktion

Der FC AUXFU wird generell ereignisgesteuert im Grundprogramm aufgerufen, wenn neue Hilfsfunktionen für den im Eingangs-Parameter übergebenen Kanal bereitstehen. Dieser FC kann durch den PLC-Anwender für seine Hilfsfunktion-Bearbeitung mit Programm-Anweisungen erweitert werden, um das zyklische Polling der Kanal-DBs bei Hilfsfunktionen zu vermeiden. Über diesen Mechanismus wird eine auftragsgesteuerte Bearbeitung der Hilfsfunktionen ermöglicht. Dieser FC wird als leerer Baustein in kompilierter Form im Grundprogramm geliefert. Hierbei wird der Parameter "Chan" vom Grundprogramm mit der Kanalnummer versorgt. Somit kennt der PLC-Anwender den Kanal in dem neue Hilfsfunktionen bereitstehen. Über die Änderungssignale der Hilfsfunktionen in diesem Kanal können die neuen Hilfsfunktionen ermittelt werden.

### Deklaration der Funktion

```

FUNCTION FC 12: VOID           //Ereignissteuerung der Hilfsfunktionen
VAR_INPUT
    Chan :    BYTE;
END_VAR
BEGIN
    BE;
END_FUNCTION
    
```

### Erläuterungen der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion AUXFU:

Signal	Art	Typ	Wertebereich	Bemerkung
Chan	E	BYTE	0 ... 9	Nr. des NC-Kanals -1

### Beispiel

```

FUNCTION FC 12: VOID           //Ereignissteuerung der Hilfsfunktionen
VAR_INPUT
    Chan :    BYTE;           //Parameter wird vom Grundprogramm versorgt
END_VAR
VAR_TEMP
    ChanDB:    INT;
END_VAR

BEGIN
L Chan;                       //KanalIndex-Nr, (0,1,2,...)
+ 21;                         //Kanal DB Offset
T ChanDB;                     //Kanal DB-Nr. speichern
AUF DB[ChanDB];              //Kanal-DB wird indirekt geöffnet
//Jetzt werden die Änderungssignale der Hilfsfunktionen abgefragt, usw.
    BE;
END_FUNCTION
    
```

## 12.15.20 FC 13: BHGDisp Displaysteuerung für Bedienhandgerät

### Funktion

Dieser Baustein übernimmt die Displaysteuerung des Bedienhandgerätes (BHG oder HT2). Die Information, die auf dem Display erscheinen soll, ist in dem Stringdatum ChrArray mit 32 Zeichen (bei Verwendung eines HT2 sind es 64 Zeichen) abgelegt. Hierzu ist beim Erstellen des Datenbausteins für diesen String eine Festtext-Zuweisung von 32 bzw. 64 Zeichen notwendig.

Pro Auftrag, der mehrere OB 1-Zyklen andauert, werden 16 Zeichen zum BHG/HT2 gesendet. Die Zuordnung der Zeichen im ChrArray zur jeweiligen Zeile ist eindeutig. Für die Zeile 1 werden die Zeichen 1 bis 16 und für die Zeile 2 die Zeichen 17 bis 32 des Stringdatums ChrArray übertragen. Zusätzlich werden beim HT2 die Zeile 3 aus den Zeichen 33 bis 48 und die Zeile 4 aus den Zeichen 49 bis 64 angezeigt.

Der Baustein überprüft, ob die notwendige Mindestlänge des ChrArray für den Betrieb des BHG bzw. HT2 vorhanden ist. Sind weniger Zeichen im Stringdatum vorhanden als angezeigt werden sollen, so wird die Zeile mit Leerzeichen aufgefüllt.

Durch den Parameter Row = 0 kann die Display-Ausgabe unterdrückt werden (z. B. wenn mehrere Variablen in einem oder mehreren PLC-Zyklen im String eingetragen werden sollen, ohne dass eine Display-Ausgabe erfolgt). Sollen mehrere Zeilen "gleichzeitig" (Übertragung der Zeichen zu den Zeilen dauert mehrere OB 1-Zyklen) aktualisiert werden (Parameter Row > 1), so werden die Zeilen nacheinander mit je 16 Zeichen pro Zeile aktualisiert.

Variable Anteile innerhalb des Strings können mit der Zahlenwandler-Funktionalität (optional) eingefügt werden. Für den Zahlenwandler ist der Parameter "Convert" auf TRUE zu setzen. Die Variable, die angezeigt werden soll, ist über den Pointer Addr referenziert. Im Parameter "DataType" ist die Formatbeschreibung dieses Parameters enthalten (siehe Parameter-Tabelle). Die Anzahl der Bytes der Variablen ist an die Formatbeschreibung gekoppelt. Die rechtsbündige Adresse innerhalb des Strings wird durch den Parameter "StringAddr" angegeben. Die Anzahl der geschriebenen Zeichen ist aus der Parametertabelle ersichtlich.

### Signale

Von den Ausgangssignalen des BHG wird das Byte 1 und die Zeichenvorgaben durch den Baustein beschaltet. Diese dürfen nicht durch das PLC-Anwenderprogramm beschrieben werden.

## Zusätzliche Parameter

### BHG

Im Anlauf-OB 100 sind am FB 1, DB 7 die Pointer-Parameter für die Ein- und Ausgangsdaten des Bedienhandgerätes zu parametrieren. Der Parameter "BHGIIn" entspricht den Eingangsdaten der PLC vom Bedienhandgerät (Empfangsdaten der PLC). Der Parameter "BHGOut" entspricht den Ausgangsdaten der PLC zum Bedienhandgerät (Sendedaten der PLC). Diese beiden Pointer sind auf den jeweiligen Datenbereichs-Anfang einzustellen (der auch bei MPI-Kopplung im SDB210 parametrier ist).

Für den Betrieb eines BHG ist am FB 1-Parameter BHG eine "2" einzutragen.

### HT2

Beim Einsatz des HT2 ist am FB 1-Parameter BHG eine "5" einzutragen. Die Pointer-Parameter der Ein- und Ausgangsdaten sind, wie oben beschrieben, ebenfalls zu versorgen.

An den Parametern BHGRecGDNo und BHGRecGBZNo ist der Wert einzutragen, der am S2 des DIP-Fix-Schalters (Drehcodierschalter) des Anschlussmoduls des HT2 konfiguriert wurde.

---

### Hinweis

#### Zahlenwandlung

Wenn der Zahlenwandler zur Informationsanzeige genutzt wird, dann ist es zur Verringerung der PLC-Zykluszeit zweckmäßig, die Wandlung nicht in jedem PLC-Zyklus durchzuführen.

Die Konvertierungsroutine kann unabhängig von der Displaysteuerung genutzt werden. Dazu ist beim Parameter Row "0" anzufragen, aber der Parameter Convert ist zu setzen. Somit wird nur der String bearbeitet und die Wandlerroutine wird ausgeführt.

#### Höhere Auflösung der Anzeige

Soll z. B. der Achsistwert mit einer höheren Auflösung dargestellt werden, so sollte man folgendes beachten:

Das Lesen der Variablen erfolgt wie bisher mit FB 2 oder FB 5. Statt Anypointer BYTE 8 als Kriterium zur Ausgabe als 64-Bit Gleitkommazahl wird REAL 2 verwendet (z. B.: P#M100.0 REAL 2).

Bei der Angabe der 64-Bit Gleitkommazahl am BHG/HT2 kann, anstelle von fest vorgegebenen Formaten, das Ausgabeformat mit bis zu max. 14 Stellen, frei auf Vor- und Nachkommastellen verteilt, gewählt werden.

---



## Deklaration der Funktion

### AWL-Darstellung

```

DATA_BLOCK "strdat"                                // DB-Nummer in Symboldatei definiert

STRUCT                                             // 32 Zeichen sind definiert
  disp :      STRING [32]:= 'character_line1 character_line2';
END_STRUCT;
BEGIN
END_DATA_BLOCK

FUNCTION FC 13: VOID
  VAR INPUT
    Row :      BYTE;                                //Displayzeile (siehe Tabelle)
    ChrArray : STRING;                                //mindestens String[32/64] übergeben
    Convert :  BOOL;                                //Zahlenwandlung aktivieren
    Addr :    POINTER;                              //Zeigt auf Variable, die konvertiert wird
    DataType : BYTE;                                //Datentyp der Variablen
    StringAddr : INT;                               //rechtsbündige Stringadresse (1...32/64)
    Digits :  BYTE;                                //Anzahl Nachkommastellen (1...3)
  END VAR
  VAR OUTPUT
    Error :    BOOL;                                //Konvertierungs- oder Stringfehler
  END VAR

```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion BHGDisp:

Signal	Art	Typ	Wertebereich	Bemerkung	
Row	E	BYTE	0 ... B#16#F	Display-Zeile	
				"binäre" Auswertung:	
				0:	keine Display-Ausgabe
				1:	Zeile 1
				2:	Zeile 2
				3:	Zeile 1 und Zeile 2 im Wechsel
				4:	Zeile 3
				5:	Zeile 1 und Zeile 3 im Wechsel
				8:	Zeile 4
B#16#F:	automatischer Wechsel aller 4 Zeilen				

Signal	Art	Typ	Wertebereich	Bemerkung	
ChrArray	E	STRING	>= string[32] [DBName].[VarName]	Hier ist der komplette Display-Inhalt abgelegt. Bei HT2 muss der String mit 64 Zeichen angelegt sein.	
Convert	E	BOOL		Aktivierung der Zahlenwandlung	
Addr	E	Pointer		Zeigt auf die Variable, die konvertiert werden soll	
DataType	E	BYTE	1 ... 8, B#16#13	Datentyp der Variable	
				1:	BOOL, 1 Zeichen
				2:	BYTE, 3 Zeichen
				3:	CHAR, 1 Zeichen
				4:	WORD, 5 Zeichen
				5:	INT, 6 Zeichen
				6:	DWORD, 7 Zeichen
7:	DINT, 8 Zeichen				
8:	REAL, 9 Zeichen (7 Ziffern plus Vorzeichen und Dezimalpunkt; Nachkommastellen siehe Parameter Digits)				
B#16#13:	String, bis 32/64 Zeichen, Addr muss Pointer auf STRING sein.				
B#16#30	REAL64, (12 Zeichen: 10 Ziffern plus Vorzeichen und Dezimalpunkt; Nachkommastellen siehe Parameter Digits)				
StringAddr	E	INT	1 ... 32/64	Rechtsbündige Adresse innerhalb der Variablen ChrArray	
Digits	E	BYTE	bei DataType REAL: 1 ... 4	Anzahl Nachkommastellen	
			bei DataType REAL64: 1 ... 9		
Error	A	BOOL		Fehler: <ul style="list-style-type: none"> <li>Angelegtes Chr/Array zu klein,</li> <li>Konvertierungsfehler,</li> <li>Zahlenüberlauf,</li> <li>StringAddr fehlerhaft</li> </ul>	

## Wertebereiche

Wertebereiche von Datentypen	
Datentyp	darstellbarer Zahlenbereich
BOOL	0, 1
BYTE	0 bis 255
WORD	0 bis 65535
INT	- 32768 bis + 32767
DWORD	0 bis 9999999
DINT	- 9999999 bis + 9999999
REAL (Digits := 1)	- 999999.9 bis + 999999.9
REAL (Digits := 2)	- 99999.99 bis + 99999.99
REAL (Digits := 3)	- 9999.999 bis + 9999.999
REAL (Digits := 4)	- 999.9999 bis + 999.9999
...	...
REAL (Digits := 9)	- 0.9999999 bis + 0.9999999

## Aufrufbeispiel

//DB mit Namen strdat in der Symboltabelle, Datenelement disp ist als String[32] (bei HT2:  
//String[64]) deklariert und komplett zugewiesen mit Zeichen

```
CALL FC 13 (
  Row :=          MB 26,
  ChrArray :=     "strdat".disp,
  Convert :=      M 90.1,
  Addr :=         P#M 20.0,           //zu konvertierende Zahl
  DataType :=     MB 28,             //Datentyp der Variablen
  StringAddr :=   MW 30,
  Digits :=       B#16#3,           //3 Nachkommastellen
  Error :=        M 90.2);
```

### 12.15.21 FC 17: YDelta Stern-Dreieck-Umschaltung

#### Funktion

Mit dem Baustein für die Stern-Dreieck-Umschaltung wird eine definierte Umschaltlogik zeitlich so gesteuert, dass diese Umschaltung auch bei laufender Spindel in beiden Richtungen vorgenommen werden kann. Der Baustein ist nur für digitale Hauptspindelantriebe verwendbar und muss für jede Spindel getrennt aufgerufen werden.

Die Umschaltung erfolgt über 2 getrennte Schütze und wird in 4 Schritten abgearbeitet:

Schritt 1:	Löschen des Nahtstellensignals DB31, ... .DBX21.5 (Motoranwahl erfolgt) im zugehörigen Achs-DB und Anmelden des Umschaltvorgangs über A mit DB31, ... .DBX21.3 (Motoranwahl).
Schritt 2:	Sobald die Rückmeldung NST DB31, ... .DBX93.7 (Impulse freigegeben) = 0 und die Quittierung der angemeldeten Motoranwahl vom Antrieb vorliegt, wird das bisher angezogene Schütz ausgeschaltet.
Schritt 3:	Nach der vom Anwender im Parameter "TimeVal" parametrisierten Zeit wird das andere Schütz eingeschaltet.
Schritt 4:	Nach einer erneuten Zeit wird die Umschaltung an den Antrieb mit NST DB31, ... .DBX21.5 (Motoranwahl erfolgt) gemeldet.

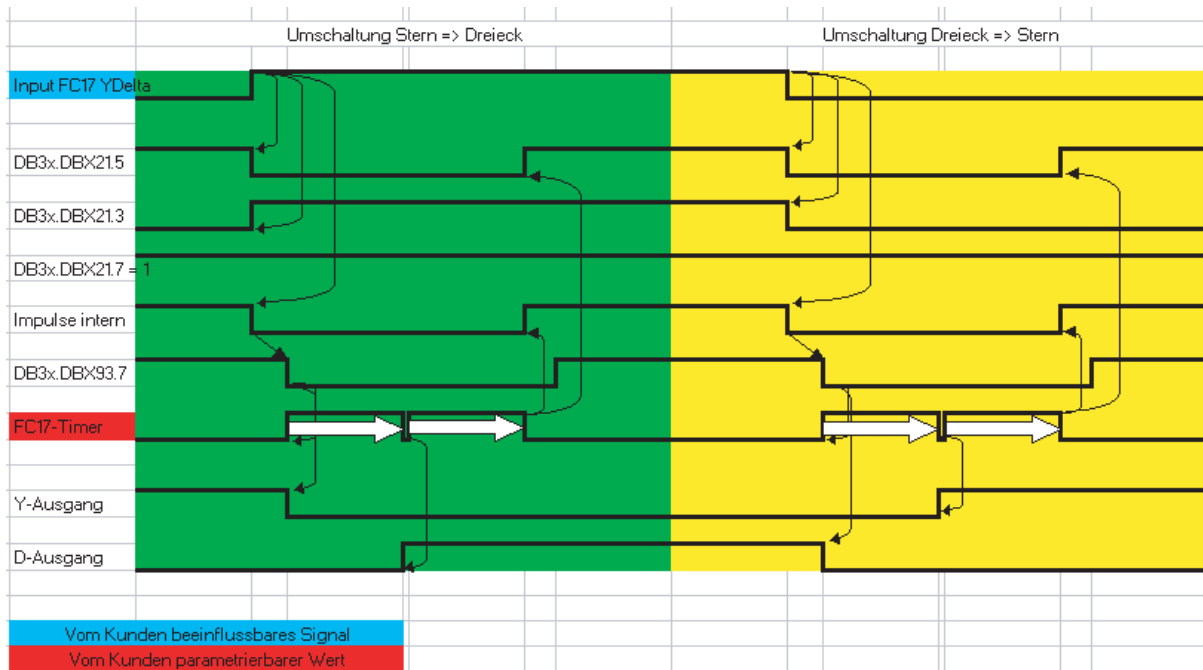


Bild 12-28 Stern-Dreieck-Umschaltung

Weitere Erläuterungen zu Motordrehzahlanpassungen siehe:

**Literatur:**

Funktionshandbuch Grundfunktionen; Spindeln (S1); Kapitel "Projektierbare Getriebearrangungen"

Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertesyst., Regelung (G2)

## Fehlermeldung

Wenn Parameter "SpindleIFNo" nicht im zulässigen Bereich liegt, erfolgt Stopp der PLC mit Ausgabe der Alarmmeldungs-Nummer 401702.

## Besonderheiten

Bei Parametrierung des "TimeVal" mit dem Wert 0 wird ein Standardwert von 100 ms verwendet. Bei einem Wert kleiner 50 ms wird der Minimalwert von 50 ms eingestellt.

Der Baustein ist absolut aufzurufen.

---

### Hinweis

Eine Umschaltung erfolgt nicht, wenn die Spindel in einer Achs-Betriebsart wie z. B. M70, SPOS ist.

---

## Randbedingungen

Mit der Stern-Dreieck-Umschaltung digitaler Hauptspindelantriebe wird ein Vorgang angestoßen der auch reglungstechnische Abläufe beinhaltet. Da die Regelung die automatische Stern-Dreieck-Umschaltung unterstützt, ergeben sich einige Randbedingungen, die zu beachten sind:

- Aufgrund des automatischen Wegschaltens der Impulse im Antrieb werden gleichzeitig mit dem NST DB31, ... .DBX93.7 (Impulse freigegeben) die NST DB31, ... .DBX61.7 (Stromregler aktiv) und DB31, ... .DBX61.6 (Drehzahlregler aktiv) weggeschaltet.
- Wird bei drehender Spindel und eingeschaltetem Lageregler der Spindel mit NST DB31, ... .DBX61.5 (Lageregler aktiv) von Stern nach Dreieck umgeschaltet, so führt dies zum Alarm 25050 "Konturüberwachung".
- Eine angestoßene Stern-Dreieck-Umschaltung mit FC 17 kann nicht vom Anwender z. B. durch betriebsmäßiges Warten auf eine erfolgreich durchgeführte Umschaltung der Stern-Dreieck-Schütze verzögert werden. Dieses Signalspiel kann vom Anwender durch eine PLC-Logik realisiert werden.

**Deklaration der Funktion**

**AWL-Darstellung**

```

VAR_INPUT
  YDelta :      BOOL;           //Stern = 0, Dreieck = 1
  SpindleIFNo : INT;           //Maschinenachs-Nummer
  TimeVal :     S5TIME;        //Zeitwert
  TimerNo :     INT;           //Timer des Anwenders für Umschaltzeit
END_VAR
VAR_OUTPUT
  Y :          BOOL;           //Stern-Schütz
  Delta :      BOOL;           //Dreieck-Schütz
END_VAR
VAR_IN_OUT
  Ref :        WORD;           //Zustandswort des Bausteins (Instanz)
END_VAR
    
```

**Erläuterung der Formalparameter**

Die folgende Tabelle zeigt alle Formalparameter der Funktion YDelta:

Signal	Art	Typ	Wertebereich	Bemerkung
YDelta	E	BOOL		= Stern = Dreieck Die Umschaltflanke des Signals löst die Umschaltung aus.
SpindleIFNo	E	INT	1 ...	Nummer der Achsschnittstelle, die als Spindel deklariert ist.
TimeVal	E	S5time	0 ...	Umschaltzeit
TimerNo	E	INT	10 ...	Zeitglied für die Programmierung der Wartezeit.
Y	A	BOOL		Ansteuerung des Stern-Schütz
Delta	A	BOOL		Ansteuerung des Dreieck-Schütz
Ref	E/A	WORD		Instanz für Zustandsinformationen. Interne Verwendung

## Aufrufbeispiel

```
CALL FC 17 (  
    YDelta :=          e 45.7,          //Stern Dreieck  
    SpindleIFNo :=    4,  
    TimeVal :=        S5T#150ms,  
    TimerNo :=        10,              //Timer 10  
    Y :=              a 52.3,          //Stern-Schütz  
    Delta :=          a 52.4,          //Dreieck-Schütz  
    Ref :=            mw 50);          //Instanz
```

## 12.15.22 FC 18: SpinCtrl Spindelsteuerung

### Funktion

Mit dem FC SpinCtrl können Spindeln und Achsen von der PLC gesteuert werden.

#### Literatur:

Funktionshandbuch Grundfunktionen; Spindeln (S1)

Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

Funktionshandbuch Erweiterungsfunktionen; Teilungsachsen (T1)

Der Baustein unterstützt dabei die Funktionen:

- Spindel positionieren
- Spindel drehen
- Spindel pendeln
- Teilungsachsen
- Positionierachsen

Jede Funktion wird durch die positive Flanke des entsprechenden Anstoß-Signals (Start, Stop) angestoßen. Das Anstoß-Signal muss solange auf logisch "1" bleiben, bis die Funktion durch InPos = "1" oder Error = "1" positiv oder negativ quittiert wurde. Die Ausgangsparameter werden mit Rücksetzen des jeweiligen Anstoß-Signals und dem Ende der Funktion abgelöscht.

Um die Spindel/Achsen über die PLC zu beeinflussen, muss sie für die PLC aktiviert werden. Dieses kann z.B. durch Aufruf des FC "SpinCtrl" mit Aktivieren des Parameters "Start" bzw. "Stop" erreicht werden. Hierbei fordert der FC "SpinCtrl" die Kontrolle der Spindel/Achse von der NC an.

Die NC meldet in der zugehörigen Spindel-/Achsen-Nahtstelle DB 31, ... im Byte 68 den Status dieser Spindel/Achse zurück (siehe Nahtstellen-Listen). Nachdem die Achse / Spindel der PLC gehört, kann über die zugehörige Achsnahtstelle der Fahrbefehl für den Aktivzustand ausgewertet werden.

Nach Beendigung ("InPos" ist TRUE, "Start" wechselt auf Null) wird die Kontrolle der Achse/Spindel vom FC "SpinCtrl" in einen neutralen Zustand geschaltet.

Alternativ kann auch das PLC-Anwenderprogramm vor Aufruf des FC "SpinCtrl" die Kontrolle für die PLC anfordern.

Hierdurch kann ein besseres Reaktionsverhalten der Spindel/Achse bei mehrfach hintereinander folgenden Aufrufen dieser Funktion erreicht werden, da der Umschaltvorgang im FC entfällt.

Die Aktivierung durch das PLC-Anwenderprogramm wird in der zugehörigen Spindel-Nahtstelle im Byte 8 durchgeführt.

Nach Rückgabe der Kontrolle kann die Spindel vom NC-Programm wieder programmiert werden.

---

#### **Hinweis**

##### **Aufrufhinweis:**

Der FC 18 ist zyklisch aufzurufen, bis das Signal "InPos" , bzw. bei Fehler das Signal "Error", einen Flankenwechsel von 1 nach 0 liefert. Erst wenn das Signal "InPos"/"Error" den Wert 0 geliefert hat, ist ein weiterer "Start" oder "Stop" für diese Spindel/Achse möglich. (mindestens ein PLC-Zyklus muss mit dem nächsten "Start" oder "Stop" gewartet werden). Dies gilt auch bei Veränderung der Zuordnung im Datenbyte 8 der axialen Nahtstelle.

##### **Abbruch:**

Ein Abbrechen der Funktion ist nicht möglich über den Parameter "Start" oder "Stop", sondern nur durch die axialen Nahtstellensignale (z. B. Restweglöschen). Ebenso liefert die axiale Nahtstelle Statussignale der Achse zurück, die gegebenenfalls auszuwerten sind (z. B. Genauhalt, Fahrbefehl).

##### **InPos bei Spindel - Drehen/Pendeln:**

Bei der Funktion "Spindel drehen" und auch bei "Spindel pendeln" ist die Bedeutung des Parameters "InPos" wie folgt festgelegt:

Solldrehzahl wird ausgegeben → Funktion wurde ohne Fehler gestartet.

Das Erreichen der geforderten Spindeldrehzahl muss über die Spindelnahtstelle ausgewertet werden.

##### **Gleichzeitigkeit:**

Mehrere Achsen können zeitgleich oder auch zeitversetzt durch den Bausteine FC 18 verfahren werden. Die Obergrenze ist über die max. Achsanzahl begrenzt. Der NCK wickelt die Funktionsanforderung der PLC (FC 18) über eigenständige Schnittstellen je Achse / Spindel ab.

##### **Achssperre:**

Bei gesetzter Achssperre bewegt sich eine über FC 18 gesteuerte Achse nicht. Es wird nur ein simulierter Istwert erzeugt. (Verhalten wie bei NC-Programmierung).

---

 <b>WARNUNG</b>
--

Wenn mehrere Bausteinaufrufe (FC 18) für die gleiche Achse / Spindel im PLC-Anwenderprogramm programmiert wurden, dann ist eine Verriegelung dieser Funktionen durch bedingte Aufrufe im Anwenderprogramm notwendig. Der bedingte Aufruf eines gestarteten (Parameter Start oder Stop = TRUE) ist solange zyklisch aufzurufen bis ein Zustandswechsel des Ausgangs-Parameters "Activ" bzw. "InPos" von 1 nach 0 erfolgt.
--



## Funktionen

### 1. Spindel positionieren:

Folgende Signale sind relevant:

Start :	Anstoß-Signal
Funct :	"1" = Spindel positionieren
Mode :	Positionierungs-Mode 1, 2, 3, 4
AxisNo :	Nummer der Maschinenachse
Pos :	Position
FRate :	Positioniergeschwindigkeit, wenn FRate = 0, wird Wert aus MD35300 \$MA_SPIND_POSCTRL_VELO (Lageregeleinschaltdrehzahl) genommen
InPos :	wird bei Erreichen der Position mit "Genauhalt Fein" auf "1" gesetzt
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

### 2. Spindel drehen:

Folgende Signale sind relevant:

Start :	Anstoß-Signal für Start drehen
Stop :	Anstoß-Signal für Stop drehen
Funct :	"2" = Spindel drehen
Mode :	Positionierungs-Mode 5 (Drehrichtung M4) Positionierungs-Mode < >5 (Drehrichtung M3)
AxisNo :	Nummer der Maschinenachse
FRate :	Spindel-Drehzahl
InPos :	Funktion wurde ohne Fehler gestartet
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

### 3. Spindel pendeln:

Folgende Signale sind relevant:

Start :	Anstoß-Signal für Start pendeln
Stop :	Anstoß-Signal für Stop pendeln
Funct :	"3" = Spindel pendeln
AxisNo :	Nummer der Maschinenachse
Pos :	Sollgetriebestufe
InPos :	Solldrehzahl wird ausgegeben
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

Die Pendeldrehzahl stammt aus dem Maschinendatum:  
MD35400 \$MA\_SPIND\_OSCILL\_DES\_VELO

MD35010 \$MA_GEAR_STEP_ CHANGE_ENABLE = 0	Funktion	MD35010 \$MA_GEAR_STEP_ CHANGE_ENABLE = 1	Funktion
Pos = 0	Pendeln	Pos = 0	
Pos = 1	Pendeln	Pos = 1	Pendeln mit Getriebestufenwechsel M41
Pos = 2	Pendeln	Pos = 2	Pendeln mit Getriebestufenwechsel M42
Pos = 3	Pendeln	Pos = 3	Pendeln mit Getriebestufenwechsel M43
Pos = 4	Pendeln	Pos = 4	Pendeln mit Getriebestufenwechsel M44
Pos = 5	Pendeln	Pos = 5	Pendeln mit Getriebestufenwechsel M45

#### 4. Teilungsachsen verfahren:

Folgende Signale sind relevant:

Start : Anstoß-Signal  
Funct : "4" = Teilungsachse

---

#### Hinweis

Bei: Funct: "4" = Teilungsachse

Die Modulwandlung ist vergleichbar mit Teilungsposition anfahren über POS [AX] =CIC (wert) im Teileprogramm.

---

Mode :	Positionierungs-Mode 0, 1, 2, 3, 4
AxisNo :	Nummer der Maschinenachse
Pos :	Teilungs-Position
FRate :	Positioniergeschwindigkeit; wenn FRate = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
InPos :	wird bei Erreichen der Position mit "Genauhalt Fein" auf "1" gesetzt
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

**5. bis 8. Achsen positionieren:**

Folgende Signale sind relevant:

Start :	Anstoß-Signal
Funct :	"5 bis 8" = Achsen positionieren
Mode :	Positionierungs-Mode 0, 1, 2, 3, 4
AxisNo :	Nummer der Maschinenachse
Pos :	Position
FRate :	Positioniergeschwindigkeit; wenn FRate = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
InPos :	wird bei Erreichen der Position mit "Genauhalt Fein" auf "1" gesetzt
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

**9. Spindel drehen mit automatischer Getriebestufenwahl:**

Folgende Signale sind relevant:

Start :	Anstoß-Signal für Start drehen
Stop :	Anstoß-Signal für Stopp drehen
Funct :	"9" = Spindel drehen mit Getriebestufenwahl
Mode :	Positionierungs-Mode 5 (Drehrichtung M4)
	Positionierungs-Mode < > 5 (Drehrichtung M3)
AxisNo :	Nummer der Maschinenachse
FRate :	Spindel-Drehzahl
InPos :	Solldrehzahl wird ausgegeben
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

**10./11. Spindel drehen mit konstanter Schnittgeschwindigkeit:**

Voraussetzung zur Ausführung ist die Aktivierung der Funktion "konstante Schnittgeschwindigkeit" durch das NC-Programm.

Folgende Signale sind relevant:

Start :	Anstoß-Signal für Start drehen
Stop :	Anstoß-Signal für Stop drehen
Funct :	"B#16#0A = Spindel drehen mit konstanter Schnittgeschwindigkeit (m/min)
Funct :	"B#16#0B = Spindel drehen mit konstanter Schnittgeschwindigkeit (feet/min)
Mode :	Positionierungs-Mode 5 (Drehrichtung M4) Positionierungs-Mode <>5 (Drehrichtung M3)
AxisNo :	Nummer der Maschinenachse
FRate :	Schnittgeschwindigkeit
InPos :	Solldrehzahl wird ausgegeben
Error :	Bei Positionsfehler = "1"
State :	Fehlercode

**Deklaration der Funktion**

```

FUNCTION FC 18: VOID                //SpinCtrl
VAR_INPUT
    Start :      BOOL;
    Stop  :      BOOL;
    Funct :      BYTE;
    Mode  :      BYTE;
    AxisNo :     INT;
    Pos   :      REAL;
    FRate :      REAL;
END_VAR
VAR_OUTPUT
    InPos :      BOOL;
    Error  :      BOOL;
    State  :      BYTE;
END_VAR
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion SpinCtrl.

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		Start Spindel-Steuerung von PLC
Stop	E	BOOL		Stop Spindel-Steuerung von PLC
Funct	E	BYTE	1 bis B#16#0B	1: Spindel positionieren 2: Spindel drehen 3: Spindel pendeln 4: Teilungsachse 5: Positionierachse metrisch 6: Positionierachse inch 7: PosAchse metrisch mit Handradüberlagerung 8: PosAchse inch mit Handradüberlagerung 9: Spindel drehen mit automatischer Getriebestufenwahl A: Spindel drehen mit konstanter Schnittgeschwindigkeit (m/min) B: Spindel drehen mit konstanter Schnittgeschwindigkeit (feet/min)
Mode	E	BYTE	0 bis 5	0: Pos auf Absolutpos 1: Pos inkrementell 2: Pos kürzester Weg 3: Pos absolut, positive Anfahrriichtung 4: Pos absolut, negative Anfahrriichtung 5: Drehrichtung wie M4
AxisNo	E	INT	1 - 31	Nr. der zu verfahrenen Achse, Spindel
Pos	E	REAL	± 0,1469368 E -38 bis ± 0,1701412 E +39	Rundachse: Grad Teilungsachse: Teilungsposition Linearachse: mm oder inch
FRate	E	REAL	± 0,1469368 E -38 bis ± 0,1701412 E +39	Rundachse und Spindel: Umdr./Min siehe unterhalb der Tabelle zum Thema FRate
InPos	A	BOOL		1 = Position erreicht, bzw. Funktion ausgeführt
Error	A	BOOL		1 = Fehler
State	A	BYTE	0 bis 255	Fehlercode

## FRate

Die Vorschubgeschwindigkeit im FC 18 kann auch angegeben werden als:

- Schnittgeschwindigkeit in der Einheit m/min bzw. ft/min
- konstante Scheibenumfangsgeschwindigkeit in m/s bzw. ft/s

Voraussetzung für diese alternativen Geschwindigkeiten ist eine Aktivierung dieser Funktion durch das NC-Programm. Rückmeldungen für die erfolgreiche Aktivierung sind in der Achsnahtstelle im Byte 84 zu finden.

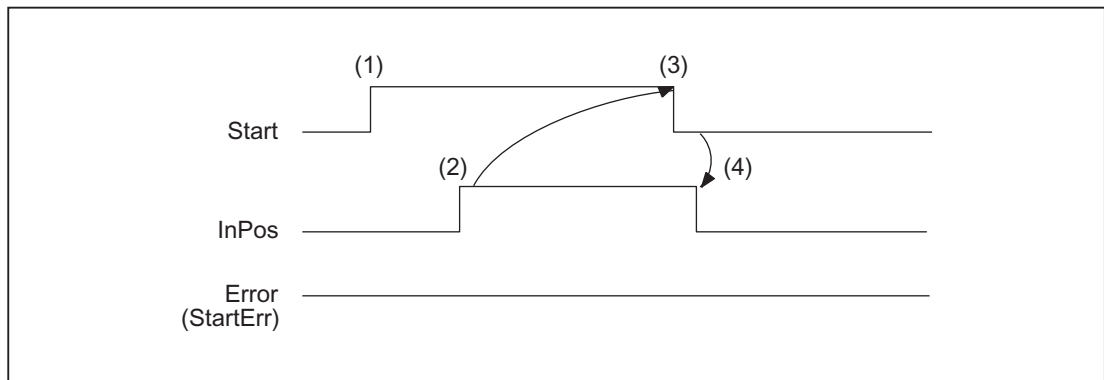
## Fehlerkennungen

Konnte eine Funktion nicht ausgeführt werden, wird dies am Zustandsparameter "Error" mit "logisch 1" angezeigt. Die Fehlerursache ist am Baustein Ausgang "State" kodiert:

State	Bedeutung
Fehler die durch PLC-Handling auftreten:	
1 B#16#1	es wurden mehrere Funktionen der Achse/Spindel gleichzeitig aktiviert
20 B#16#14	es wurde eine Funktion gestartet, ohne dass die Position erreicht wurde
30 B#16#1e	Die Achse/Spindel wurde vor Ende der Bewegung an die NC abgegeben
40 B#16#28	Die Achse ist durch NC Programm programmiert, NCK interner Fehler
50 B#16#32	fest zugeordnete PLC-Achse: verfährt (JOG) oder referiert
60 B#16#3C	fest zugeordnete PLC-Achse: Kanalzustand lässt keinen Start zu
Fehler die durch Behandlung des NCK auftreten. Die Alarmnummern sind im Diagnosehandbuch beschrieben:	
100 B#16#64	Falsche Position bei Achse/Spindel programmiert (entspricht Alarmnummer 16830)
101 B#16#65	Programmierte Drehzahl ist zu hoch
102 B#16#66	Falscher Wertebereich konstante Schnittgeschwindigkeit (entspricht Alarmnummer 14840)
104 B#16#68	Folgespindel: unerlaubte Programmierung (entspricht Alarmnummer 22030)
105 B#16#69	Kein Messsystem vorhanden (entspricht Alarmnummer 16770)
106 B#16#6a	Positioniervorgang der Achse noch aktiv (entspricht Alarmnummer 22052)
107 B#16#6b	Referenzmarke nicht gefunden (entspricht Alarmnummer 22051)
108 B#16#6c	Kein Übergang von der Drehzahlregelung in die Lageregelung (entspricht Alarmnummer 22050)
109 B#16#6d	Referenzmarke nicht gefunden (entspricht Alarmnummer 22051)
110 B#16#6e	Geschwindigkeit/Drehzahl ist negativ
111 B#16#6f	Soll-Drehzahl ist Null
112 B#16#70	ungültige Getriebestufe
115 B#16#73	programmierte Position wurde nicht erreicht

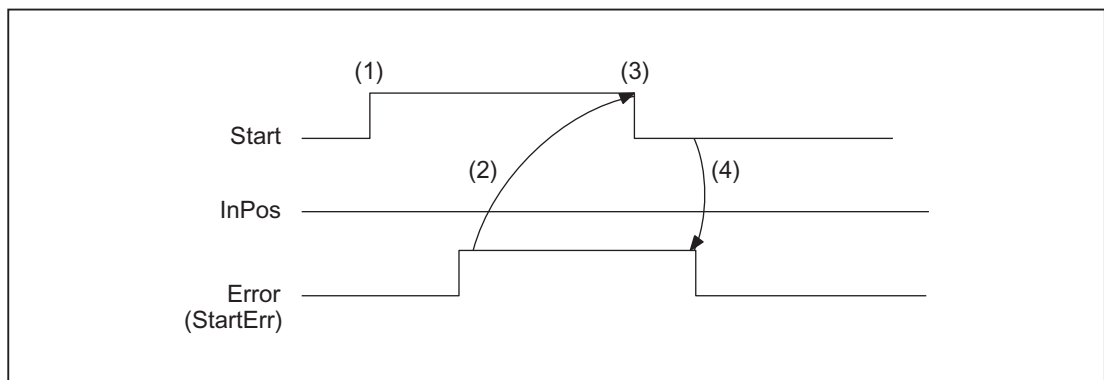
State	Bedeutung
117 B#16#75	in der NC ist G96/G961 <b>nicht aktiv</b>
118 B#16#76	in der NC ist G96/G961 <b>noch aktiv</b>
120 B#16#78	Achse ist keine Teilungsachse (entspricht NCK-Alarm 20072)
121 B#16#79	Teilungsposition- Fehler (entspricht NCK-Alarm 17510)
125 B#16#7d	DC (kürzester Weg) nicht möglich (entspricht NCK-Alarm 16800)
126 B#16#7e	Absolutwert Minus nicht möglich (entspricht NCK-Alarm 16820)
127 B#16#7f	Absolutwert Plus nicht möglich (entspricht NCK-Alarm 16810)
128 B#16#80	Keine Planachse für Durchmesserprogrammierung vorhanden (entspricht NCK-Alarm 16510)
130 B#16#82	Softwareendschalter Plus (entspricht NCK-Alarm 20070)
131 B#16#83	Softwareendschalter Minus (entspricht NCK-Alarm 20070)
132 B#16#84	Arbeitsfeldbegrenzung Plus (entspricht NCK-Alarm 20071)
133 B#16#85	Arbeitsfeldbegrenzung Minus (entspricht NCK-Alarm 20071)
134 B#16#85	Frame ist unzulässig für Teilungsachse
135 B#16#87	Teilungsachse mit "Hirth-Verzahnung" ist aktiv (entspricht NCK-Alarm 17501)
136 B#16#88	Teilungsachse mit "Hirth-Verzahnung" aktiv und Achse nicht referiert (entspricht NCK-Alarm 17503)
137 B#16#89	Spindelbetrieb für transformierte Spindel/Achse nicht möglich (entspricht NCK-Alarm 22290)
138 B#16#8A	für die Achse wird die entsprechende wirksame koordinatensystem-spezifische Arbeitsfeldbegrenzung Plus verletzt (entspricht NCK-Alarm 20082)
139 B#16#8B	für die Achse wird die entsprechende wirksame koordinatensystem-spezifische Arbeitsfeldbegrenzung Minus verletzt entspricht NCK-Alarm 20082)
System- oder andere schwere Alarme:	
200 B#16#c8	entspricht System-Alarmnummer 450007

## Impulsdiagramm



- (1) Funktionsanstoß durch positive Flanke durch Start oder Stop
- (2) Positive Quittung: Funktion ausgeführt / Position erreicht
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC

## Impulsdiagramm (Fehlerfall)



- (1) Funktionsanstoß durch positive Flanke durch Start oder Stop
- (2) Negative Quittung: Fehler aufgetreten
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC



## Aufrufbeispiele

### 1. Spindel positionieren:

```
//positive Quittung setzt Start zurück:
U M112.0;           //InPos
R M 100.0;         //Start
//negative Quittung , nach Fehlerauswertung (State: MB114) mit T12 Start rücksetzen
U M113.0;         //Error
U E 6.4;          //Taste T12
R M 100.0;         //Start
//Starten mit T13
U E 6.3;          //Taste T13
UN M 112.0;       //neuen Start erst wenn InPos bzw. Error = 0
UN M 113.0;
S M 100..0;

CALL FC 18 (
    Start :=      M100.0,
    Stop  :=      FALSE,
    Funct :=      B#16#1,           //Spindel positionieren
    Mode  :=      B#16#2,           //kürzester Weg
    AxisNo :=     5,
    Pos   :=      MD104,
    FRate :=      MD108,
    InPos :=      M112.0,
    Error :=      M113.0,
    State :=      MB114);
```

### 2. Start Spindel drehen:

```
CALL FC 18 (
    Start :=      M100.0,
    Stop  :=      FALSE,
    Funct :=      B#16q#,           //Spindel drehen
    Mode  :=      B#16#5,           //Drehrichtung wie M4
    AxisNo :=     5,
    Pos   :=      0.0,
    FRate :=      MD108,
    InPos :=      M112.0,
    Error :=      M113.0,
    State :=      MB114);
```

### 3. Start Spindel pendeln:

```
CALL FC 18 (  
  Start :=      M100.0,  
  Stop :=      FALSE,  
  Funct :=     B#16#3,           //Spindel pendeln  
  Mode :=     B#16#0,  
  AxisNo :=    5,  
  Pos :=      0.0,  
  FRate :=    MD108,  
  InPos :=    M112.0,  
  Error :=    M113.0,  
  State :=    MB114);
```

### 4. Teilungsachse verfahren

```
CALL FC 18 (  
  Start :=      M100.0,  
  Stop :=      FALSE,           //nicht genutzt  
  Funct :=     B#16#4,           //Teilungsachse verfahren  
  Mode :=     B#16#0,           //absolut positionieren  
  AxisNo :=    4,  
  Pos :=      MD104,           //Vorgabe in REAL: 1.0;2.0;..  
  FRate :=    MD108,  
  InPos :=    M112.0,  
  Error :=    M113.0,  
  State :=    MB114);
```

### 5. Achsen positionieren

```
CALL FC 18 (  
  Start :=      M100.0,  
  Stop :=      FALSE,           //nicht genutzt  
  Funct :=     B#16#5,           //Achsen positionieren  
  Mode :=     B#16#1,           //inkrementell positionieren  
  AxisNo :=    6,  
  Pos :=      MD104,  
  FRate :=    MD108,  
  InPos :=    M112.0,  
  Error :=    M113.0,  
  State :=    MB114);
```

## 12.15.23 FC 19: MCP\_IFM Übertragung der MCP-Signale an die Nahtstelle

### Funktion

Mit dem FC MCP\_IFM (M-Variante) werden von der Maschinensteuertafel einer Breite von 19 Zoll z. B. MCP 483 an die entsprechenden Signale der NCK-/PLC-Nahtstelle übertragen:

- Betriebsarten
- Achsanwahlen
- WKS/MKS-Umschaltung
- Verfahrstasten
- Overrides
- Schlüsselschalter

Im Grundprogramm (FC 2) werden weiterhin die Handradanwahlen, Betriebsarten und weitere Bediensignale von der Bedientafel (HMI) an die NCK-/PLC-Nahtstelle so übertragen, dass bei den Betriebsarten eine wahlweise Anwahl von der MCP oder von der HMI her möglich ist.

Die Übertragung der HMI-Signale an die Nahtstelle kann durch Setzen des Parameters "MMCToIF" im FB 1 (DB 7) auf den Wert "FALSE" abgeschaltet werden.

Für **Vorschub-Override**, **Achsfahr**- und **INC-Tasten** gelten abhängig von der aktiven Betriebsart bzw. vom angewählten Koordinatensystem folgende Festlegungen:

- **Vorschub-Override:**
  - Der Vorschub-Override wird auf die Nahtstelle des angewählten Kanals und auf die Nahtstelle der Achsen transferiert.
  - Die Vorschub-Override-Signale werden zusätzlich zum Nahtstellenbyte "Eilgangkorrektur" (DBB 5) an den NC-Kanal übergeben, wenn das HMI-Signal "Vorschubkorrektur für Eilgang wirksam" gesetzt ist (Ausnahme: Schalterstellung "Null"). Weiterhin wird mit diesem HMI-Signal auch "Eilgangkorrektur wirksam" gesetzt.
- **Maschinenfunktion INC- und Achsfahrtasten:**
  - Bei angewähltem MKS werden die Signale auf die Nahtstelle der angewählten Maschinenachse transferiert.
  - Bei angewähltem WKS werden die Signale auf die Geo-Achs-Nahtstelle des parametrisierten Kanals transferiert.
  - Bei Umschaltung zwischen MKS und WKS erfolgt generell eine Abwahl der bis dahin angewählten Achse.

Die **Handrad-Anwahlsignale von HMI** werden decodiert und in der zugehörigen Maschinen-Achs-Nahtstelle oder in der Geo-Achs-Nahtstelle des jeweiligen Handrads aktiviert (nur wenn im FB 1 der Parameter "HWheelMMC: = TRUE" ist).

Die Ansteuerung der zugehörigen LED der Maschinensteuertafel wird aus der Rückmeldung entsprechender Anwahlen abgeleitet.

Vorschub- und Spindel-Start/Stop werden nicht an die Nahtstelle übertragen, sondern als Signal "FeedHold" bzw. "SpindleHold" selbsthaltend ausgegeben. Der Anwender kann diese Signale mit weiteren Signalen verknüpfen, die zu Vorschub- oder Spindel-Halt führen sollen (dies kann z. B. über die entsprechenden Eingangssignale des FC 10: AL\_MSG erfolgen). Zusätzlich werden die zugehörigen LED mit angesteuert.

Bei Ausfall der Maschinensteuertafel werden die Signale, die von dieser kommen, mit Null vorbesetzt; ebenso die Ausgangssignale "FeedHold" und "SpindleHold".

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 19 bzw. auch FC 24, FC 25, FC 26 zulässig. Hierbei steuert der erste Aufruf im Zyklus die LED-Anzeigen an. Weiterhin werden im ersten Aufruf alle Aktionen des parametrisierten Bausteins durchgeführt. Bei den weiteren Aufrufen findet nur noch eine reduzierte Bearbeitung von Kanal und BAG-Schnittstelle statt. Die Geometrieachsen werden nur im ersten Aufruf des Bausteins im Zyklus mit Richtungsvorgaben versorgt.

Die Einzelsatz-An-/Abwahl wird nur durch den ersten Aufruf im Zyklus beeinflusst.

Die zweite Maschinensteuertafel kann bearbeitet werden, wenn der Parameter "BAGNo" um B#16#10 erhöht ist. Bei der Parametrierung ist die BAG Nummer in dem unteren Nibble (untere 4 Bits) enthalten.

"BAGNo" = 0 oder B#16#10 bedeutet keine Bearbeitung der BAG Signale.

"ChanNo" = 0 bedeutet keine Bearbeitung der Kanal-Signale.

Die INC-Anwahlen werden in die BAG-Schnittstelle übertragen. Die Aktivierung für diese Vorgabe findet über den DB10.DBX57.0 (INC-Eingänge im BAG-Bereich aktiv) durch diesen Baustein einmalig nach Hochlauf statt.

Weiterhin können zwei Maschinensteuertafeln parallel durch diesen Baustein bearbeitet werden. Hierbei ist der Aufruf des Bausteins für die 2. Maschinensteuertafel im OB 1-Zyklus zeitlich hinter den Aufruf für die 1. MCP zu setzen. Eine Unterstützung von 2 MCP ist in den Maschinensteuertafel-Bausteinen bis zu bestimmten Grenzen vorhanden (nicht unterstützt werden vom Standard gegenseitige Verriegelungen der Achsanwahlen bei gleich zugeordneten Achsen bei 2 MCP).

## Flexible Achskonfiguration

Es ist eine Flexibilität bezüglich der Zuordnung Achsanwahlen bzw. Richtungstasten von Maschinen-Achsnummern verfügbar.

Der Einsatz von 2 Maschinensteuertafeln, die gleichzeitig betrieben werden sollen, wird durch die MCP Bausteine insbesondere für den Anwendungsfall 2 Kanäle, 2 BAG besser unterstützt. Bei den Achs-Tabellen der jeweiligen MCP ist zu beachten, dass die Achsnummern auch in der parametrisierten BAG Nummer des MCP-Bausteins angegeben sind.

Für diese Flexibilität existieren Tabellen für Achsnummern im DB 10.

Für die **erste** Maschinensteuertafel (MCP) beginnt die Tabelle ab dem Byte 8 (symbolischer Name: MCP1AxisTbl[1..22]) und für die **zweite** Maschinensteuertafel (MCP) ab dem Byte 32 (symbolischer Name: MCP2AxisTbl[1..22]). Hier sind die Maschinen-Achsnummern Byte-weise einzutragen.

Der Eintrag 0 ist in der Achstabelle zulässig. Eine Prüfung auf eine unzulässige Achsnummer findet nicht statt und kann bei Falscheintrag zum PLC-Stop führen.

Für den **FC 19** kann auch eine Begrenzung der **maximal möglichen Achsanwahlen** erfolgen. Diese Obergrenze wird für die erste Maschinensteuertafel im DB10.DBW30 (symbolischer Name: MCP1MaxAxis) bzw. für die zweite Maschinensteuertafel im DB10.DBW54 (symbolischer Name: MCP2MaxAxis) eingestellt.

Der voreingestellte Wert ist 0, damit wirkt die konfigurierte maximale Achszahl. Die Achsnummern und die Begrenzung können auch dynamisch angepasst werden. Danach muss eine erneute Achsanwahl bei FC 19 erfolgen. Während des Fahrens von Achsen über die jeweiligen Richtungstasten darf keine Umschaltung der Achsnummern erfolgen. Voreingestellt ist der Kompatibilitätsmodus mit den Achsnummern **1 bis 9** für beide MCP und die Begrenzung auf die konfigurierte Achszahl.

## Beispiel

Es sollen mehr als 9 Achsen mit dem FC 19 über Zusatzapplikation angesteuert werden. Folgende Vorgehensweise ist erforderlich:

- Freie Taste an MCP reservieren
- Diese Taste als Flipflop auswerten
- Ausgang des FlipFlops als positiver und negativer Flanke auswerten
- Bei positiver Flanke einen Satz Achsnummern in Achstabelle (DB 10) schreiben und LED über dieser Taste einschalten
- Bei negativer Flanke einen anderen Satz Achsnummern in Achstabelle (DB 10) schreiben und LED über dieser Taste ausschalten

## Deklaration der Funktion

```

FUNCTION FC 19: VOID                                //symbolischer Name: MCP_IFM

  VAR_INPUT
    BAGNo :          BYTE;
    ChanNo :         BYTE;
    SpindleIFNo :    BYTE;
  END_VAR

  VAR_OUTPUT
    FeedHold :       BOOL;
    SpindleHold :    BOOL;
  END_VAR

BEGIN
END_FUNCTION

```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "MCP\_IFM":

Signal	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	0 - b#16#0A und b#16#10 - b#16#1A	BAG-Nr., in die die Betriebsartensignale übertragen werden. BAGNo >= b#16#10 bedeutet Zugriff auf die 2. Maschinensteuertafel.
ChanNo	E	BYTE	0 - B#16#0A	Kanal-Nr. für die Kanalsignale
SpindleIFNo	E	BYTE	0 - 31 (B#16#1F)	Nummer der Achsschnittstelle, die als Spindel deklariert ist.
FeedHold	A	BOOL		Vorschub Halt von MCP, selbsthaltend
SpindleHold	A	BOOL		Spindel Halt von MCP, selbsthaltend

### Anwahlsignale der MCP auf die Anwender-Nahtstelle

Tabelle 12- 1 Schlüsselschalter

Quelle: MCP - Schalter	Ziel: Nahtstellen-DB
Stellung 0	DB10.DBX56.4
Stellung 1	DB10.DBX56.5
Stellung 2	DB10.DBX56.6
Stellung 3	DB10.DBX56.7

Tabelle 12- 2 Betriebsarten und Maschinenfunktionen

Quelle: MCP - Taster	Ziel: Nahtstellen-DB (Parameter BAGNo) Darstellung für BAG 1
AUTOMATIC	DB11.DBX0.0
MDA	DB11.DBX0.1
JOG	DB11.DBX0.2
REPOS	DB11.DBX1.1
REF	DB11.DBX1.2
TEACH IN	DB11.DBX1.0
INC 1 ... 10 000, INC Var.	DB11.DBX2.0 - 2.5

Tabelle 12- 3 Richtungstasten Eilgangüberlagerung

Quelle: MCP - Taster	Ziel: Nahtstellen-DB (Parameter ChanNo)
Richtungstaste +	DB21, ... .DBX12.7
Richtungstaste -	DB21, ... .DBX12.6
Eilgangüberlagerung	DB21, ... .DBX12.5
Richtungstaste +	DB21, ... .DBX16.7
Richtungstaste -	DB21, ... .DBX16.6
Eilgangüberlagerung	DB21, ... .DBX16.5
Richtungstaste +	DB21, ... .DBX20.7
Richtungstaste -	DB21, ... .DBX20.6
Eilgangüberlagerung	DB21, ... .DBX20.5

Quelle: MCP - Taster	Ziel: Nahtstellen-DB (alle Achs-DB)
Richtungstaste +	DB31, ... .DBX4.7
Richtungstaste -	DB31, ... .DBX4.6
Eilgangüberlagerung	DB31, ... .DBX4.5

Die Übertragung ist abhängig von der angewählten Achse. Für nicht angewählte Achsen werden die zugehörigen Nahtstellenbits gelöscht.

Tabelle 12- 4 Override

Quelle: MCP - Schalter	Ziel: Nahtstellen-DB (Parameter ChanNo)
Vorschuboverride	DB21, ... .DBB4

Quelle: MCP - Schalter	Ziel: Nahtstellen-DB (alle Achs-DB)
Vorschuboverride	DB31, ... .DBB0 (angewählte Achsnummer) Durch Vorschuboverride der 1. MCP werden alle Achsen beaufschlagt.
Spindeloverride	DB31, ... .DBB19 (Parameter SpindleIFNo)

Tabelle 12- 5 Kanalsignale

<b>Quelle: MCP - Tasten</b>	<b>Ziel: Nahtstellen-DB (Parameter ChanNo)</b>
NC-Start	DB21, ... .DBX7.1
NC-Stop	DB21, ... .DBX7.3
RESET	DB21, ... .DBX7.7
Einzelstart	DB21, ... .DBX0.4

Tabelle 12- 6 Vorschub, Spindelsignale

<b>Quelle: MCP - Tasten</b>	<b>Ziel: Ausgangsparameter vom FC</b>
Vorschub Halt Vorschub Freigabe	Parameter: "FeedHold" speichernd verknüpft, LED werden angesteuert
Spindel Halt Spindel Freigabe	Parameter: "SpindleHold" speichernd verknüpft, LED werden angesteuert

## Rückmeldungen der Anwender-Nahtstelle zur Ansteuerung von Anzeigen

Tabelle 12- 7 Betriebsarten und Maschinenfunktionen

<b>Ziel: MCP - LED</b>	<b>Quelle: Nahtstellen-DB (Parameter BAGNo) Darstellung für BAG 1</b>
AUTOMATIC	DB11.DBX6.0
MDA	DB11.DBX6.1
JOG	DB11.DBX6.2
REPOS	DB11.DBX7.1
REF	DB11.DBX7.2
TEACH IN	DB11.DBX7.0

<b>Ziel: MCP - LED</b>	<b>Quelle: Nahtstellen-DB (Parameter BAGNo) Darstellung für BAG 1</b>
INC 1 ... 10 000, INC Var.	DB11.DBX8.0 - 8.5



Tabelle 12- 8 Kanalsignale

Ziel: MCP - LED	Quelle: Nahtstellen-DB (Parameter ChanNo)
NC-Start	DB21, ... .DBX35.0
NC-Stop	DB21, ... .DBX35.2 oder DB21, ... .DBX35.3
Einzelsatz	DB21, ... .DBX0.4

**Hinweis**

LED der Richtungstasten werden durch Betätigen der Richtungstasten angesteuert.

Achsanwahl- und WKS/MKS-LED werden durch Betätigen der jeweiligen Taster angesteuert.

**Aufrufbeispiel**

```
CALL FC 19(           //Maschinensteuertafel-M-Variante Signale an Nahtstelle
  BAGNo :=           B#16#1,           //BAG Nr. 1
  ChanNo :=          B#16#1,           //Kanal Nr. 1
  SpindleIFNo :=    B#16#4,           //Spindel Schnittstellen-Nummer = 4
  FeedHold :=       m22.0,           //Vorschub Halt Signal selbsthaltend
  SpindleHold :=    db2.dbx151.0);    //Spindel Halt selbsthaltend in
                                           //Meldungs-DB
```

Mit dieser Parametrierung werden die Signale an die 1. BAG, den 1. Kanal und an alle Achsen übertragen. Zusätzlich wird der Spindel-Override in die 4. Achs-/Spindel-Nahtstelle übertragen. Das Vorschub-Halt-Signal wird an Merker 22.0 und das Spindel-Halt-Signal an den Datenbaustein DB 2, Datenbit 151.0 übergeben.

**Rangierung der Achsanwahlen**

Um eine flexible Zuordnung der Achsanwahltasten zur jeweiligen Achse oder Spindel zu ermöglichen, ist **kein Umschreiben oder Neuschreiben** des FC 19 erforderlich. Es muss nur die Achsnummer in der Achstabelle DB10.DBB8 und folgende wie gewünscht eingetragen werden. Es muss nur die Achsnummer in der Achstabelle DB10.DBB8 und folgende wie gewünscht eingetragen werden.

**Beispiel:**

Die Spindel ist als 4. Achse definiert und soll über die Achstaste 9 angewählt werden.

Lösung:

Im DB10 Byte (8 +(9-1)) ist der Wert 4 für die 4. Achse einzutragen.

```
CALL FC 19(           //Signale an Nahtstelle
  BAGNo :=           B#16#1,      //BAG Nr. 1
  ChanNo :=          B#16#1,      //Kanal Nr. 1
  SpindleIFNo :=     B#16#4,      //Spindel Schnittstellen-Nummer = 4

  FeedHold :=        m30.0,       //Vorschub Halt Signal selbsthaltend
  SpindleHold :=     m30.1);      //Spindel Halt selbsthaltend
```

## 12.15.24 FC 21: Transfer Datenaustausch PLC-NCK

### Funktion

Mit Aufruf des Bausteins Transfer werden Daten zwischen PLC und NCK entsprechend des angewählten Funktionscodes ausgetauscht. Die Daten werden sofort bei Aufruf des FC 21 übertragen, nicht erst am Zyklusbeginn.

Die Aktivierung des Bausteins erfolgt über das "Enable" - Signal.  
Der FC 21 wird nur bei "Enable" = "1" durchlaufen.

Folgende Funktionen für den Datenaustausch zwischen PLC und NCK werden unterstützt:

1. Signale Synchronaktionen an NCK - Kanal
2. Signale Synchronaktionen von NCK - Kanal
3. Schneller Datenaustausch PLC-NCK (Funktion lesen im NCK)
4. Schnellen Datenaustausch PLC-NCK (Funktion schreiben im NCK)
5. Steuerungssignale an NCK - Kanal aktualisieren
6. Steuerungssignale an Achsen aktualisieren (Datenbyte 2 der Anwender Nahtstelle)
7. Steuerungssignale an Achsen aktualisieren (Datenbyte 4 der Anwender Nahtstelle)

## Deklaration der Funktion

### AWL-Darstellung

```

VAR_INPUT
    Enable : BOOL ;
    Funct  : BYTE ;
    S7Var  : ANY  ;
    IVar1  : INT  ;
    IVar2  : INT  ;

END_VAR

VAR_OUTPUT
    Error   : BOOL ;
    ErrCode : INT  ;

END_VAR
    
```

## Erläuterungen der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "Transfer":

Signal	Art	Typ	Wertebereich	Bemerkung	
Enable	E	BOOL		1 = FC 21 aktiv	
Funct	E	BYTE	1 ... 7	1:	Synchronaktionen an Kanal
				2:	Synchronaktionen von Kanal
				3:	Daten lesen
				4:	Daten schreiben
				5:	Steuersignale an Kanal
				6, 7:	Steuersignale an Achse
S7Var	E	ANY	S7-Datenbereich	Abhängig von "Funct"	
IVAR1	E	INT	0 ...	Abhängig von "Funct"	
IVAR2	E	INT	1 ...	Abhängig von "Funct"	
Error	A	BOOL			
ErrCode	A	INT		Abhängig von "Funct"	

**Funktion 1, 2: Signale Synchronaktionen an / von Kanal**

Synchronaktionen können von PLC verriegelt bzw. freigegeben werden.  
 Der Datenbereich liegt in der Anwender-Nahtstelle DB21, ... .DBB300 ...307 (an Kanal) und DB21, ... .DBB308 ...315 (von Kanal). Der Parameter "S7Var" wird bei dieser Funktion nicht ausgewertet, muss aber mit einem Aktualparameter belegt werden (siehe Aufrufbeispiel). Die Daten werden sofort während des FC 21-Durchlaufs an/von NC übergeben.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1 = FC 21 aktiv
Funct	E	BYTE	1, 2	1: Synchronaktionen an Kanal
				2: Synchronaktion von Kanal
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	1..MaxKanal	Kanal-Nummer
Error	A	BOOL		
ErrCode	A	INT		1: "Funct" ungültig
				10: Kanal-Nr. ungültig

**Aufrufbeispiel:**

```

FUNCTION FC 100: VOID

VAR_TEMP
    myAny : ANY ;
END_VAR

BEGIN
NETWORK

//Synchronaktionen mit ID3, ID10 und ID31 deaktivieren im NC-Kanal 1:
SYAK :   AUF   DB21;
         SET;
         S     DBX 300.2;    //ID3
         S     DBX 301.1;    //ID10
         S     DBX 303.6;    //ID31
         L     B#16#1;
         T     MB11;
         SPA   TRAN;
    
```

```
//Synchronaktionen von NCK-Kanal 1:
SYVK :   L B#16#2;
         T MB11;
TRAN:   CALL          FC    21 (
        Enable        := M 10.0,          //wenn True, FC 21 aktiv
        Funct         := MB 11,
        S7Var         := #myAny,          //nicht genutzt
        IVAR1         := 1,              //Kanal-Nr
        IVAR2         := 0,
        Error         := M 10.1,
        ErrCode       := MW 12);
END_FUNCTION
```

### Funktion 3, 4: Schneller Datenaustausch PLC-NCK

#### Allgemeines

Für den schnellen Austausch von Informationen zwischen PLC und NCK existiert ein eigener interner Datenbereich. Die Dimension des internen Datenfeldes ist auf 4096 Bytes festgelegt. Die Zugriffe (Lesen/Schreiben) von PLC aus erfolgen über FC 21. Die Belegung dieses Bereichs (Struktur) muss im NC-Teileprogramm und PLC-Anwenderprogramm identisch vereinbart werden.

Vom NC-Teileprogramm kann auf diese Daten mit den Befehlen \$A\_DBB[x], \$A\_DBW[x], \$A\_DBD[x], \$A\_DBR[x] zugegriffen werden (siehe Listenhandbuch Systemvariablen).

Die konkrete Adresse im Datenfeld wird über einen Byte-Offset (0 bis 4095) im Parameter IVAR1 angegeben. Dabei muss die Ausrichtung entsprechend dem Datenformat gewählt werden, d. h. ein Dword beginnt auf einer 4-Byte-Grenze und ein Word auf einer 2-Byte-Grenze. Bytes können auf einem beliebigen Offset innerhalb des Datenfeldes liegen, Einzelbitzugriffe werden nicht unterstützt und vom FC 21 als Bytezugriff umgesetzt. Die Datentypinformation und Anzahl der Daten wird aus dem über S7Var übergebenen ANY-Parameter entnommen.

Die Datenkonsistenz wird nur für 1- und 2-Byte Zugriffe sowohl beim NCK als auch bei der PLC ohne zusätzliche programmiertechnische Maßnahmen sichergestellt. Bei der 2-Byte Konsistenz trifft dies nur beim Datentyp WORD bzw. INT zu, nicht aber beim Datentyp BYTE.

Bei Verwendung größerer Datentypen oder Übertragung von Feldern, die konsistent übertragen werden sollen, muss ein Semaphor-Byte im Parameter IVAR2 angegeben werden, über das der FC 21 die Gültigkeit bzw. Konsistenz eines Blockes erkennen kann. Dieses Handling muss auf der NC-Seite, d. h. im Teileprogramm, durch schreiben bzw. löschen des Semaphor-Bytes unterstützt werden. Das Semaphor-Byte liegt auch in dem gleichen Datenfeld, wie die eigentlichen Nutzdaten.

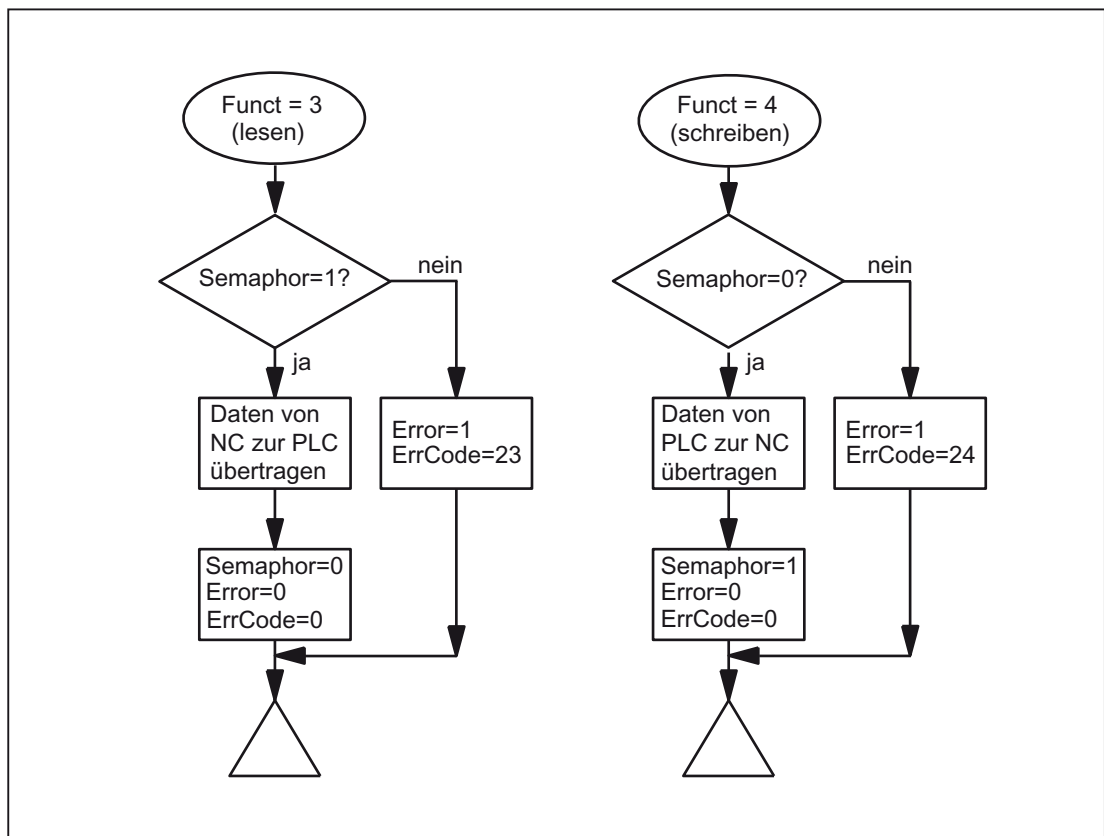
Ein Wert zwischen 0 und 4095 im IVAR2 bezeichnet das Semaphor-Byte.

Bei der PLC wird über den FC 21 in dem gleichen Aufruf, in denen die Nutzdaten übertragen werden sollen, das Semaphorbyte gelesen und beschrieben. Der PLC Programmierer muss nur für die Bereitstellung einer Semaphor-Variablen sorgen. Bei dem Zugriff von der NC über das Teileprogramm muss der Semaphor-Mechanismus entsprechend dem unten abgebildeten Flussdiagramm über Einzelanweisungen programmiert werden. Die Sequenz muss für Lesen bzw. Schreiben von Variablen unterschiedlich ausgeführt sein.

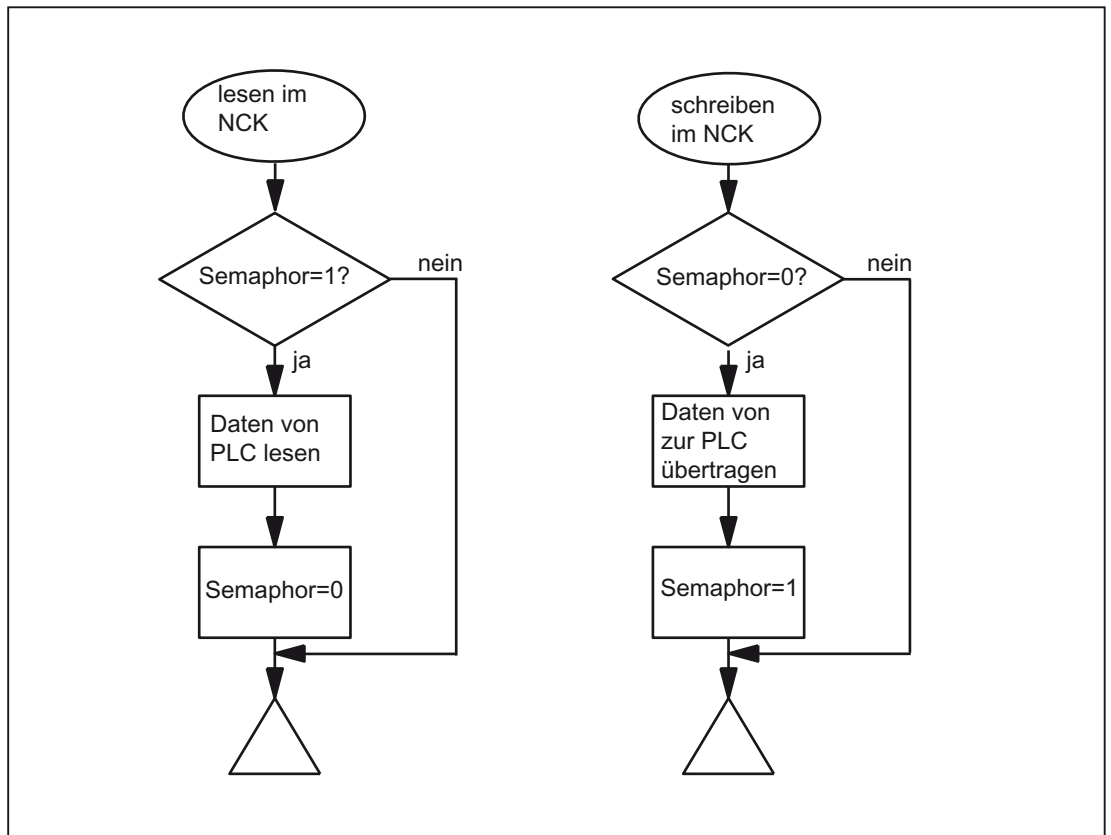
Es werden nur Einzelvariable bzw. ARRAYs direkt durch die Semaphortechnik unterstützt. Eine Übertragung von Strukturen ist in Einzelaufträge zu zerteilen. Hierbei ist für die Datenkonsistenz dieser Struktur mit einem, vom Anwender zu programmierenden, Semaphormechanismus selber zu sorgen.

Wird IVAR2 = -1 gesetzt, erfolgt die Datenübertragung ohne Semaphor.

**Datenaustausch mit Semaphor in PLC (Prinzipschaltung des FC21)**



Prinzipieller Aufbau im NCK:



### Variablen-Wertebereiche

Folgende Signale sind relevant:

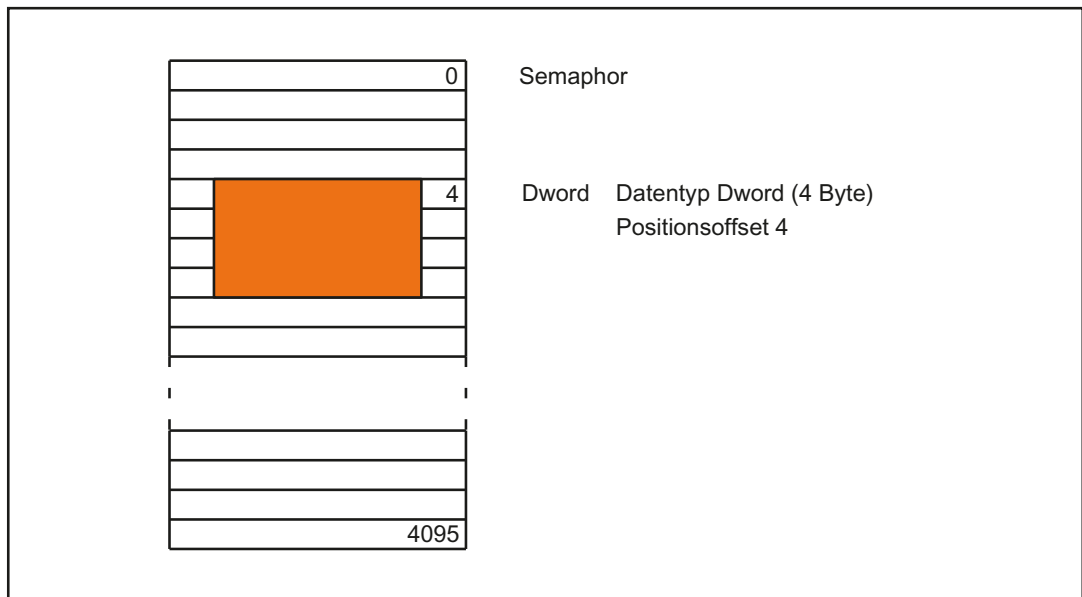
Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		= FC 21 aktiv
Funct	E	BYTE	3, 4	3: Daten lesen
				4: Daten schreiben
S7Var	E	ANY	S7-Datenbereich, außer Lokaldaten	Quell-/Ziel-Datenbereich
IVAR1	E	INT	0 ... 4095	Positionsoffset
IVAR2	E	INT	-1 ... 4095	Semaphor-Byte Transfer ohne Semaphor: -1
Error	A	BOOL		
ErrCode	A	INT		20: Ausrichtungsfehler
				21: unerlaubter Positionsoffset
				22: unerlaubtes Semaphor-Byte
				23: keine neue Daten zu lesen
				24: kann keine Daten schreiben
				25: Lokaldaten bei S7Var parametrier

**Aufrufbeispiel:**

**1. Doppelwort vom Positionsoffset 4 mit Semaphore im Byte 0 lesen und im MD100 speichern:**

Datentyp Dword (4 Byte)

Positionsoffset 4



```

CALL FC      21 (
    Enable    := M 10.0,           //wenn TRUE, FC 21 aktiv
    Funct     := B#16#3,           //Daten lesen
    S7Var     := P#M 100.0 DWORD 1,
    IVAR1     := 4,
    IVAR2     := 0,
    Error     := M 10.1,
    ErrCode   := MW12);
UN   M10.1;           //Enable so lange 1, bis Wert gelesen
R    M10.0;
    
```

**Beispiele: NCK-Programmierung aus Synchronaktionen**

Datentransfer von NC zu PLC mit Schreiben der Daten aus Synchronaktionen heraus, Byte0 dient als Semaphore:

```
ID=1 WHENEVER $A_DBB[0] == 0 DO $A_DBR[4] = $AA_IM[X] $A_DBB[0] = 1
```

Datentransfer von PLC zu NC mit Lesen der Daten aus Synchronaktionen heraus, Byte1 dient als Semaphore:

```
ID=2 WHENEVER $A_DBB[1] == 1 DO $R1 = $A_DBR[12] $A_DBB[1] = 0
```



## 2. Wort vom Positionsoffset 8 ohne Semaphor lesen und im MW 104 speichern:

```
CALL FC 21 (
    Enable :=M 10.0, //wenn TRUE, FC 21 aktiv
    Funct :=B#16#3, //Daten lesen
    S7Var :=P#M 104.0 WORD 1,
    IVAR1 :=8,
    IVAR2 :=-1,
    Error :=M 10.1,
    ErrCode :=MW12);
```

### Funktion 5: Steuersignale an Kanal aktualisieren

Diese Funktion dient einer schnellen Übertragung von wichtigen Steuersignalen, zwischen der zyklischen Datenübertragung. Die Datenbytes 6 und 7 der Anwender-Nahtstelle DB21, ... werden an die NC übertragen. Der Kanal wird im Parameter "IVAR1" angegeben. Hiermit kann z. B. die Vorschubsperrung, Einlesesperrung außerhalb des PLC Zyklus übertragen werden.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1= FC 21 aktiv
Funct	E	BYTE	5	5: Steuersignale an Kanal
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	1. MaxKanal	Kanal - Nummer
Error	A	BOOL		
ErrCode	A	INT		1: "Funct" ungültig
				10: Kanal - Nr. ungültig

### Funktion 6: Steuersignale an Achsen aktualisieren

Die Funktion 6 dient einer schnellen Übertragung von wichtigen Steuersignalen, zwischen der zyklischen Datenübertragung. Das **Datenbyte 2** der Anwender-Nahtstelle DB31, ... wird an die NC übertragen. Die Übertragung wird für alle aktivierten Achsen vorgenommen. Hiermit kann z. B. die Reglerfreigabe außerhalb des PLC-Zyklus übertragen werden.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1= FC 21 aktiv
Funct	E	BYTE	6	6: Steuersignale an Achsen
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	0	
Error	A	BOOL		
ErrCode	A	INT		1: "Funct" ungültig

### Funktion 7: Steuersignale an Achsen aktualisieren

Die Funktion 7 dient einer schnellen Übertragung von wichtigen Steuersignalen, zwischen der zyklischen Datenübertragung. Das **Datenbyte 4** der Anwender-Nahtstelle DB31, ... wird an die NC übertragen. Die Übertragung wird für alle aktivierten Achsen vorgenommen. Hiermit kann z. B. der Vorschubhalt außerhalb des PLC-Zyklus übertragen werden.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1= FC 21 aktiv
Funct	E	BYTE	7	7: Steuersignale an Achsen
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	0	
Error	A	BOOL		
ErrCode	A	INT		1: "Funct" ungültig

### 12.15.25 FC 22: TM\_DIR Richtungsauswahl für Werkzeugverwaltung

#### Funktion

Der Baustein TM\_DIR liefert den kürzesten Weg für die Positionierung eines Magazins oder eines Revolvers anhand der Ist- und Sollposition.

Solange der Eingang **Start** "1-Signal" besitzt, werden alle Ausgangsparameter zyklisch aktualisiert. Hierbei können in darauffolgenden PLC-Zyklen Veränderungen der Eingangsparameter (z. B. Positionswerte) vorgenommen werden.

Bei 0-Pegel des Startsignals sind die Ausgangssignale undefiniert.

Bei Richtungsauswahl mit Sonderpositionierung - Eingang "Offset" > 0 - wird eine neue Sollposition aus Soll-, Sonderposition, sowie der Anzahl Magazinplätze, nach der Formel berechnet:

Neue Sollpos. = (Sollpos. - (Sonderpos. - 1)) neg. modulo # Plätze

Die neue Sollposition entspricht der Platz-Nr., auf die das Magazin positioniert werden muss, damit die vom Anwender geforderte Sollposition auf der Platz-Nr. der Sonderposition steht. Die Richtungsoptimierung ist sowohl mit als auch ohne Sonderpositionierung aktiv.

Pro Magazin ist der Baustein einmal mit entsprechender Parametrierung aufzurufen.

 **WARNUNG**

Der Baustein darf nur in Verbindung mit der Werkzeugverwaltung aufgerufen werden.

**Hinweis**

Weitere Informationen zur Werkzeugverwaltung (auch in Bezug zur PLC) sind in der Funktionsbeschreibung Werkzeugverwaltung enthalten. Weiterhin stehen noch PI-Dienste für die Werkzeugverwaltung über den FB 4, FC 7 und FC 8 (siehe auch die entsprechenden Kapitel in dieser Dokumentation) zur Verfügung.

**Deklaration der Funktion**

**AWL-Darstellung**

```
FUNCTION FC 22 : VOID
//NAME :          TM_DIR
VAR_INPUT
    MagNo :          INT;
    ReqPos :          INT;
    ActPos :          INT;
    Offset :          BYTE;
    Start :           BOOL;
END_VAR
VAR_OUTPUT
    Cw :              BOOL;
    Ccw :              BOOL;
    InPos :            BOOL;
    Diff :              INT;
    Error :             BOOL;
END_VAR
BEGIN
END_FUNCTION
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "TM\_DIR":

Signal	Art	Typ	Wertebereich	Bemerkung
MagNo	E	INT	1 ...	Magazinnummer
ReqPos	E	INT	1 ...	Sollplatz
ActPos	E	INT	1 ...	Istplatz
Offset	E	BYTE	0 ...	Offset für Sonderpositionierung
Start	E	BOOL		Start der Berechnung
Cw	A	BOOL		1 = Magazin im Uhrzeigersinn bewegen
Ccw	A	BOOL		1 = Magazin gegen den Uhrzeigersinn bewegen
InPos	A	BOOL		1 = Position erreicht
Diff	A	INT	0 ...	Differenzweg (kürzester Weg)
Error	A	BOOL		1 = Fehler

### Aufrufbeispiel

```
CALL FC 22(
    MagNo := 2,           //Magazin Nummer
    ReqPos := mw 20,     //Zielposition
    ActPos := mw 22,     //Aktuelle Position
    Offset := b#16#0,    //Offset für Sonderpositionierung
    Start := m 30.4,     //Start-Anstoß
                        //Rückgabeparameter
    Cw := m 30.0,        //Magazin im Uhrzeigersinn
                        //bewegen
    Ccw := m 30.1,      //Magazin gegen Uhrzeigersinn
                        //bewegen
    InPos := m 30.2,    //Magazin in Position
    Diff := mw 32,      //Differenzweg
    Error := m 30.3     //Fehler aufgetreten
);
```

## 12.15.26 FC 24: MCP\_IFM2 Übertragung der MCP-Signale an die Nahtstelle

### Funktion

Mit dem FC MCP\_IFM2 (M-Variante schmale Maschinensteuertafel z. B. MCP 310) werden von der Maschinensteuertafel (MCP) an die entsprechenden Signale der NCK-/PLC-Nahtstelle übertragen:

- Betriebsarten
- Achsanwahlen
- WKS/MKS Umschaltung
- Verfahrtafeln
- Overrides bzw. Overridenachbildung

Im Grundprogramm (FC 2) werden weiterhin die Handradanwahlen, Betriebsarten und weitere Bediensignale von der Bedientafelfront (BT) bzw. vom HMI an die NCK-/PLC-Nahtstelle so übertragen, dass bei den Betriebsarten eine wahlweise Anwahl von der MCP oder von der BT her möglich ist.

Die Übertragung der HMI-Signale an die Nahtstelle kann durch Setzen des Parameters im FB 1 (DB 7) "MMCToIF" auf den Wert FALSE abgeschaltet werden. Die Ein- und Ausschaltung von "MMCToIF" kann auch im zyklischen Programm durch Setzen oder Rücksetzen (z. B. R gp\_par.MMCToIF) erfolgen.

Für **Vorschub-Override**, **Achsfahr-** und **INC-Tasten** gelten abhängig von der aktiven Betriebsart bzw. vom angewählten Koordinatensystem folgende Festlegungen:

- **Vorschub-Override:**
  - Der Vorschub-Override wird auf die Nahtstelle des angewählten Kanals und auf die Nahtstelle der Achsen transferiert.
  - Die Vorschub-Override-Signale werden zusätzlich zum Nahtstellenbyte "Eilgangkorrektur" (DBB 5) an den NC-Kanal übergeben, wenn das HMI-Signal "Vorschubkorrektur für Eilgang wirksam" gesetzt ist (Ausnahme: Schalterstellung "Null"). Weiterhin wird mit diesem HMI-Signal auch "Eilgangkorrektur wirksam" gesetzt.
- **Maschinenfunktion INC- und Achsfahrtasten:**
  - Bei angewähltem MKS werden die Signale auf die Nahtstelle der angewählten Maschinenachsetransferiert.
  - Bei angewähltem WKS werden die Signale auf die Geo-Achs-Nahtstelle des parametrisierten Kanals transferiert.
  - Bei Umschaltung zwischen MKS und WKS erfolgt generell eine Abwahl der bis dahin angewählten Achse.

Die **Handrad-Anwahlsignale von HMI** werden ausdecodiert und in der zugehörigen Maschinenachs-Nahtstelle oder in der Geometrieachs-Nahtstelle des jeweiligen Handrades aktiviert (nur wenn im FB 1 der Parameter "HWheelMMC": = TRUE ist).

Die Ansteuerung der zugehörigen LED's der Maschinensteuertafel wird aus der Rückmeldung entsprechender Anwahlen abgeleitet.

Vorschub- und Spindel-Start/Stop werden nicht an die Nahtstelle übertragen, sondern als Signal "FeedHold" bzw. "SpindleHold" selbsthaltend ausgegeben. Der Anwender kann diese Signale mit weiteren Signalen verknüpfen, die zu Vorschub- oder Spindel-Halt führen sollen (dies kann z. B. über die entsprechenden Eingangssignale des FC 10: AL\_MSG erfolgen). Zusätzlich werden die zugehörigen LEDs mit angesteuert.

Die **Spindelrichtung** (+, -) wird auch nicht direkt geschaltet, sondern als Ausgangsparameter "SpindleDir" zur Verfügung gestellt. Hierüber kann z. B. der FC 18 parametrierbar werden. Zusätzlich wird eine Freigabe der Spindel geschaltet über den Parameter "SpindleHold". Eine Möglichkeit, die Spindel direkt zu bewegen, besteht darin, die Spindel über entsprechende Achsanwahl vorzuwählen und über die (Achs-) Richtungstasten zu verfahren.

Bei Ausfall der Maschinensteuertafel werden die Signale, die von dieser kommen, mit Null vorbesetzt; ebenso die Ausgangssignale "FeedHold" und "SpindleHold".

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 24 bzw. auch FC 19, FC 25, FC 26 zulässig. Hierbei steuert der erste Aufruf im Zyklus die LED-Anzeigen an. Weiterhin werden im ersten Aufruf alle Aktionen des parametrierbaren Bausteins durchgeführt. Bei den weiteren Aufrufen findet nur noch eine reduzierte Bearbeitung von Kanal und BAG-Schnittstelle statt. Die Geometrieachsen werden nur im ersten Aufruf des Bausteins im Zyklus mit Richtungsvorgaben versorgt.

Die Einzelsatz-An-/Abwahl wird nur durch den ersten Aufruf im Zyklus beeinflusst.

Die zweite Maschinensteuertafel kann bearbeitet werden, wenn der Parameter "BAGNo" um B#16#10 erhöht ist. Bei der Parametrierung ist die BAG Nummer in dem unteren Nibble (untere 4 Bits) enthalten.

"BAGNo" = 0 oder B#16#10 bedeutet keine Bearbeitung der BAG Signale.

ChanNo = 0 bedeutet keine Bearbeitung der Kanal-Signale.

Die INC Anwahlen werden in die BAG-Schnittstelle übertragen. Die Aktivierung für diese Vorgabe findet über den DB10.DBX57.0 (INC-Eingänge im BAG-Bereich aktiv) durch diesen Baustein einmalig nach Hochlauf statt.

Weiterhin können 2 Maschinensteuertafeln parallel durch diesen Baustein bearbeitet werden. Hierbei ist der Aufruf des Bausteins für die 2. Maschinensteuertafel im OB 1-Zyklus zeitlich hinter den Aufruf für die 1. MCP zu setzen. Eine Unterstützung von 2 MCP ist in den Maschinensteuertafel-Bausteinen bis zu bestimmten Grenzen vorhanden (nicht unterstützt werden vom Standard gegenseitige Verriegelungen der Achsanwahlen bei gleich zugeordneten Achsen bei 2 MCP).

## Flexible Achskonfiguration

Es ist eine Flexibilität bezüglich der Zuordnung Achsanwahlen bzw. Richtungstasten von Maschinen-Achsnummern verfügbar.

Der Einsatz von 2 Maschinensteuertafeln, die gleichzeitig betrieben werden sollen, wird durch die MCP Bausteine insbesondere für den Anwendungsfall 2 Kanäle, 2 BAGs besser unterstützt. Bei den Achs-Tabellen der jeweiligen MCP ist zu beachten, dass die Achsnummern auch in der parametrisierten BAG Nummer des MCP-Bausteins angegeben sind.

Für diese Flexibilität existieren Tabellen für Achsnummern im DB 10.

Für die 1. Maschinensteuertafel (MCP) beginnt die Tabelle ab dem Byte 8 (symbolischer Name: MCP1AxisTbl[1..22]) und für die 2. Maschinensteuertafel (MCP) ab dem Byte 32 (symbolischer Name: MCP2AxisTbl[1..22]). Hier sind die Maschinen-Achsnummern byteweise einzutragen. Der Eintrag 0 ist in der Achstabelle zulässig. Eine Prüfung auf eine unzulässige Achsnummer findet nicht statt und kann bei Falscheintrag zum PLC-Stop führen.

Für den **FC 24** kann auch eine Begrenzung der **maximal möglichen Achsanwahlen** erfolgen. Diese Obergrenze wird für die 1. Maschinensteuertafel im DB10.DBW30 (symbolischer Name: MCP1MaxAxis) bzw. für die 2. Maschinensteuertafel im DB10.DBW54 (symbolischer Name: MCP2MaxAxis) für die jeweilige MCP eingestellt werden.

Der voreingestellte Wert ist 0, damit wirkt die konfigurierte maximale Achszahl. Die Achsnummern und die Begrenzung können auch dynamisch angepasst werden. Danach muss eine erneute Achsanwahl bei FC 24 erfolgen. Während des Fahrens von Achsen über die jeweiligen Richtungstasten darf keine Umschaltung der Achsnummern erfolgen. Voreingestellt ist der Kompatibilitätsmodus mit den Achsnummern **1 bis 6** für beide MCP und die Begrenzung auf die konfigurierte Achszahl.

## Deklaration der Funktion

```

FUNCTION FC 24: VOID
//NAME :                MCP_IFM2

VAR_INPUT
    BAGNo :              BYTE;
    ChanNo :              BYTE;
    SpindleIFNo :        BYTE;
END_VAR

VAR_OUTPUT
    FeedHold :            BOOL;
    SpindleHold :         BOOL;
    SindleDir :           BOOL;
END_VAR

BEGIN
END_FUNCTION

```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "MCP\_IFM2":

Signal	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	0 - b#16#0A und b#16#10 - b#16#1A	BAG-Nr., in die die Betriebsartensignale übertragen werden. BAGNo >= b#16#10 bedeutet Zugriff auf die 2. Maschinensteuertafel.
ChanNo	E	BYTE	0 - B#16#0A	Kanal-Nr. für die Kanalsignale.
SpindleIFNo	E	BYTE	0 - 31 (B#16#1F)	Nummer der Achsschnittstelle, die als Spindel deklariert ist.
FeedHold	A	BOOL		Vorschub Halt von MCP, selbsthaltend
SpindleHold	A	BOOL		Spindel Halt von MCP, selbsthaltend
SpindleDir	A	BOOL		Spindel-Drehrichtung
				0: entspricht + (links)
				1: entspricht - (rechts)

### Aufrufbeispiel

```
CALL FC 24 (
//schmale Maschinensteuertafel-M-Variante
//Signale an Nahtstelle
BAGNo := B#16#1, //BAG Nr. 1
ChanNo := B#16#1, //Kanal Nr. 1
SpindleIFNo := B#16#4, //Spindel Schnittstellen- Nummer = 4
FeedHold := m22.0, //Vorschub Halt Signal selbsthaltend
SpindleHold := db2.dbx151.0, //Spindel Halt selbsthaltend in Meldungs-DB
SpindleDir := m22.1); //Rückgabe Spindelrichtung
```

Mit dieser Parametrierung werden die Signale an die 1. BAG, den 1. Kanal und an alle Achsen übertragen. Zusätzlich wird der Spindel-Override in die 4. Achs-/Spindel-Nahtstelle übertragen. Das Vorschub-Halt-Signal wird an Merker 22.0 und das Spindel-Halt-Signal an den Datenbaustein DB 2, Datenbit 151.0 übergeben. Mit der Rückmeldung der Spindelrichtung über den Parameter "SpindleDir" kann eine Richtungsvorgabe für einen zusätzlich aufzurufenden FC 18 erfolgen.



## 12.15.27 FC 25: MCP\_IFT Übertragung der MCP-/BT-Signale an die Nahtstelle

### Funktion

Mit dem FC MCP\_IFT (T-Variante) werden von der Maschinensteuertafel einer Breite von 19 Zoll z. B. MCP 483 an die entsprechenden Signale der NCK/PLC-Nahtstelle übertragen:

- Betriebsarten
- Richtungstasten von vier Achsen
- WKS/MKS-Umschaltung
- Overrides
- Schlüsselschalter

Im Grundprogramm (FC 2) werden weiterhin die Handradanwahlen, Betriebsarten und weitere Bediensignale von der Bedientafelfront (BT) bzw. vom HMI an die NCK/PLC-Nahtstelle so übertragen, dass bei den Betriebsarten eine wahlweise Anwahl von der MCP oder von der BT her möglich ist.

Die Übertragung der HMI-Signale an die Nahtstelle kann durch Setzen des Parameters im FB 1 (DB 7) "MMCToIF" durch den Wert FALSE abgeschaltet werden.

Für **Vorschub-Override**, **Achsfahr**- und **INC-Tasten** gelten abhängig von der aktiven Betriebsart bzw. vom angewählten Koordinatensystem folgende Festlegungen:

- **Vorschub-Override:**
  - Der Vorschub-Override wird auf die Nahtstelle des angewählten Kanals und auf die Nahtstelle der Achsen transferiert.
  - Die Vorschub-Override-Signale werden zusätzlich zum Nahtstellenbyte "Eilgangkorrektur" (DBB 5) an den NC-Kanal übergeben, wenn das HMI-Signal "Vorschubkorrektur für Eilgang wirksam" gesetzt ist (Ausnahme: Schalterstellung "Null"). Weiterhin wird mit diesem HMI Signal auch "Eilgangkorrektur wirksam" gesetzt.
- **Maschinenfunktion INC- und Achsfahrtasten:**
  - Bei angewähltem MKS werden die Signale auf die Nahtstelle der angewählten Maschinenachse transferiert.
  - Bei angewähltem WKS werden die Signale auf die Geo-Achs-Nahtstelle des parametrisierten Kanals transferiert.

Die **Handrad-Anwahlsignale von HMI** werden ausdecodiert und in der zugehörigen Maschinenachs-Nahtstelle oder in der Geometrieachs-Nahtstelle des jeweiligen Handrades aktiviert (nur wenn im FB 1 der Parameter "HWheelMMC": = TRUE ist).

Die Ansteuerung der zugehörigen LEDs der Maschinensteuertafel wird aus der Rückmeldung entsprechender Anwahlen abgeleitet.

Vorschub- und Spindel-Start/Stopp werden nicht an die Nahtstelle übertragen, sondern als Signal "FeedHold" bzw. "SpindleHold" selbsthaltend ausgehen. Der Anwender kann diese Signale mit weiteren Signalen verknüpfen, die zu Vorschub- oder Spindel-Halt führen sollen (dies kann z. B. über die entsprechenden Eingangssignale des FC 10: AL\_MSG erfolgen). Zusätzlich werden die zugehörigen LEDs mit angesteuert.

Bei Ausfall der Maschinensteuertafel werden die Signale, die von dieser kommen, mit Null vorbesetzt; ebenso die Ausgangssignale "FeedHold" und "SpindleHold".

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 25 bzw. auch FC 19, FC 24, FC 26 zulässig. Hierbei steuert der erste Aufruf im Zyklus die LED-Anzeigen an. Weiterhin werden im ersten Aufruf alle Aktionen des parametrisierten Bausteins durchgeführt. Bei den weiteren Aufrufen findet nur noch eine reduzierte Bearbeitung von Kanal und BAG-Schnittstelle statt. Die Geometrieachsen werden nur im ersten Aufruf des Bausteins im Zyklus mit Richtungsvorgaben versorgt.

Die Einzelsatz-An-/Abwahl wird nur durch den ersten Zyklus beeinflusst.

Die zweite Maschinensteuertafel kann bearbeitet werden, wenn der Parameter "BAGNo" um B#16#10 erhöht ist. Bei der Parametrierung ist die BAG Nummer in dem unteren Nibble (untere 4 Bits) enthalten.

"BAGNo" = 0 oder B#16#10 bedeutet keine Bearbeitung der BAG Signale.

ChanNo = 0 bedeutet keine Bearbeitung der Kanal-Signale.

## Flexible Achskonfiguration

Es ist eine Flexibilität bezüglich der Zuordnung Achsanwahlen bzw. Richtungstasten von Maschinenachsnummern verfügbar.

Der Einsatz von zwei gleichzeitig betriebenen Maschinensteuertafeln wird durch die MCP Bausteine, insbesondere für den Anwendungsfall "2 Kanäle, 2 BAG" unterstützt. Hierbei ist der Aufruf des Bausteins für die zweite Maschinensteuertafel im OB 1-Zyklus zeitlich hinter den Aufruf für die 1. MCP zu setzen. Bei den Achstabellen der jeweiligen MCP ist zu beachten, dass die Achsnummern auch in der parametrisierten BAG Nummer des MCP-Bausteins angegeben sind.

Für diese Flexibilität existieren Tabellen für Achsnummern im DB 10.

Für die 1. Maschinensteuertafel (MCP) beginnt die Tabelle ab Byte 8 (symbolischer Name: MCP1AxisTbl[1..22]) und für die 2. Maschinensteuertafel (MCP) ab Byte 32 (symbolischer Name: MCP2AxisTbl[1..22]). Hier sind die Maschinenachsnummern byteweise einzutragen. Der Eintrag 0 ist in der Achstabelle zulässig. Eine Prüfung auf eine unzulässige Achsnummer findet nicht statt und kann bei Falscheintrag zum PLC-Stop führen.

Die Begrenzung der **möglichen Achsanwahlen bei FC 25** erfolgt über die 0-Werte in der Achstabelle. Die Achsnummern können auch dynamisch angepasst werden. Während des manuellen Verfahrens von Achsen über Richtungstasten, darf keine Umschaltung der Achsnummern erfolgen. Voreingestellt ist der Kompatibilitätsmodus mit den Achsnummern **1 bis 4** für beide MCP und die Begrenzung auf die konfigurierte Achszahl.

---

### Hinweis

Ergänzende Informationen siehe "FC 19: MCP\_IFM Übertragung der MCP-Signale an die Nahtstelle (Seite 1131)".

---

## Deklaration der Funktion

```

FUNCTION FC 25: VOID
//NAME :           MCP_IFT

VAR_INPUT
    BAGNo :         BYTE;
    ChanNo :        BYTE;
    SpindleIFNo :   BYTE;
END_VAR

VAR_OUTPUT
    FeedHold :      BOOL;
    SpindleHold :   BOOL;
END_VAR

BEGIN
END_FUNCTION
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "MCP\_IFT":

Signal	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	B#16#00 - B#16#0A	1. MCP: BAG-Schnittstelle in welche die Betriebsartensignale übertragen werden
			B#16#10 - B#16#1A	2. MCP: BAG-Schnittstelle in welche die Betriebsartensignale übertragen werden
ChanNo	E	BYTE	B#16#00 - B#16#0A	Kanal-Nr. für die Kanalsignale.
SpindleIFNo	E	BYTE	B#16#00 - B#16#1F	Achsschnittstelle in welche die Spindelaten übertragen werden
FeedHold	A	BOOL		Vorschub Halt von MCP, selbsthaltend
SpindleHold	A	BOOL		Spindel Halt von MCP, selbsthaltend

### Aufrufbeispiel

```
CALL FC 25(                                     //Maschinensteuertafel-T-Variante
                                     //Signale an Nahtstelle
    BAGNo :=          B#16#1,                 //BAG Nr. 1
    ChanNo :=         B#16#1,                 //Kanal Nr. 1
    SpindleIFNo :=   B#16#4,                 //Spindel Schnittstellen-Nummer = 4
    FeedHold :=      m22.0,                  //Vorschub Halt Signal selbsthaltend
    SpindleHold :=   db2.dbx151.0);          //Spindel Halt selbsthaltend in Meldungs-DB
```

Mit dieser Parametrierung werden die Signale an die 1. BAG, den 1. Kanal und an alle Achsen übertragen. Zusätzlich wird der Spindel-Override in die 4. Achs-/Spindel-Nahtstelle übertragen. Das Vorschub-Halt-Signal wird an Merker 22.0 und das Spindel-Halt-Signal an den Datenbaustein DB 2, Datenbit 151.0 übergeben.

### 12.15.28 FC 26: HPU\_MCP Übertragung der HT8-Signale an die Nahtstelle

#### Funktionsdeklaration

```
FUNCTION FC 26: VOID
//NAME :          HPU_MCP

VAR_INPUT
    BAGNo :      BYTE;
    ChanNo :     BYTE;
END_VAR

BEGIN
END_FUNCTION
```

## Parameter

Parameter	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	1. MCP: B#16#00 - B#16#0A	Oberes Nibble: Nummer der MCP deren Signale übertragen werden sollen. 0 = 1. MCP, 1 = 2. MCP
			2. MCP: B#16#10 - B#16#1A	Unteres Nibble: Nummer der BAG in welche die BAG-spez. Nahtstellensignale übertragen werden sollen. Bei BAG-Nummer 0 erfolgt keine Bearbeitung der BAG-spez. Signale.
ChanNo	E	BYTE	B#16#00 - B#16#0A	Nummer des Kanals in welchen die Kanal-spez. Nahtstellensignale übertragen werden sollen. Bei Kanal-Nummer 0 erfolgt keine Bearbeitung der Kanal-spez. Signale.
Art: E = Eingangsparameter, A = Ausgangsparameter				

## Aufrufbeispiele

Aufruf des FC 26 für die erste MCP, die erste BAG und den ersten Kanal der NC.

```
CALL FC 26 (                                     //Maschinensteuertafel des HT8
            BAGNo :=      B#16#01,             //1.MCP, 1.BAG
            ChanNo :=     B#16#01);           //1.Kanal
```

Aufruf des FC 26 für die zweite MCP, die zweite BAG und den dritten Kanal der NC.

```
CALL FC 26 (                                     //Maschinensteuertafel des HT8
            BAGNo :=      B#16#12,             //2.MCP, 2.BAG
            ChanNo :=     B#16#03 );          //3.Kanal
```

## Allgemeine Funktionsbeschreibung

Die Funktion FC 26 "HPU\_MCP (Maschinensteuertafel-Signale des Bedienhandgerätes HT8" überträgt die HT8-spezifischen Signale der folgenden Funktionen zwischen dem im Funktionsbaustein FB 1 parametrisierten HT 8-Ein/Ausgangsdatenbereich (Parameter: MCPxIn und MCPxOut) und der NC/PLC-Nahtstelle:

- Betriebsarten
- Maschinenfunktion INC
- Koordinatensystem WKS oder MKS
- Axiale Verfahrtasten
- Achsanwahl
- Vorschub-Override
- Eilgang-Override
- Schlüsselschalter-Informationen

---

### Hinweis

#### Betriebsartenumschaltung durch HT 8 und/oder HMI

Die Funktion FC 2 "GP\_HP Grundprogramm, zyklischer Teil" überträgt die Signale der Betriebsarten-Umschaltung prinzipiell so, dass eine wahlweise Anwahl vom MCP des HT 8 und des HMI möglich ist. Die Übertragung der HMI-Signale an die NC/PLC-Nahtstelle kann im Funktionsbaustein FB 1 mit Parameter "MMCToIF" = FALSE aber auch ausgeschaltet werden.

#### Aktive Achsen

Über das HT 8 können maximal 6 Achsen gleichzeitig angesprochen werden. Die Auswahl der Achsen ist durch den Anwender/Maschinenhersteller im PLC-Anwenderprogramm zu realisieren.

---

## Flexible Achskonfiguration

Die Funktion FC 26 ermöglicht eine flexible Zuordnung der Maschinenachsen zu den Verfahrtasten bzw. zur Achsanwahl. Im DB 10 stehen dazu 2 Tabellen zur Verfügung:

- Maschinenachstabelle 1. MCP: DB10.DBB8 bis DBB13 (Tabelle der Maschinenachsnr.)  
Symbolischer Name: MCP1AxisTbl[1..22]
- Maschinenachstabelle 2. MCP: DB10.DBB32 bis DBB37 (Tabelle der Maschinenachsnr.)  
Symbolischer Name: MCP2AxisTbl[1..22]

In die Tabellen sind die Achsnummern n (mit n = 1, 2, ...) der aktiven Maschinenachsen byteweise einzutragen. Bei nicht verwendeten Tabellenplätzen muss der Wert 0 eingetragen werden.

Die Tabellenlänge kann dem FC 26 vorgegeben werden:

- 1. MCP: DB10.DBB30 (Obergrenze der Maschinenachstabelle)
- 2. MCP: DB10.DBB54 (Obergrenze der Maschinenachstabelle)

Ein Wert von z. B. 4 bedeutet, dass vom FC 26 nur die ersten 4 Tabelleneinträge bzw. Maschinenachsen berücksichtigt werden. Der Maximalwert für den FC 26 ist 6. Bei Wert 0 oder Werten größer 6 wird implizit der Maximalwert genommen.

---

#### Hinweis

Folgende Randbedingungen sind zu beachten:

- Eine Überprüfung auf zulässige Maschinenachsennummern findet nicht statt. Ungültige Maschinenachsennummer können zum Stopp der PLC führen.
  - Die Maschinenachsennummern können dynamisch geändert werden. Das Beschreiben der Tabelle darf nicht durchgeführt werden, wenn eine Maschinenachse aktuell über eine Verfahraste verfahren wird.
- 

### Übertragung der Verfahrastensignale in Abhängigkeit vom aktiven Koordinatensystem

Die Verfahrastensignale für 6 Achsen liegen im HT 8-Eingangsdatenbereich unter:

- EB n + 2, Bit 0 - Bit 5 (positive Verfahrachse)
- EB n + 3, Bit 0 - Bit 5 (negative Verfahrachse)

Das Umschalten des Koordinatensystems erfolgt über das Eingangssignal:

- EB n + 0, Bit 0 (MKS/WKS)

Das Eingangssignal wird im FC 26 mittels Flankenmerker ausgewertet. Das aktive Koordinatensystem wird in folgendem Ausgangssignal angezeigt:

- AB n + 0, Bit 0 (MKS/WKS) mit 0 = MKS, 1 = WKS

Bei aktivem MKS werden die Verfahrastensignale der Achsen 1 - 6 in die achsspezifischen Nahtstellen (DB31, ... .DBX4.6 und DBX4.7 (Verfahrastensignale +/-)) der in den Maschinenachstabellen (DB10.DBB8 bis DBB13 bzw. DBB32 bis DBB37) angegebenen Achsen übertragen.

Bei aktivem WKS wird davon ausgegangen, dass die Achsen 1 - 3 der Maschinenachstabelle Geometrieachsen sind. Daher werden die Verfahrastensignale:

- der Achsen **1 - 3** (EB n + 2 / 3, Bit 0 - Bit 2) in die Nahtstelle der Geometrieachsen in DB21, ... .DBB 12 + (n \* 4), mit n = 0, 1, 2), Bit 6 und Bit 7 (Verfahrastensignale +/-) des über Parameter "ChanNo" angegebenen Kanals übertragen.  
Die Zuordnung der Verfahrastensignale der Achsen 1, 2 und 3 auf die Geometrieachsen 1, 2 und 3 des Kanals ist fest und kann nicht verändert werden.
- der Achsen **4 - 6** (EB n + 2 / 3, Bit 3 - Bit 5) in die achsspezifischen Nahtstellen (DB31, ... .DBX4.6 und DBX4.7 (Verfahrastensignale +/-)) der in der Maschinenachstabelle eingetragenen Achsen 4 - 6 (DB10.DBB11 bis DBB13 bzw. DBB35 bis DBB37) übertragen.

### Kein Verfahren von Maschinenachsen im WKS

Bei aktivem WKS (AB n + 0, Bit 0 = 1) kann das Verfahren von Maschinenachsen verriegelt werden. Dazu sind im PLC-Anwenderprogramm folgende Ausgangssignale zu setzen:

- AB n + 3, Bit 7 = 1 (Bei WKS: keine Maschinenachsen)

Anforderung an den FC 26, keine Verfahrtastensignale für Maschinenachsen zu übertragen. Die Verfahrtastensignale für die Achsen 1 - 3 der Maschinenachstabelle werden auf die Geometrieachsen 1 - 3 des angegebenen Kanals übertragen. Die Verfahrtastensignale für die Achsen 4 - 6 der Maschinenachstabelle werden nicht übertragen.

- AB n + 2, Bit 6 (Achsen 7 - n angewählt)

Anforderung an den FC 26 keine Verfahrtastensignale zu übertragen, da die Achsen 1 - 6 der Maschinenachstabelle umgeschaltet wurden. Die Achsen 1 - 3 somit keine Geometrieachsen sondern ebenfalls Maschinenachsen sind.

### Vorschub-Override

Der Wert des HT8-Korrekturschalters wird als Vorschub-Override in die kanalspezifische Nahtstelle DB 21, ... .DBB4 (Vorschubkorrektur) des programmierten Kanals (Parameter: "ChanNo") und in die achsspezifische Nahtstellen DB31, ... .DBB0 (Vorschubkorrektur) der in der Tabelle DB10.DBB8 bis DBB13 (Maschinenachsnummer) programmierten Achsen übertragen.

### Eilgang-Override

Ist für den programmierten Kanals (Parameter: "ChanNo") das Signal DB21, ... .DBX25.3 = 1 (Vorschubkorrektur für Eilgang) gesetzt, wird der Wert des HT8-Korrekturschalters als Eilgang-Override in dessen kanalspez. Nahtstelle in DB 21, ... .DBB5 (Eilgangkorrektur) übertragen und zusätzlich das Signal DB21, ... .DBX6.6 = 1 (Eilgangkorrektur wirksam) gesetzt.



## Maschinenfunktion INC

Die HT8-Signale der Maschinenfunktion INC werden abhängig vom aktiven Koordinatensystem MKS oder WKS unterschiedlich übertragen:

- Aktives Koordinatensystem: MKS

Die angewählte Maschinenfunktion INC wird für alle 6 Achsen in die achsspez. Nahtstellen in DB31, ... .DBX5.0 bis DBX5.5 (Maschinenfunktion) der in der Tabelle in DB10.DBB8 bis DBB13 (Maschinenachsnummer) programmierten Achsen übertragen.

- Aktives Koordinatensystem: WKS

Für die Achsen 1 bis 3 werden die Signale der Maschinenfunktion INC in die kanalspez. Nahtstelle in DB21, ... .DBX13.0 bis DBX13.5 (Maschinenfunktion) des programmierten Kanals (Parameter: "ChanNo") übertragen.

Für die Achsen 4 bis 6 werden die Signale der Maschinenfunktion INC in die achsspez. Nahtstellen DB31, ... .DBX5.0 bis DBX5.5 (Maschinenfunktion) der in der Tabelle in DB10.DBB11 bis DBB13 (Maschinenachsnummer) programmierten Achsen übertragen.

Die Anwahlsignale der INC-Maschinenfunktionen werden in die BAG-spezifische Nahtstelle DB11 DBB 2 + (n \* 20), Bit 0 bis Bit 5 (mit n = 0, 1, 2, ...) übertragen. Der FC 26 informiert den NCK über die Aktivierung der BAG-Nahtstelle für die INC-Maschinenfunktionen einmalig nach dem Hochlauf über DB10.DBX57.0 (INC-Eingänge im BAG-Bereich aktiv).

## Handradanwahl

Die Handradanwahlsignale werden vom HMI ausgewertet und in die entsprechenden NC/PLC-Nahtstellensignale der Maschinen- oder Geometrieachsen übertragen:

- Geometrieachsen: DB21, ... DBB 12 + (n \* 4), Bit 0 bis Bit 2 (mit n = 0, 1, 2)
- Maschinenachsen: DB31, ... .DBX4.0 bis DBX4.2

Voraussetzung: FB 1-Parameter: "HWheelIMMC" = TRUE

## Mehrfachaufruf in einem PLC-Zyklus

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 26 zulässig. Beim ersten Aufruf im PLC-Zyklus werden:

- Alle Aktionen des parametrisierten Bausteins durchgeführt
- Die LED-Signale im Ausgangsbereich geschrieben
- Bei angewähltem WKS, die Verfahrtastensignale der Geometrieachsen geschrieben
- Die Signale für Einzelsatz-An- und Abwahl bearbeitet

Bei weiteren Aufrufen des FC 26 erfolgt nur noch eine reduzierte Bearbeitung der Kanal- und BAG-Schnittstelle.

### Bearbeitung von zwei MCP

Wird die Funktion FC 26 im zyklischen Ablauf des PLC-Programms (Organisationsbaustein OB 1) zweimal für zwei MCP aufgerufen, muss der Aufruf für die zweite MCP zeitlich nach dem Aufruf für die erste MCP erfolgen.

---

#### Hinweis

Kann eine Achse von zwei MCP aus verfahren werden, liegt die Realisierung einer gegenseitigen Verriegelung in der Verantwortung des Anwenders (Maschinenherstellers).

---

### Ausfall der MCP des HT8

Bei Ausfall der MCP des HT8 werden alle Eingangssignale auf den Wert 0 gesetzt.

#### 12.15.28.1 Übersicht der NC/PLC-Nahtstellensignale von HT 8

#### Betriebsarten und Maschinenfunktionen

Quelle: MCP	Ziel: Programmierte BAG (Parameter BAGNo) Darstellung für BAG 1
AUTOMATIC	DB11.DBX0.0
MDA	DB11.DBX0.1
JOG	DB11.DBX0.2
REPOS	DB11.DBX1.1
REF	DB11.DBX1.2
TEACH IN	DB11.DBX1.0
INC 1 ... 10 000, INC Var.	DB11.DBX2.0 - DBX 2.5

## Verfahrtasten und Eilgang-Override

Quelle: MCP	Ziel: Geometrie-Achsen des prog. Kanals (Parameter: ChanNo)
Verfahrtaste +	DB21, ... .DBX12.7
Verfahrtaste -	DB21, ... .DBX12.6
Eilgang-Override	DB21, ... .DBX12.5
Verfahrtaste +	DB21, ... .DBX16.7
Verfahrtaste -	DB21, ... .DBX16.6
Eilgang-Override	DB21, ... .DBX16.5
Verfahrtaste +	DB21, ... .DBX20.7
Verfahrtaste -	DB21, ... .DBX20.6
Eilgang-Override	DB21, ... .DBX20.5

Quelle: MCP	Ziel: Prog. Achsen entsprechend der Tabelle in DB 10, DBB 8 - 13 (1. MCP) bzw. DBB 32 - 37 (2. MCP)
Verfahrtaste +	DB31, ... .DBX4.7
Verfahrtaste -	DB31, ... .DBX4.6
Eilgang-Override	DB31, ... .DBX4.5

## Override

Quelle: MCP	Ziel: Programmierter Kanal (Parameter: ChanNo)
Vorschub-Override	DB21, ... .DBB4

Quelle: MCP	Ziel: Prog. Achsen entsprechend der Tabelle in DB 10, DBB 8 - 13 (1. MCP) bzw. DBB 32 - 37 (2. MCP)
Vorschub-Override	DB31, ... .DBB0

## Kanalsignale

Quelle: MCP	Ziel: Programmierter Kanal (Parameter: ChanNo)
NC-Start	DB21, ... .DBX7.1
NC-Stop	DB21, ... .DBX7.3
RESET	DB21, ... .DBX7.7
Einzelstart	DB21, ... .DBX0.4

## 12.15.28.2 Übersicht der NC/PLC-Nahtstellensignale an HT 8

### Betriebsarten und Maschinenfunktionen

Ziel: MCP	Quelle: Nahtstellen-DB (Parameter BAGNo) Darstellung für BAG 1
AUTOMATIC	DB11.DBX6.0
MDA	DB11.DBX6.1
JOG	DB11.DBX6.2
REPOS	DB11.DBX7.1
REF	DB11.DBX7.2
TEACH IN	DB11.DBX7.0

## 12.15.29 FC 19, FC 24, FC 25, FC 26 Beschreibung Quellcode

### Aufgabe

Maschinensteuertafel an Anwendernahtstelle (FC 19 M-Variante, FC 24 schmale Variante, FC 25 T-Variante, FC 26 HT8-Variante)

### Zugehörige Bausteine

- DB 7, Anzahl BAG, Kanäle, Achsen
- DB 7, Pointer der Maschinensteuertafeln,
- DB 8, Ablage für den nächsten Zyklus

### Verwendete Ressourcen

Keine.

### Allgemeines

Die Bausteine FC 19 (M-Version), FC 24 (schmale Version), FC 25 (T-Version) und FC 26 (HT8-Version) übertragen die Signale der Maschinensteuertafel von und zur Anwendernahtstelle. In der Eingangsparametrierung wird durch "BAGNo" die BAG ausgewählt, die von dem Baustein bearbeitet wird. Über den Parameter "BAGNo" wird auch die Nummer der Maschinensteuertafel selektiert (Bit 4). Bei dem Parameter "ChanNo" wird der Kanal ausgewählt, der bearbeitet wird.

Nicht FC 26:

Der Parameter "SpindleIFNo" definiert die Achs-Nahtstelle der Spindel. Der Spindel-Override wird auf diese Spindelnahstelle transferiert. Die Eingangsparameter werden bezüglich Falschparametrierung überprüft.

Nicht FC 26:

Die Ausgangsparameter "FeedHold" und "SpindleHold" werden aus den 4 Tasten Vorschub/Spindel-Sperre, Vorschub/Spindel-Freigabe gebildet und mit "logisch 1" bei Sperre zurückgeliefert.

Informationen für den nächsten Zyklus werden abhängig von der Maschinensteuertafel-Nummer im DB 8, Byte 0 bis 3 bzw. Byte 62 bis 65 abgespeichert. Diese Informationen sind Flankenmerker, Vorschubwert und angewählte Achsnummer. Die Bausteine werden mit Nutzdaten durch die Pointer-Parameter im DB 7 "MCP1In" und "MCP1Out" ("MCP2In" und "MCP2Out") versorgt. Die Pointer werden indirekt über einen weiteren Zeiger aus dem VAR-Teil des DB 7 adressiert um die Absolutadressierung zu vermeiden. Dieser zusätzliche Zeiger ist im FB 1 symbolisch ermittelt worden.

## Bausteinbeschreibung

Alle 4 Bausteine sind ähnlich strukturiert aufgebaut und für die einzelnen Teilaufgaben gegliedert:

Im **Netzwerk Input** werden verschiedene Parameter in lokale Variablen kopiert. Hierbei werden auch die Maschinensteuersignale (Nutzdaten Ein-/Ausgangsbereich) über die verschiedenen Zeiger im DB 7 (gp\_par) umkopiert. Mit diesen lokalen Variablen wird aus Effizienzgründen im Baustein gearbeitet. Für den Anlauf werden einige Werte initialisiert.

Im **Netzwerk Global\_IN** wird die MKS/WKS Umschaltung mit Flankenauswertung, Achsanwahlen, Richtungstasten und Eilgangüberlagerung für die weitere Bearbeitung im Baustein ermittelt. In diesem Programmteil sollten anwenderspezifische Veränderungen stattfinden, die sich hauptsächlich an den Achsanwahlen orientieren.

Im **Netzwerk NC** werden nur die Schlüsselschalter-Informationen kopiert.

Das **Netzwerk BAG** transferiert die Betriebsarten der Taster als dynamische Signale zum NCK. Die INC Rückmeldungen der NC werden temporär für die entsprechenden LEDs abgelegt. Bei BAG-Nummer 0 wird dieses Netzwerk nicht bearbeitet. Einer zu großen Nummer hat eine Meldung 401901 bzw. 402501 zur Folge und schaltet nach Stopp um.

Im **Netzwerk Channel** werden die Funktionen NC-Start, Stopp, RESET und Einzelsatz mit entsprechenden Rückmeldungen aktiviert. Die Richtungstasten der GEO-Achsen werden bei entsprechender Vorwahl versorgt, anderenfalls entsorgt. Bei Kanal-Nummer 0 wird dieses Netzwerk nicht bearbeitet. Eine zu große Nummer hat eine Meldung 401902 bzw. 402502 zur Folge und es wird nach Stopp geschaltet.

Das **Netzwerk Spindle** überträgt den Spindeloverride in die über "SpindleIFNo" parametrisierte Nahtstelle.

Das **Netzwerk Axes** überträgt den Vorschub-Override auf die angewählte Achsnahtstelle. Die Richtungstasten werden der angewählten Achse/Spindel zugeordnet. Bei einer vorher angewählten Achse wird die Richtungsinformation auf 0 gesetzt.

Im **Netzwerk Global\_OUT** werden die Ausgangsparameter aufbereitet und die LED-Signale der INC-Maschinenfunktion gebildet.

Das **Netzwerk Output** überträgt die Ausgangssignale der Maschinensteuertafel vom Abbild der VAR\_TEMP auf die logische Adresse. Weiterhin werden die Daten für den nächsten Zyklus gerettet.

## Erweiterung Achsanwahl

Im Netzwerk Global\_IN ist für eine Anwahl von mehr als 9 Achsen einzugreifen. Falls hier weitere Tasten und LED der Maschinensteuertafel verwendet werden, sind folgende Maßnahmen zu treffen:

1. Der Befehl UD DW#16#Wert (Kommentar: Clear all axis LED's for display) löscht alle definierten LEDs für Achsanwahlen. Derzeit werden mit der Bitmaske genau die 9 Achsanwahl-LEDs bearbeitet.
2. Der Befehl UW W#16# (Kommentar: " Masking all the axis selection buttons") prüft ob eine neue Richtungsanwahl erfolgte. Hier muss die Bitleiste angepasst werden.
3. Die Sprungleiste (SPL) ist mit neuen Sprungmarken zu erweitern. Die neuen Sprungmarken sind vor der Marke m009 einzufügen in abwärtszählender Reihenfolge. Bei den neuen Sprungmarken sind Anwahlinformationen, wie bei Marke m009, m008 dargestellt, zu erweitern.

---

### Hinweis

Die Bausteine werden als AWL-Quellen bei Bedarf zur Verfügung gestellt. Sie entsprechen aber nicht zwingend dem aktuellen Stand der Bausteine. Die aktuelle Implementierung in C ist in einigen Details schon weiter entwickelt. Es ist sinnvoll die gewünschten zusätzlichen Anforderungen an die Bausteine zu spezifizieren und diese Anforderung über den Vertrieb an die Projektleitung zu geben.

---

## 12.15.30 FC 1005: AG\_SEND Übergibt Daten an Ethernet-CP

### Funktion

Der FC-Baustein AG\_SEND übergibt Daten an den Ethernet-CP zur Übertragung über eine projektierte Verbindung.

Die genannten Funktionen entsprechen den Funktionen der Bibliothek "SIMATIC\_NET\_CP" der S7-300 CPU in STEP 7. Generell gilt für diese Funktion die Online-Hilfe dieser Funktionen und deshalb wird hier auf eine ausführliche Beschreibung verzichtet.

Die Funktionen AG\_SEND, AG\_RECV können über die integrierte "CP 840D sl" zum Datenaustausch mit einer weiteren Station genutzt werden.

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt die Formalparameter der Funktion AG\_SEND an.

Signal	Art	Typ	Wertebereich	Bemerkung
ACT	E	BOOL		Auftragsanstoß
ID	E	INT		Verbindungs-ID
LADDR	E	WORD		Baugruppen-Anfangsadresse (SINUMERIK-Besonderheit; siehe Beschreibung unter der Tabelle)
SEND	E	ANY		Angabe von Adresse und Länge. Die Adresse des Datenbereichs verweist alternativ auf: <ul style="list-style-type: none"> <li>• Merkerbereich</li> <li>• Datenbausteinbereich</li> </ul>
LEN	E	INT		Anzahl Bytes, die mit dem Auftrag aus dem Datenbereich gesendet werden sollen
DONE	A	BOOL		Auftrag fehlerfrei beendet
ERROR	A	BOOL		Fehleranzeige
STATUS	A	WORD		Statusanzeige

Bei Nutzung der Funktionen AG\_SEND und AG\_RECV findet über den Ethernet-Bus der CP ein Datentransport zur Gegenstelle statt. Die Gegenstelle wird in STEP 7 im Tool "NetPro" projiziert.

Die Besonderheit beim Aufruf der Funktionen liegt in der Angabe des Parameters "LADDR" an den genannten Bausteinen. Bei der SINUMERIK 840D sl muss der Parameter "LADDR" mit dem Wert W#16#8110 beschaltet sein.

Im Grundprogramm ist diese Funktion unter der FC-Nummer 1005 verfügbar (dieser FC entspricht der FC-Nummer FC 5 in der Library "SIMATIC\_NET\_CP").

Der Baustein kann auch in einer SIMATIC-CPU 3xx mit CP343-1 verwendet werden.

Unterstützt werden die Protokolle TCP und UDP. TCP ist zu bevorzugen.

---

### Hinweis

Bei der Funktion AG\_SEND muss der Parameter ACT so lange TRUE sein, bis ein Ergebnis in DONE oder ERROR gemeldet wird.

---

### 12.15.31 FC 1006: AG\_RECV Empfängt Daten von Ethernet-CP

#### Funktion

Der FC-Baustein AG\_RECV übernimmt vom Ethernet-CP die über eine projektierte Verbindung übertragenen Daten.

Die genannten Funktionen entsprechen den Funktionen der Bibliothek "SIMATIC\_NET\_CP" der S7-300 CPU in STEP 7. Generell gilt für diese Funktion die Online-Hilfe dieser Funktionen und deshalb wird hier auf eine ausführliche Beschreibung verzichtet.

Die Funktionen AG\_SEND, AG\_RECV können über die integrierte "CP 840D sl" zum Datenaustausch mit einer weiteren Station genutzt werden.

#### Erläuterung der Formalparameter

Die folgende Tabelle zeigt die Formalparameter der Funktion AG\_RECV an.

Signal	Art	Typ	Wertebereich	Bemerkung
ID	E	INT		Verbindungs-ID
LADDR	E	WORD		Baugruppen-Anfangsadresse (SINUMERIK-Besonderheit; siehe Beschreibung unter der Tabelle)
RECV	E	ANY		Angabe von Adresse und Länge. Die Adresse des Datenbereichs verweist alternativ auf: <ul style="list-style-type: none"> <li>• Merkerbereich</li> <li>• Datenbausteinbereich</li> </ul>
NDR	A	BOOL		Neue Daten übernommen
ERROR	A	BOOL		Fehleranzeige
STATUS	A	WORD		Statusanzeige
LEN	A	INT		Anzahl Bytes, die vom Ethernet-CP in den Datenbereich übernommen wurden

Bei Nutzung der Funktionen AG\_SEND und AG\_RECV findet über den Ethernet-Bus der CP ein Datentransport zur Gegenstelle statt. Die Gegenstelle wird in STEP 7 im Tool "NetPro" projektiert.

Die Besonderheit beim Aufruf der Funktionen liegt in der Angabe des Parameters "LADDR" an den genannten Bausteinen. Bei der SINUMERIK 840D sl muss der Parameter "LADDR" mit dem Wert W#16#8110 beschaltet sein.

Im Grundprogramm ist diese Funktion unter der FC-Nummer 1006 verfügbar (dieser FC entspricht der FC-Nummer FC 6 in der Library "SIMATIC\_NET\_CP").

Der Baustein kann auch in einer SIMATIC-CPU 3xx mit CP343-1 verwendet werden.

Unterstützt werden die Protokolle TCP und UDP. TCP ist zu bevorzugen.



## 12.16 Signal-/Datenbeschreibungen

### 12.16.1 Nahtstellensignale NCK/PLC, HMI/PLC, MSTT/PLC

#### Literatur

Die Nahtstellensignale NCK/PLC, HMI/PLC und MCP/PLC sind aufgelistet im Listenhandbuch:

Listen sl (Buch2)

Die darin enthaltenen Literaturkürzel (nach den Signalbezeichnungen) verweisen auf die jeweilige Funktionsbeschreibung, in der ein Signal beschrieben ist.

Die NCK-Signale, die vom Grundprogramm ausgewertet und aufbereitet zur Anwendernahtstelle übergeben werden, sind in den nachfolgenden Kapiteln aufgeführt.

### 12.16.2 Dekodierte M-Signale

Die im Teileprogramm, ASUP oder Synchronaktionen programmierten M-Funktionen werden von der NC kanalspezifisch an die PLC übertragen:

- M-Funktionen aus Kanal 1: DB 21
- M-Funktionen aus Kanal 2: DB 22
- usw.

Die Signaldauer beträgt 1 PLC-Zykluszeit.

---

#### Hinweis

Folgende spindelspezifischen M-Funktionen werden nicht ausdekodiert: M3, M4, M5 und M70.

---

Adresse in DB 21, ...	Variable	Typ	Kommentar
DBX 194.0 ... 7	M_Fkt_M0 ... M7	BOOL	M-Signale M0 ... M7
DBX 195.0 ... 7	M_Fkt_M8 ... M15	BOOL	M-Signale M8 ... M15
DBX 196.0 ... 7	M_Fkt_M16 ... M23	BOOL	M-Signale M16 ... M23
DBX 197.0 ... 7	M_Fkt_M24 ... M31	BOOL	M-Signale M24 ... M31
DBX 198.0 ... 7	M_Fkt_M32 ... M39	BOOL	M-Signale M32 ... M39
DBX 199.0 ... 7	M_Fkt_M40 ... M47	BOOL	M-Signale M40 ... M47
DBX 200.0 ... 7	M_Fkt_M48 ... M55	BOOL	M-Signale M48 ... M55
DBX 201.0 ... 7	M_Fkt_M56 ... M63	BOOL	M-Signale M56 ... M63
DBX 202.0 ... 7	M_Fkt_M64 ... M71	BOOL	M-Signale M64 ... M71
DBX 203.0 ... 7	M_Fkt_M72 ... M79	BOOL	M-Signale M72 ... M79
DBX 204.0 ... 7	M_Fkt_M80 ... M87	BOOL	M-Signale M80 ... M87
DBX 205.0 ... 7	M_Fkt_M88 ... M95	BOOL	M-Signale M88 ... M95
DBX 206.0 ... 3	M_Fkt_M96 ... M99	BOOL	M-Signale M96 ... M99

#### Hinweis

M02/M30 als Hilfsfunktionsausgabe an die PLC sagt nicht aus, dass das Teileprogramm beendet wurde. Um das Ende eines Teileprogramms im Kanal sicher zu ermitteln, muss das folgende Nahtstellensignal ausgewertet werden:

DB21, ... .DBX33.5 (M02/M30 aktiv)

Der Kanalzustand muss RESET sein. Die Hilfsfunktionsausgabe könnte aus einem asynchronen Unterprogramm (ASUP) oder einer Synchronaktion stammen und hat in diesen Fällen nichts mit dem tatsächlichen Teileprogrammende zu tun.

### 12.16.3 G-Funktionen

Die im Teileprogramm, ASUP oder Synchronaktionen programmierten G-Funktionen werden von der NC kanalspezifisch an die PLC übertragen:

- G-Funktionen aus Kanal 1: DB 21
- G-Funktionen aus Kanal 2: DB 22
- usw.

Die Signaldauer beträgt 1 PLC-Zyklus.

#### POWER ON

Nach POWER ON wird in der NC/PLC-Nahtstelle für alle G-Gruppen der Wert Null, d. h. aktive G-Gruppe undefiniert, vorgegeben.

#### Teileprogramm-Ende bzw. -Abbruch

Nach Teileprogramm-Ende bzw. -Abbruch bleibt der letzte Zustand der G-Gruppe erhalten.

**NC-START**

Nach NC-START werden die Werte der im Maschinendatum:  
 MD22510 \$NC\_GCODE\_GROUPS\_TO\_PLC  
 angegebenen 8 G-Gruppen entsprechend der über Maschinendaten festgelegten  
 Grundstellung sowie der im Teileprogramm programmierten Werte überschrieben.

Adresse in DB 21, ...	Variable	Typ	Grundstellung	Kommentar
DBB 208	G_FKT_GR_1	BYTE	0	Aktive G-Funktion der Gruppe 1
DBB 209	G_FKT_GR_2	BYTE	0	Aktive G-Funktion der Gruppe 2
DBB 210	G_FKT_GR_3	BYTE	0	Aktive G-Funktion der Gruppe 3
DBB 211	G_FKT_GR_4	BYTE	0	Aktive G-Funktion der Gruppe 4
DBB 212	G_FKT_GR_5	BYTE	0	Aktive G-Funktion der Gruppe 5
DBB 213	G_FKT_GR_6	BYTE	0	Aktive G-Funktion der Gruppe 6
DBB 214	G_FKT_GR_7	BYTE	0	Aktive G-Funktion der Gruppe 7
DBB 215	G_FKT_GR_8	BYTE	0	Aktive G-Funktion der Gruppe 8
DBB 216	G_FKT_GR_9	BYTE	0	Aktive G-Funktion der Gruppe 9
DBB 217	G_FKT_GR_10	BYTE	0	Aktive G-Funktion der Gruppe 10
DBB 218	G_FKT_GR_11	BYTE	0	Aktive G-Funktion der Gruppe 11
DBB 219	G_FKT_GR_12	BYTE	0	Aktive G-Funktion der Gruppe 12
DBB 220	G_FKT_GR_13	BYTE	0	Aktive G-Funktion der Gruppe 13
DBB 221	G_FKT_GR_14	BYTE	0	Aktive G-Funktion der Gruppe 14
DBB 222	G_FKT_GR_15	BYTE	0	Aktive G-Funktion der Gruppe 15
DBB 223	G_FKT_GR_16	BYTE	0	Aktive G-Funktion der Gruppe 16
DBB 224	G_FKT_GR_17	BYTE	0	Aktive G-Funktion der Gruppe 17
DBB 225	G_FKT_GR_18	BYTE	0	Aktive G-Funktion der Gruppe 18
DBB 226	G_FKT_GR_19	BYTE	0	Aktive G-Funktion der Gruppe 19
DBB 227	G_FKT_GR_20	BYTE	0	Aktive G-Funktion der Gruppe 20
DBB 228	G_FKT_GR_21	BYTE	0	Aktive G-Funktion der Gruppe 21
DBB 229	G_FKT_GR_22	BYTE	0	Aktive G-Funktion der Gruppe 22
DBB 230	G_FKT_GR_23	BYTE	0	Aktive G-Funktion der Gruppe 23
DBB 231	G_FKT_GR_24	BYTE	0	Aktive G-Funktion der Gruppe 24

Adresse in DB 21, ...	Variable	Typ	Grundstellung	Kommentar
DBB 232	G_FKT_GR_25	BYTE	0	Aktive G-Funktion der Gruppe 25
DBB 233	G_FKT_GR_26	BYTE	0	Aktive G-Funktion der Gruppe 26
DBB 234	G_FKT_GR_27	BYTE	0	Aktive G-Funktion der Gruppe 27
DBB 235	G_FKT_GR_28	BYTE	0	Aktive G-Funktion der Gruppe 28
DBB 236	G_FKT_GR_29	BYTE	0	Aktive G-Funktion der Gruppe 29
DBB 237	G_FKT_GR_30	BYTE	0	Aktive G-Funktion der Gruppe 30
DBB 238	G_FKT_GR_31	BYTE	0	Aktive G-Funktion der Gruppe 31
DBB 239	G_FKT_GR_32	BYTE	0	Aktive G-Funktion der Gruppe 32
DBB 240	G_FKT_GR_33	BYTE	0	Aktive G-Funktion der Gruppe 33
DBB 241	G_FKT_GR_34	BYTE	0	Aktive G-Funktion der Gruppe 34
DBB 242	G_FKT_GR_35	BYTE	0	Aktive G-Funktion der Gruppe 35
DBB 243	G_FKT_GR_36	BYTE	0	Aktive G-Funktion der Gruppe 36
DBB 244	G_FKT_GR_37	BYTE	0	Aktive G-Funktion der Gruppe 37
DBB 245	G_FKT_GR_38	BYTE	0	Aktive G-Funktion der Gruppe 38
DBB 246	G_FKT_GR_39	BYTE	0	Aktive G-Funktion der Gruppe 39
...	...	...	...	...
DBB 271	G_FKT_GR_64	BYTE	0	Aktive G-Funktion der Gruppe 64

Eine vollständige Auflistung aller G-Funktionen findet sich in:

**Literatur:**

Programmierhandbuch Grundlagen; Kapitel: "Liste der G-Funktionen/Wegbedingungen"

### 12.16.4 Meldesignale im DB 2

Über den DB 2 wird dem Anwender die Möglichkeit gegeben, für einzelne Signale Meldungen auf der Bedientafelfront anzuzeigen. Die Signale sind - wie in den Listen der Nahtstellensignale dargestellt - in vordefinierte Gruppen unterteilt. Bei Auftreten oder Verschwinden bzw. Quittieren einer Meldung wird die in der Spalte Meldenummer aufgeführte Nummer zum HMI übertragen. Zu jeder Meldenummer kann im HMI ein Text hinterlegt werden.

**Literatur:**

- Listen sl (Buch2); Kapitel "PLC-Meldungen (DB 2)"
- Inbetriebnahmehandbuch; Kapitel "Alarm- und Meldungstexte"

---

**Hinweis**

Die Anzahl der Anwenderbereiche kann über den FB 1 parametrisiert werden.

Nach Änderung der Konfiguration (FB 1: MsgUser) müssen DB 2 / DB 3 gelöscht werden.

---

## 12.17 Programmiertipps mit STEP 7

Im Folgenden sollen einige Hinweise gegeben werden, um komplexere Abläufe in STEP 7 zu programmieren. Es handelt sich hierbei im Wesentlichen um den Umgang mit dem Datentyp POINTER bzw. ANY.

Grundlegende Hinweise zum Aufbau vom Datentyp POINTER und ANY siehe:

### Literatur:

STEP 7-Handbuch; Kapitel: "Entwerfen von Anwenderprogrammen" > "Register der CPU und Speichern von Daten"

### 12.17.1 Daten kopieren

#### Kopiervarianten

Für das schnelle Kopieren von Daten von einem DB in einen anderen empfiehlt sich:

- **bei größeren Datenmengen** der Einsatz der Systemfunktionen SFC BLKMOV bzw. SFC FILL, da hier ein wesentlich schnelleres Kopieren stattfinden kann.
- **bei kleineren Datenmengen** die untenstehende Routine, da die Versorgung der ANY-Parameter an den SFCs zusätzliche Zeit kostet.

#### Beispiel

Code		Kommentar
		// DB xx.[AR1] ist Quelle
		// DI yy.[AR2] ist Ziel
AUF	DB 100;	//Quelle DB
LAR1	P#20.0;	//Anfangsadresse Quelle auf Datenbyte 20
AUF	DI 101;	//Ziel DB
LAR2	P#50.0;	//Anfangsadresse Ziel auf Datenbyte 50
		//AR1, AR2, DB, DI vorher geladen
L	4;	//8 Byte transferieren
M001:		
L	DBW [AR1,P#0.0];	//Wortweise kopieren
T	DIW [AR2,P#0.0];	
+AR1	P#2.0;	
+AR2	P#2.0;	
TAK;		
LOOP	M001;	

## 12.17.2 ANY und POINTER

In den folgenden Programmbeispielen sollen Programmiermechanismen dargestellt werden. Hierbei soll der Zugriff auf Ein-/Aus- und DurchgangsvARIABLEN (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT) vom Datentyp POINTER bzw. ANY innerhalb eines FC bzw. FB aufgezeigt werden. Die Zugriffe sind so dargestellt, dass eine teilsymbolische Programmierform verwendet werden kann.

### 12.17.2.1 Verwendung von POINTER und ANY im FC

#### Funktion

Der FC 99 hat Eingangsparameter, die als POINTER bzw. ANY definiert sind.

In dem Beispiel wird ein Rumpfprogramm gezeigt, um auf die Teilkomponenten des POINTER bzw. ANY zuzugreifen. Hierbei wird der über den POINTER, ANY parametrisierte DB aufgeschlagen und der Adressoffset als bereichsübergreifender Zeiger im Adressregister AR1 hinterlegt. Somit kann auf Datenelemente der Variablen (in der Regel Strukturen, Arrays) zugegriffen werden, die über den POINTER, ANY adressiert sind.

Dieser Zugriff ist im Beispiel am Ende der jeweiligen Programmsequenz dargestellt. Bei dem Datentyp ANY besteht zusätzlich die Möglichkeit, anhand des Datentyps und der Anzahl von Elementen eine Überprüfung bzw. Verzweigung beim Zugriff der Variablen durchzuführen.

#### Beispiel

Code	Kommentar
FUNCTION FC 99: VOID	
VAR_INPUT	
Row : BYTE ;	
Convert : BOOL ;	//Zahlenwandlung aktivieren
Addr : POINTER;	//Zeigt auf Variable
Addr1 : ANY;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Anzahl : WORD ;	
typ : BYTE;	
END_VAR	
BEGIN	

Code	Kommentar
NETWORK	
TITLE =	
	//POINTER
L P##Addr;	
LAR1 ;	//Pointer holen
L W [AR1,P#0.0];	//DB-Nummer holen
T #dbchr;	
L D [AR1,P#2.0];	//Offsetteil des Pointers
LAR1 ;	
AUF DB [#dbchr];	//Db der Variablen aufschlagen
L B [AR1,P#40.0];	//hole Byte-Wert über Pointer mit dem //Adressoffset 40 //ANY
L P##Addr1;	
LAR1 ;	//ANY holen
L B [AR1,P#1.0];	//Typ hole
T #typ;	
L W [AR1,P#2.0];	//Anzahl holen
T #Anzahl;	
L W [AR1,P#4.0];	//DB-Nummer holen
T #dbchr;	
L D [AR1,P#6.0];	//Offsetteil des Pointers
LAR1 ;	
AUF DB [#dbchr];	//Db der Variablen aufschlagen
L B [AR1,P#0.0];	//hole Byte-Wert über ANY

### 12.17.2.2 Verwendung von POINTER und ANY im FB

#### Funktion

Der FB 99 hat Eingangsparameter, die als POINTER bzw. ANY definiert sind.

In dem Beispiel wird ein Rumpfprogramm gezeigt um auf die Teilkomponenten des POINTER bzw. ANY zuzugreifen. Hierbei wird der über den POINTER, ANY parametrisierte DB aufgeschlagen und der Adressoffset als bereichsübergreifender Zeiger im Adressregister AR1 hinterlegt. Somit kann auf Datenelemente der Variable (in der Regel Strukturen, Arrays) zugegriffen werden, die über den POINTER, ANY adressiert sind.

Dieser Zugriff ist im Beispiel am Ende der jeweiligen Programmsequenz dargestellt. Bei dem Datentyp ANY besteht zusätzlich die Möglichkeit anhand des Datentyps und der Anzahl von Elementen eine Überprüfung bzw. Verzweigung beim Zugriff der Variablen durchzuführen.

### Beispiel

Code	Kommentar
FUNCTIONBLOCK FB 99	
VAR_INPUT	
Row : BYTE ;	
Convert : BOOL ;	//Zahlenwandlung aktivieren
Addr : POINTER ;	//Zeigt auf Variable
Addr1 : ANY;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Anzahl : WORD ;	
typ : BYTE;	
END_VAR	
BEGIN	
NETWORK	
TITLE =	
	//POINTER
L                    P##Addr;	
LAR1;	//Pointer holen aus Instanz-DB
L                    DIW [AR1,P#0.0];	//DB-Nummer holen
T                    #dbchr;	
L                    DID [AR1,P#2.0];	//Offsetteil des Pointers
LAR1 ;	
AUF                  DB [#dbchr];	//Db der Variablen aufschlagen
L                    B [AR1,P#40.0];	//hole Byte-Wert über Pointer mit dem
	//Adressoffset 40
	//ANY
L                    P##Addr1;	
LAR1 ;	//ANY holen aus Instanz-DB
L                    DIB [AR1,P#1.0];	//Typ holen
T                    #typ;	
L                    DIW [AR1,P#2.0];	//Anzahl holen
T                    #Anzahl;	
L                    DIW [AR1,P#4.0];	//DB-Nummer holen
T                    #dbchr;	
L                    DID [AR1,P#6.0];	//Offsetteil des Pointers
LAR1 ;	
AUF                  DB [#dbchr];	//Db der Variablen aufschlagen
L                    B [AR1,P#0.0];	//hole Byte-Wert über ANY



### 12.17.2.3 Variable POINTER bzw. ANY für Übergabe an FC oder FB

#### Variable POINTER bzw. ANY

In STEP 7 ab Version 1 besteht die Möglichkeit, einen POINTER bzw. ANY in der VAR\_TEMP zu definieren.

Die Versorgung eines ANY soll im Folgenden an 2 Beispielen dargestellt werden.

#### Beispiel 1: ANY-Parameter über eine Auswahlliste an einen anderen FB (FC) übergeben

An einem FB (FC) sind **mehrere** ANY-Parameter definiert. Es soll nun über eine Auswahlliste ein bestimmter ANY-Parameter an einen anderen FB (FC) übergeben werden. Dieses funktioniert nur über einen ANY in der VAR\_TEMP. Im Parameter "WelcherAny" kann 1 bis 4 parametrisiert sein, um Addr1 bis Addr4 auszuwählen.

---

#### Hinweis

Im Baustein wird das Adressregister AR2 verwendet. Dieses Adressregister AR2 wird bei auch bei Multi-Instanz-DB verwendet. Deshalb darf dieser FB **nicht** als Multi-Instanz-DB erklärt sein.

---

Code	Kommentar
FUNCTIONBLOCK FB 100	
CODE_VERSION1	//ab STEP 7 Version 2, um Multi-Instanz-DB //zu deaktivieren
VAR_INPUT	
WelcherAny : INT;	
Addr1 : ANY;	//feste Reihenfolge beachten
Addr2 : ANY;	
Addr3 : ANY;	
Addr4 : ANY;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Anzahl : WORD ;	
typ : BYTE;	
Temp_addr : ANY;	
END_VAR	
BEGIN	
NETWORK	

Code	Kommentar
TITLE =	
L           WelcherAny;	
DEC 1;	
L           P#10.0;	//10 Byte pro ANY
*I;	
LAR2;	
L           P##Addr1;	
+AR2;	//Anfangsadresse der ANYs addieren
L           P##Temp_addr;	
LAR1 ;	//Pointer holen aus VAR_TEMP
L           DID [AR2,P#0.0];	//Zeigerwerte übertragen in VAR_TEM
T           LD [AR1,P#0.0];	
L           DID [AR2,P#4.0];	
T           LD [AR1,P#4.0];	
L           DIW [AR2,P#8.0];	
T           LW [AR1,P#8.0];	
CALL FB 101, DB 100	
(ANYPAR := #Temp_addr);	//ANYPAR ist der Datentyp ANY

### Beispiel 2: Vorher zusammengebauten ANY-Parameter an einen anderen FB (FC) übergeben

Es soll ein ANY-Parameter, der vorher zusammengebaut wurde, an einen anderen FB (FC) übergeben werden. Dieses funktioniert nur über einen ANY in der VAR\_TEMP.

Code	Kommentar
FUNCTIONBLOCK FB 100	
VAR_INPUT	
DBNummer: INT;	
DOffset : INT;	
Datentyp: INT;	
Anzahl: INT;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Temp_addr : ANY;	
END_VAR	
BEGIN	
NETWORK	
TITLE =	
L           P##Temp_addr;	

Code	Kommentar
LAR1 ;	//Pointer holen aus VAR_TEMP
L        B#16#10;	//Kennung ANY
T        LB [AR1,P#0.0];	
L        Datentyp;	
T        LB [AR1,P#1.0];	
L        Anzahl;	
T        LW [AR1,P#2.0];	
L        DBNummer;	
T        LW [AR1,P#4.0];	
L        DBOffset;	
SLD 3;	//Offset ist ein Bitoffset
T        LD [AR1,P#6.0];	
CALL FB 101, DB 100	
(ANYPAR := #Temp_addr);	//ANYPAR ist der Datentyp ANY

### 12.17.3 Multi-Instanz-DB

#### Funktion

In STEP 7 ab Version 2 können FBs Multi-Instanz-fähig, d. h. mit Multi-Instanz-DBs versehen sein. Multi-Instanz-DBs zeichnen sich dadurch aus, dass ein Datenbaustein für verschiedene Instanzen von FBs verwendet werden kann (siehe Dokumentation STEP 7). Somit kann das Mengengerüst der DBs optimiert werden.

Das Aktivieren von Multi-Instanz-DBs sollte nur dann erfolgen, wenn diese auch genutzt werden, da sich Laufzeit und Codegröße der FBs erhöhen.

---

#### Hinweis

Bei komplexeren Programmen in FBs, die Zeiger und Adressregister verwenden, sind bei FBs, die Multi-Instanz-fähig sein sollen, vom Programmierer bestimmte Regeln einzuhalten.

Bei Multi-Instanzen wird die Anfangsadresse der Variablen (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR) mit dem DI-Datenbaustein-Register und dem Adressregister AR2 übergeben. Bei Zugriffen innerhalb des Multi-Instanz-fähigen FB regelt der Compiler die Zugriffe auf diese Variablen über das Adressregister AR2 selbstständig. Wenn allerdings komplexere Programmteile in diesem FB auch mit Adressregistern arbeiten müssen (z. B. Daten kopieren), dann ist vor dem Ändern des AR2 der bisherige Inhalt zu retten. Das AR2-Register muss vor einem Zugriff auf eine Instanz-Variable (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR) wieder mit dem ursprünglichen Inhalt restauriert sein. Das AR2-Register der Instanz ist am sinnvollsten in eine Lokal-Variable (VAR\_TEMP) zu retten.

Der Befehl "Zeiger auf eine Instanzvariable laden" liefert einen Zeigerwert ab dem Beginn der Instanzdaten. Um auf diese Variable über Zeiger zugreifen zu können, ist der Offset, der im AR2 steht, hinzuzurechnen.

---

## Beispiel

Code	Kommentar
FUNCTION_BLOCK FB 99	
VAR_INPUT	
varin: INT;	
END_VAR	
VAR	
variable1: ARRAY[0..9] OF INT;	
variable2: INT;	
END_VAR	
BEGIN	
L        P##variable1;	//Pointer auf Anfang des ARRAYs
	//Im Accu steht jetzt der Wert 8500 0010
	//Im AR2 steht auch ein Bereichs-übergreifender
	//Zeiger. Wenn Bereichs-übergreifend gearbeitet
	//werden soll, dann ist bei der Addition dieser
	//beiden Zeiger ein Bereich auszublenden.
UD        DW#16#00FF_FFFF,	//Ausblenden des Bereichs
LAR1	//Ins AR1 laden
TAR2;	
+AR1 AR2;	//AR2 Instanz Offset hinzuaddieren
	//Jetzt kann über AR1 indirekt auf das ARRAY
	//von variable1 zugegriffen werden.
L        DIW [AR1, P#0.0];	//Z. B. Zugriff auf erstes Element
END_FUNCTION_BLOCK	

### 12.17.4 Strings

Der Datentyp STRING wird durch bestimmte Dienste des Grundprogramms benötigt. Aus diesem Grund sollen ein paar zusätzliche Informationen zum Aufbau von Strings und dem generellen Umgang bei Parameterzuweisungen beschrieben werden.

#### Aufbau von Strings

Ein Datum mit dem Datentyp STRING ist generell in einem Datenbaustein abgelegt (definiert). Bei der Definition gibt es 2 Varianten:

1. Einer Variablen wird nur der Datentyp STRING zugewiesen. Hierbei generiert der STEP 7-Compiler eine Länge von 254 Zeichen.
2. Einer Variablen wird der Datentyp STRING mit einer Längenangabe in eckigen Klammern zugewiesen (z. B. [32]). Hierbei generiert der STEP 7-Compiler eine String-Länge entsprechend der Vorgabe.

Für eine Variable vom Datentyp STRING werden immer 2 Byte mehr abgelegt, als die Definition ausweist. Hierbei ist in dem 1. Byte die Anzahl der maximal möglichen Zeichen durch den STEP 7 -Compiler abgelegt. Im 2. Byte ist die Anzahl der genutzten Zeichen enthalten. Normalerweise wird hier die Nutzlänge des zugewiesenen Strings durch den STEP 7-Compiler abgelegt. Ab dem 3. Byte sind die Zeichen abgelegt (1 Byte pro Zeichen).

Eine Zuweisung von STRING-Parametern an Bausteine des Grundprogramms erfolgt generell über den Datentyp POINTER bzw. ANY. Die Zuweisung soll generell über symbolische Programmierung erfolgen. Hierbei ist der Datenbaustein, in dem sich der zu parametrierende String befindet, in die Symbolliste aufzunehmen. Anschließend ist die Zuweisung an den Grundprogramm-Baustein mit symbolischen Namen des Datenbausteins, anschließenden Punkt und symbolischen Namen der STRING-Variable durchzuführen.

## 12.17.5 Ermittlung von Offsetadressen auf Datenbaustein-Strukturen

### Funktion

Eine weitere häufig vorkommende Aufgabe ist die symbolische Ermittlung einer Offsetadresse innerhalb eines strukturierten DBs. In diesem DB liegt z. B. an beliebiger Stelle ein ARRAY oder eine STRUKTUR. Man möchte, nachdem das Adressregister symbolisch mit der Anfangsadresse geladen wurde, über ein Adressregister auf einzelne Elemente des ARRAY oder der STRUKTUR zugreifen. Ein Weg wie man das Adressregister symbolisch laden kann, führt über einen FC, der als Eingangsparameter einen Pointer hat. Diesem Eingangsparameter des FC wird nun im Programm symbolisch die Adresse des ARRAY oder der STRUKTUR zugewiesen. Der Programmcode im FC ermittelt nun die Offsetadresse aus dem Eingangsparameter und übergibt die Offsetadresse im Adressregister (AR1) an die rufende Funktion. Somit ist auch bei indirekten Zugriffen eine Symbolische Adressierung möglich.

### Beispiel

Code	Kommentar
FUNCTION FC 99: VOID	
VAR_INPUT	
Addr : POINTER ;	//Zeigt auf Variable
END_VAR	
BEGIN	
NETWORK	
TITLE =	
L        P##Addr;	
LAR1 ;	//Pointer holen von Addr
L        D [AR1,P#2.0];	//Offsetteil des Pointers der Variable
LAR1 ;	
END_FUNCTION	

## 12.17.6 Aufrufe von FBs

### Funktion

Zur Optimierung von Ablaufgeschwindigkeiten ist es sinnvoll, alle Funktionsbausteinaufrufe mit vielen statischen Parametern, wie z. B. die vom Grundprogramm zur Verfügung gestellten Bausteine FB 2, 3, 4, 5 und 7, schon im Anlauf mit den zugehörigen Instanzparametern aufzurufen. Im Anlauf (OB 100) sollte die Vorbesetzung von Parametern erfolgen, die anschließend im zyklischen Teil (OB 1) nicht mehr verändert werden. Diese festen Parameterwerte werden in dem zyklischen Aufruf nicht mehr parametrieren, da sie bereits im Anlauf in den Instanz-DB geschrieben worden sind.

### Beispiel: Parametrierung des FB 2 mit dem Instanz-DB 110

Am folgenden Beispiel soll gezeigt werden, wie eine sinnvolle Aufteilung in OB 100-Anteil und OB 1-Anteil zu realisieren ist.

Als erstes ist der übliche Aufruf im zyklischen Programm dargestellt:

```
CALL FB 2, DB 110(
    Req :=          M 100.0,
    NumVar :=       2,                               //2 Variablen lesen
    Addr1 :=        NCVAR.C1_RP_rpa0_0
    Line1 :         W#16#1
    Addr2 :=        NCVAR.C1_RP_rpa0_0
    Line2 .         W#16#2
    Error :=        M1.0,
    NDR :=          M1.1,
    State :=        MW 2,
    RD1 :=          P#M 4.0 REAL 1,
    RD2 :=          P#M 24.0 REAL 1,
```

Ab hier folgt die modifizierte Version des Programmaufrufs.  
Hier ist der Aufruf im OB 100 dargestellt:

```
CALL FB 2, DB 110(
    Req :=          FALSE,
    NumVar :=       2,                               //2 Variablen lesen
    Addr1 :=        NCVAR.C1_RP_rpa0_0
    Line1 :         W#16#1
    Addr2 :=        NCVAR.C1_RP_rpa0_0
    Line2 .         W#16#2
    RD1 :=          P#M 4.0 REAL 1,
    RD2 :=          P#M 24.0 REAL 1,
```

Hier ist der noch verbleibende Aufruf im OB 1 dargestellt:

```
CALL FB 2, DB 110 (  
    Req :=          M0.0,  
    Error :=       M1.0,  
    NDR :=         M1.1,  
    State :=       MW 2,
```

---

### Hinweis

Durch diese Maßnahme wird eine kürzere Durchlaufzeit im OB 1 erreicht, da die statischen Parameterwerte nicht in jedem OB 1-Durchlauf in den Instanz-DB kopiert werden.

---

### Ersparnis dieser Variante:

Der zyklische Kopieraufwand von 3 Integerwerten und 4 ANY-Parametern bezogen auf den Instanz-DB, welches aus 3-mal Laden einer Konstante mit einen 3-maligen Transfer in den Instanz-Datenbaustein resultiert. Bei jedem ANY-Transfer werden 4-mal Konstanten mit anschließendem Transfer in den Datenbaustein geladen.

## 12.18 Datenlisten

### 12.18.1 Maschinendaten

#### 12.18.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10100	PLC_CYCLIC_TIMEOUT	Zyklische PLC-Überwachungszeit
14504	MAXNUM_USER_DATA_INT	Anzahl der Anwenderdaten (INT)
14506	MAXNUM_USER_DATA_HEX	Anzahl der Anwenderdaten (HEX)
14508	MAXNUM_USER_DATA_FLOAT	Anzahl der Anwenderdaten (FLOAT)
14510	USER_DATA_INT	Anwenderdatum (INT)
14512	USER_DATA_HEX	Anwenderdatum (HEX)
14514	USER_DATA_FLOAT[n]	Anwenderdatum (FLOAT)

#### Hinweis

Ein Maschinendatum im Integer-/Hexformat wird in der NC als DWORD gehandhabt. Ein Maschinendatum im Fließkomma-Format wird in der NC als FLOAT (8 Byte IEEE) gehandhabt. Sie werden jeweils in der NC/PLC-Nahtstelle abgelegt und können vom PLC-Anwenderprogramm bereits im Hochlauf der PLC aus dem DB 20 gelesen werden.

#### 12.18.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
28150	MM_NUM_VDIVAR_ELEMENTS	Anzahl Elemente für das Schreiben von PLC-Variablen



## P4: PLC für SINUMERIK 828D

### 13.1 Überblick

#### 13.1.1 PLC-Firmware

Die PLC der SINUMERIK 828D ist eine integrierte PLC auf Basis des Befehlssatzes der SIMATIC S7-200.

Die Programmierung der PLC-Anwenderprogramme erfolgt in erster Linie an einem Windows-PC mit dem "Programming Tool PLC828". Zusätzlich kann die PLC auch über die Bedienoberfläche der Steuerung diagnostiziert und editiert werden. Hierzu steht ein "Ladder-Add-On-Tool" auf der Steuerung zur Verfügung.

Folgende Besonderheiten sind zu beachten:

- Die Programmierung des PLC-Anwenderprogramms erfolgt durchgehend in Kontaktplan (KOP).
- Es wird eine Untermenge der Programmiersprache S7-200 unterstützt.
- Beim Laden in CPU werden neben dem ablaufrelevanten Code auch die kompletten Projektdaten (einschließlich Symbolik und Kommentaren) in die Steuerung geladen. Damit enthält die Steuerung immer das passende Projekt zum aktuellen laufenden PLC-Anwenderprogramm.
- Beim Laden aus CPU werden die kompletten Projektdaten (einschließlich Symbolik und Kommentaren) in das Programming Tool PLC828 geladen und sind mit diesem bearbeitbar.
- Der Anwender muss seine Daten und Prozessinformationen typgerecht verwalten. Bei allen Zugriffen auf die Daten muss konsequent der vereinbarte Datentyp verwendet werden.

### 13.1.2 PLC-Anwendernahstelle

Die Anwendernahstelle wird von der PLC-Firmware eingerichtet, welche auch den Austausch aller Signale und Daten zwischen der PLC einerseits und NCK und HMI andererseits organisiert.

Die Anwendernahstelle besteht aus den Teilen:

- Datenschnittstelle mit zyklischem Austausch (siehe "Datenschnittstelle (Seite 1259)")
- Funktionsschnittstelle mit funktions- bzw. auftragsbezogenem Datenaustausch (siehe "Funktionsschnittstelle (Seite 1265)")

Die strukturierten Daten dieser Schnittstellen (remanente und nichtremanente) werden von der Firmware durch Zuordnung auf Datenbausteine dem Anwender zur Verfügung gestellt: "Kommunikationspartner" des PLC-Anwenderprogramms sind die NC (NCK, Werkzeugverwaltung, NC-Kanal, Achsen, Spindeln, ...) und das HMI.

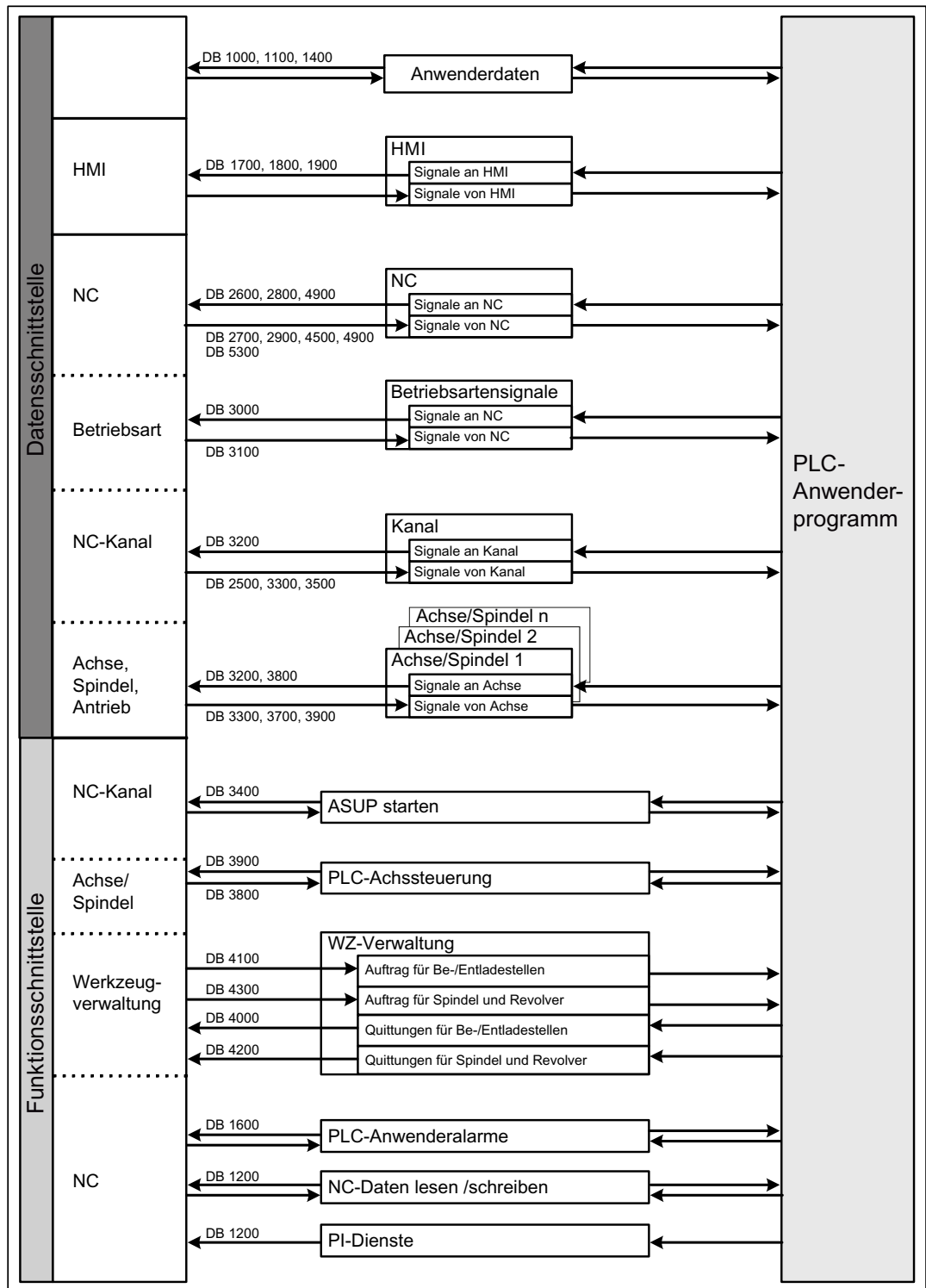


Bild 13-1 Überblick über die Anwenderstation der PLC 828D

### 13.1.2.1 Zyklisch ausgetauschte Daten

Ausgetauscht werden Daten zwischen PLC und NC einerseits sowie zwischen PLC und HMI andererseits.

Daten **an** die PLC werden von der Firmware am Zyklusanfang des Anwenderprogramms bereitgestellt. Dadurch wird gesichert, dass z. B. die Signale vom NCK über einen Zyklus konstant bleiben.

Daten **von** der PLC werden von der Firmware am Zyklusende des Anwenderprogramms an NCK oder HMI weitergeleitet.

#### Nahtstelle PLC ↔ NCK

Die zyklischen Daten enthalten z. B. Statussignale ("Programm läuft", "Programm unterbrochen") und Steuersignale (Start, Stopp) und Hilfs- und G-Funktionen.

Die Daten sind gegliedert in Signale für:

- Betriebsarten
- Kanäle
- Achsen/Spindeln
- Allgemeine NCK-Signale

#### Nahtstelle PLC ↔ HMI

Dies sind Signale für:

- Programmanwahl über Listen
- Messenger Steuerkommando
- Allgemeine Signale von/an HMI
- Signale von/an Wartungsplaner
- Signale von Bedientafel (remanenter Bereich)
- Allgemeine Anwahl-/Statussignale von/an HMI (remanenter Bereich)

### 13.1.2.2 Alarmer und Meldungen

Die Anwendernahtstelle im DB1600 bietet die Möglichkeit, Fehler- und Betriebsmeldungen auf dem HMI zur Anzeige zu bringen.

Durch die Firmware werden eingetragene Signale ausgewertet und als kommende und gehende Alarmer und Meldungen an den HMI gesendet und dort zur Anzeige gebracht. Die Fehlertexte werden vom HMI verwaltet.

### 13.1.2.3 Remanente Daten

Für die remanenten Daten gibt es die Anwender-Datenbausteine DB9000 - DB9063 und den Datenbereich DB1400.DBW0 - DBW127. Dort kann der Anwender alle Daten ablegen, die über POWER OFF/ON ihre Gültigkeit behalten sollen. Die remanenten Daten werden im nichtflüchtigen Speicher gesichert, jedoch nicht bei der Datensicherung.

### 13.1.2.4 Nichtremanente Daten

Nichtremanente Daten (z. B. Merker, Zeiten und Zähler) werden bei jedem Steuerungshochlauf gelöscht.

### 13.1.2.5 PLC-Maschinendaten

Die PLC-Maschinendaten befinden sich im Bereich der NCK-Maschinendaten. Diese Daten werden bei POWER ON von der PLC-Firmware in den DB4500 der PLC-Anwendernahtstelle übertragen und können dort vom PLC-Anwenderprogramm ausgewertet werden.

## Literatur

Listenhandbuch SINUMERIK 828D

### 13.1.3 PLC-Eckdaten

Die integrierte PLC verfügt über einen Programmspeicher von 24000 PLC-Anweisungen, welche komplett in einem festen PLC-Zyklus abgearbeitet werden.

Im optional nutzbaren Interruptprogramm INT0 können maximal 500 Anweisungen abgearbeitet werden. Es wird servosynchron ausgeführt und ermöglicht schnellste Reaktionen auf Prozessereignisse. Deshalb werden interruptfähige PLC-Peripheriebaugruppen nicht gebraucht.

Daten	Anzahl	Besonderheiten
Hauptprogramm (MAIN)	1	
Unterprogramm (SBRx)	256	
Interrupts	2	
Zeitgesteuerter Interrupt	1	servosynchrones Interruptprogramm
Alarmer / Meldungen	248	
Merker	4096	nicht remanent
Zähler	64	nicht remanent
Zeiten, davon:	128	nicht remanent
10 ms Inkrementintervall	112	
100 ms Inkrementintervall	16	
Anwenderdatenbaustein mit je max. 512 Bytes	64	Adressbereich DB9000 bis DB9063
Datenübertragung NCK ↔ PLC		Über fest parametrierbare Schnittstelle

### 13.1.4 PLC Peripherie, schnelle Onboard-Ein-/Ausgänge

Zu den Eigenschaften der schnellen Onboard-Ein-/Ausgänge und deren Reaktionszeiten, siehe Kapitel "Schnelle Onboard-Ein- und Ausgänge (Seite 1199)".

Erläuterungen zu den Peripheriemodulen, den Maschinensteuertafeln sowie der Zuordnung der Onboard-Ein-/Ausgänge zur PLC siehe:

**Literatur:**

Gerätehandbuch PPU SINUMERIK 828D

## 13.1.5 PLC-Toolbox

### 13.1.5.1 Stern/Dreieck-Umschaltung

Für die Stern/Dreieck-Umschaltung wird in der PLC-Toolbox folgender Baustein mitgeliefert:

- StarDelta

---

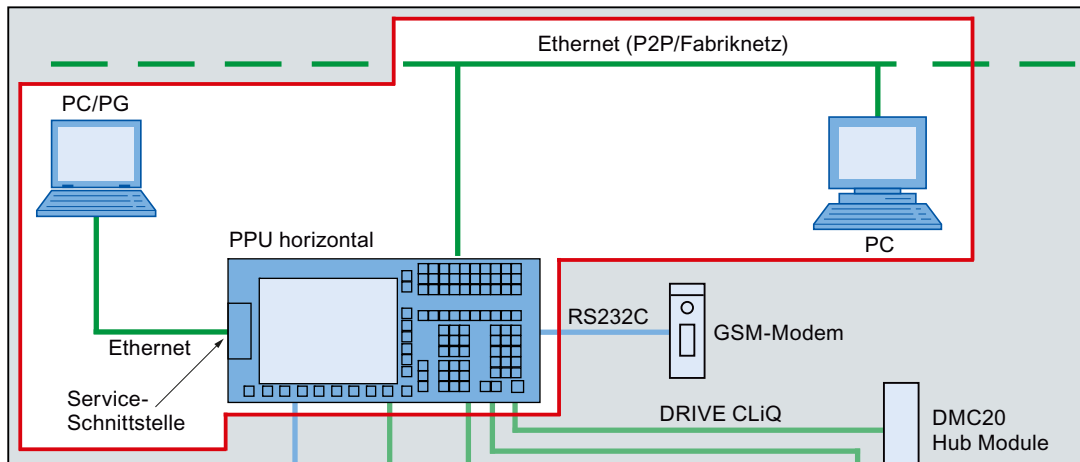
#### **Hinweis**

Mit diesem Baustein kann die Stern/Dreieck-Umschaltung auch von Spindelmotoren 1PH8 mit SMI an SINAMICS S120 durchgeführt werden.

---

## 13.2 Programming Tool PLC828

Das "Programming Tool PLC828" ist das Werkzeug, mit welchem die PLC-Programme komfortabel erstellt werden können. Es ist ein Windows-Programm und muss auf einem Windows-PC installiert werden. Für den Online-Zugang zur Steuerung muss dieser PC mit Ethernetanschluss ausgerüstet und mit der Steuerung über das Fabriknetz oder die Serviceschnittstelle verbunden sein.



### Literatur:

Inbetriebnahmehandbuch SINUMERIK 828D

Mit dem Aufruf des Programming Tools PLC828 wird – ohne Angabe eines bereits existierenden Projekts – implizit ein neues Projekt mit Default-Namen "Projekt1" angelegt. Dieses Projekt kann sofort zum Erstellen des PLC-Anwenderprogramms genutzt und dann unter einem beliebigen Namen gespeichert und in die Steuerung geladen werden.

Bereits existierende Projekte können auf die Windows-typische Weise geöffnet werden, ebenfalls Windows-typisch steht über F1 eine komfortable Online-Hilfe bereit.



## 13.3 Programmierung

### 13.3.1 Einführung

#### 13.3.1.1 Wichtige Begriffe

Beim Aufruf des Programming Tools PLC828 wird implizit ein neues Projekt angelegt oder aber ein vorgegebenes geöffnet.

Was aber ist ein Projekt, was ein Programm, ein Datenbaustein? Was sind Datenklassen?

Die wichtigsten Begriffe sollen im Folgenden kurz erklärt werden. Weitere und umfassendere Ausführungen siehe:

##### Literatur:

- Programming Tool PLC828 Online-Hilfe
- S7-200 Systemhandbuch

#### Projekt

Ein Projekt ist die größtmögliche Organisationseinheit für die PLC-Arbeiten des Anwenders. Es ist vergleichbar mit einem Container und kann:

- Programm- und Datenbausteine, Symbol- und Statustabellen, Querverweise sowie Schnittstellen- und Debug-Einstellungen aufnehmen.
- auf Datenträger gespeichert oder von dort geladen werden.
- in die CPU geladen und von dieser auch wieder geholt werden.

#### Datenklassen

Datenklassen sind vor allem Eigenschaften der Aktualwerte solcher Datenbausteine, die von den Anwendern explizit in das Projekt aufgenommen werden. (Nicht gemeint sind also systemimmanente Datenbausteine z. B. der Anwendernahtstelle.)

Die Datenklassen "Manufacturer", "Individual" und "User" wurden in die SINUMERIK 828D eingeführt, um:

- die DB-Aktualwerte speziellen Anwendergruppen zuordnen zu können.
- die zu einer Anwendergruppe (also Datenklasse) gehörenden Werte in die CPU oder von dieser laden zu können.
- Fehlersuche und Wartbarkeit zu erleichtern.

Alle Programme (mit Ausnahme der beiden Interruptprogramme INT100 und INT101) und alle Datenbausteinstrukturen (also der innere Aufbau, der "Typ" der Datenbausteine) und Datenbausteinanfangswerte haben die Datenklasse "Manufacturer".

Die Interruptprogramme INT100 und INT101 haben den Typ "Individual".

## Programm

Ein Programm (auch Programmorganisationseinheit "POE") ist ein Baustein für eine Folge von Befehlen (einschließlich Kommentaren), die der Anwender zur Lösung seiner Aufgabe mittels KOP-Editor aneinanderfügt. Drei Arten dieser POEs stehen dem Anwender zur Verfügung:

- Das Hauptprogramm (MAIN)  
Davon gibt es nur eines. Es wird vom System im PLC-Zyklus aufgerufen.
- 256 optionale Unterprogramme (SBR\_xyz)  
Unterprogramme dienen dem Strukturieren und Kapseln von Funktionalität.  
Sie können – einmal codiert – mehrfach aufgerufen werden.
- Drei optionale Interruptprogramme  
Diese werden im PLC-Zyklus an anderer Stelle höherprior abgearbeitet und sind Sonderaufgaben vorbehalten.

## Datenbaustein

Ein Datenbaustein ist ein Baustein für Daten (Anfangswerte, Aktualwerte) und Kommentare mit folgenden Eigenschaften:

- Die Daten werden genau in der vom Anwender vorgegebenen Reihenfolge gespeichert. Damit ist sozusagen die innere Struktur des Datenbausteins definiert und wenn mehrere Datenbausteine mit der gleichen inneren Struktur (d. h. des gleichen Typs) angelegt werden, steht ein bestimmtes Datum immer auf seinem bestimmten Platz. Dieser Platz wird Offset genannt und ist die relative Länge in Byte vom Anfang des Datenbausteins (DB) bis zum eigentlichen Datum.
- Den Daten können Anfangswerte zugewiesen werden, die sie nach dem erstmaligen Laden in die PLC annehmen.
- Die Istwerte (Aktualwerte) der Daten können online aus der Steuerung gelesen, verändert und auch mit dem Projekt gespeichert werden.

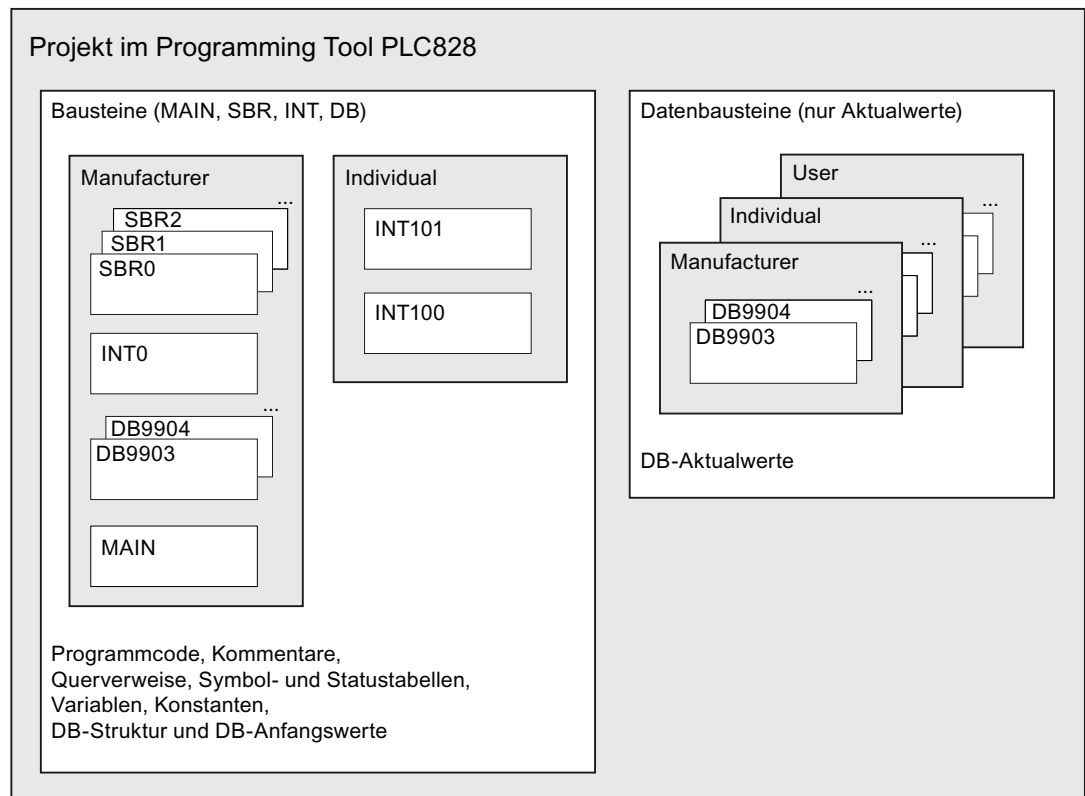


Bild 13-2 Beispiel einer Projektstruktur

## Symoltabellen

Die Symboltabelle dient der Verwendung symbolischer Adressen. Symbole erleichtern häufig die Programmierung und erhöhen die Übersichtlichkeit der Programme. Im übersetzten Programm, das in das Zielsystem geladen wird, werden alle Symbole in absolute Adressen umgewandelt. Die Angaben der Symboltabelle werden in das Zielsystem geladen.

## Statustabellen

In den Statustabellen können Sie beobachten, wie sich die Prozesswerte durch die Programmausführung ändern. Statustabellen werden nicht in das Zielsystem geladen. Sie dienen nur dazu, die Aktivitäten des Zielsystems (bzw. des simulierten Zielsystems) zu beobachten.

## Querverweise

Im Querverweisfenster werden tabellarisch dargestellt:

- Die symbolisch oder absolut adressierten Operanden und ihre Verwendungsorte
- Die verwendeten Bytes
- Die verwendeten Bits

Die Querverweise und die Angaben zu den verwendeten Elementen werden nicht in das Zielsystem geladen.

### 13.3.1.2 Projekt anlegen/öffnen

#### Anlegen eines neuen Projekts



- Doppelklicken Sie auf die Schaltfläche "Programming Tool PLC828".

oder

- Wählen Sie im Windows-Startmenü den Befehl "Programming Tool PLC828" > "Programming Tool PLC828".

Das Programming Tool PLC828 wird gestartet und ein neues Projekt geöffnet.

#### Öffnen eines vorhandenen Projekts

- Wählen Sie im Programming Tool PLC828 im Menü "Datei" einen der folgenden Befehle:
  - "Öffnen"

Navigieren Sie zu einem vorhandenen Projekt und öffnen Sie es.

- "Dateiname"

Haben Sie erst vor kurzem in einem Projekt gearbeitet, ist dieses Projekt im Menü "Datei" aufgelistet und Sie können es direkt öffnen, ohne es vorher im Dialogfeld auswählen zu müssen.

- Sie können auch im Windows Explorer zum gewünschten Verzeichnis navigieren und das Projekt direkt von dort öffnen, ohne vorher das Programming Tool PLC 828 zu starten. Ihr Projekt ist in einer einzigen Datei mit der Erweiterung **\*.ptp** enthalten.

---

**Hinweis**

Wenn Sie ein Projekt angelegt haben, können Sie damit beginnen, Ihr Programm zu schreiben. Sie sollten zuvor jedoch folgende Aufgaben ausführen:

- **Bereichsprüfung nach dem Zielsystem**

Sie können den Typ des Zielsystems auswählen, bevor Sie Ihr Programm schreiben, damit das Programming Tool PLC828 den Bereich der Parameter entsprechend dem Zielsystem prüfen kann. (Wenn Sie für Ihr Projekt einen CPU-Typ ausgewählt haben, werden die Operationen, die für Ihr Zielsystem nicht verwendet werden können, im Operationsbaum mit einem roten x gekennzeichnet.)

- **Einrichten der Arbeitsumgebung**

Sie können Ihre Arbeitsumgebung auf verschiedene Arten einrichten ("Windows look and feel"). Näheres finden Sie in der Online-Hilfe unter "Einrichten der Darstellung vom Programming Tool".

---

### 13.3.1.3 Programmorganisation durch das Programming Tool

Das Programming Tool PLC828 organisiert Ihr Programm im Programm-Editor in einzelnen Registern pro POE. Standardmäßig werden beim Anlegen eines neuen Projekts das Hauptprogramm MAIN und ein Unterprogramm SBR\_0 angelegt. MAIN ist immer im ersten Register, es folgen alle Unterprogramme und Interruptprogramme, die Sie erstellt haben.



#### MAIN

MAIN kann nicht umbenannt werden und ist fest der Datenklasse "Manufacturer" zugeordnet.

## Unterprogramme

Unterprogramme sind nützlich, wenn Sie eine Funktion mehrfach ausführen möchten. Damit Sie die Logik nicht an jeder Stelle, an der Sie die Funktion ausführen möchten, ins Hauptprogramm aufnehmen müssen, schreiben Sie die Logik einmal in ein Unterprogramm und rufen dieses Unterprogramm so oft auf, wie es während der Bearbeitung des Hauptprogramms erforderlich ist. Unterprogramme können aussagekräftig benannt werden, auch sie sind der Datenklasse "Manufacturer" zugeordnet.

Vorteile:

- Der funktionelle Ablauf im Hauptprogramm wird sehr übersichtlich.
- Unterprogramme sind gut portierbar. So können Sie leicht eine Funktion ausgrenzen und sie ohne viel Aufwand aus anderen Programmen mit anderen Parameterwerten aufrufen.

---

### Hinweis

Die Verwendung von globalen Variablen schränkt die Portabilität von Unterprogrammen ein, weil die Zuordnung von Adressen im Variablenspeicher des einen Programms mit der Zuordnung der Adressen im anderen Programm in Konflikt stehen kann. Unterprogramme, die für alle Adresszuordnungen auf lokale Variablen zugreifen, sind dagegen sehr gut portierbar, weil keine Gefahr von Adresskonflikten besteht.

---

## Interruptprogramme

Interruptprogramme werden verwendet, um spezielle Prozessbedingungen oder -forderungen beherrschen zu können. Die PLC828 unterscheidet folgende:

Bedingung	Programm	Beschreibung
Kürzeste Reaktionszeit: Ausführung im <b>Servotakt</b>	INT_0	Für sehr schnelle Reaktionen auf Ereignisse, welche dies unbedingt erfordern. In INT_0 können die Direktzugriffsbefehle, die entweder auf ein internes, im Servotakt aktualisiertes E/A-Abbild oder nur auf die Hardware der Onboard-EA zugreifen, sinnvoll genutzt werden. Datenklasse: "Manufacturer" <b>Hinweis:</b> Die Größe des Bausteins INT_0 ist auf 500 Anweisungen begrenzt.
Ausführung <b>vor</b> MAIN, nach dem Lesen der Eingänge	INT_100	Besonders zum Rangieren von Daten, auf die im weiteren Zyklus lesend zugegriffen wird. Datenklasse: "Individual"
Ausführung <b>nach</b> MAIN, vor dem Schreiben der Ausgänge	INT_101	Besonders zum Rangieren von Daten, die vorher im PLC-Zyklus geschrieben worden sind, aber auf andere oder weitere Ausgänge ausgegeben werden sollen. Datenklasse: "Individual"

Interruptprogramme können nicht umbenannt werden, aber sie können Datenklassen zugeordnet werden.

Das Einfügen der Interruptprogramme in das Projekt erfolgt durch Doppelklick auf den entsprechenden Namen im Operationsbaum: "Bibliotheken" > "Interruptprogramme".

Die PLC-Firmware sorgt dann automatisch für den Aufruf.



### Beenden von Programmorganisationseinheiten

Durch die beschriebene Programmorganisation (jede POE belegt ein eigenes Register) ist es eindeutig, wo MAIN und die einzelnen Unter- und Interruptprogramme enden, sie bedürfen daher keiner besonderen Endekennung.

### Siehe auch

Adressierung (Seite 1204)

Speicherart (Seite 1201)

### 13.3.1.4 Schnelle Onboard-Ein- und Ausgänge

Die PLC der PPU-Baugruppe kann 8 schnelle Eingänge und 6 schnelle Ausgänge nutzen, die "onboard" über die Steckverbinder X242 (DIN1-DIN4, DOUT1-DOUT4) und X252 (DIN9-DIN12, DOUT9-DOUT10) bereitgestellt werden. Sie benötigen somit keine Peripheriemodule und können daher besonders schnell verarbeitet werden. Standardmäßig sind diese Ein-/Ausgänge der NC zugeordnet. Die Zuordnung zur NC bzw. PLC erfolgt über folgende Maschinendaten:

MD10366 \$MN\_HW\_ASSIGN\_DIG\_FASTIN[<n>  
(HW-Zuordnung für externe digitale Eingänge)

MD10368 \$MN\_HW\_ASSIGN\_DIG\_FASTOUT[<n>  
(HW-Zuordnung für externe digitale Ausgänge)

<n>: Index für Adressierung der externen digitalen E/A-Bytes (0 bis 3) bzw. der externen analogen Ein-/Ausgänge (0 bis 7)

#### NC/PLC-Zuordnung

Steckverbinder	Maschinendaten	E/A-Zuordnung
X242	MD10366[0] = MD10368[0] = 10101 <sub>H</sub>	DIN1 - DIN4 und DOUT1 - DOUT4 an NC
	MD10366[0] = MD10368[0] = 10001 <sub>H</sub>	DIN1 - DIN4 und DOUT1 - DOUT4 an PLC
X252	MD10366[1] = MD10368[1] = 10101 <sub>H</sub>	DIN9 - DIN12 und DOUT9 - DOUT10 an NC
	MD10366[1] = MD10368[1] = 10001 <sub>H</sub>	DIN9 - DIN12 und DOUT9 - DOUT10 an PLC

**Adressierung**

<b>Adressierung:</b>	<b>Adresse</b>
Onboard-Eingänge:	I256.0 ... I256.3
Onboard-Ausgänge:	Q256.0 ... Q256.3

**Zugriff**

Diesen Ein-/Ausgängen ist kein Abbildspeicher zugeordnet, Schreiben und Lesen erfolgt direkt auf oder von der Hardware. Für schnellstmögliche Zugriffe (servotaktsynchron) wird die Verwendung der Direktoperationsbefehle im Interruptprogramm **INT0** empfohlen:

<b>Befehl</b>	<b>Symbol</b>
direkter Schließer	-  I  -
direkter Öffner	-  / I  -
Bitwert direkt setzen	-(SI)-
Bitwert direkt rücksetzen	-(RI)-
Bitwert direkt zuweisen	-(I)-

**Reaktionszeiten**

Abhängig von Lagereglertakt, Ausführungsort und verwendete digitale Ein-/Ausgänge ergeben sich folgende Reaktionszeiten:

<b>Lagereglertakt</b>		<b>1,5 ms</b>	<b>3 ms</b>
<b>Verarbeitung über:</b>		<b>Reaktionszeit <sup>1)</sup></b>	
	Unterprogramm SBRx im zyklischen Betrieb (MAIN, OB1), Digitale Ein-/Ausgänge: Peripheriemodul PP 72/48...	12,5 ms	14 ms
	Interruptprogramm INT0 (servosynchron) ), Digitale Ein-/Ausgänge: Onboard Ein-/Ausgänge der PPU	3 ms	4,5 ms
<sup>1)</sup> Signal an Eingangsklemme → Verarbeitung im PLC-Programm → Signal an Ausgangsklemme			



## 13.3.2 Speicher des Zielsystems

### 13.3.2.1 Speicherart

Art	Beschreibung	Zugriff im Bitformat	Zugriff im Byteformat	Zugriff im Wortformat	Zugriff im Doppelwortformat	Kann remanent sein
I	Digitale Eingänge und Prozessabbild der Eingänge	lesen / schreiben	lesen / schreiben	lesen / schreiben	lesen / schreiben	nein
Q	Digitale Ausgänge und Prozessabbild der Ausgänge	lesen / schreiben	lesen / schreiben	lesen / schreiben	lesen / schreiben	nein
M	Interne Merker	lesen / schreiben	lesen / schreiben	lesen / schreiben	lesen / schreiben	nein
SM	Sondermerker (SM0.0 - SM0.6 sind schreibgeschützt)	lesen / schreiben	lesen / schreiben	lesen / schreiben	lesen / schreiben	nein
V	Variablenspeicher	lesen / schreiben	lesen / schreiben	lesen / schreiben	lesen / schreiben	ja
T	Aktuelle Werte von Zeiten und Zeitbits	Zeitbit lesen / schreiben	nein	akt. Wert der Zeit lesen / schreiben	nein	nein
C	Aktuelle Werte der Zähler und Zählbits	Zählerbit lesen / schreiben	nein	akt. Wert des Zählers lesen / schreiben	nein	nein
AC	Akkumulatoren	nein	lesen / schreiben	lesen / schreiben	lesen / schreiben	nein
L	Lokaldatenspeicher	lesen / schreiben	lesen / schreiben	lesen / schreiben	lesen / schreiben	nein

### 13.3.2.2 Adressbereiche des Zielsystems

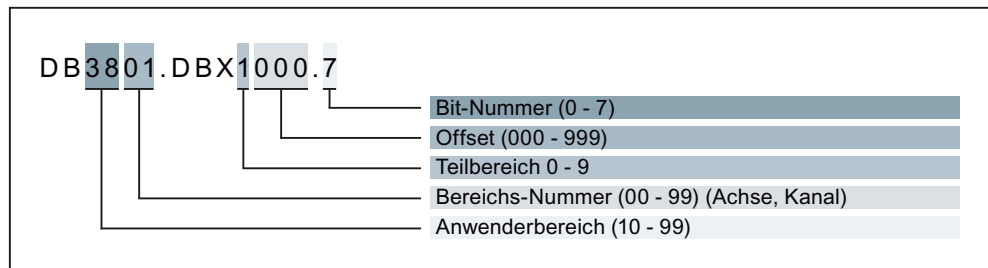
Unter Adressbereich einer Speicherart versteht man den durch kleinst- und größtmögliche Adressnummern aufgespannten adressierbaren Bereich dieser Art Speicher. Dabei hängen kleinst- und größtmögliche Adressnummer von der jeweiligen Speicherart ab.

Wenn Sie Operationen ausführen, die mit dem Zielsystem kommunizieren, erkennt das Programming Tool PLC828 die Variante Ihrer CPU. Beim Erstellen Ihres Programms müssen Sie darauf achten, dass Sie nur für diese CPU gültige Adressbereiche verwenden. Wenn Sie versuchen, ein Programm zu laden, das auf Adressbereiche zugreift, die für diese CPU ungültig sind, wird eine Fehlermeldung ausgegeben.

## Adressbereiche

Zugriffsart	Speicherart	Min. und max. Adressnummern
Bit (Byte.Bit)	I	0.0 - 255.7 <sup>1)</sup> 256.0 - 256.3 <sup>2)</sup>
	Q	0.0 - 255.7 <sup>1)</sup> 256.0 - 256.3 <sup>2)</sup>
	M	0.0 - 999.7
	SM	0.0 - 0.6
	T	0 - 15 (100 ms) 16 - 127 (10 ms)
	C	0 - 63
	L	0.0 - 59.7
Byte	IB	0 - 255 <sup>1)</sup>
	QB	0 - 255 <sup>1)</sup>
	MB	0 - 999
	SMB	0
	LB	0 - 59
	AC	0 - 3
Wort	IW	0 - 254 <sup>1)</sup>
	QW	0 - 254 <sup>1)</sup>
	MW	0 - 998
	T	0 - 15 (100 ms) 16 - 127 (10 ms)
	C	0 - 63
	LW	0 - 58
	AC	0 - 3
Doppelwort	ID	0 - 252 <sup>1)</sup>
	QD	0 - 252 <sup>1)</sup>
	MD	0 996
	LD	0 - 56
	AC	0 - 3
Anwendernahtstelle	DB	1000 - 7999 <sup>3)</sup>
Anwender-Datenbausteine	DB	9000 - 9063 <sup>3)</sup>
Spezielle Datenbausteine	DB	9900 - 9999 <sup>3)</sup>

- 1) Bereich der Abbild-Variablen! Die Zuordnung dieser Variablen zu den physischen Eingängen und Ausgängen ist der Systemübersicht im Inbetriebnahmehandbuch SINUMERIK 828D zu entnehmen.
- 2) Diese Adressen bedienen direkt (d. h. ohne Abbildspeicher) die 8 Onboard-Eingänge und die 6 Onboard-Ausgänge, wenn diese der PLC zugeordnet sind.
- 3) Aus Gründen der Vereinfachung sind in der Tabelle nur die DB-Nummern angegeben. Ihre Adressierung ist DB-strukturabhängig und erfolgt nach folgendem Schema:



Zugriff	Beispiel	Erklärung
Bit	DB3801.DBX1000.7	Bit 7 des Bytes mit Offset 0 im Teilbereich 1 für Achse 2 im Anwenderbereich 38
Byte	DB3801.DBB0	Byte mit Offset 0 im Teilbereich 0 für Achse 2 im Anwenderbereich 38
Word	DB4500.DBW2	Word mit Offset 2 im Teilbereich 0 im Bereich 0 im Anwenderbereich 45
Double Word	DB2500.DBD3004	Doppelwort mit Offset 4 im Teilbereich 3 im Bereich 0 im Anwenderbereich 25

Die Zugriffsart ist Bestandteil der Adressnotation und darf nicht mit Datentyp gleichgesetzt oder verwechselt werden (siehe "Datentypen (Seite 1207)").

#### Hinweis

Der erlaubte **Offset** einer Adresse ist abhängig vom Zugriff:

- Bit- oder Bytezugriff: jeder Offset erlaubt.  
Variablen der Größe Byte werden in einem DB nahtlos aneinandergelegt.
- Wortzugriff: Offset muss durch 2 teilbar sein.  
Variablen der Größe Word (2 Byte) werden immer auf geraden Offsets abgelegt.
- Doppelwortzugriff: Offset muss durch 4 teilbar sein.  
Variablen der Größe Double Word (4 Byte) werden immer auf durch 4 teilbaren Offsets abgelegt.

Siehe auch "Datentypen (Seite 1207)".

**Anwendernahtstelle** ist eine Schnittstelle aus Datenbausteinen, die von der Firmware auf dem Zielsystem angelegt werden. Sie dient dem Datenaustausch zwischen der PLC einerseits und NCK und HMI andererseits (→ Listenhandbuch SINUMERIK 828D).

Weil von der Firmware angelegt und daher zum System gehörend, brauchen diese DB weder aus der CPU noch in die CPU geladen werden.

**Anwenderdatenbausteine** werden ausschließlich vom Anwender angelegt. Wird beabsichtigt, auf Bausteine gleicher Struktur indirekt zuzugreifen, sollten diese nummernmäßig hintereinander liegen.

**Spezielle Datenbausteine** sind in ihrer Struktur vom System fest vorgegeben und im Programming Tool PLC828 in "Bibliotheken" zu finden. Ob sie allerdings in das Anwenderprogramm eingebunden, einer Datenklasse zugeordnet und in die CPU geladen werden, liegt allein in der konzeptionellen Verantwortung des Anwenders (wenn z. B. ohne Wartungsplaner gearbeitet wird, brauchen die diesbezüglichen DB nicht eingebunden werden).

### 13.3.2.3 Adressierung

#### Direkte Adressierung

Bei der direkten Adressierung müssen *Speicherart* sowie *Adressnummer* angegeben werden (siehe auch "Adressbereiche des Zielsystems (Seite 1201)").

Auf den CPU-Speicher (V, I, Q, M und SM) kann man bitweise sowie im Byte-, Wort- und Doppelwortformat zugreifen. Die direkte Adresse setzt sich zusammen aus Speicherart und einer gültigen Adressnummer.

Zum Zugreifen auf ein Bit in einem Speicherbereich geben Sie die Byteadresse und die Nummer des Bits an. Trennen Sie das Byte und das Bit durch einen Dezimalpunkt.

Beispiele:

DB9900.DBX20.0	Bit 0 im Byte 20 des DB9900
MB21,	Merkerbyte 21
QD16,	Ausgangsdoppelwort 16
I1.7,	Bit 7 im Eingangsbyte 1

#### Indirekte Adressierung

Die indirekte Adressierung kann nur bei Datenbausteinen der gleichen Struktur (des gleichen Typs) angewendet werden. Sie dient dazu, dass bei Zugriffen auf Datenbausteinen die Nummer des Datenbausteins variabel adressiert werden kann. Die Nummer des Datenbausteins muss in einem der Akkumulatoren AC0 ... AC3 stehen (siehe "Datenbausteine (Seite 1211)").

## Absolute und symbolische Adressierung

Sie können die Operanden in den Anweisungen in Ihrem Programm absolut oder symbolisch angeben.

Eine *absolute Adresse* gibt Speicherart sowie Adressnummer an.

Ein *symbolische Adresse* (kurz: Symbol) gibt die Adresse mittels eines Namens (einer Kombination aus alphanumerischen Zeichen) an.

*Globale Symbole* werden in der Symboltabelle ihren absoluten Adresswerten zugeordnet und sind projektweit gültig (global). Diese Zuordnung kann jederzeit erfolgen.

*Lokale Symbole* werden in der lokalen Variablen-tabelle des jeweiligen Programms zugeordnet und sind nur in diesem Programm gültig (lokal).

Beispiele für die Anzeige von Adressen im Programm-Editor:

I0.0	Eine absolute Adresse gibt Speicherart und Adressnummer an.
#Input1	Das Zeichen # ist einem lokalen Symbol vorangestellt.
INPUT1	Globales Symbol
???	Rote Fragezeichen kennzeichnen eine noch nicht definierte Adresse (Sie müssen die Adresse angeben, bevor Sie das Programm übersetzen können).

Beide Adressierungsarten – absolute und symbolische – sind an die jeweilige Ansicht gekoppelt:

Menü "Ansicht" > "Symbolische Adressierung (Strg +Y)"

Diese Einstellung sollte immer mit Menü "Ansicht > "Symboltabelle (Strg +T)" eingestellt werden.

Empfehlung: Entscheiden Sie sich für eine der beiden Adressierungsarten und behalten Sie diese bei.

## Globaler und lokaler Geltungsbereich

Symbolische Adressen, die in der Symboltabelle zugeordnet wurden, haben globalen Geltungsbereich. Symbolische Adressen, die in einer lokalen Variablen-tabelle zugeordnet wurden, haben lokalen Geltungsbereich.

## Lokale Variablen

Lokale Variablen werden in der lokalen Variablen-tabelle der jeweiligen POE zugeordnet und sind vom Geltungsbereich auf die POE beschränkt, in der sie angelegt wurden. Jede Programmorganisationseinheit verfügt über eine eigene lokale Variablen-tabelle.

Beispiel:

Sie definieren eine Variable mit dem Namen INPUT1 in der lokalen Variablen-tabelle eines Unterprogramms mit dem Namen SBR1.

Wenn Sie aus SBR1 auf INPUT1 verweisen, erkennt der Programm-Editor INPUT1 als lokale Variable von SBR1.

Wenn Sie jedoch an einer anderen Stelle im Programm auf INPUT1 verweisen (z. B. in MAIN oder in einem zweiten Unterprogramm), erkennt der Programm-Editor INPUT1 nicht als lokale Variable und behandelt INPUT1 als nicht definiertes globales Symbol.

---

### Hinweis

#### Namensgebung bei lokalen und globalen Symbolen

Wenn Sie den gleichen Namen für eine Adresse auf lokaler und auf globaler Ebene verwenden, hat die lokale Verwendung Vorrang. Das heißt, wenn der Programm-Editor eine Definition für den Namen in der lokalen Variablen-tabelle eines bestimmten Programmbausteins findet, wird diese Definition verwendet. Wird keine Definition gefunden, prüft der Programm-Editor die Symbol-tabelle.

Beispiel:

Sie definieren das globale Symbol "PumpeEin". Sie definieren "PumpeEin" auch als lokale Variable in SBR2, jedoch nicht in SBR1.

Wird das Programm übersetzt, wird die lokale Definition für "PumpeEin" in SBR2 verwendet. Die globale Definition wird für "PumpeEin" in SBR1 verwendet.

---

### Hinweis

#### Verwendung von lokalen und globalen Symbolen

Lokale Variable verwenden den temporären lokalen Speicher des Zielsystems. Unterprogramme, die nur lokale Variablen und Übergabeparameter verwenden, sind gut portierbar und können flexibel eingesetzt werden.

Wenn Sie einen Parameter in mehreren Programm-Organisationseinheiten verwenden möchten, ist es sinnvoll, diesen Parameter als globales Symbol in der Symbol-tabelle und nicht in der lokalen Variablen-tabelle zu definieren, weil Sie den Parameter dann in jede lokale Variablen-tabelle der einzelnen POE aufnehmen müssten.

---

### Hinweis

#### Initialisierung lokaler Variablen

Da lokale Variablen temporären Speicher belegen, müssen Sie die lokalen Variablen in der POE jedes Mal, wenn die POE aufgerufen wird, initialisieren. Sie können **nicht** davon ausgehen, dass eine lokale Variable einen Datenwert von einem POE-Aufruf zum nächsten behält.

---

### 13.3.2.4 Datentypen

Bei der Definition von Symbolen in der globalen Symboltabelle braucht ein Datentyp nicht explizit angegeben zu werden, da er implizit durch das dem Symbol zugeordnete Datum gegeben ist.

Wenn Sie in der lokalen Variablen-tabelle Werte zuordnen, müssen Sie für jede lokale Variable einen Datentyp angeben.

Indem Sie für einen Wert explizit einen Datentyp angeben, geben Sie dem Programming Tool PLC828 eindeutige Anweisungen, wie viel Speicher für den Wert zugeordnet werden muss (z. B. kann der Wert 100 als BYTE, WORD oder DWORD gespeichert werden) und wie der Wert dargestellt werden soll (z. B. soll 0 als BOOL oder als numerischer Wert interpretiert werden?).

Die Operationen und parametrisierten Unterprogramme werden über eine genaue Definition erkannt. Diese Definition wird auch als Signatur bezeichnet. Bei allen genormten Operationen werden die für die Operanden der Operation zulässigen Datentypen in der Signatur angegeben. Bei parametrisierten Unterprogrammen wird die Signatur des Unterprogramms vom Anwender über die lokale Variablen-tabelle erstellt.

### Datentypprüfung

Das Programming Tool PLC828 bietet eine einfache Datentypprüfung. Wird ein Datentyp für eine lokale oder globale Variable angegeben, prüft die Software, dass der Datentyp des Operanden der Signatur der Operation entspricht.

Elementare Datentypen	Beschreibung	Speicherbereich
BOOL (Bit)	Boolesch	0 ... 1
BYTE	Byte ohne Vorzeichen	0 ... 255
WORD	Ganze Zahl (16 Bit)	-32768 ... +32767
DWORD (Double Word)	Ganze Zahl (32 Bit)	-2147483648 ... +2147483647
REAL	32-Bit Gleitpunkt	+/- 10 <sup>-37</sup> ... +/- 10 <sup>+38</sup>

Komplexe Datentypen	Beschreibung	Speicherbereich
TON	Einschaltverzögerung	100 ms T0 ... T15 10 ms ab T16
TOF	Ausschaltverzögerung	100 ms T0 ... T15 10 ms ab T16
TONR	Einschaltverzögerung, speichernd	100 ms T0 ... T15 10 ms ab T16
CTU	Vorwärtszähler	C0 ... C63
CTD	Rückwärtszähler	C0 ... C63
CTUD	Vorwärts-/Rückwärtszähler	C0 ... C63

Das Programming Tool PLC828 verfügt über zwei Stufen der Datentypprüfung:

1. Einfache Datentypprüfung

Bei der einfachen Datentypprüfung werden automatisch, wenn einem Symbol oder einer Variablen ein Datentyp zugeordnet wird, alle Datentypen zugeordnet, die der Bitgröße des anwenderdefinierten Datentyps entsprechen. Wenn Sie beispielsweise DINT als Datentyp angeben, wird der lokalen Variablen automatisch auch der Datentyp DWORD zugeordnet, weil beide Datentypen 32-Bit-Typen sind. Der Datentyp REAL wird nicht automatisch zugewiesen, obwohl es sich auch um einen 32-Bit-Datentyp handelt. Der Datentyp REAL ist so definiert, dass er keine äquivalenten Datentypen besitzt: er ist immer eindeutig. Die einfache Datentypprüfung wird nur bei der Verwendung von lokalen Variablen durchgeführt.

Anwenderdefinierte Datentypen	Äquivalenter Datentyp
BOOL	BOOL
BYTE	BYTE
WORD	WORD, INT
INT	WORD, INT
DWORD	DWORD, INT
DINT	DWORD, DINT
REAL	REAL

2. Keine Datenprüfung

Dieser Modus ist nur für globale Variable verfügbar, zu denen keine Datentypen angegeben werden können. Ist keine Datentypprüfung aktiv, werden dem Symbol automatisch alle Datentypen gleicher Größe zugewiesen.

Beispiel:

Ein Symbol, dem die Adresse DB1400.DBD4 zugeordnet ist, werden die folgenden Datentypen automatisch von der Programmiersoftware zugewiesen: DWORD, DINT und REAL.

Anwenderdefinierte Adresse	Zugeordneter äquivalenter Datentyp
DB1400.DBX0.0	BOOL
DB1400.DBB0	BYTE
DB1400.DBW2	WORD, INT
DB1400.DBD4	DWORD, DINT, REAL



## Vorteile der Datentypprüfung

Die Datentypprüfung hilft Ihnen, weit verbreitete Programmierfehler zu vermeiden. Unterstützt eine Operation Zahlen mit Vorzeichen, kennzeichnet das Programming Tool PLC828 die Verwendung von vorzeichenlosen Zahlen in Operanden von Operationen.

Beispiel:

Der Vergleich < I ist eine Operation mit Vorzeichen. -1 ist kleiner als 0 für Operanden mit Vorzeichen. Unterstützt die Operation < I jedoch vorzeichenlose Datentypen, muss durch die Programmierung sichergestellt werden, dass Folgendes nicht auftritt: Zur Laufzeit des Programms ist ein vorzeichenloser Wert von 40,000 tatsächlich kleiner als 0 für die Operation < I. Ist nicht sichergestellt, dass die vorzeichenlosen Zahlen für Operationen mit Vorzeichen die positiven und negativen Grenzwerte nicht überschreiten, kann es zu unvorhersehbaren Ereignissen in Ihrem Programm oder in der Funktionsweise der Steuerung kommen.

## Arbeiten mit Operationen zum Umwandeln des Datentyps

Umwandlungsoperationen wandeln einen Datentyp in einen anderen um. Das Programming Tool PLC828 unterstützt die folgenden Umwandlungsoperationen zum Übertragen von Werten zwischen den einfachen Datentypen.

Umwandlung der Datentypen	Umwandlungsoperationen	Vollständige Datentypprüfung Zulässige Operanden	Datenprüfung Zulässige Operanden
INT in BCD	I_BCD	IN: INT OUT: INT	IN: WORD, INT OUT: WORD, INT
BCD in INT	BCD_I	IN: INT OUT: INT	IN: WORD, INT OUT: WORD, INT
DINT in REAL	DI_R	IN: DINT OUT: REAL	IN: DWORD, DINT OUT: REAL
REAL in DINT (ROUND)	TRUNC	IN: REAL OUT: DINT	IN: REAL OUT: DWORD, DINT

### 13.3.2.5 Konstanten

#### Bereiche der Konstanten

Größe der Daten	Bereich ohne Vorzeichen		Bereich mit Vorzeichen	
	Dezimal:	Hexadezimal:	Dezimal:	Hexadezimal:
B (Byte)	0 ... 255	0 ... FF	-128 ... +127	80 ... 7F
W (Wort)	0 ... 65535	0 ... FFFF	-32768 ... +32767	8000 ... 7FFF
D (Doppelwort)	0 ... 4294967295	0 ... FFFF FFFF	-2147483648 ... +2147483647	8000 0000 ... 7FFF FFFF
Größe	Dezimale Realzahl (positiv)		Dezimale Realzahl (negativ)	
D (Doppelwort)	+1.175495E-38 ... +3.402823E+38		-1.175495E-38 ... -3.402823E+38	

#### Formatkennzeichen von Konstanten

Ihr Programm kann in vielen Operationen Konstanten im Byte-, Wort- oder Doppelwortformat einsetzen. Formatkennzeichen geben an, wie ein konstanter Wert angezeigt wird (im Binär-, Dezimal-, Hexadezimal- oder ASCII-Format).

Konstanten im Programm werden als Dezimalzahlen betrachtet, sofern kein Formatkennzeichen angegeben ist:

- 2# für Dualzahlen
- 16# für Hexadezimalzahlen

Beispiel für binäre Konstanten:

Beispiel	Numerische Basis	Trennzeichen	Konstante
2#1101	2	#	1101

Beispiel für hexadezimale Konstanten:

Beispiel	Numerische Basis	Trennzeichen	Konstante
16#3FB3	16	#	3FB3

---

#### Hinweis

Unterstriche können die Lesbarkeit in importierten ASCII-Dateien erhöhen.

Beispiel: 16#A\_B\_C\_D

---

### 13.3.2.6 Datenbausteine

#### Datenbausteinrentypen

Es sind drei Typen von Datenbausteinen zu unterscheiden (siehe auch "Adressbereiche des Zielsystems (Seite 1201)"):

- Datenbausteine der Anwendernahtstelle

Sie dienen der Kommunikation aus dem Anwenderprogramm mit den einzelnen Steuerungskomponenten und werden vom System angelegt. Der Anwender bedient über Lese- und Schreibzugriffe die Nahtstelle.

---

#### Hinweis

Nach Notationsform und Adressierung zwar Datenbausteine, unterscheiden diese DBs jedoch nicht zwischen Anfangs- und Aktualwerten, und sie sind auch nicht Bestandteil des PLC-Anwenderprojekts. Die Erläuterungen des Abschnitts "Eigenschaften von Datenbausteinen" gelten für diese DBs nicht.

---

- Spezielle Datenbausteine

Sie dienen speziellen Aufgaben (z. B. Werkzeugverwaltung, Serviceplaner) und sind vorgefertigt im Programming Tool PLC828 verfügbar. Soll die jeweilige Funktionalität genutzt werden, müssen die entsprechenden DBs in das Anwenderprogramm eingebunden werden.

- Anwenderdatenbausteine

Diese werden vom Anwender in ihrer Struktur festgelegt und in das Anwenderprogramm eingebunden.

#### Eigenschaften von Datenbausteinen

Ein Datenbaustein ist ein Baustein für Daten (Anfangswerte, Aktualwerte) und Kommentare mit folgenden Eigenschaften:

- Die Daten werden genau in der vom Anwender vorgegebenen Reihenfolge gespeichert. Damit ist sozusagen die innere Struktur des Datenbausteins definiert und wenn mehrere Datenbausteine mit der gleichen inneren Struktur (d. h. des gleichen Typs) angelegt werden, steht ein bestimmtes Datum immer auf seinem bestimmten Platz. Dieser Platz wird Offset genannt und ist die relative Länge in Byte vom Anfang des Datenbausteins (DB) bis zum eigentlichen Datum.
- Den Daten können Anfangswerte zugewiesen werden. Mit diesen Anfangswerten werden beim erstmaligen Laden in die CPU die Istwerte (Aktualwerten) des DB initialisiert. Die Anfangswerte werden auch in der CPU gespeichert.
- Die Istwerte (Aktualwerte) der Daten können online aus der Steuerung gelesen, verändert und auch mit dem Projekt gespeichert werden.

Die Strukturen der Datenbausteine sind wie die POE Bestandteil des Projekts. Sie werden mit dem Projekt übersetzt, gespeichert, importiert oder exportiert. Außerdem werden sie mit dem Projekt in das bzw. aus dem Zielsystem geladen. Die Datenbausteine selbst enthalten nur die Aktualwerte. Diese können unabhängig vom Projekt in das bzw. aus dem Zielsystem geladen werden. Dabei ist es zwingend notwendig, dass die Struktur des zu ladenden Datenbausteins im Zielsystem mit dem des im Programming Tool PLC828 geöffneten Projekts übereinstimmt. Datenbausteine werden wie POE in einer eigenen Symboltabelle aufgelistet.

Wenn Sie für Ihr Projekt den CPU-Typ wechseln, gehen die vorhandenen Datenbausteine nicht verloren. Wählen Sie allerdings eine CPU, in der Datenbausteine nicht verfügbar sind, müssen Sie beachten, dass die Variablen aus dem Datenbaustein nun in ihrer absoluten Adresse dargestellt werden (z. B. DB9000.DBB0). Ob diese Adresse gültig ist, wird erst beim Starten des Zielsystems geprüft, und das Programm wird gegebenenfalls nicht ausgeführt.

Sie können Ihre Datenbausteine umbenennen und die Eigenschaften ändern, indem Sie mit der rechten Maustaste im Operationsbaum auf den entsprechenden Datenbaustein klicken. Wählen Sie "Eigenschaften". Es öffnet sich das Dialogfenster "Eigenschaften Datenbaustein". Hier können Sie Name, Bausteinnummer und Datenklasse ändern sowie einen Autor und einen Kommentar hinzufügen.

---

#### Hinweis

Wenn Sie nur den Namen des Datenbausteins ändern wollen, klicken sie im Operationsbaum auf das entsprechende Objekt mit der rechten Maustaste und wählen Sie "Umbenennen".

Sie können Ihren Datenbaustein auch bequem in einem anderen Programm (z. B. Microsoft Excel) bearbeiten.

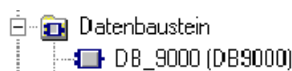
---

### Editieren von Datenbausteinen im Programming Tool PLC828



Zum Editieren von Datenbausteinen im Programming-Tool PLC828:

- Klicken Sie in der Navigationsleiste auf die Schaltfläche "Datenbaustein".
- Wählen Sie den Menübefehl "Ansicht" > "Datenbaustein".
- Doppelklicken Sie im Operationsbaum auf die Schaltfläche "Datenbaustein" und dann auf den zu bearbeitenden Datenbaustein:



## Zuordnen von Adressen und Anfangswerten im Datenbaustein

Die Struktur des DB (die Reihenfolge der einzelnen DB-Variablen) legen Sie in der Deklarationssicht fest. Den Variablen des Datenbausteins können Anfangswerte zugewiesen werden, die mit der Datenklasse bzw. dem Projekt gespeichert werden und mit denen beim erstmaligen Laden in die CPU die Aktualwerte des DBs initialisiert werden.

Wenn Sie eine Variable definieren, legen Sie deren Namen und Datentyp fest. Der Anfangswert ist standardmäßig auf Null/OFF gesetzt, Sie können ihn aber auch verändern. Kommentare können Sie optional einfügen. Das Programming Tool PLC828 vergibt automatisch die Adresse. Jede Adresse wird nach der Größe des Datentyps ausgerichtet, dabei können Lücken entstehen.

Beispiel:

	Adresse	Name	Datentyp	Format	Anfangswert	
1	0.0	dwDatum1	DWORD	Hexadezimal	16#00000000	K1
2	4.0	wDatum2	WORD	Ohne Vorzeichen	0	K2
3	6.0	bDatum3	BYTE	Ohne Vorzeichen	0	K3
4	7.0	boDatum4	BOOL	Bit	OFF	K4
5	7.1	boDatum5	BOOL	Bit	OFF	K5
6	8.0	rDatum6	REAL	Gleitpunkt	0.5	K6
7	12.0	iDatum7	INT	Mit Vorzeichen	+0	K7

DB\_9000 DB\_9001 **DB\_9002** DB\_9003 TM\_CTS TM\_VTS TM\_A

Die maximale Größe eines Datenbausteins ist beschränkt (512 Byte), wird sie überschritten, markiert das Programming Tool PLC828 die Adressen der überzähligen Variablen mit einer roten Wellenlinie. Beim Übersetzen gibt das Programming Tool PLC828 einen Fehler aus.

## Zuweisen von Aktualwerten

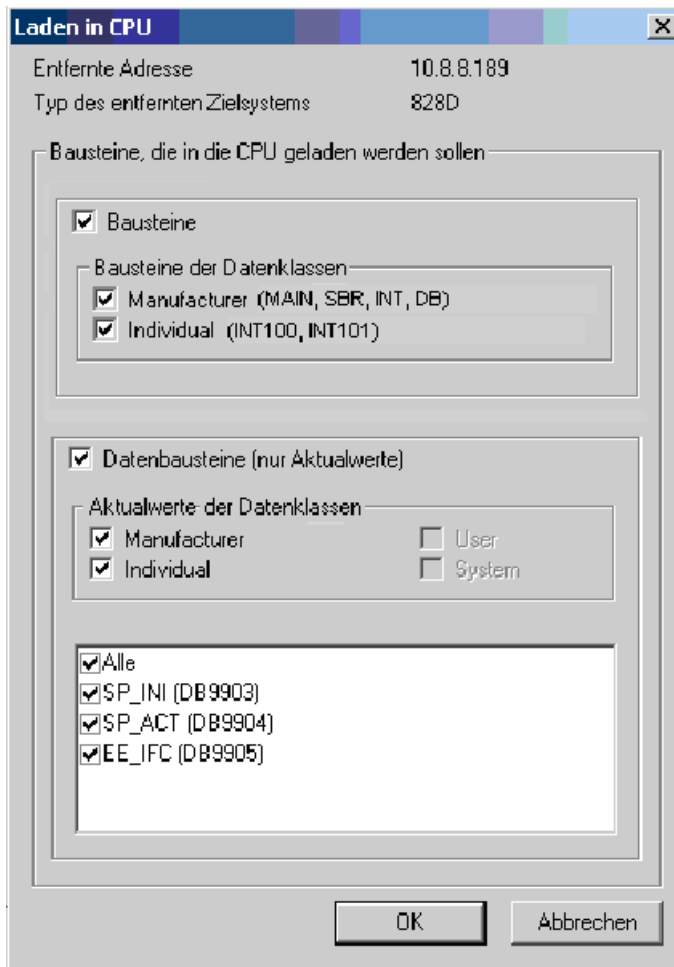
Aktualwerte werden mit ihrer Datenklasse (Manufacturer, Individual oder User) gespeichert. Sie sind nur in der Datensicht eingblendet und änderbar, wohingegen die Struktur der Variablen (Name, Datentyp, Anfangswert und Kommentar) schreibgeschützt ist.

Nach dem Laden der Aktualwerte des Datenbausteins (nicht des gesamten Projekts) werden diese Werte im Zielsystem wirksam.

Beispiel:

	Adresse	Name	Datentyp	Format	Anfangswert	Aktualwert	
1	0.0	dwDatum1	DWORD	Hexadezimal	16#00000000	16#01234567	K1
2	4.0	wDatum2	WORD	Ohne Vorzeichen	0	0	K2
3	6.0	bDatum3	BYTE	Ohne Vorzeichen	0	0	K3
4	7.0	boDatum4	BOOL	Bit	OFF	OFF	K4
5	7.1	boDatum5	BOOL	Bit	OFF	OFF	K5
6	8.0	rDatum6	REAL	Gleitpunkt	0.5	0.5	K6
7	12.0	iDatum7	INT	Mit Vorzeichen	+0	+0	K7

DB\_9000 DB\_9001 **DB\_9002** DB\_9003 TM\_CTS TM\_VTS TM\_A



Dabei muss die Struktur des Datenbausteins im Projekt mit der Struktur des Datenbausteins im Zielsystem übereinstimmen. Haben Sie die Struktur des Datenbausteins verändert oder den DB neu angelegt, muss das gesamte Projekt neu geladen werden.

Sie können auf folgende Arten zur Datensicht gelangen:



- Klicken Sie in der Standard-Funktionsleiste auf das Symbol "Datenbaustein-Werte anzeigen".

oder

- Wählen Sie den Menübefehl "Ansicht" > "Datenbaustein-Werte anzeigen".

## Übernehmen der Anfangswerte

Beim Übernehmen der Anfangswerte werden alle Aktualwerte eines Datenbausteins überschrieben. Sollten Sie bei einer Variablen keinen Anfangswert gewählt haben, so setzt das Programming Tool PLC828 auf Null/OFF. Andere Datenbausteine werden dabei nicht verändert.

Um alle Aktualwerte im aktuellen Datenbaustein auf die Anfangswerte zurückzusetzen:

- Klicken Sie in der Tabelle auf die rechte Maustaste und wählen Sie im Kontextmenü "Anfangswerte übernehmen".

oder

- Wählen Sie den Menübefehl "Bearbeiten" > "Anfangswerte übernehmen".

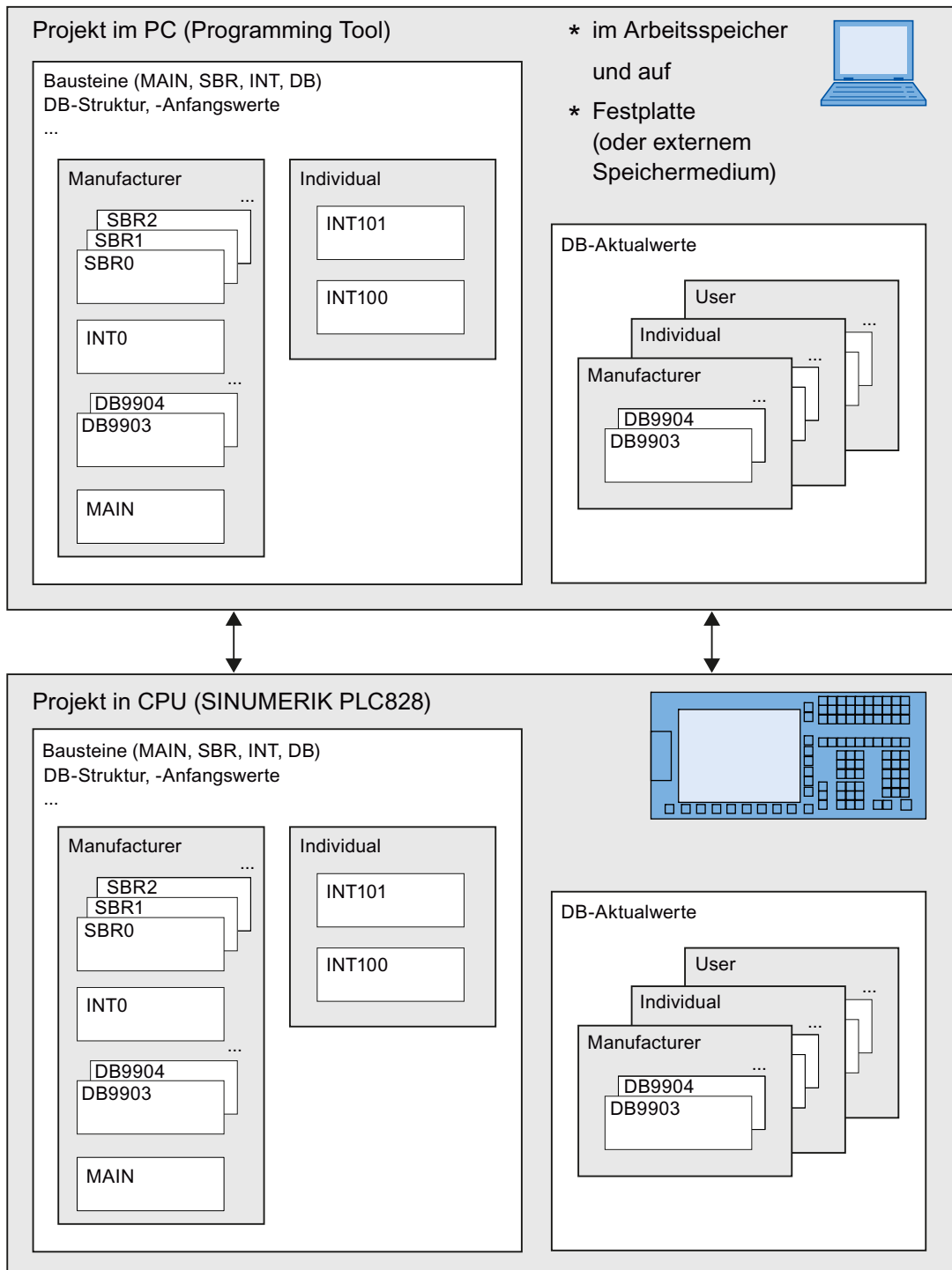
## Laden und Speichern der Datenbausteine

**Speichern** ist der Vorgang des Schreibens der DB (je nach Wahl Struktur + Anfangswerte + weitere Bausteine und Tabellen oder Aktualwerte oder beide gemeinsam) vom PC-Arbeitsspeicher auf Festplatte oder ein anderes externes Speichermedium. Diese Werte stehen nach dem Öffnen des Projekts wieder zur Verfügung.

**Laden in CPU** ist der Vorgang des Schreibens der DB (je nach Wahl Struktur + Anfangswerte + weitere Bausteine und Tabellen oder Aktualwerte oder beide gemeinsam) vom PC-Arbeitsspeicher in die PLC828.

**Laden aus der CPU** ist der umgekehrte Vorgang.

Die Ladevorgänge sind dabei Datenklassen-selektiv.





### Direkte Adressierung bei Datenbausteinen

- Absolute Adresseingabe

Die direkte absolute Adresse der Variablen in ihrem Datenbaustein setzt sich bei der absoluten Eingabe aus der Nummer des Datenbausteins (z. B. DB9000) einem Punkt und der Adresse der Variablen zusammen.

Die Anwahl erfolgt durch Menübefehl "Ansicht" > "DB Adressdarstellung (Ctrl+B)":



Beispiel:

DB9000.DBB0

- Symbolische Adresseingabe

In der Symboltabelle können den Variablen der Datenbausteine Namen gegeben werden. Beim Verwenden der Variable reicht dann die Angabe dieses Namens aus der Symboltabelle.

Ist in der Symboltabelle der DB-Variablen kein Name zugeordnet, setzt sich die symbolische Adresse aus dem DB-Namen (z. B. myDB\_9000) und dem Namen der Variablen im Datenbaustein zusammen (z. B. VKorrektur1).

Adresse	Name der Variablen im Datenbaustein	Eintrag in der Symboltabelle	Absolute Darstellung	Symbolische Darstellung
DB9000.DBX0.0	VKorrektur1	FeedCorrAx1	DB9000.DBX0.0	FeedCorrAx1
DB9000.DBX0.0	VKorrektur1		DB9000.DBX0.0	myDB_9000.VKorrektur1

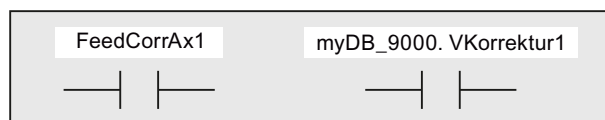


Bild 13-3 Symbolische Adressdarstellung

Das Umschalten zwischen absoluter und symbolischer Darstellung erfolgt mittels Menübefehl "Ansicht" > "Symbolische Adressierung (Ctrl+Y)".

### Indirekte Adressierung bei Datenbausteinen

Existieren in Ihrem Programm Datenbausteine gleicher Struktur, so lässt sich eventuell die Programmierung vereinfachen. Mit Hilfe der Akkumulatoren AC0 bis AC3 können die Datenbausteine indirekt adressiert werden. Sie teilen dem Programm durch den Akkumulator mit, welcher Datenbaustein behandelt werden soll. Der Wert im AC wird dann als Index behandelt.

So können Sie z. B. bei Achs-DB den Programmtext um einiges verringern, indem Sie nicht für jede Achse ein eigenes Programm schreiben, sondern über die verschiedenen Datenbausteine und den Index (AC) auf die entsprechende Achse zugreifen. Da der Wert im AC als Index behandelt wird, beginnt er für die erste Achse bei 0. Eine indirekte Adressierung ist nur über ACs möglich. Im Programmstatus des Programming Tools PLC828 ist die Anzeige des aktuellen Datenwerts nicht möglich. Die absolute Adresse kann nicht bestimmt werden. Die indirekte Adressierung kann nicht über V-Adressen erfolgen.

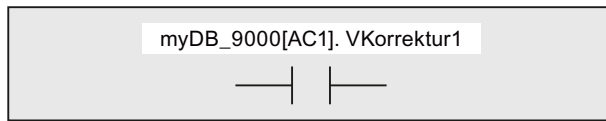


Bild 13-4 Indirekte Adressierung

**Absolute Eingabe**

DB3800[AC1].DBX2.1  
DB9000[AC0].DBW0

**Symbolische Eingabe**

ToAxis[AC1].ControlEnable  
Prototyp1[AC0].MyWord1

Zur Indexierung nicht erlaubt ist das Verwenden von Konstanten:

DB3800[1].DBX2.1                      ToAxis[5].ControlEnable

Die indirekte Adressierung mit V-Adressen ist ebenfalls nicht erlaubt:

V3800[5]0002.1                      V380[5]0002.1  
V3800[AC0]0002.1                      V38[AC0]0002.1

**Ausschneiden, Kopieren und Einfügen im Datenbaustein-Editor**

Sie können die Daten (Zeilen, Spalten, Zellen) Ihres Datenbausteins in der Microsoft-Zwischenablage speichern und somit in anderen Programmen bearbeiten. Sie können dabei wie folgt vorgehen:

- Im Kontextmenü (rechte Maustaste) über die Befehle "Ausschneiden"/"Kopieren" und "Einfügen".
- Über die Tastatur durch Betätigen der Tastenkombinationen <Strg+X>/<Strg+C> und <Strg+V>.
- Im Hauptmenü über die Befehle "Bearbeiten" > "Ausschneiden"/"Kopieren" und "Bearbeiten" > "Einfügen".

Ausschneiden:    Markierte Daten werden in die Zwischenablage kopiert und gelöscht.  
Kopieren:        Markierte Daten werden in die Zwischenablage kopiert und nicht gelöscht.  
Einfügen:        Sofern in der Zwischenablage vorhanden, werden Daten eingefügt.  
Löschen:         Markierter Datenbaustein/-teil wird gelöscht, er wird nicht in der Zwischenablage gespeichert.

Auf diese Weise können Sie Ihren Datenbaustein bequem z. B. in Microsoft Excel bearbeiten:

	A	B	C	D	E	F	G
1	DBB0	mybyte1	BYTE	Ohne Vorzeichen	0	0	Kommentar für
2	DBB1	mybyte2	BYTE	Ohne Vorzeichen	0	0	Kommentar für
3	DBW2	myword1	WORD	Ohne Vorzeichen	0	0	Kommentar für
4	DBW4	myint1	INT	Mit Vorzeichen	0	0	Kommentar für
5	DBX6.0	mybit1	BOOL	Bit	OFF	OFF	Kommentar für
6	DBX6.1	mybit2	BOOL	Bit	OFF	OFF	Kommentar für
7	DBX6.2	mybit3	BOOL	Bit	OFF	OFF	Kommentar für
8	DBX6.3	mybit4	BOOL	Bit	OFF	OFF	Kommentar für
9	DBX6.4	mybit5	BOOL	Bit	OFF	OFF	Kommentar für
10	DBX6.5	mybit6	BOOL	Bit	OFF	OFF	Kommentar für
11	DBX6.6	mybit7	BOOL	Bit	OFF	OFF	Kommentar für
12	DBX6.7	mybit8	BOOL	Bit	OFF	OFF	Kommentar für
13	DBD8	mydword1	DWORD	Ohne Vorzeichen	0	0	Kommentar für
14	DBD12	mydint1	DINT	Mit Vorzeichen	0	0	Kommentar für
15	DBD16	myreal1	REAL	Gleitpunkt	0.0	0.0	Kommentar für

### Verwenden spezieller Datenbausteine

Das Programming Tool PLC828 bietet Ihnen die Möglichkeit, spezielle Datenbausteine für den Werkzeugwechsel, Wartungsplaner und den Aggregatmanager zu verwenden. Diese finden Sie im Operationsbaum unter "Bibliotheken" > "spezielle Datenbausteine".

Diese Datenbausteine haben eine feste Struktur. Sie können mit einem Doppelklick oder über Kopieren und Einfügen (im Operationsbaum) verwendet werden.

Folgende spezielle Datenbausteine stehen zur Verfügung:

TM_CTS	(DB9900)	Konstante Transferschrift-Tabelle für Werkzeugwechsel
TM_VTS	(DB9901)	Variable Transferschrift-Tabelle für Werkzeugwechsel
TM_ACK	(DB9902)	Quittungsschritt-Tabelle für Werkzeugwechsel
SP_INI	(DB9903)	Startdaten für Wartungsplaner
SP_ACT	(DB9904)	Istdaten für Wartungsplaner
EE_IFC	(DB9905)	Schnittstelle für Aggregate-Manager

## Beheben von Fehlern

Fehler beim Eingeben (z. B. im KOP Editor oder der Symboltabelle) werden vom Programming Tool PLC828 markiert. So kann z. B. eine unzulässige Syntax oder das Verwenden ungültiger Werte Eintragsfehler hervorrufen. Sie müssen auftretende Fehler korrigieren, bevor sie fehlerfrei übersetzen und das Programm in die CPU laden können.

Jeder Datenbaustein wird in den Übersetzungsvorgang des Projekts einbezogen.

Die Übersetzung wird gestartet:

- über den Menübefehl "Zielsystem" > "Übersetzen"

oder

- durch Klicken auf das Symbol.



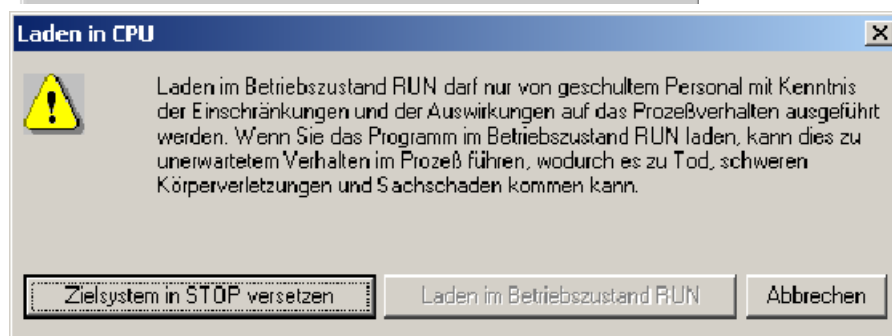
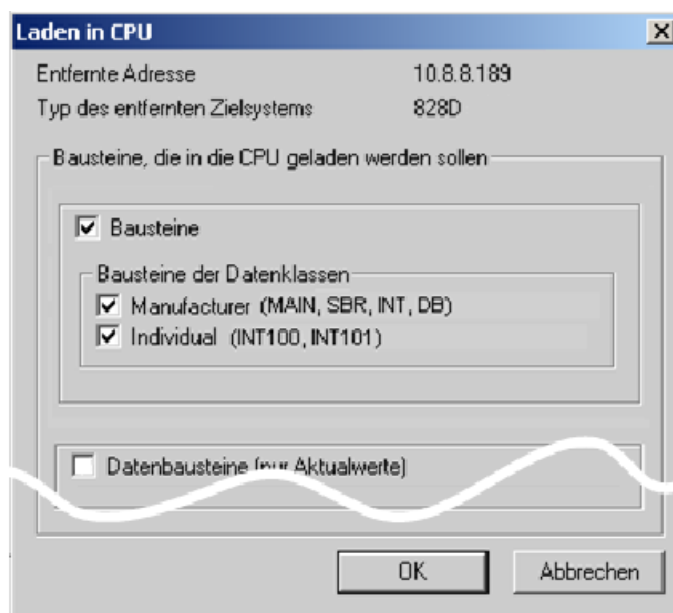
Treten beim Übersetzen des Datenbausteins Fehler auf, werden diese im Ausgabefenster angezeigt. Positionieren Sie den Cursor auf eine Fehlermeldung im Ausgabefenster und doppelklicken Sie, damit Ihnen die fehlerhafte Zeile im Datenbaustein angezeigt wird.

## Laden des Datenbausteins ins Zielsystem

An dieser Stelle sei auch auf das Kapitel "Datenklassen (Seite 1226)" verwiesen, in welchem das Laden und Speichern (nicht nur von Datenbausteinen) ausführlich erläutert wird.

- Nach strukturellen Änderungen eines Datenbaustein muss dieser bei gestoppter PLC ins Zielsystem geladen werden. Erst im anschließenden Betriebszustand RUN werden die Änderungen wirksam. Stellt das Zielsystem fest, dass ein Datenbaustein neu ist oder geändert wurde, setzt es für diesen Datenbaustein die Anfangswerte als erste Aktualwerte.

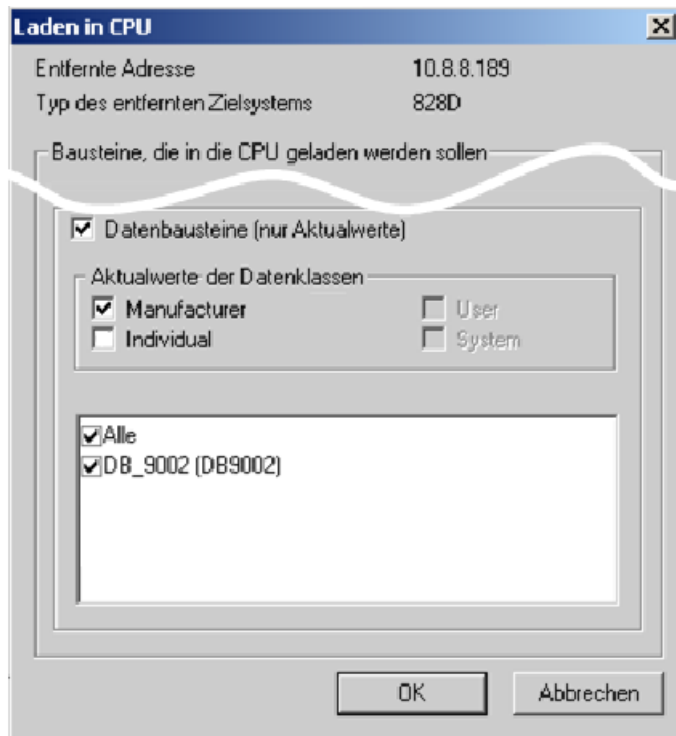
Sie sollten unbedingt die drei Checkboxen der Datenklassen aktiviert lassen. Warum? Wenn der geänderte DB auch aus der Datenklasse "Individual" aufgerufen wird, dann entstehen durch deren Abwahl Programmfehler: INT100 oder INT101 greifen dann nämlich noch auf die "alte" Struktur des DB zu.



- Nach Änderungen der Aktualwerte eines Datenbausteins müssen diese anschließend ebenfalls ins Zielsystem geladen werden. (Die Struktur des Datenbausteins im Zielsystem muss dabei der Struktur des Datenbausteins im Projekt entsprechen). Die Aktualwerte können auf zwei Möglichkeiten in die CPU geladen werden:



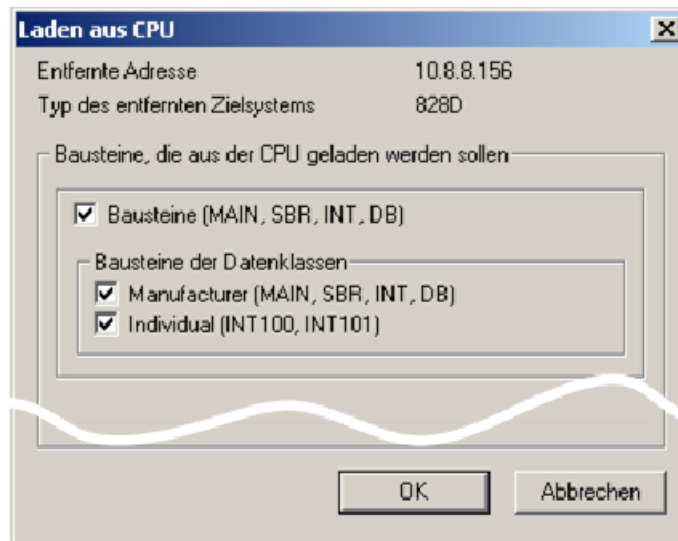
- Durch Klicken auf das Symbol "Alle Schreiben".
- Durch Laden des DB im Betriebszustand RUN:



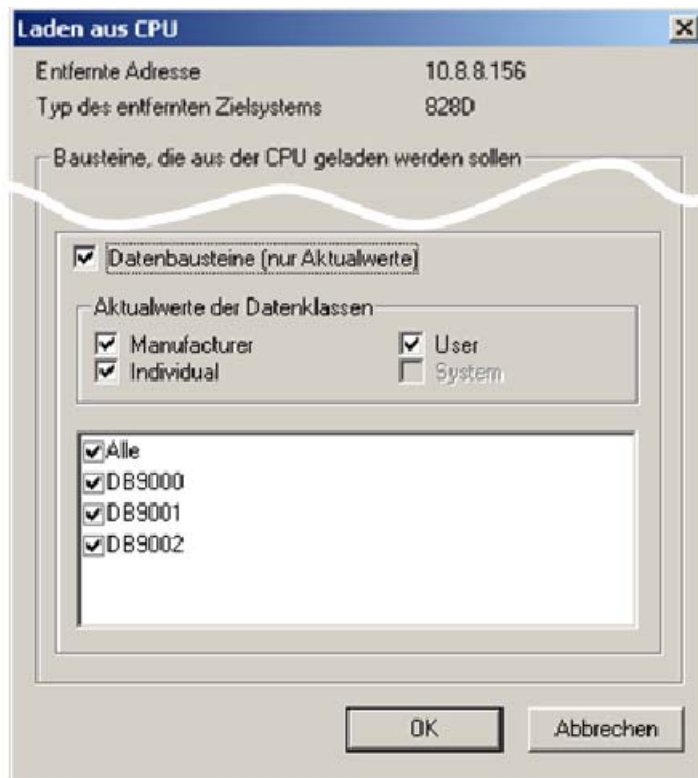
## Laden des Datenbausteins aus dem Zielsystem

An dieser Stelle sei auch auf das Kapitel "Datenklassen (Seite 1226)" verwiesen, in welchem das Laden und Speichern (nicht nur von Datenbausteinen) ausführlich erläutert wird.

- Sie müssen zunächst ein Projekt im Programming Tool PLC828 öffnen, bevor Sie den Programmbaustein (Projekt) und damit seine Struktur aus dem Zielsystem laden können. Weil Struktur und Anfangswerte der Datenbausteine fest der Datenklasse "Manufacturer" zugeordnet sind, braucht auch nur das Optionskästchen dieser Datenklasse aktiviert zu werden.



- Möchten Sie nur die Aktualwerte eines Datenbausteins aus dem Zielsystem laden, öffnen Sie zunächst das entsprechende Projekt im Programming Tool PLC828 oder laden Sie es aus dem Zielsystem. Nun können Sie die Aktualwerte laden. Entspricht die Struktur des Datenbausteins im Zielsystem nicht der des Datenbausteins des geöffneten Projekts (oder besitzt das geöffnete Projekt keinen Datenbaustein), dann können Sie den Datenbaustein nicht laden.





### 13.3.2.7 Sondermerker und ihre Funktionen

Sondermerker SMB0 enthält sieben Bits (SM0.0 ... SM0.6), die von der PLC-Firmware am Ende eines jeden Zyklus aktualisiert werden. Mit diesen Bits können Sie verschiedene Funktionen in Ihrem Programm realisieren.

SM-Bits (schreibgeschützt)	Beschreibung
SM0.0	Dieses Bit ist immer eingeschaltet.
SM0.1	Dieses Bit ist im ersten Zyklus eingeschaltet. Es wird z. B. zum Aufrufen eines Initialisierungs-Unterprogramms verwendet.
SM0.2	Dieses Bit wird für die Dauer eines Zyklus eingeschaltet, wenn remanente Daten verlorengegangen sind. Es kann entweder als Fehlermerker oder als Mechanismus zum Aufrufen von besonderen Anlaufsequenzen verwendet werden.
SM0.3	Dieses Bit wird für die Dauer eines Zyklus eingeschaltet, wenn die Betriebsart RUN nach dem Einschalten (Netz Ein) eingestellt wird. Damit kann vor dem Betrieb für eine Aufwärmzeit der Anlage gesorgt werden.
SM0.4	Dieses Bit sorgt für einen Takt der 30 Sekunden eingeschaltet und 30 Sekunden ausgeschaltet ist. Und zwar für eine Zykluszeit von 1 Minute. Damit verfügen Sie über eine leicht zu programmierende Verzögerungszeit bzw. eine Taktzeit von 1 Minute.
SM0.5	Dieses Bit sorgt für einen Takt, der 0,5 Sekunden eingeschaltet und 0,5 Sekunden ausgeschaltet ist. Und zwar für eine Zykluszeit von 1 Sekunde. Damit verfügen Sie über eine leicht zu programmierende Verzögerungszeit bzw. eine Taktzeit von 1 Sekunde.
SM0.6	Dieses Bit stellt einen Zyklustakt dar. Es ist ein Zyklus eingeschaltet, der nächste Zyklus ausgeschaltet. Sie können dieses Bit als Zykluszähleingang verwenden.
SM0.7	Reserviert

### 13.3.3 Operationsumfang

Die PLC der SINUMERIK 828D stellt folgende Operationsgruppen bereit:

- Bitverknüpfungsoperationen
- Festpunktarithmetik
- Interruptoperationen
- Gleitpunktarithmetik
- Programmsteuerungsoperationen
- Schiebe-/Rotieroperationen
- Übertragungsoperationen
- Umwandlungsoperationen
- Vergleichsoperationen
- Verknüpfungsoperationen
- Zähler
- Zeiten
- Unterprogramme

Details entnehmen Sie bitte der Online-Hilfe des Programming Tools PLC828 und dem S7-200 Systemhandbuch.

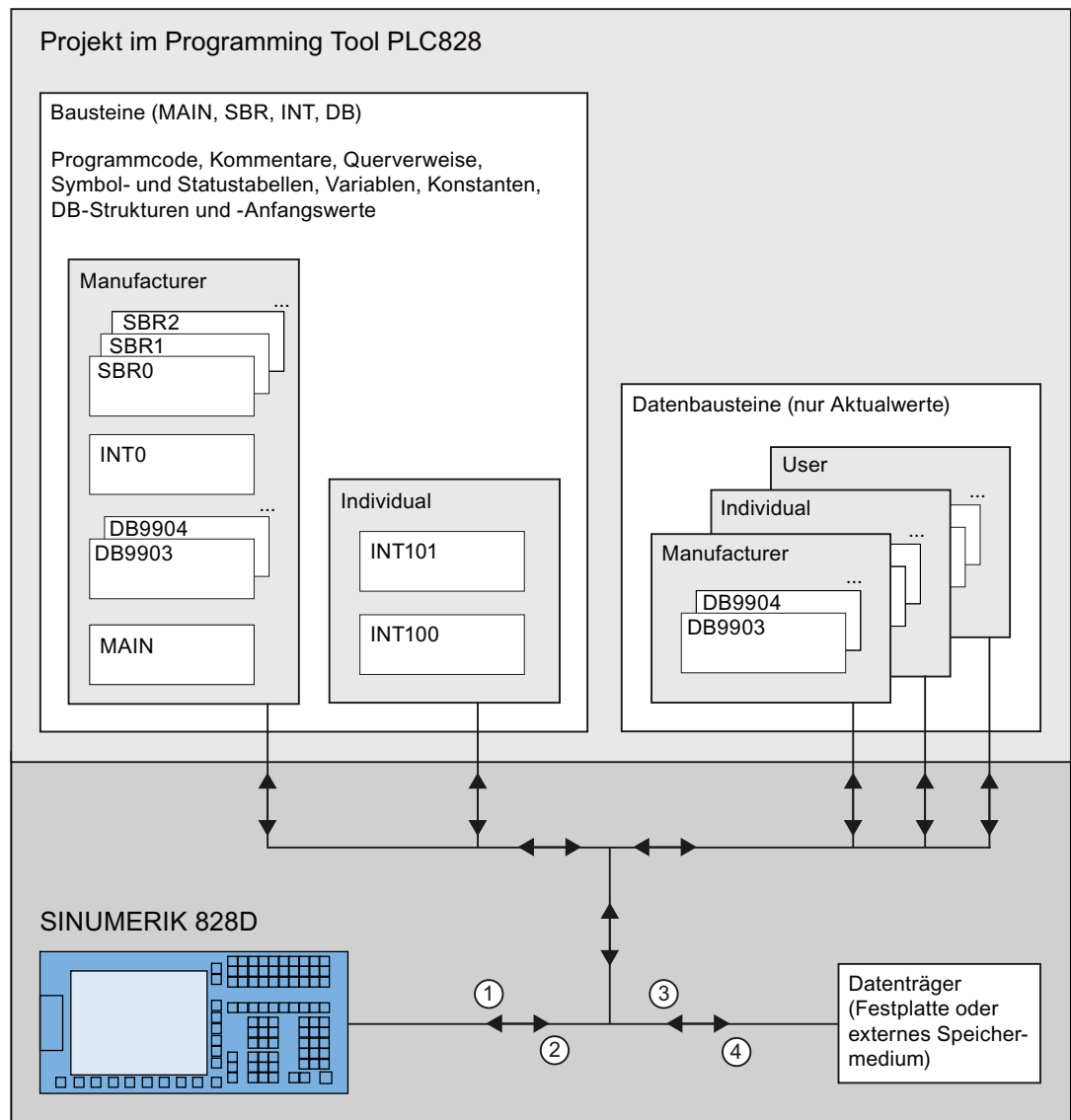
### 13.3.4 Datenklassen

#### 13.3.4.1 Definition der Datenklassen

##### Übersicht

Datenklassen sind als Projekt-Subcontainer anwenderbezogene Organisationseinheiten für Programme und Datenbausteine.

Ihre besondere Bedeutung liegt in der gegenseitigen Abgrenzung der Inhalte: Die ihnen zugeordneten Daten und Programme werden als Gruppe, eben als Datenklasse handhabbar. Dies gilt für das Laden in die Steuerung und aus der Steuerung sowie für die Datensicherung mittels Export und Import.



- ① Laden in CPU
- ② Laden aus CPU
- ③ Projekt öffnen
- ④ Projekt speichern (unter)

Bild 13-5 Projektaufbau im Programming Tool PLC828 und Struktur der Datenklassen-Transportwege

Aus Anwendersicht gibt es drei Datenklassen:

- Manufacturer:
  - POEs (MAIN, alle Unterprogramme und das Interruptprogramm INT0);
  - die Datenbausteine: innere Struktur (Typ-Info) plus Anfangswerte;
  - die Datenbausteine: Aktualwerte, sofern vom Anwender zugeordnet;
- Individual:
  - die Interruptprogramme INT100 und INT101
  - die Datenbausteine: Aktualwerte, sofern vom Anwender zugeordnet;
- User:
  - die Datenbausteine: Aktualwerte, sofern vom Anwender zugeordnet;

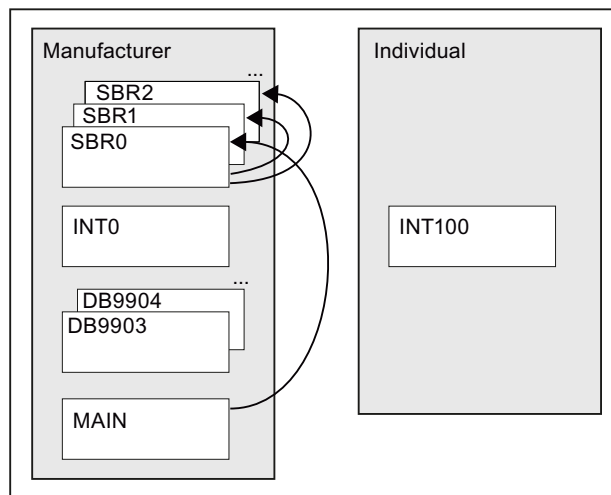
Im Dialog für Laden werden immer alle vorhandenen Datenklassen automatisch angewählt. Damit werden immer alle zum Projekt gehörenden Daten und Bausteine geladen, wenn sie nicht vom Anwender abgewählt werden.

## Beispiel

An einer Serienmaschine, die einschließlich PLC-Anwenderprogramm vom Maschinenhersteller in Betrieb genommen wurde, fällt ein PLC-Eingang (IO-Baugruppe) aus. Der Eingang wird auf einen freien Eingang umverdrahtet. Damit nicht vom Service/Einrichter das PLC-Anwenderprogramm des Maschinenherstellers wegen des Umverdrahtens aufwändig geändert werden muss, kann das Interruptprogramm INT100 genutzt werden: Es läuft vor dem Hauptprogramm (MAIN) und schreibt den umverdrahteten Eingang im Abbild auf den ursprünglichen Eingang. INT100 ist der Datenklasse "Individual" zugeordnet und wird mit dieser Datenklasse in die CPU geladen.

Der Projektbaustein des Maschinenherstellers hat die Datenklasse "Manufacturer" und wird mit dieser Datenklasse in die CPU geladen.

Aktualisiert der Maschinenhersteller sein PLC-Anwenderprogramm, wird das Korrekturprogramm davon nicht beeinflusst. Die Korrekturen bleiben unabhängig und weiterhin wirksam.



Das Interruptprogramm INT100 kann in seiner Datenklasse unabhängig geladen werden und oben beschriebene Korrekturfunktion ausführen.

### 13.3.4.2 Baustein einer Datenklasse zuordnen

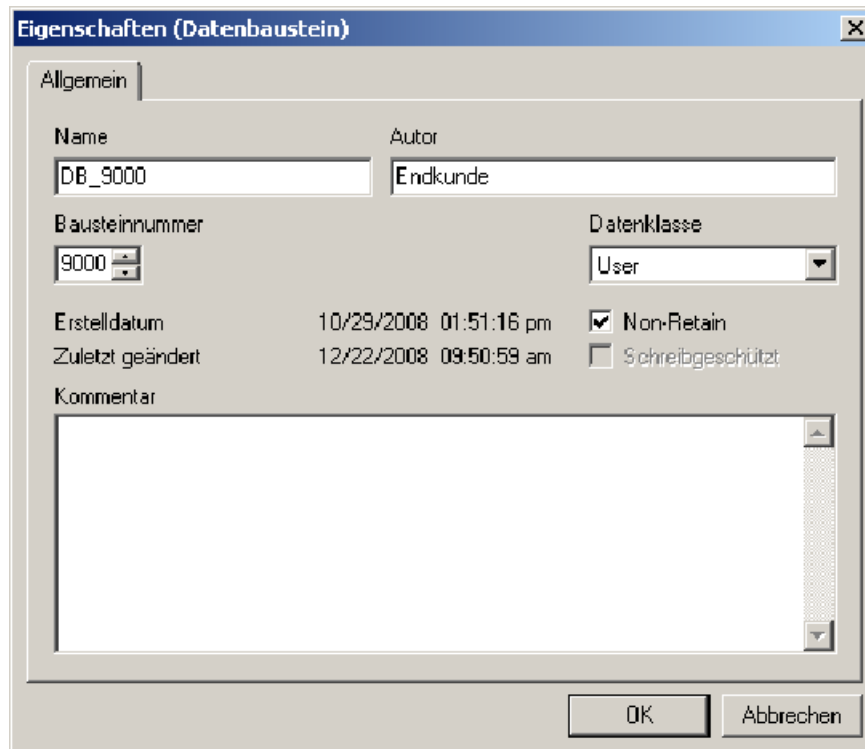
Ein Datenbaustein wird in seinem Eigenschaftsdialog einer Datenklasse zugeordnet.

#### Vorgehensweise

1. Klicken Sie mit der rechten Maustaste im Operationsbaum auf den entsprechenden Baustein und wählen Sie "Eigenschaften":



2. Ordnen Sie dem Baustein eine der drei möglichen Datenklassen zu:



Für den DB9000 wurde hier neben der Datenklasse "User" noch die Eigenschaft "Non-Retain" angewählt. Datenbausteine mit diesem Attribut werden nach jedem Netz-Aus- und Netz-Einschalten auf die Anfangswerte zurückgesetzt.

## Siehe auch

Definition der Datenklassen (Seite 1226)

### 13.3.4.3 Datenklasse(n) laden in CPU

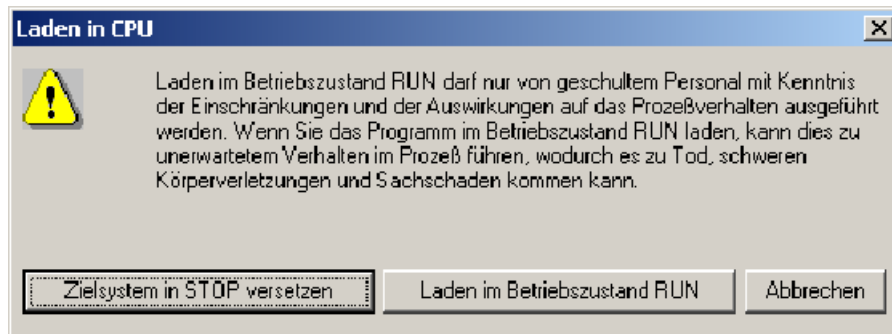
#### Vorgehensweise

1. Wählen Sie im Fenster "Laden in CPU" die Datenklasse(n) aus, deren Bausteine geladen werden sollen:



2. Wählen Sie die Option "Bausteine (MAIN, SUBR, INT, DB)", wenn Programm- oder Datenbausteinänderungen ins Zielsystem geladen werden sollen.
3. Wählen Sie die Option "Datenbausteine (nur Aktualwerte)", wenn Aktualwerte eines oder mehrerer Datenbausteine ins Zielsystem geladen werden sollen.

4. Wählen Sie im Meldungsfenster "Laden in CPU" eine der beiden folgenden Optionen:
- "Zielsystem in Stopp versetzen"  
→ Wenn strukturell geänderte Programme oder Anfangswerte von Datenbausteinen in das Zielsystem geladen werden sollen.
- oder**
- "Laden im Betriebszustand RUN"  
→ Wenn sich weder Programm- noch Datenbausteinstrukturen geändert haben.



Nach dem Laden muss folgende Meldung kommen:



5. Bestätigen Sie die Meldung.
6. Schalten Sie nach "Laden in Stopp" – wenn gewünscht – die Steuerung wieder in RUN.

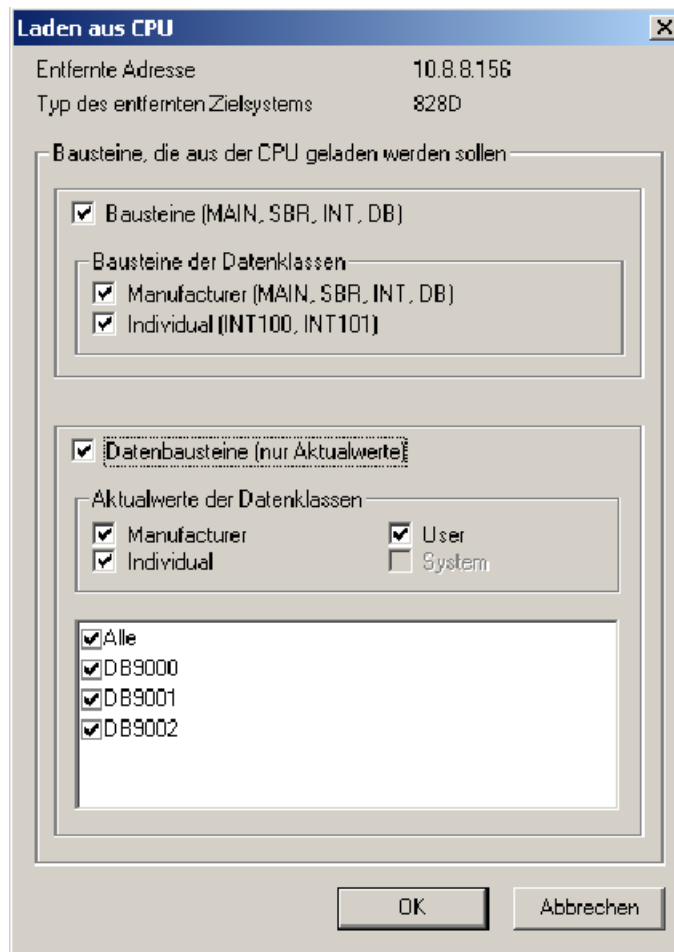




### 13.3.4.4 Datenklasse(n) laden aus CPU

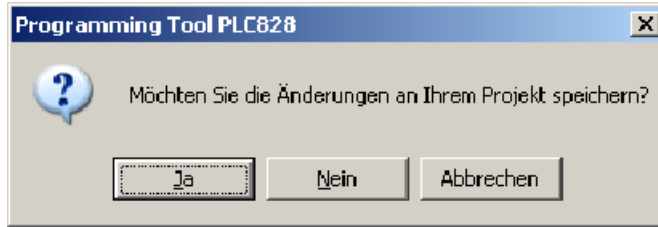
#### Vorgehensweise

1. Wählen Sie im Fenster "Laden aus CPU" die Datenklasse(n) aus, deren Bausteine geladen werden sollen:



2. Wählen Sie die Option "Bausteine (MAIN, SUBR, INT, DB)", wenn Programm- oder Datenbausteinänderungen in das Zielsystem geladen werden sollen.
3. Wählen Sie die Option "Datenbausteine (nur Aktualwerte)", wenn Aktualwerte eines oder mehrerer Datenbausteine ins Zielsystem geladen werden sollen.

- Speichern Sie gegebenenfalls das im Programming Tool PLC828 geöffnete Projekt, es wird beim "Laden aus CPU" überschrieben:



Ein neues Projekt mit dem Namen des im Zielsystem befindlichen Projekts wird angelegt.

- Nach einer Warnung, dass durch das "Laden aus CPU" eventuell das existierende Projekt verändert wird, werden die angeforderten Daten aus dem Zielsystem geladen (siehe auch Abbildung "Projektaufbau im Programming Tool PLC828 und Struktur der Datenklassen-Transportwege" in "Definition der Datenklassen (Seite 1226)").

#### 13.3.4.5 Vergleich zwischen Offline- und Online-Projekt

Im Dialog "Vergleichen..."(Menü "Zielsystem" > "Vergleichen...") können die Datenklassen ausgewählt werden, deren Bausteine verglichen werden sollen.

Existieren Datenklassen nur im Offline-Projekt oder nur in der CPU (Online-Projekt), werden diese entsprechend gekennzeichnet. Als Resultat des Vergleichs wird dargestellt, welche Unterschiede zwischen den offline und online vorhandenen Programmbausteinen (SBRs, INTs) bzw. Datenbausteinen (DBs) bestehen:

**Projekt mit Zielsystem vergleichen** [X]

Entfernte Adresse: 10.88.156  
 Typ des entfernten Zielsystems: 828D

Zu vergleichende Bausteine

Bausteine (MAIN, SBR, INT, DB)

Bausteine der Datenklassen

Manufacturer (MAIN, SBR, INT, DB)  
 Individual (INT100, INT101)

Datenbausteine (nur Aktualwerte)

Aktualwerte der Datenklassen

Manufacturer  User  
 Individual  System

Alle  
 DB\_9000 (DB9000)  
 DB\_9001 (DB9001)  
 DB\_9002 (DB9002)

Ergebnis des Vergleichs

Bausteine:

Datenbausteine:

Beginnen Schließen

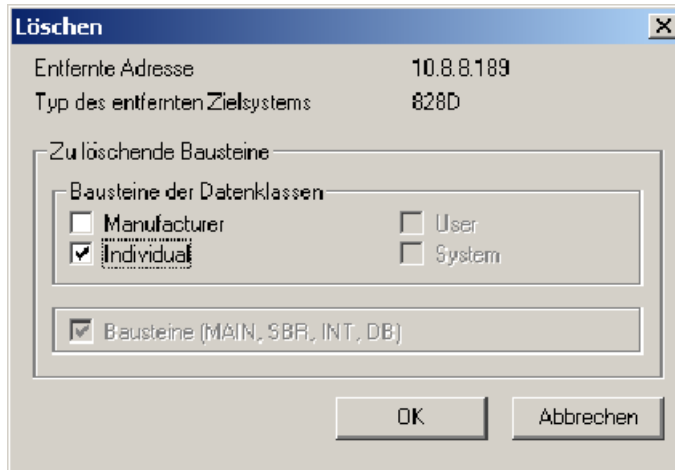
Beispiel:

Im unteren Feld "Datenbausteine" werden die Unterschiede zwischen den im Projekt gespeicherten (offline) Aktualwerten (z. B. +22) und den auf dem Zielsystem vorliegenden (online) Aktualwerten (z. B. +11) angezeigt.

### 13.3.4.6 Löschen im Zielsystem

Löschen ist nur im Betriebszustand "STOP" erlaubt.

Löschen von "Manufacturer" führt beim Übergang in den Betriebszustand "RUN" zum Anlauffehler. Dabei kehrt das Zielsystem in den sicheren Betriebszustand "STOP" zurück.



Siehe auch

Definition der Datenklassen (Seite 1226)

### 13.3.5 Operanden umverdrahten

Funktion

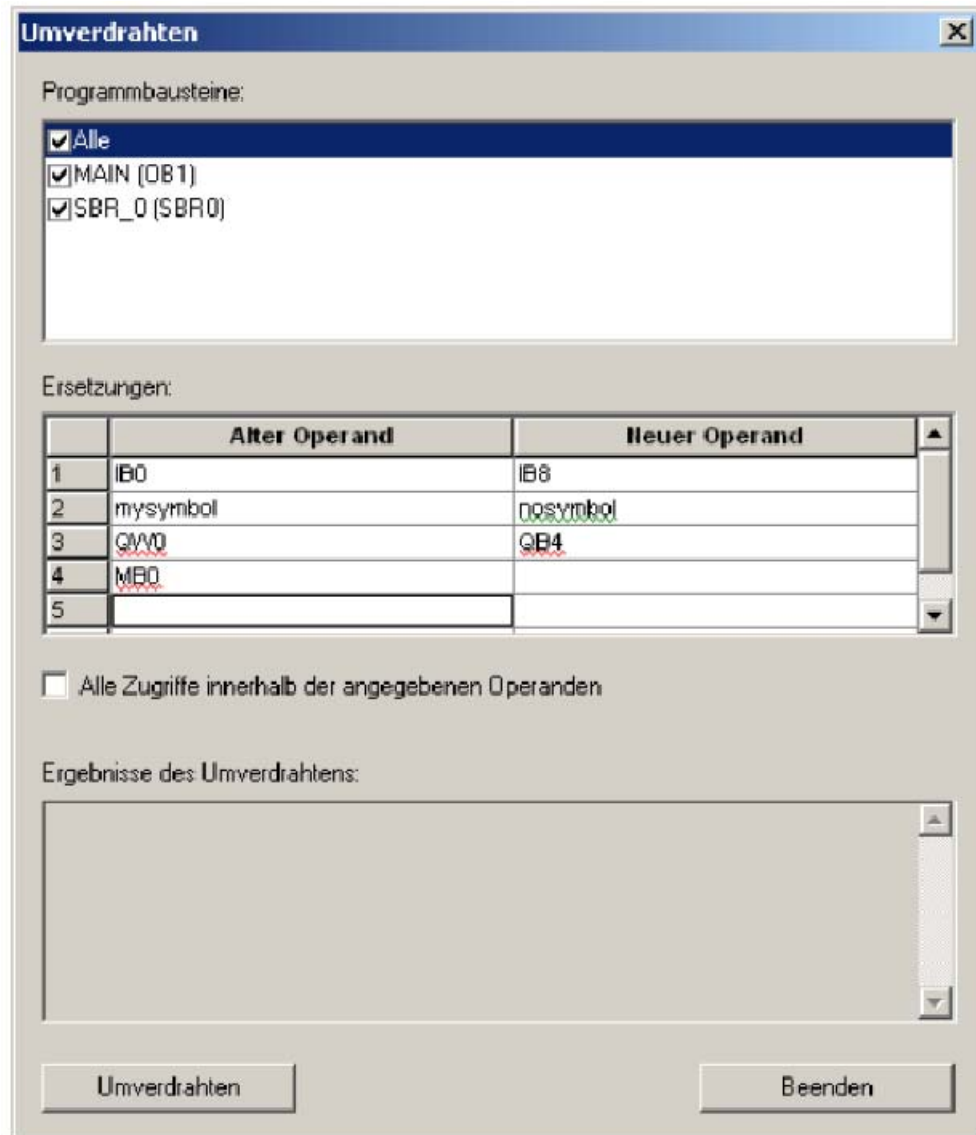
Mit der Funktion "Umverdrahten" können Operanden im PLC-Anwenderprogramm zentral geändert werden, z. B. IW0 nach IW8. Damit lassen sich Anwenderprogramme schnell an einen geänderten Peripherieausbau anpassen.

**Beispiel:**

Der Kunde schreibt für eine Maschinenreihe ein PLC-Anwenderprogramm. Da aufgrund des unterschiedlichen Maschinenausbau sich der Peripherieausbau einiger Maschinen unterscheidet, mussten bisher die Operanden im Anwenderprogramm einzeln geändert werden. Das konnten unter Umständen mehrere Hundert Operanden sein. Mit Hilfe des Dialogs "Umverdrahten" kann nunmehr für diese Maschinen eine Liste der zu ändernden Operanden, z. B. Ein- und Ausgänge, eingegeben werden. Diese wird mit der Funktion "Umverdrahten" abgearbeitet und die Operanden werden im Anwenderprogramm geändert.

## Vorgehensweise

Verwenden Sie den Dialog "Umverdrahten", um Operanden umzuverdrahten:



Gehen Sie wie folgt vor:

1. Öffnen Sie den Dialog "Umverdrahten" im KOP-Editor über das Kontextmenü ("Umverdrahten ..."). bzw. über die Menüleiste ("Bearbeiten > Umverdrahten ...").
2. Wählen Sie in der Liste "Programmbausteine" (Liste aller im Projekt verfügbaren POEs) die POEs aus, in denen das Umverdrahten durchgeführt werden soll.

3. Geben Sie in der Liste "Ersetzungen" jeweils den alten und den neuen Operanden für das Umverdrahten ein.

**Erlaubte Operanden sind:**

- Eingänge
- Ausgänge
- Merker
- Sondermerker
- Variablenspeicher / Datenbausteine
- Zeiten
- Zähler

**Editieren der Operandenliste**

Mit Hilfe des Kontextmenüs (rechte Maustaste) werden folgende Funktionen unterstützt:

- Ausschneiden (Ctrl+X)
- Kopieren (Ctrl+C)
- Einfügen (Ctrl+V)
- Alles markieren (Ctrl+A)
- Reihe einfügen (Ctrl+I)
- Markierung Löschen

Damit ist es möglich, die Liste oder Teile der Liste aus anderen bzw. in andere Applikationen, z. B. Microsoft Excel, zu kopieren.

**"Alter Operand"**

In diese Spalte tragen Sie den Namen oder die Adresse des Operanden ein, den Sie umverdrahten wollen.

**"Neuer Operand"**

In diese Spalte tragen Sie den neuen Namen oder die neue Adresse des Operanden ein. Beachten Sie dabei, dass der Typ des neuen Operanden dem des alten Operanden entspricht, z. B. alter Operand IW0 und neuer Operand IW4, nicht IB4, oder alter Operand DB9000.DBB0 und neuer Operand MB0, nicht MW0.

**Prüfen der Gültigkeit der Operanden**

Existiert der Name des Operanden (Symbol) im geöffneten Projekt nicht, wird dieser mit einer grünen Wellenlinie gekennzeichnet.

Stimmen die Typen des alten und des neuen Operanden nicht überein oder wurde nur ein Operand (alter oder neuer Operand) eingetragen, wird dieser Operand mit einer roten Wellenlinie gekennzeichnet.

4. Wählen Sie die Option "Alle Zugriffe innerhalb der angegebenen Operanden" an oder ab.  
Wenn die Option eingeschaltet ist, werden Operandenbereiche (BYTE, WORD, DWORD) umverdrahtet.

Beispiel:

Sie geben als Operandenbereiche IW0 und IW4 an. Dann werden die Operanden I0.0 ... I1.7 auf die Operanden I4.0 ... I5.7 umverdrahtet. Operanden aus dem umverdrahteten Bereich (z. B. I0.1) können anschließend nicht mehr einzeln in die Tabelle eingetragen werden.

5. Klicken Sie auf die Schaltfläche "Umverdrahten", um die Funktion zu starten.

---

#### Hinweis

Wollen Sie den Dialog verlassen, ohne dass die Funktion "Umverdrahten" aktiviert wird, dann benutzen Sie die Schaltfläche "Beenden".

---

Nach Ausführen des Umverdrahtens werden die Ergebnisse in der Liste "Ergebnisse des Umverdrahtens" angezeigt. Diese Liste enthält die Operandenliste mit den Spalten "Alter Operand" und "Neuer Operand". Dazu werden die einzelnen Bausteine aufgeführt mit der Anzahl Verdrahtungen, welche jeweils im Baustein durchgeführt worden sind. Mit Hilfe des Kontextmenüs (rechte Maustaste) können die Ergebnisse in andere Applikationen kopiert werden, z. B. Microsoft Word.

---

#### Hinweis

##### Folgendes ist beim Umverdrahten zu beachten:

- Name oder Nummer einer POE können nicht mit der Funktion "Umverdrahten" geändert werden. Nutzen Sie dafür im Operationsbaum im POE-Kontextmenü (Klick mit rechter Maustaste z. B. auf SBR\_0) die Funktionen "Umbenennen" oder "Eigenschaften ...".
  - Zeiten können nur wieder nach Zeiten (z. B. alter Operand T0, neuer Operand T16) und Zähler nur wieder nach Zähler (z. B. alter Operand C0, neuer Operand C1) umverdrahtet werden.
-

## 13.4 Test- und Beobachtungsfunktionen

### 13.4.1 Programmstatus

Nachdem Sie die Kommunikation zwischen Ihrem Programmiergerät, auf dem das Programming Tool PLC828 installiert ist, und einem Zielsystem hergestellt und ein Programm in das Zielsystem geladen haben, können Sie mit den Diagnosefunktionen vom Programming Tool PLC828 arbeiten und neue Programme testen sowie Programme überwachen, die bereits in Bearbeitung sind.

Im Folgenden werden diese Hilfethemen erläutert.

#### 13.4.1.1 Statusdefinition

Der Begriff "Status" bezieht sich auf die Anzeige der aktuellen Werte der Operanden während der Ausführung des Programms im Zielsystem. Sie können Statusinformationen in einer Statustabelle oder durch Einschalten des Programmstatus im Programm-Editor anzeigen lassen.

Die folgende Grafik zeigt ein Beispiel für Statusinformationen in der Statustabelle und im Programm-Editor vom Programming Tool PLC828:



The screenshot shows the SIMATIC Manager interface for a PLC828. The main window displays a ladder logic network (Netzwerk 1) with the following components: a normally open contact labeled 'Start\_1', a normally closed contact labeled 'Stopp\_1', another normally closed contact labeled 'Beh\_voll', and a coil labeled 'Pumpe\_1'. Below the network, the status table (Statustabelle) is visible, showing the current and new values for the variables.

	Adresse	Format	Aktueller Wert	Neuer Wert
1	Start_1	Bit	2#0	
2	Stopp_1	Bit	2#0	
3	Beh_voll	Bit	2#0	
4	Pumpe_1	Bit	2#1	

### Hinweis

Beachten Sie, dass unnötige Projektkomponenten (z. B. der Operationsbaum und das Ausgabefenster) weggelassen wurden, um mehr Platz zum Anzeigen der gewünschten Komponenten (KOP Programm-Editor, Statustabelle, Funktionsleiste zum Testen und Symbol für die Statustabelle in der Navigationsleiste) zu haben. Mit Hilfe des Menüs "Ansicht" können Sie sich die Umgebung im Programming Tool PLC828 entsprechend Ihren Aufgaben einrichten, d. h. Sie lassen sich nur die Projektkomponenten anzeigen, die Sie benötigen.

### 13.4.1.2 Voraussetzungen der Statusaktualisierung

Bevor Sie den Status aktualisieren können, um Ihr Programm zu überwachen und zu testen, müssen Sie folgende Aufgaben ausführen:

- Ihr Programm muss sich fehlerfrei übersetzen lassen.
- Sie müssen die Kommunikation zwischen dem Programming Tool PLC828 und dem Zielsystem eingerichtet haben.
- Ihr Programm muss fehlerfrei in das Zielsystem geladen sein.
- Sie sollten auch nach dem Laden des Programms ins Zielsystem dieses wieder in den Betriebszustand RUN versetzen. Andernfalls wird der Operandenstatus angezeigt, doch das Zielsystem kann das Programm nicht ausführen, so dass Sie nicht die Verknüpfungsoperationen angezeigt bekommen, die Sie erwarten.

### 13.4.1.3 Einfluss des Betriebszustandes des Zielsystems

Es richtet sich nach dem Betriebszustand Ihres Zielsystems, welche Arten von Überwachungs- und Testfunktionen Sie ausführen können.

Auch wenn Ihr Programm im Betriebszustand STOP nicht ausgeführt wird, beobachtet das Betriebssystem des Zielsystems weiterhin das Zielsystem (Status von RAM und E/A), übermittelt den Datenstatus an das Programming Tool PLC828. Befindet sich das Zielsystem im Betriebszustand STOP, können Sie folgende Funktionen ausführen:

- Sie können sich im Tabellenstatus oder im Programmstatus die aktuellen Werte der Operanden anzeigen lassen. (Dies ist gleichbedeutend mit der Funktion "Einfaches Lesen", da das Programm nicht bearbeitet wird.)
- Sie können im Tabellenstatus Werte schreiben.
- Sie können eine bestimmte Anzahl Zyklen ausführen und sich die Auswirkungen in einer Statustabelle und/oder im Programmstatus anzeigen lassen.

Befindet sich das Zielsystem im Betriebszustand RUN, können Sie die Funktionen "Erster Zyklus" oder "Mehrere Zyklen" nicht ausführen. Sie können Werte in einer Statustabelle schreiben, Sie können auch folgende Funktionen ausführen (nicht im Betriebszustand STOP):

- Sie können im Tabellenstatus fortlaufende Aktualisierungen durchführen. (Möchten Sie nur eine Aktualisierung ausführen, müssen Sie den Tabellenstatus ausschalten, damit Sie den Befehl "Einfaches Lesen" ausführen können.)
- Sie können im Programmstatus fortlaufende Aktualisierungen durchführen.

#### 13.4.1.4 Kommunikation und Zyklus

In einem fortlaufenden Zyklus liest das Zielsystem die Eingänge, es führt das Programm aus, schreibt in die Ausgänge und führt Systemfunktionen sowie die Kommunikation aus. Dieser Zyklus läuft mit extrem hoher Geschwindigkeit viele Male pro Sekunde ab. Auch wenn das Programming Tool PLC828 Statusanforderungen in schnellen Folgen aufgibt, ist es wichtig, dass Sie erkennen, dass Sie nicht jedes einzelne Ereignis, das sich im Zielsystem abspielt, beobachten können. Wenn Sie mit dem Programmstatus oder dem Tabellenstatus die Datenwerte aus einem Programm des Zielsystems lesen, fragen Sie die Daten in Stichproben ab. Die Aktualisierungsrate der aus dem Zielsystem gelesenen Statuswerte richtet sich nach der Baudrate der Kommunikation.

#### 13.4.1.5 Status aktualisieren

##### Programmstatus in KOP

Wenn Sie den Programmstatus im Programm-Editor in KOP überwachen, wird der Status am Ende eines jeden Zyklus aktualisiert. Wird ein Operand von mehreren Operationen bearbeitet, werden die Zwischenwerte des Operanden vom Status nicht angezeigt. Der Programmstatus in KOP zeigt nur die Werte der Operanden am Ende eines jeden Programmzyklus an.

### Status aktualisieren (Vorgehensweise)

Sie können den Status auf verschiedene Arten aktualisieren:



- Öffnen Sie den Programm-Editor und schalten Sie den Programmstatus ein (Menü "Testen" > "Programmstatus"), um sich die fortlaufende Aktualisierung des Status im Betriebszustand RUN des Zielsystems anzusehen.

Beachten Sie, dass "fortlaufend" nicht mit Echtzeit gleichzusetzen ist, sondern bedeutet, dass das Programmiergerät das Zielsystem schnell nach Statusinformationen abfragt und diese auf Ihrem Bildschirm anzeigt, so dass die Anzeige, so schnell es die eingerichtete Kommunikation zulässt, aktualisiert wird. Es kann sein, dass einige sich schnell ändernde Werte nicht erfasst und auf Ihrem Bildschirm angezeigt werden. Es kann auch sein, dass sich diese Werte so schnell ändern, dass Sie sie nicht lesen können. Wenn Sie das Zielsystem in den Betriebszustand STOP versetzen, können Sie den Status einmal aktualisieren. Wenn sich das Zielsystem im Betriebszustand STOP befindet, können Sie sich mit dem Befehl "Mehrere Zyklen" einen oder mehrere Zyklen anzeigen lassen. Mit der Funktion "Erster Zyklus" können Sie sich einen einzelnen Zyklus ansehen, wobei der Merker des ersten Zyklus aktiviert ist.



- Öffnen Sie das Fenster der Statustabelle und schalten Sie den Tabellenstatus ein, um sich die fortlaufende Aktualisierung des Status im Betriebszustand RUN des Zielsystems anzusehen.



- Schalten Sie den Tabellenstatus aus und nutzen Sie die Funktion "Einfaches Lesen", wenn Sie nur eine Aktualisierung des Status benötigen und das Zielsystem dafür nicht in den Betriebszustand STOP versetzen wollen.

Wenn Sie das Zielsystem in den Betriebszustand STOP versetzen und den Tabellenstatus einschalten, können Sie auch eine Aktualisierung des Status erhalten. Außerdem können Sie die Funktionen "Mehrere Zyklen" und "Erster Zyklus" einsetzen, wenn Sie sich die Statustabelle anzeigen lassen.

#### 13.4.1.6 Simulieren von Prozessbedingungen

Sie können Prozessbedingungen simulieren, indem Sie neue Werte in die Operanden schreiben. Verwenden Sie hierzu die Statustabelle.

#### 13.4.1.7 Prüfen von Querverweisen und verwendeten Elementen

Wenn Sie Ihr Programm testen, kann es sein, dass Sie Parameter ergänzen, löschen oder ändern möchten.

Im Fenster "Querverweise" können Sie ermitteln, wie die Parameter derzeit in Ihrem Programm zugeordnet sind. Auf diese Weise vermeiden Sie es, Werte doppelt zuzuweisen.

## 13.4.2 Programmstatus im KOP-Programmeditor

### 13.4.2.1 Programmstatus anzeigen

---

#### Hinweis

Wenn Sie im Betriebszustand STOP ein Programm ins Zielsystem geladen haben, müssen Sie das Zielsystem wieder in den Betriebszustand RUN versetzen, bevor Sie sich fortlaufende Aktualisierungen des Programmstatus anzeigen lassen können!

---

#### Vorgehensweise

Zum Einschalten des Programmstatus gehen Sie auf eine der folgenden Arten vor:

- Wählen Sie den Menübefehl "Testen" > "Programmstatus".

oder



- Klicken Sie zum Testen in der Funktionsleiste auf die Schaltfläche "Programmstatus".

Im Programm-Editor wird der Programmstatus angezeigt.

Start\_1



Boolesche Operationen (Kontakte, Spulen) werden als farbige Blöcke angezeigt, wenn der Operandenwert 1 beträgt (das Bit ist eingeschaltet).

Der Wert von nicht-booleschen Operanden wird gezeigt und aktualisiert, so schnell es die Kommunikation zulässt.

---

#### Hinweis

Wenn Sie den Programmstatus einschalten, werden viele anderen Funktionen im Programming Tool deaktiviert. Sie können z. B. Ihr Programm nicht ändern, es sei denn, Sie schalten den Programmstatus wieder aus. Andere Funktionen, z. B. das Wechseln der Anzeige von einem Programm-Editor zu einem anderen, bewirken, dass der Programmstatus automatisch ausgeschaltet wird. Wenn Sie sich den Status wieder anzeigen lassen möchten, müssen Sie den Befehl "Programmstatus" erneut anwählen.

---

---

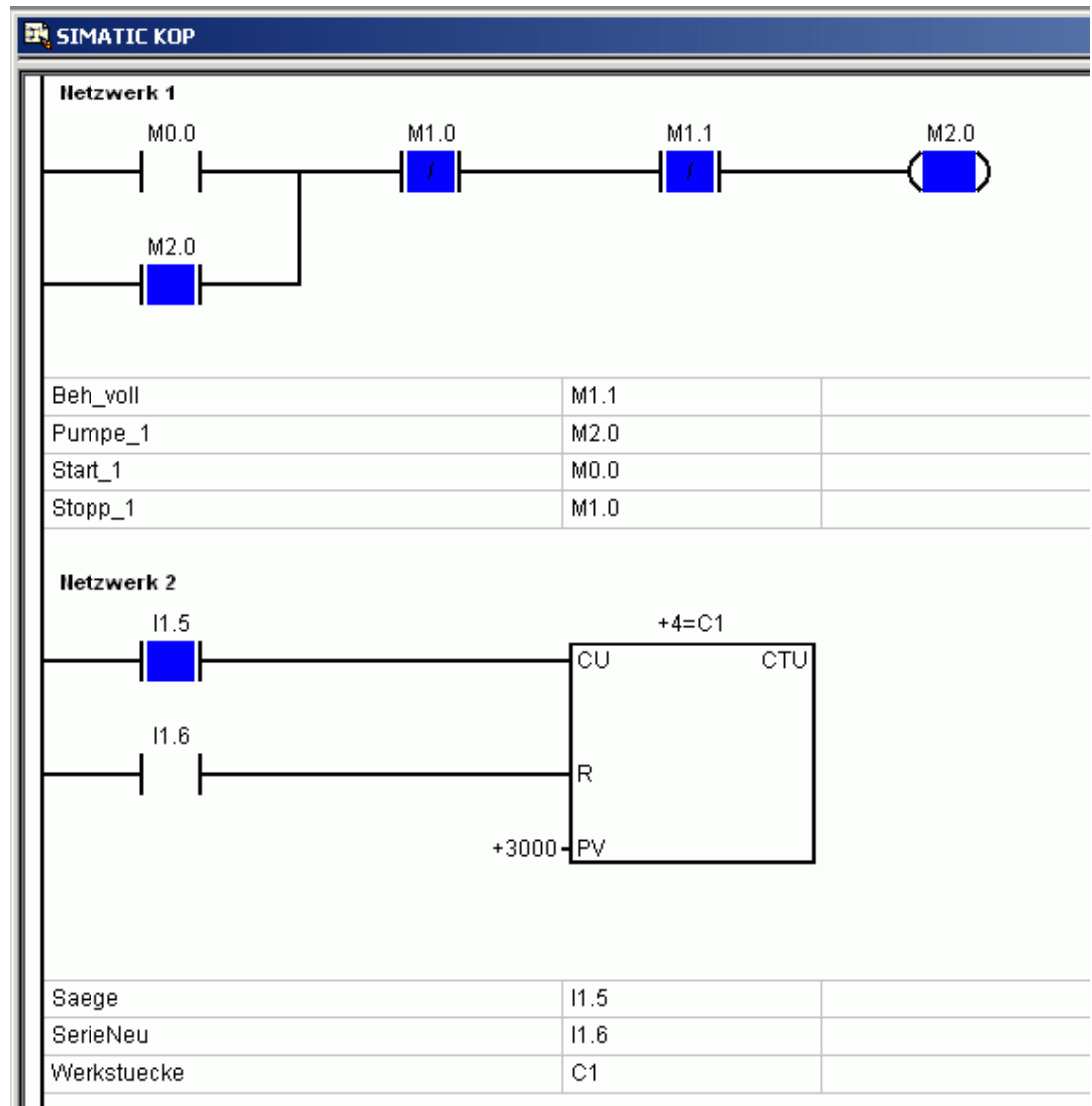
### Hinweis

Wenn Sie Probleme haben, den Programmstatus einzuschalten, bedenken Sie folgende Voraussetzungen:

- Sie müssen die Kommunikation eingerichtet haben (damit Sie Ihr Programm ins Zielsystem laden können).
  - Sie müssen die richtige CPU-Variante ausgewählt haben, damit Sie Ihr Programm ins Zielsystem laden können.
  - Ihr Programm muss sich fehlerfrei übersetzen lassen.
  - Ihr Programm muss sich fehlerfrei in das Zielsystem laden lassen.
  - Damit fortlaufende Aktualisierungen des Status angezeigt werden können, muss sich Ihr Zielsystem im Betriebszustand RUN befinden. Andernfalls werden nur Änderungen an Ein- und Ausgängen (sofern vorhanden) angezeigt. Da das Programm im Zielsystem nicht ausgeführt wird, haben Änderungen an den Ein- und Ausgängen nicht die Auswirkungen, die Sie auf die Programmlogik im angezeigten Programmstatus erwarten würden.
  - Wenn Sie sich einen Programmbereich anzeigen lassen, der nicht ausgeführt wird (z. B. ein Interruptprogramm oder Unterprogramm oder ein Bereich, der aufgrund einer Sprungoperation ausgelassen wird), gibt es keine Statusanzeige, weil der Code nicht abgefragt wird.
-

### 13.4.2.2 Eigenschaften der Anzeige

Wenn Sie sich den Programmstatus in KOP anzeigen lassen, werden boolesche Operationen als farbige Blöcke angezeigt, wenn der Wert des Operanden gleich 1 ist (Bit ist eingeschaltet). Der aktuelle Datenwert von anderen Operanden wird neben dem Operanden (oder anstelle des Operanden) angezeigt. Die Anzeige wird aktualisiert, wenn Änderungen aus dem Zielsystem gelesen werden.



### 13.4.2.3 Einschränkungen

Der Vorteil daran, sich den Status im Programm-Editor anzeigen zu lassen, liegt darin, dass Sie eine grafische Darstellung der Ereignisse in Ihrem Programm angezeigt bekommen. Doch stehen Ihnen nicht alle Werkzeuge wie in der Statustabelle zur Verfügung (z. B. die Funktionen "Einfaches Lesen", "Alle schreiben").

Es ist wichtig, dass Sie die Einschränkungen des Programmstatus kennen, wenn Sie Ihr Programm testen und überwachen.

## Signalfluss

**Die Farbmarkierung für "Signalfluss" bedeutet nicht, dass immer Signalfluss vorhanden ist!**

Da im Programmstatus in KOP nur die Werte vom Ende des Zyklus angezeigt werden, kann es manchmal schwierig sein, auszuwerten, was die Darstellung "Signalfluss" wirklich bedeutet. Boolesche Kontakte und Spulen werden im Programmstatus in KOP entsprechend dem Wert des Bit-Operanden farbig gekennzeichnet. Ist der Bitwert gleich 1 (Bit ist eingeschaltet), dann wird die Operation farbig markiert. Dies heißt jedoch nicht unbedingt, dass die Operation ausgeführt wurde. Es gibt mehrere Bedingungen, die zu einer missverständlichen Anzeige des Signalflusses führen können:

- Befindet sich das Zielsystem beim Auswerten des Status im Betriebszustand STOP, können Kontakte aktiviert sein, doch Spulen und Boxen werden nicht eingeschaltet, weil das Programm nicht ausgeführt wird.
- Enthält das Programm eine Sprungoperation, kann es sein, dass die Netzwerke, die Sie untersuchen, nicht die erwarteten Ergebnisse anzeigen, weil das Zielsystem diese Operationen während der Programmbearbeitung übersprungen hat.
- Ähnlich verhält es sich, wenn Sie sich ein Unterprogramm ansehen. Die booleschen Operanden können aktiviert sein, doch die Logik des Unterprogramms kann nur ausgeführt werden, wenn das Unterprogramm aktiviert wird. Wurde das Unterprogramm nicht vom Hauptprogramm aufgerufen, wurde keine Logik der Netzwerke ausgeführt, ganz gleich, was die Bitwerte der Operationen anzeigen.

## Status am Zyklusende in KOP

Wenn Sie sich den Programmstatus im Programm-Editor in KOP anzeigen lassen, wird der Status am Ende eines jeden Zyklus aktualisiert. Das Anzeigen Ihres Programms in einer anderen Programmiersprache hat keine Auswirkungen auf das eigentliche Programm.



### 13.4.2.4 Anzeige des Programmstatus anpassen

#### Anzeige des Programmstatus im Programm-Editor anpassen

Vorgehensweise:

1. Wählen Sie den Menübefehl "Extras > Optionen".
2. Öffnen Sie das Register "Status".

Sie können nun folgende Einstellungen bearbeiten:

- **Zoomfaktor**

Zum Bearbeiten der Skalierung.

Tastenkombination: Zum schnellen Einstellen des Zoomfaktors im Programmstatus können Sie Tastenkombinationen verwenden:

- Drücken Sie die Strg-Taste und die Plus-Taste im Ziffernblock der Tastatur, um die Anzeigegröße zu vergrößern.
- Drücken Sie die Strg-Taste und die Minus-Taste im Ziffernblock der Tastatur, um die Anzeigegröße zu verkleinern.

- **Feldbreite und -höhe**

Zum Bearbeiten der Gittereinstellungen.

Sie können die Feldbreite vergrößern, damit die Informationen angezeigt werden, ohne dass sie abgeschnitten werden. Sie können die Feldhöhe verkleinern, damit mehr Netzwerke auf dem Bildschirm Platz finden.

- **Signalfluss**

Sie können die Farbe ändern, die anzeigt, dass boolesche Operanden aktiviert sind (Bitwert gleich 1).

- **Operandenanzeige**

Sie können die Operanden innerhalb oder außerhalb der Operationen anzeigen. Sie können auch den Statuswert anzeigen lassen, ohne den Namen oder die Adresse des Operanden anzuzeigen.

#### Anordnung der Fenster im Programming Tool PLC828 anpassen

Um mehr Platz für das Fenster des Programm-Editors zu schaffen oder um es gleichzeitig mit einem anderen Fenster, wie z. B. der Statustabelle, der Symboltabelle oder den Querverweisen, anzuzeigen, können Sie die Anordnung und Größe aller im Programming Tool PLC828 angezeigten Fenster wie folgt anpassen:

1. Selektieren Sie das Fenster, dessen Anzeige Sie anpassen möchten.
2. Verkleinern/Vergrößern und Positionieren Sie das Fenster mit der Maus oder mit den entsprechenden Tastenkombinationen Ihrer Tastatur.

- oder -

Verwenden Sie die für die Fensteranzeige zur Verfügung stehenden Befehle aus den Menüs "Ansicht" und "Fenster".

### 13.4.3 Anzeigen des Status in einer Statustabelle

#### 13.4.3.1 Eigenschaften einer Statustabelle

##### Anwendung

Sie können, nachdem Sie Ihr Programm in das Zielsystem geladen haben, eine oder mehrere Statustabellen zum Überwachen und Testen der Programmbearbeitung erstellen.

Befindet sich das Zielsystem im Betriebszustand RUN, wird das Programm fortlaufend ausgeführt. Sie können den Tabellenstatus einschalten, damit die Statuswerte in der Tabelle ununterbrochen aktualisiert werden. Als Alternative können Sie mit der Funktion "Einfaches Lesen" einen "Schnappschuss" der Statuswerte in der Tabelle erzeugen, ohne dass Sie die Statustabelle einschalten müssen.

Während Sie sich eine Statustabelle ansehen, können Sie das Zielsystem auch in den Betriebszustand STOP versetzen und nur den ersten bzw. eine bestimmte Anzahl von Zyklen ausführen, in denen Sie die Programmbearbeitung überwachen.

---

##### Hinweis

Beachten Sie, dass Sie Ihre Tabelle nicht ändern können, wenn der Tabellenstatus eingeschaltet ist! Schalten Sie den Tabellenstatus aus, wenn Sie die Tabelle bearbeiten möchten.

---

## Öffnen / Einschalten einer Statustabelle

Das Öffnen einer Statustabelle ist nicht das gleiche wie das Einschalten einer Statustabelle. Sie können eine Statustabelle öffnen und auswerten oder ändern, doch wenn Sie nicht den Befehl "Einfaches Lesen" (im Menü "Testen" oder in der Funktionsleiste) ausführen bzw. den Tabellenstatus (im Menü "Testen" oder in der Funktionsleiste) einschalten, werden keine Statusinformationen in der Spalte "Aktueller Wert" angezeigt.

	Adresse	Format	Aktueller Wert	Neuer Wert
1	Start_1	Bit	2#1	
2	Start_2	Bit	2#0	
3	Stopp_1	Bit	2#1	
4	Stopp_2	Bit	2#0	
5	Beh_voll	Bit	2#0	
6	Beh_leer	Bit	2#0	
7	Rücksetzen	Bit	2#0	
8		Mit Vorzeichen		
9	Pumpe_1	Bit	2#1	
10	Pumpe_2	Bit	2#0	
11	Mischermotor	Bit	2#0	
12	Dampfventil	Bit	2#0	
13	Abl_Ventil	Bit	2#1	
14	Abl_Pumpe	Bit	2#1	
15		Mit Vorzeichen		
16	Füllstand_erreicht	Bit	2#1	
17	Mischzeit	Mit Vorzeichen	+32767	
18	Zykluszähler	Mit Vorzeichen	+0	

Bild 13-6 Beispiel für eine Statustabelle

## Funktion "Einfaches Lesen"

Wenn Sie die Funktion "Einfaches Lesen" verwenden (nur bei ausgeschaltetem Tabellenstatus verfügbar), um eine Statustabelle auszuwerten, werden die aktuellen Werte des Zielsystems übernommen und in der Spalte "Aktueller Wert" angezeigt. Doch die Werte werden nicht aktualisiert, während das Zielsystem das Programm ausführt.

## Funktion "Tabellenstatus"

Wenn Sie den Tabellenstatus einschalten (im Menü "Testen" oder in der Funktionsleiste), werden die aktuellen Werte des Zielsystems regelmäßig aktualisiert. Werden Änderungen vom Zielsystem empfangen, wird die Spalte "Aktueller Wert" aktualisiert.

## Spalte "Neuer Wert"

Mit der Spalte "Neuer Wert" können Sie bestimmte Werte im Zielsystem zuordnen (schreiben).

### 13.4.3.2 Statustabelle öffnen

Öffnen Sie eine Statustabelle, um den Inhalt der Tabelle auszuwerten oder zu ändern.

#### Vorgehensweise



- Klicken Sie auf die Schaltfläche "Statustabelle" in der Navigationsleiste.  
oder
- Wählen Sie den Menübefehl "Ansicht" > "Statustabelle".  
oder
- Öffnen Sie das Verzeichnis der Statustabelle im Operationsbaum und doppelklicken Sie auf das Symbol einer Tabelle.
  
- Umfasst Ihr Projekt mehr als eine Statustabelle, können Sie mit Hilfe der Register für die Statustabellen am unteren Rand des Fensters zwischen den einzelnen Tabellen umschalten:



---

#### Hinweis

Wenn Sie eine Statustabelle öffnen, wird noch nicht der Status angezeigt. Sie müssen die Statustabelle einschalten, damit die Statusinformationen aktualisiert werden (siehe "Statustabelle einschalten (Seite 1257)").

---

### 13.4.3.3 Mit mehreren Statustabellen arbeiten

#### Zusätzliche Statustabellen einfügen

Zum Einfügen zusätzlicher Statustabellen:

- Klicken Sie im Operationsbaum mit der rechten Maustaste auf den Ordner "Statustabelle" und wählen Sie im Popup-Menü den Befehl "Statustabelle einfügen".  
oder
- Öffnen Sie das Fenster "Statustabelle" und rufen Sie das Menü "Bearbeiten" auf oder klicken Sie mit der rechten Maustaste und wählen Sie den Befehl "Inhalte einfügen" > "Tabelle".

### Zwischen den Statustabellen wechseln

Nachdem Sie eine neue Statustabelle eingefügt haben, erscheint am unteren Rand des Fensters "Statustabelle" ein neues Register:



Wenn Sie zwischen den Statustabellen wechseln möchten:

- Klicken Sie auf das Register der gewünschten Statustabelle.

### Verdeckte Register anzeigen

Manchmal wird ein Register durch die Schaltflächen zum Blättern auf der rechten Seite verdeckt. Ist ein Register nicht sichtbar, gehen Sie wie folgt vor:

- Ziehen Sie die Teilung zwischen Registerbereich und Schaltflächen zum Blättern, um weitere Register anzuzeigen.

#### 13.4.3.4 Statustabelle anlegen

In eine Statustabelle können Sie Adressen eintragen, um Werte aus Ihrem Programm zu beobachten und zu steuern. Werte von Zeiten und Zählern können als Bits oder Wörter angezeigt werden. Wenn Sie sich den Wert einer Zeit oder eines Zählers als Bit anzeigen lassen, dann wird der Zustand des Ausgangs angezeigt (ein bzw. aus). Wenn Sie sich den Wert einer Zeit oder eines Zählers als Wort anzeigen lassen, dann wird der aktuelle Wert verwendet.

### Vorgehensweise

Zum Anlegen einer Statustabelle gehen Sie folgendermaßen vor:

1. Tragen Sie in der Spalte "Adresse" die Adressen der gewünschten Werte ein.

Gültig sind alle Speicherarten mit Ausnahme von Akkumulatoren und Datenkonstanten.

Zum Bearbeiten eines Adressfeldes:

- Wählen Sie das gewünschte Feld mit den Cursortasten bzw. der Maus aus.
- Wenn Sie Daten eingeben, werden vorhandene Daten gelöscht und die neuen Zeichen eingetragen.
- Wenn Sie mit der Maus doppelklicken oder die Taste <F2> drücken, wird das Feld markiert. Sie können dann den Cursor mit den Cursortasten an die Stelle bewegen, die Sie bearbeiten möchten.

Adresse	
1	I0.0

2. Handelt es sich bei dem Element um ein Bit (z. B. I, Q oder M), wird in der zweiten Spalte Bitformat angezeigt. Handelt es sich bei dem Element um ein Byte, Wort oder Doppelwort, markieren Sie das Feld in der Spalte "Format" und doppelklicken Sie oder drücken Sie die Leertaste bzw. ENTER, um durch die gültigen Formate zu blättern, bis das richtige Format angezeigt wird.

Format
Mit Vorzeichen ▾

---

#### Hinweis

Sie können Adressen in der Symboltabelle markieren und in die Statustabelle kopieren, um Ihre Tabelle schneller zu erstellen.

Sie können sich den Status mehrfach anzeigen lassen. So können Sie Elemente in logische Gruppen aufteilen, um sich jede Gruppe in einer einzelnen Tabelle anzeigen zu lassen. Sie vermeiden auf diese Weise, dass Sie durch extrem lange Listen blättern müssen.

---

### 13.4.3.5 Statustabelle bearbeiten

#### Anzeigen

Zum Blättern durch die möglichen Datenformate für eine bestimmte Adresse:

- Wählen Sie das Feld "Datenformat" und drücken Sie wiederholt die Eingabetaste.

Zum Anzeigen aller verfügbaren Datenformate:

- Öffnen Sie das aufklappbare Listenfeld.

## Ändern

Zum Einstellen der Breite einer Spalte:

- Positionieren Sie den Mauszeiger auf dem Rand einer Spalte, bis der Cursor sein Aussehen ändert, und ziehen die Spalte größer oder kleiner.

## Auswählen

Zum Auswählen einer ganzen Reihe (zum Ausschneiden oder Kopieren):

- Klicken Sie die Nummer der Reihe einmal an.

Zum Auswählen der gesamten Statustabelle:

- Klicken Sie einmal die obere linke Ecke oberhalb der Reihennummern an.

## Einfügen

Zum Einfügen einer neuen Reihe:

1. Wählen Sie ein Feld oder eine Reihe in der Statustabelle.
2. Öffnen Sie das Menü "Bearbeiten" oder klicken Sie mit der rechten Maustaste auf das Feld (damit das Kontextmenü angezeigt wird).
3. Wählen Sie den Befehl "Inhalte einfügen" > "Reihe".

Die neue Reihe wird oberhalb der Cursorposition in die Statustabelle eingefügt.  
Nachfolgende Reihen werden um eine Reihe nach unten verschoben.

Zum Einfügen einer neuen Reihe mit der folgenden Adresse und dem gleichen Datenformat:

- Wählen Sie ein Adressfeld und drücken Sie die Eingabetaste.

Zum Einfügen einer Reihe am unteren Ende der Statustabelle:

- Ordnen Sie den Cursor in einem Feld in der letzten Reihe an und drücken Sie die Taste <Pfeil nach unten>.

### Löschen

Zum Löschen eines Feldes oder einer Reihe:

1. Markieren Sie das Feld bzw. die Reihe und klicken Sie mit der rechten Maustaste.
2. Wählen Sie den Menübefehl "Löschen" > "Markierung".

Wenn Sie eine Reihe löschen, rücken die nachfolgenden Reihen eine Reihe nach oben.

### Navigieren

Zum Springen ins nächste Feld der Tabelle:

- Drücken Sie die <TAB>-Taste.

#### 13.4.3.6 Datenformate

Das Datenformat, das Sie einem Wert zuordnen, legt fest, wie der Wert in der Statustabelle dargestellt werden kann.

	Adresse	Format	Aktueller Wert	Neuer Wert
11	DB_9002.dwDatum1	Hexadezimal	16#3F8E3831	
12	DB_9002.wDatum2	Ohne Vorzeichen	65471	
13	DB_9002.bDatum3	ASCII	'A'	
14	DB_9002.bDatum3	Ohne Vorzeichen	65	
15	DB_9002.boDatum4	Bit	2#1	
16	DB_9002.rDatum6	Gleitpunkt	0.5	
17	DB_9002.iDatum7	Mit Vorzeichen	-65	

In der Abbildung ist der DB9002 symbolisch adressiert dargestellt. Am Beispiel der Bausteinvariablen iDatum7 sind die möglichen Anzeigeformate für Variablen des Typs INTEGER bzw. WORD aufgezeigt.

#### Hinweis

Bit- und Binärwerte werden beide durch die Zahl 2 und das Symbol # eingeleitet.

Hexadezimalwerte werden durch die Zahl 16 und das Symbol # eingeleitet.

Bitwerte haben eine Ziffer. Binärwerte haben acht Ziffern.

Werte mit und ohne Vorzeichen verwenden die Basis 10 (dezimal).



### 13.4.3.7 Statustabelle einschalten

Schalten Sie die Statustabelle ein, damit die Statusinformationen aktualisiert werden.

#### Vorgehensweise

Wenn Sie die Statusinformationen in der Statustabelle fortlaufend aktualisieren lassen möchten, schalten Sie den Tabellenstatus ein:

- Wählen Sie hierzu den Menübefehl "Testen" > "Tabellenstatus".

oder



- Klicken Sie auf die entsprechende Schaltfläche in der Funktionsleiste.

Wenn Sie nur einen "Schnappschuss" der Werte benötigen, führen Sie die Funktion "Einfaches Lesen" aus:

- Wählen Sie hierzu den Menübefehl "Testen" > "Einfaches Lesen".

oder



- Klicken Sie auf die entsprechende Schaltfläche in der Funktionsleiste.

---

#### Hinweis

Bei eingeschaltetem Tabellenstatus ist die Funktion "Einfaches Lesen" deaktiviert.

---

#### Hinweis

Das Einschalten der Statustabelle bewirkt nichts, wenn die Tabelle noch leer ist: Sie müssen sich zuerst Ihre Statustabelle anlegen, indem Sie Programmwerte (Operanden) in die Spalte "Adresse" eintragen und für jede Adresse einen Datentyp in die Spalte "Format" eingeben (siehe "Statustabelle anlegen (Seite 1253)" und "Datenformate (Seite 1256)").

---

### 13.4.3.8 Arbeiten mit Testfunktionen in der Statustabelle

Sie greifen auf die Testfunktionen (Einfaches Lesen, Alle schreiben,) über das Menü "Testen" oder über die Funktionsleiste mit den Testfunktionen zu.

#### Einfaches Lesen

Setzen Sie das einfache Lesen ein, wenn Sie einen "Schnappschuss" benötigen, also eine einzelne Aktualisierung des Programmstatus von allen Werten.



Standardmäßig fragt der Tabellenstatus das Zielsystem fortlaufend nach Status-Aktualisierungen ab. Wenn Sie auf eine Statustabelle klicken und der Tabellenstatus ist ausgeschaltet, dann ist die Schaltfläche zum einfachen Lesen aktiviert.

### Alle schreiben



Nachdem Sie Werte in die Spalte "Neuer Wert" in der Statustabelle eingetragen haben, schreiben Sie die gewünschten Änderungen mit dem Befehl "Alle schreiben" in das Zielsystem.

### 13.4.4 Zyklen ausführen

Sie können angeben, dass das Zielsystem Ihr Programm für eine bestimmte Anzahl von Zyklen bearbeiten soll (von 1 Zyklus bis zu 65535 Zyklen). Wenn Sie angeben, dass das Zielsystem eine bestimmte Anzahl Zyklen ausführen soll, können Sie die Bearbeitung der Prozessvariablen beobachten.

Im ersten Zyklus ist der Wert von SM0.1 = 1 (EIN).

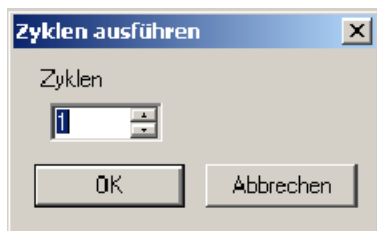
#### Ausführen eines einzelnen Zyklus

1. Das Zielsystem muss sich im Betriebszustand STOP befinden. Ist es nicht bereits in STOP, versetzen Sie das Zielsystem in den Betriebszustand STOP.
2. Wählen Sie den Menübefehl "Testen" > "Erster Zyklus".

#### Ausführen mehrerer Zyklen

1. Das Zielsystem muss sich im Betriebszustand STOP befinden. Ist es nicht bereits in STOP, versetzen Sie das Zielsystem in den Betriebszustand STOP.
2. Zum Ausführen mehrerer Zyklen wählen Sie den Menübefehl "Testen" > "Mehrere Zyklen".

Daraufhin wird das Dialogfeld "Zyklen ausführen" angezeigt:



3. Geben Sie an, wie viele Zyklen ausgeführt werden sollen, und bestätigen Sie mit "OK".

---

#### Hinweis

Achten Sie darauf, dass Sie das Zielsystem wieder in den Betriebszustand RUN versetzen, wenn Sie zur normalen Funktionsweise der Programmbearbeitung zurückkehren möchten:

- Wählen Sie hierzu die Schaltfläche "RUN" in der Funktionsleiste.

oder

- Wählen Sie den Menübefehl "Zielsystem" > "RUN".
-

## 13.5 Datenschnittstelle

Es werden Daten zwischen PLC und NC einerseits sowie zwischen PLC und HMI andererseits zyklisch ausgetauscht. Dies bedeutet insbesondere, dass vom HMI kommende und für die NC bestimmte Daten vom AWP rangiert werden müssen, damit diese wirksam werden.

**Daten an die PLC** werden von der Firmware am Zyklusanfang des AWP bereitgestellt. Dadurch wird gesichert, dass z. B. die Signale vom NCK über einen Zyklus konstant bleiben.

**Daten von der PLC** werden von der Firmware am Zyklusende des AWP an NCK oder HMI weitergeleitet.

Alle Daten dieser Nahtstelle sind im Gerätehandbuch für die SINUMERIK 828D, PPU, aufgeführt.

### 13.5.1 Nahtstelle PLC-NCK

Diese zyklischen Daten enthalten z. B. Statussignale ("Programm läuft", "Programm unterbrochen"), Steuersignale (Start, Stopp) und Hilfs- und G-Funktionen.

Die Daten sind gegliedert in Signale für:

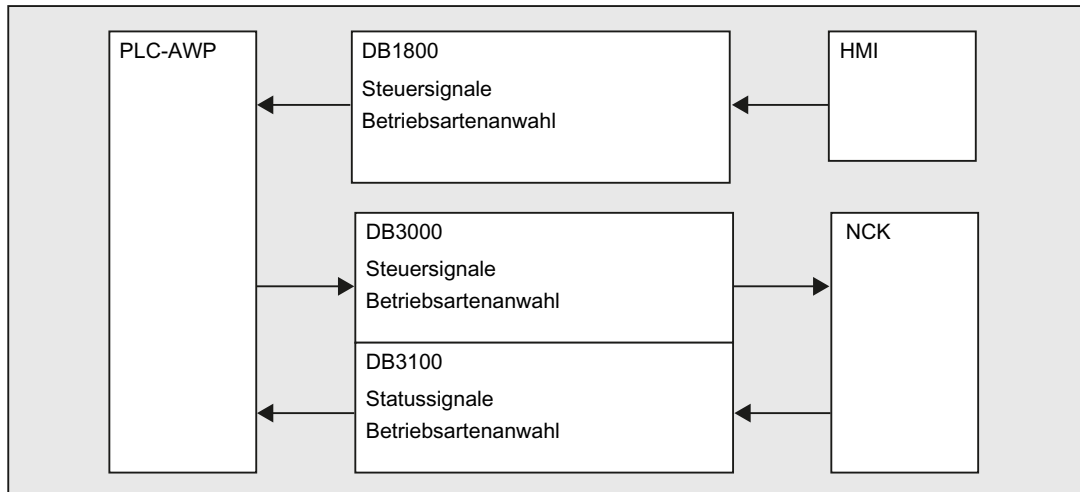
- Betriebsartensignale
- NC-Kanal-Signale
- Achs- und Spindelsignale
- Allgemeine NCK-Signale
- Schneller Datenaustausch PLC-NCK

### 13.5.1.1 Betriebsartensignale

#### DB3000, 3100

Zum NCK werden die von der Maschinensteuertafel oder vom HMI vorgegebenen Betriebsartensignale übertragen.

Vom NCK werden deren aktuelle Zustände an die PLC gemeldet.



### 13.5.1.2 NC-Kanal-Signale

#### DB2500, 3200, 3300, 3500

Die Signale gliedern sich in:

- Steuer-/Status-Signale mit normaler zyklischer Übertragung, siehe "Betriebsartensignale (Seite 1260) ".
- Hilfs- und G-Funktionen

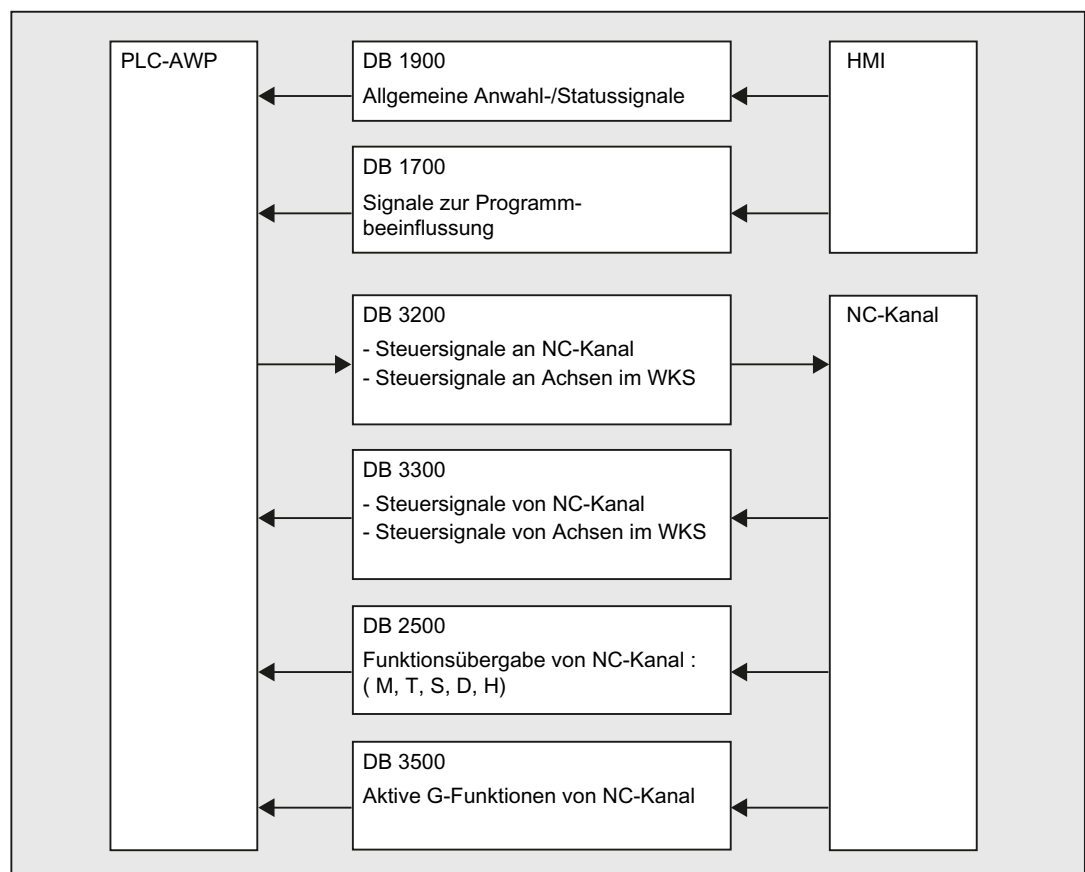
Diese werden auf zweierlei Weise in die Nahtstellen-DBs eingetragen.

Zunächst werden sie mit den Änderungssignalen eingetragen.

Die M-Signale M0 bis M99 werden zusätzlich dekodiert und die zugehörigen Nahtstellenbits für einen Zyklus gesetzt.

Bei den G-Befehlen werden nur die durch die Maschinendaten ausgewählten Gruppen in den Nahtstellen-DB eingetragen.

Die S-Werte werden zusätzlich zusammen mit den zugehörigen M-Signalen (M03, M04, M05) in die spindelspezifische Nahtstelle eingetragen. Ebenso werden achsspezifische Vorschübe in die entsprechende achsspezifische Nahtstelle eingetragen.



### 13.5.1.3 Achs- und Spindelsignale

#### DB3200, 3300, 3700, 3800, 3900

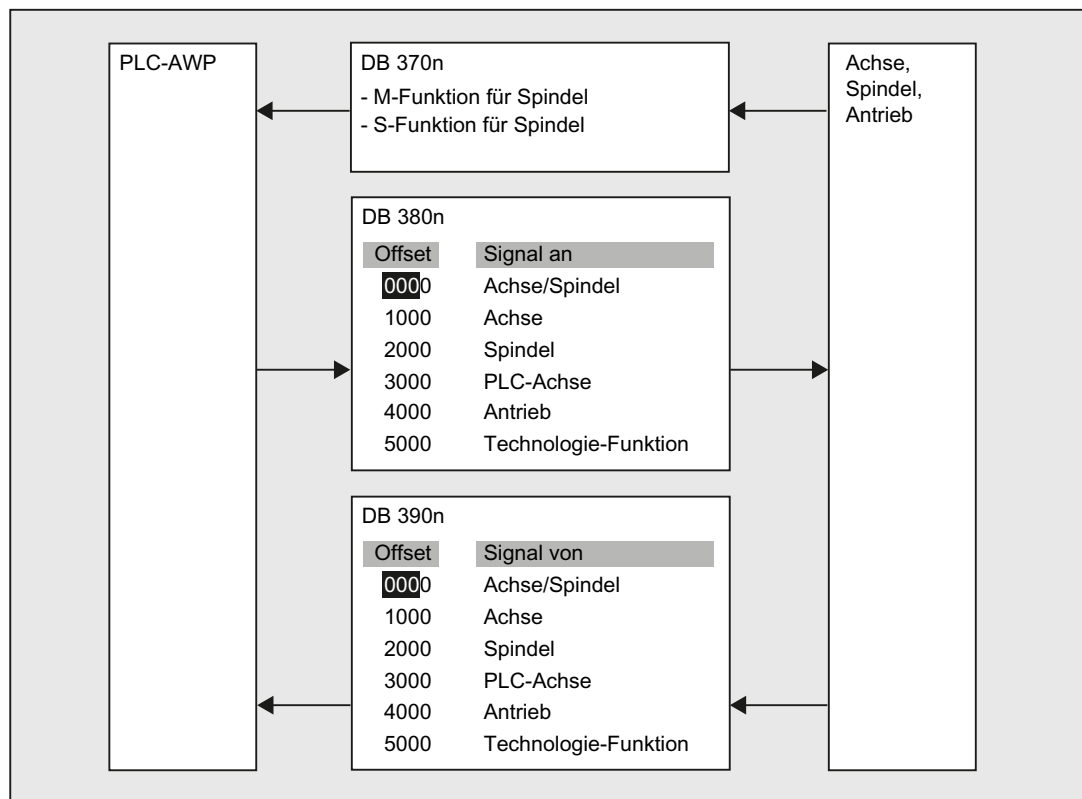
Die achs- und spindelspezifischen Signale sind in folgende Gruppen aufgeteilt:

- Gemeinsame Achs-/Spindelsignale
- Achssignale
- Spindelsignale
- Antriebssignale

Die Signale werden bis auf die im Folgenden beschriebenen Ausnahmen zyklisch übertragen. Zu den Ausnahmen gehören axialer F-Wert, M- und S-Wert.

Ein axialer F-Wert wird über den M-, S-, F-Verteiler dann eingetragen, wenn er im Zuge der NC-Programmbearbeitung an die PLC übertragen wird.

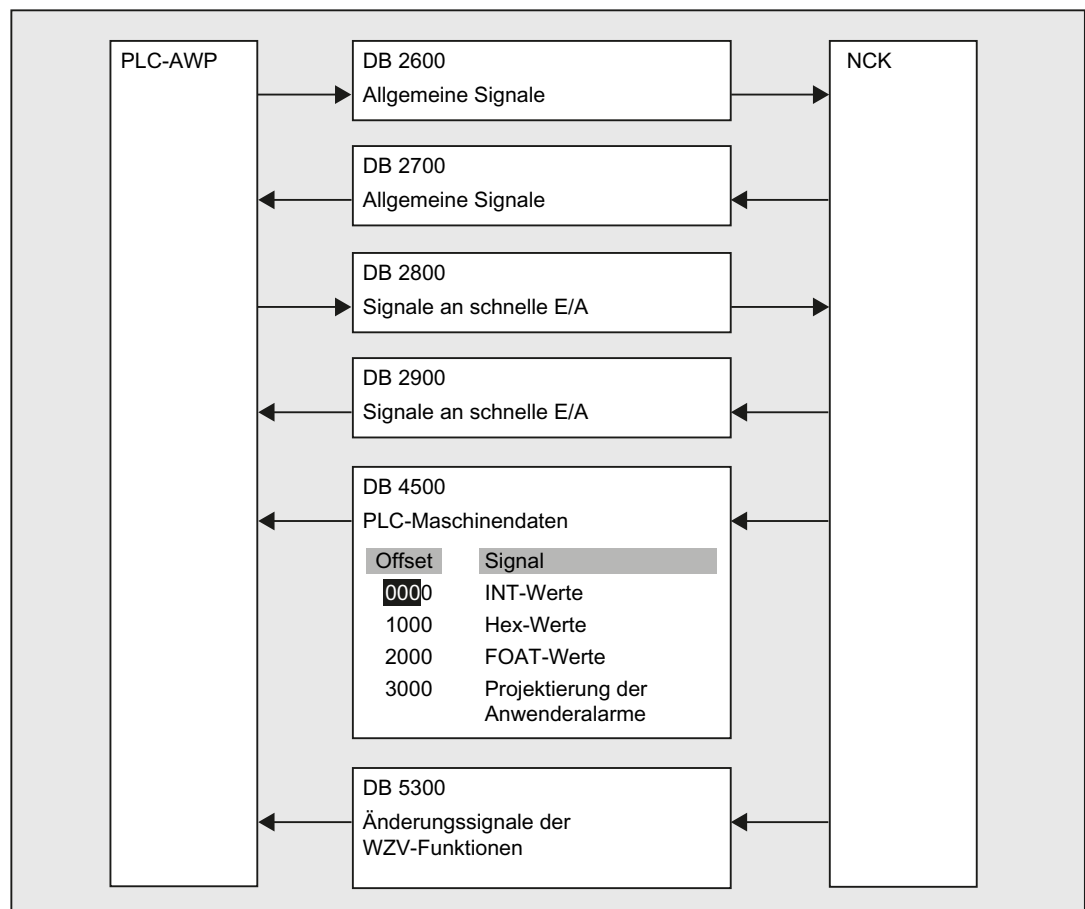
M- und S-Wert werden ebenfalls dann über den M-, S-, F-Verteiler eingetragen, wenn einer oder beide Werte zur Abarbeitung kommen.



### 13.5.1.4 Allgemeine NCK-Signale

#### DB2600, 2700, 2800, 2900, 4500, 5300

- Sollwerte an die digitalen und analogen Ein-/Ausgänge des NCK
- Istwerte von den digitalen und analogen Ein-/Ausgängen des NCK
- Schlüsselschalter- und Not-Aus-Signale
- Bereitschafts- und Statussignale des NCK



### 13.5.1.5 Schneller Datenaustausch PLC-NCK

#### DB4900

Für den schnellen Austausch von Informationen zwischen PLC und NCK ist der Datenbaustein DB4900 mit einer Größe von 1024 Byte vorgesehen.

Die Belegung dieses Bereichs (Struktur) muss im NC-Teilprogramm und PLC-Anwenderprogramm identisch vereinbart werden.

Vom NC-Teilprogramm kann auf diese Daten mit den Befehlen `$A_DBB[x]`, `$A_DBW[x]`, `$A_DBD[x]` und `$A_DBR[x]`;  $0 \leq x \leq 1023$  zugegriffen werden (siehe Listenhandbuch Systemvariablen).

Dabei muss die Ausrichtung der Daten entsprechend ihrem Format gewählt werden, d. h. ein Dword beginnt auf einer 4-Byte-Grenze und ein Word auf einer 2-Byte-Grenze. Bytes können auf einem beliebigen Offset innerhalb des Datenfeldes liegen.

Die Datenkonsistenz wird für Byte-, Word- und Dword-Zugriffe sichergestellt. Bei Übertragung mehrerer Daten ist die Konsistenz anwenderseitig über Semaphore zu sichern, über welche die Gültigkeit bzw. Konsistenz eines Blockes erkannt werden kann.

### 13.5.2 Nahtstelle PLC-HMI

#### DB1700, 1800, 1900

Diese Signale sind bereits in den Bildern des Kapitels Nahtstelle PLC-NCK (Seite 1259) mit angegeben worden.

Noch einmal sei auf das unter Datenschnittstelle (Seite 1259) Gesagte hingewiesen:

Vom HMI kommende und für die NC bestimmte Daten werden nicht automatisch in den NC-Nahtstellenbereich eingetragen. Vielmehr müssen diese Signale und Daten vom AWP rangiert werden.

Es handelt sich um folgende Signale:

- Programmanwahl über Listen
- Messenger Steuerkommando
- Allgemeine Signale von/an HMI
- Signale von/an Wartungsplaner
- Signale von Bedientafel (remanenter Bereich)
- Allgemeine Anwahl-/Statussignale von/an HMI (remanenter Bereich)



## 13.6 Funktionsschnittstelle

### 13.6.1 NC-Variable lesen/schreiben

#### 13.6.1.1 Anwenderschnittstelle

Über die NC/PLC-Nahtstelle "NC-Variable lesen/schreiben" können über das PLC-Anwenderprogramm maximal acht NC-Variablen gleichzeitig gelesen oder geschrieben werden.

Im Rahmen eines Auftrags (lesen/schreiben) sind folgende Schritte auszuführen:

1. Auftragspezifizierung (Seite 1265)
2. Auftragsmanagement: Auftrag starten (Seite 1267)
3. Auftragsmanagement: Warten auf Auftragsende (Seite 1268)
4. Auftragsmanagement: Auftragsabschluss (Seite 1268)
5. Auftragsauswertung (Seite 1270)

Ablaufdiagramm eines Auftrags: siehe "Auftragsmanagement: Ablaufdiagramm (Seite 1269)"

#### 13.6.1.2 Auftragsspezifizierung

Jede Variable, die in einem Auftrag bearbeitet werden soll, muss in der **variablenspezifischen Auftragsschnittstelle** über ihre Parameter spezifiziert werden. Die allgemeinen Bezeichner werden später für jede Variable, die von der Schnittstelle bedient werden kann, konkretisiert.

DB120x <sup>1)</sup>	NC-Daten lesen/schreiben (PLC → NCK)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB1000	Variablenindex							
DBB1001	Bereichsnummer							
DBW1002	Spaltenindex NCK-Variable x							
DBW1004	Zeilenindex NCK-Variable x							
DBW1006	-							
DBD1008	Schreiben: Daten an NCK-Variable x (Datentyp der Variablen: 1...4 Byte)							
1) DB120x, mit x = 0 ... 7 entsprechend Variable 1 ... 8.								

<p><b>ACHTUNG</b></p> <p><b>Antriebsspezifischen Variablen</b></p> <p>Beim Schreiben/Lesen von antriebsspezifischen Variablen dürfen in einem Auftrag über nur Variablen von genau einem Antriebsobjekt SERVO adressiert werden. Das Antriebsobjekt SERVO muss einer Maschinenachse der NC zugeordnet sein. Der Zeilenindex entspricht der logischen Antriebsnummer.</p> <p><b>Fehlerfall</b></p> <p>Im Fehlerfall, Schreiben/Lesen von Variablen von unterschiedlichen Antriebsobjekten, oder gleichzeitig von einem Kanal und einem Antriebsobjekt, erfolgt eine Fehlermeldung:</p> <p>DB1200.DBX3000.1 == 1 (Fehler aufgetreten)</p>
---

**Beispiel: Eine Variable vom Typ "Platztyp" als vierte Variable lesen**

```

DB1203.DBB1000: 7
DB1203.DBB1001: -
DB1203.DBW1002: <Platznummer>
DB1203.DBW1004: <Magazinnummer>
DB1203.DBW1006: -
DB1203.DBD1008: -
    
```

**Beispiel: Eine Variable als vierte Variable schreiben**

Zum Schreiben eines Datums an die NC, muss der Wert in das Doppelwort DBD1008 eingetragen werden:

```

DB1203.DBB1000: <Variablenindex>
DB1203.DBB1001: <Bereichsnummer>
DB1203.DBW1002: <Spaltenindex>
DB1203.DBW1004: <Zeilenindex>
DB1203.DBW1006: -
DB1203.DBD1008: <Wert>
    
```

### 13.6.1.3 Auftragsmanagement: Auftrag starten

Folgende Daten sind vom Anwender in die **globale Auftragsschnittstelle** zu schreiben:

DB120x <sup>1)</sup>	NC-Daten lesen/schreiben (PLC → NCK)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 0							Auftrags- typ	Auftrag: Start
DBB 1	Anzahl der im Auftrag zu bearbeitenden Variablen							
1) DB120x, mit x = 0 ... 7 entsprechend Variable 1 ... 8.								

#### Auftragstyp

- Variable **lesen**: DB1200.DBX0.1 = 0
- Variable **schreiben**: DB1200.DBX0.1 = 1

#### Auftrag starten

Zum Starten des Auftrag über die angegebene Anzahl Variablen ist das Start-Signal zu setzen:

DB1200.DBX0.0 = 1

#### Hinweis

Ein neuer Auftrag kann nur gestartet werden, wenn der vorhergehende Auftrag abgeschlossen wurde. Siehe Kapitel "Auftragsmanagement: Warten auf Auftragsende (Seite 1268)".

Die Dauer der Ausführung eines Auftrages kann mehrere PLC-Zyklen betragen und je nach Auslastung schwanken. Damit ist diese Funktion nicht zeitlich determiniert.

### 13.6.1.4 Auftragsmanagement: Warten auf Auftragsende

Das Auftragsende wird von der NC in der **globalen Ergebnisschnittstelle** immer für den **gesamten** Auftrag zurückgemeldet. Die Signale sind vom PLC-Anwender nur lesbar.

DB120x <sup>1)</sup>	NC-Daten lesen/schreiben (NCK -> PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 2000							Fehler im Auftrag	Auftrag beendet
1) DB120x, mit x = 0 ... 7 entsprechend Variable 1 ... 8.								

#### Auftragsstatus

- Auftragsende ohne Fehler  
DB1200.DBX2000.0 == 1 UND DB1200.DBX2000.1 == 0
- Auftragsende mit Fehler  
DB1200.DBX2000.0 == 1 UND DB1200.DBX2000.1 == 1

#### Mögliche Fehlerursachen

- Anzahl Variablen (DB1200.DBB1) außerhalb des gültigen Bereichs
- Variablenindex (DB1200.DBB1000) außerhalb des gültigen Bereichs
- Gleichzeitiges Lesen/Schreiben von NC-Daten von unterschiedlichen Antriebsobjekten Servo

### 13.6.1.5 Auftragsmanagement: Auftragsabschluss

#### Anforderung

Um den Auftrag abzuschließen, muss vom PLC-Anwenderprogramm, nach dem Erkennen des Auftragsendes, das Start-Signal des Auftrags zurückgesetzt werden:

DB1200.DBX0.1 = 0

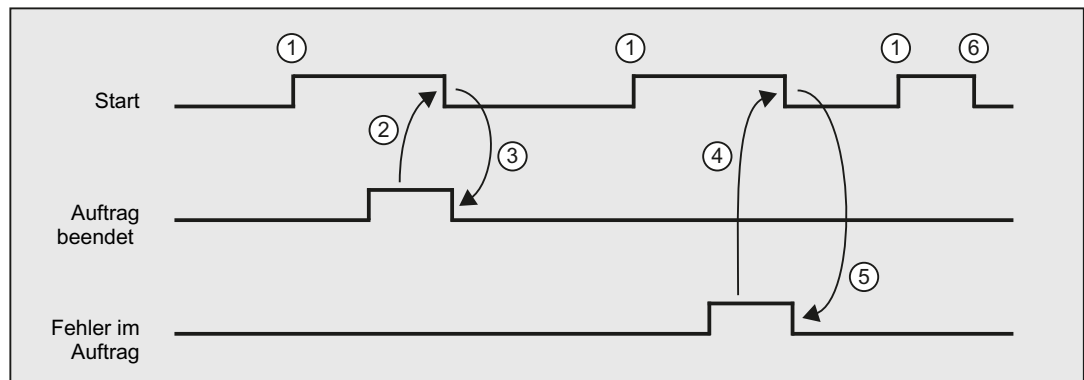
#### Rückmeldung

Als Rückmeldung werden von der NC die Status-Signale zurückgesetzt:

- DB1200.DBX2000.0 == 0
- DB1200.DBX2000.1 == 0

Der Auftrag ist damit abgeschlossen.

## 13.6.1.6 Auftragsmanagement: Ablaufdiagramm



- ① Auftrag starten:  
DB1200.DBX0.0 (Start) = 1
- ② Warten auf Auftragsende:  
DB1200.DBX2000.0 (Auftrag beendet) == 1 UND  
DB1200.DBX2000.1 (Fehler im Auftrag) == 0  
⇒ Auftragsanforderung zurücksetzen:  
DB1200.DBX0.0 = 0 (Start)
- ③ Durch DB1200.DBX0.0 == 0 (Start) wird der Auftrag vom PLC-Grundprogramm abgeschlossen:  
DB1200.DBX2000.0 (Auftrag beendet) = 0
- ④ Warten auf Auftragsende:  
DB1200.DBX2000.0 (Auftrag beendet) == 0 UND  
DB1200.DBX2000.1 (Fehler im Auftrag) == 1  
⇒ Fehlerbehandlung durchführen  
⇒ Auftragsanforderung zurücksetzen:  
DB1200.DBX0.0 (Start) = 0
- ⑤ Durch DB1200.DBX0.0 == 0 (Start) wird der Auftrag vom PLC-Grundprogramm abgeschlossen:  
DB1200.DBX2000.1 (Fehler im Auftrag) = 0
- ⑥ Wird DB1200.DBX0.0 (Start) zurückgesetzt, bevor das Auftragsende vom PLC-Grundprogramm gemeldet wurde, wird der Auftrag ohne weitere Rückmeldung ausgeführt.

### 13.6.1.7 Auftragsauswertung

Die variablenspezifische Ergebnisschnittstelle muss für jede im Auftrag bearbeiteten Variablen ausgewertet werden.

DB120x <sup>1)</sup>	NC-Dienste (NC → PLC)							
DBB3000	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
							Fehler aufgetreten	Variable gültig
DBB3001	Zugriffsergebnis							
DBW3002								
DBD3004	Gelesener Wert der Variablen							
1) DB120x, mit x = 0 ... 7 entsprechend Variable 1 ... 8.								

### Auftragsstatus

#### Auftrag ohne Fehler

- DB1200.DBX3000.0 == 1 (Variable gültig) **UND**
- DB1200.DBX3000.1 == 0 (kein Fehler aufgetreten)

Ergebnis:

- DB1200.DBB3001 == 0 (Zugriffsergebnis: "kein Fehler")
- DB1200.DBD3004 == <gelesener Wert>

#### Auftrag mit Fehler

- DB1200.DBX3000.0 == 0 (Variable nicht gültig) **UND**
- DB1200.DBX3000.1 == 1 (Fehler aufgetreten)

Ergebnis:

- DB1200.DBB3001: mögliche Zugriffsergebnisse siehe unten

### Zugriffsergebnis: DBB3001

Wert	Bedeutung
0	kein Fehler
3	Objektzugriff nicht erlaubt
5	ungültige Adresse
10	Objekt existiert nicht

### 13.6.1.8 Bedienbare Variablen

#### Variable cuttEdgeParam

#### Korrekturwertparameter und Schneidenliste mit D-Nummern zu einem Werkzeug

Die Bedeutung der einzelnen Parameter hängt vom Typ des jeweiligen Werkzeugs ab. Zur Zeit sind 25 Parameter für jede Werkzeugschneide reserviert (jedoch nur ein Teil davon ist mit Werten belegt). Um für zukünftige Erweiterungen flexibel zu sein, sollte jedoch nicht mit einem festen Wert von 25 Parametern sondern mit dem Variablenwert 'numCuttEdgeParams' (Variablenindex 2) gerechnet werden.

Eine detaillierte Beschreibung der Werkzeugparameter ist dem Kapitel W1: Werkzeugkorrektur (Seite 1567) zu entnehmen.

	Variable cuttEdgeParam [r/w]
DB120x.DBB1000	1
DB120x.DBB1001	-
DB120x.DBW1002	(Schneiden-Nr. - 1) * numCuttEdgeParams + ParameterNr (WORD)
DB120x.DBW1004	T-Nummer (1...32000) (WORD)
DB120x.DBD1008	Schreiben: Daten an NCK-Variable x (Datentyp der Variablen: REAL)
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

#### Variable numCuttEdgeParams

#### Anzahl P-Elemente einer Schneide

	Variable numCuttEdgeParams [r]
DB120x.DBB1000	2
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: WORD)

**Variable linShift**

**Translation einer einstellbaren Nullpunktverschiebung (kanalspezifische einstellbare Frames)**

Diese gibt es nur, wenn MD18601 MM\_NUM\_GLOBAL\_USER\_FRAMES > 0 ist.

Folgende Frameindizes gibt es:

- 0: ACTFRAME = aktuelle resultierende Nullpunktverschiebung
- 1: IFRAME = aktuelle einstellbare Nullpunktverschiebung
- 2: PFRAME = aktuelle programmierbare Nullpunktverschiebung
- 3: EXTFRAME = aktuelle externe Nullpunktverschiebung
- 4: TOTFRAME = aktuelle Gesamtnullpunktverschiebung = Summe aus ACTFRAME und EXTFRAME
- 5: ACTBFRAME = aktueller Gesamt-Basisframe
- 6: SETFRAME = aktueller 1. Systemframe (Istwertsetzen, Ankratzen)
- 7: EXTSFRAME = aktueller 2. Systemframe (Istwertsetzen, Ankratzen)
- 8: PARTFRAME = aktueller 3. Systemframe (TCARR und PAROT bei orientierbarem Werkzeugträger)
- 9: TOOLFRAME = aktueller 4. Systemframe (TOROT und TOFRAME)
- 10: MEASFRAME = Ergebnisframe für die Werkstück- und Werkzeugvermessung
- 11: WPFRAME = aktueller 5. Systemframe (Werkstückbezugspunkte)
- 12: CYCFRAME = aktueller 6. Systemframe (Zyklen)

Der max. Frameindex ist 12.

Der Wert von numMachAxes ist in der Variablen mit Variablenindex 4 enthalten.

	Variable linShift [r]
DB120x.DBB1000	3
DB120x.DBB1001	-
DB120x.DBW1002	Frameindex * numMachAxes + Achsnummer
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)



**Variable numMachAxes****Nummer der höchsten existierenden Kanalachse**

Wenn es keine Kanalachslücken gibt, ist dies auch die Anzahl der vorhandenen Achsen im Kanal.

	Variable numMachAxes [r]
DB120x.DBB1000	4
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: WORD)

**Variable rpa****R-Parameter**

	Variable rpa [r/w]
DB120x.DBB1000	5
DB120x.DBB1001	-
DB120x.DBW1002	R-Nummer + 1
DB120x.DBW1004	-
DB120x.DBD1008	Schreiben: Daten an NCK-Variable x (Datentyp der Variablen: REAL)
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variable actLineNumber**

**Zeilennummer des aktuellen NC-Satzes (ab 1)**

- 0: vor Programmstart
- 1: nicht verfügbar wegen Fehler
- 2: nicht verfügbar wegen DISPLOF

	Variable actLineNumber [r]
DB120x.DBB1000	6
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: INT)

**Variable Magazindaten: Platzdaten**

**Platztyp (\$TC\_MPP2)**

Platztyp: \$TC_MPP2			
Adresse	Beschreibung	Gültige Werte	
DB120x.DBB1000	Variablen Index	7	
DB120x.DBB1001	-	-	
DB120x.DBW1002	Platznummer	1 ... 31999	
DB120x.DBW1004	Magazinnummer	1 ... 9999	
DB120x.DBD1008	-	-	
DB120x.DBW3004	Ergebnis: Wert der NCK-Variable x	> 0	Platztyp für virtuellen Platz
	Datentyp: WORD	0	"match all" (Zwischenspeicher)
		9999	undefiniert (kein virtueller Platz)

## Platzzustand (\$TC\_MPP4)

Platzzustand: \$TC_MPP4			
Adresse	Beschreibung	Gültige Werte	
DB120x.DBB1000	Variablen Index	8	
DB120x.DBB1001	-	-	
DB120x.DBW1002	Platznummer	1 ... 31999	
DB120x.DBW1004	Magazinnummer	1 ... 9999	
DB120x.DBD1008	-	-	
DB120x.DBW3004	Ergebnis: Wert der NCK-Variable x	1	gesperrt
		2	frei (<> belegt)
		4	reserviert für Werkzeug im Zwischenspeicher
		8	reserviert für zu beladendes Werkzeug
		16	belegt im linken Halbplatz
		32	belegt im rechten Halbplatz
		64	belegt im oberen Halbplatz
		128	belegt im unteren Halbplatz

## T-Nummer des Werkzeuges auf diesem Platz (\$TC\_MPP6)

T-Nummer des Werkzeuges auf diesem Platz: \$TC_MPP6		
Adresse	Beschreibung	Gültige Werte
DB120x.DBB1000	Variablen Index	9
DB120x.DBB1001	-	-
DB120x.DBW1002	Platznummer	1 ... 31999
DB120x.DBW1004	Magazinnummer	1 ... 9999
DB120x.DBD1008	-	-
DB120x.DBW3004	Ergebnis: T-Nummer des Werkzeuges auf diesem Platz	T-Nummer des Werkzeuges

**Variable r0078[1]**

CO: Stromistwert momentenbildend [Aeff]

Index: [1] = Geglättet mit p0045

	Variable r0078[0...1] [r]
DB120x.DBB1000	10
DB120x.DBB1001	Nummer des Antriebsmoduls
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variable r0079[1]**

CO: Drehmomentsollwert am Ausgang des Drehzahlreglers (vor der Taktinterpolation) [Nm]

Index: [1] = Geglättet mit p0045

	Variable r0079[0...1] [r]
DB120x.DBB1000	11
DB120x.DBB1001	Nummer des Antriebsmoduls
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variable r0081**

CO: Momentenausnutzung in Prozent

	Variable r0081 [r]
DB120x.DBB1000	12
DB120x.DBB1001	Nummer des Antriebsmoduls
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variable r0082[1]**

CO: Wirkleistungswert [kW]

Index: [1] = Geglättet mit p0045

	Variable r0082[0...2] [r]
DB120x.DBB1000	13
DB120x.DBB1001	Nummer des Antriebsmoduls
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variablen: Temperaturkompensation****Variable TEMP\_COMP\_ABS\_VALUE (SD43900)**

Positionsunabhängiger Temperaturkompensationswert

	Variable TEMP_COMP_ABS_VALUE [r/w]
DB120x.DBB1000	14
DB120x.DBB1001	Nr. der Achse (1, 2, ...)
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	Schreiben: Daten an NCK-Variable x (Datentyp der Variablen: REAL)
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variable TEMP\_COMP\_SLOPE (SD43910)**

Steigungswinkel für positionsabhängige Temperaturkompensation

	Variable TEMP_COMP_SLOPE [r/w]
DB120x.DBB1000	15
DB120x.DBB1001	Nr. der Achse (1, 2, ...)
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	Schreiben: Daten an NCK-Variable x (Datentyp der Variablen: REAL)
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variable TEMP\_COMP\_REF\_POSITION (SD43920)**

Bezugsposition für positionsabhängige Temperaturkompensation

	Variable TEMP_COMP_REF_POSITION [r/w]
DB120x.DBB1000	16
DB120x.DBB1001	Nr. der Achse (1, 2, ...)
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	Schreiben: Daten an NCK-Variable x (Datentyp der Variablen: REAL)
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

**Variable TOOL\_TEMP\_COMP (SD42960[...])**

Temperaturkompensation bezogen auf das Werkzeug

	Variable TEMP_COMP_REF_POSITION [r/w]
DB120x.DBB1000	17
DB120x.DBB1001	-
DB120x.DBW1002	Index + 1 (1, 2, 3)
DB120x.DBW1004	-
DB120x.DBD1008	Schreiben: Daten an NCK-Variable x (Datentyp der Variablen: REAL)
DB120x.DBW3004	Lesen: Daten von NCK-Variable x (Datentyp der Variablen: REAL)

## 13.6.2 Programminstanz-Dienste (PI-Dienste)

### 13.6.2.1 Anwenderschnittstelle

#### Auftragsspezifizierung

PI-Dienste werden über deren **Auftragsschnittstelle** (DB1200 ab Offset 4000) über ihre Parameter spezifiziert.

DB1200		PI-Dienst [r/w]						
		Nahtstelle PLC -> NCK						
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 4000								Start
DBB 4001	PI-Index							
DBB 4002								
DBB 4003								
DBW 4004	PI-Parameter 1							
DBW 4006	PI-Parameter 2							
DBW 4008	PI-Parameter 3							
DBW 4010	PI-Parameter 4							
DBW 4012	PI-Parameter 5							
DBW 4014	PI-Parameter 6							
DBW 4016	PI-Parameter 7							
DBW 4018	PI-Parameter 8							
DBW 4020	PI-Parameter 9							
DBW 4022	PI-Parameter 10							

#### PI-Index

DB1200.DBB4001 spezifiziert den konkreten PI-Dienst.

#### PI-Parameter n

Ab DB1200.DBW4004 sind PI-Parameter n dienstspezifisch anzugeben.

#### Start

DB1200.DBX4000.1 = 1: Auftrag für die angegebene Zahl Variablen gestartet.

### Auftragsrückmeldung

Die PLC-Firmware gibt Rückmeldungen über Erfolg oder Nichterfolg des gestarteten PI-Dienstes in der **Ergebnisschnittstelle** (DB1200 ab Offset 5000).

Das Auftragsende wird durch eines von zwei Signalen gemeldet:

DB1200.DBX5000.0 == 1 oder DB1200.DBX5000.1 == 1

Die Signale werden vom PLC-Betriebssystem geschrieben und sind vom Anwender nur lesbar. Ein Auftrag ist abgeschlossen, wenn beide Quittungssignale Null sind. Sie werden Null, wenn der Anwender das Signal "Start" (DB1200.DBX4000.1) nach Auftragsende rücksetzt.

DB1200		PI-Dienst [r]						
		Nahtstelle NCK -> PLC						
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 5000							Fehler im Auftrag	Auftrag beendet
DBB 5001								
DBB 5002								

### Auftrag beendet

DB1200.DBX5000.0 = 1 Auftragsbearbeitung ohne Fehler beendet.

DB1200.DBX5000.0 = 0 sonst. wird Null, wenn Anwender "Start" rücksetzt.

### Fehler im Auftrag

DB1200.DBX5000.1 = 1 Auftrag war fehlerhaft, Ausführung beendet.

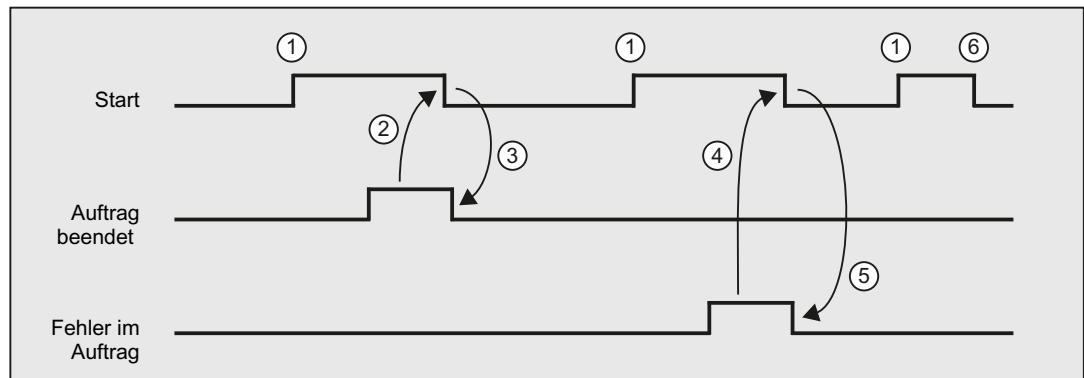
DB1200.DBX5000.1 = 0 sonst. wird Null, wenn Anwender "Start" rücksetzt.

### Mögliche Fehlerursachen

- Index des PI-Dienstes (DB1200.DBB4001) außerhalb des gültigen Bereichs
- Parameterfehler



## PI-Dienste: Taktdiagramm



- ① Anwender setzt Signal "Start", Auftragsausführung beginnt.
- ② Nachdem die PLC-Firmware "Auftrag beendet" signalisiert, setzt der Anwender das Signal "Start" zurück.
- ③ Durch Rücksetzen des Signals "Start" setzt die PLC-Firmware das Signal "Auftrag beendet" zurück.
- ④ Nachdem die PLC-Firmware "Fehler im Auftrag" signalisiert, setzt der Anwender das Signal "Start" zurück.
- ⑤ Durch Rücksetzen des Signals "Start" setzt die PLC-Firmware das Signal "Fehler im Auftrag" zurück.
- ⑥ Wird das Signal "Start" vom Anwender versehentlich zurückgesetzt, ehe eines der Signale "Auftrag beendet" oder "Fehler im Auftrag" kommt, so werden die Ergebnissignale für diesen Auftrag nicht aktualisiert. Der Auftrag wird aber dennoch ausgeführt.

### 13.6.2.2 PI-Dienste

#### PI-Dienst ASUP

#### Funktion

Mit dem PI-Dienst "ASUP" ist es von der PLC aus möglich, den Interruptnummern 1 und 2 jeweils ein Programm zuzuordnen. Diese Programme müssen in der NC im Verzeichnis für den Maschinenhersteller (CMA) mit einem fest vorgegebenen Programmnamen vorliegen:

Interrupt:	Verzeichnis	Programmname
1	_N_CMA_DIR	PLCASUP1_SPF
2	_N_CMA_DIR	PLCASUP2_SPF

Sind die oben genannten Programme nicht vorhanden, müssen sie im Maschinenhersteller-Verzeichnis (CMA) der NC neu erstellt werden. Anschließend muss auf der NC ein Power On-Reset ausgelöst werden.

Der PI-Dienst muss nur einmal nach Neustart ausgeführt werden. Die Zuordnung von Interrupt zu Programm bleibt erhalten.

PI-Dienst: ASUP		
Adresse	Beschreibung	Gültige Werte
DB1200.DBW4001	PI-Index	1, 2
DB1200.DBW4004	LIFTFAST	0, 1
DB1200.DBW4006	BLSYNC	0, 1

PI-Index	Funktion	Interrupt-Priorität
1	Zuordnung: Interrupt 1 zu _N_CMA_DIR/PLCASUP1_SPF	1
2	Zuordnung: Interrupt 2 zu _N_CMA_DIR/PLCASUP2_SPF	2

**Randbedingungen**

- Der PI-Dienst darf nur ausgeführt werden, wenn sich der Kanal im RESET-Zustand befindet.
- Wenn für den ereignisgesteuerten Programmaufruf "Hochlauf" als auslösendes Ereignis projiziert ist, darf der PI-Dienst nur nach Ende des ereignisgesteuerten Programms gestartet werden.

**PI-Dienst LOGOUT**

**Kennwort rücksetzen**

Das zuletzt an den NCK übergebene Kennwort wird zurückgesetzt.

PI-Index	Funktion
3	Kennwort rücksetzen

**Relevante PI-Parameter für PI-Dienste 3**

Keine

**PI-Dienst \_N\_DASAVE**

**Dienst zur Datenrettung von SRAM auf FLASH**

PI-Index	Funktion
4	Datenrettung von SRAM auf FLASH

**Relevante PI-Parameter für PI-Dienste 4**

Keine

## PI-Dienst TMMVTL

### Funktion

Mit dem PI-Dienst TMMVTL ist es von der PLC aus möglich, einen Auftrag zum Umsetzen eines Werkzeugs anzustoßen. Nach fehlerfreiem PI-Start führt die WZV für das WZ auf dem angegebenen Quell-Platz eine Leerplatzsuche im Ziel-Magazin durch. Anschließend erhält die PLC über den DB41xx.DBB0 einen Auftrag zum Umsetzen des Werkzeugs.

PI-Dienst: TMMVTL		
Adresse	Beschreibung	Gültige Werte
DB1200.DBW4001	PI Index	5
DB1200.DBW4004	Werkzeugnummer	-1, 1 ... 31999
DB1200.DBW4006	Quell-Platznummer	-1, 1 ... 31999
DB1200.DBW4008	Quell-Magazinnummer	-1, 1 ... 9999
DB1200.DBW4010	Ziel-Platznummer	-1, 1 ... 31999
DB1200.DBW4012	Ziel-Magazinnummer	-1, 1 ... 9999

- Das Werkzeug kann wahlweise über T-Nummer oder über Platz- und Magazinnummer angegeben werden. Eine nicht benutzte Spezifikation trägt den Wert -1.
- Mit Ziel-Platznummer = -1 wird entsprechend der eingestellten Suchstrategie im gesamten Magazin ein Leerplatz für das Werkzeug gesucht. Ist ein Zielplatz angegeben wird geprüft ob der Platz mit der angegebenen Ziel-Platznummer für das Werkzeug frei und geeignet ist.
- Bei Ziel-Magazinnummer = -1 wird für Werkzeuge auf einem Zwischenspeicher entsprechend der sich aus \$TC\_MDP2 ergebenden Zuordnung gesucht.

### Beispiele

- Bei Verwendung von Zwischenspeichern zum Rücktransport des Werkzeugs (zum Beispiel Toolboy und/oder Shifter) kann im Zuge des asynchronen Rücktransportes eine explizite Leerplatzsuche im Magazin notwendig sein. In diesem Fall muss sich die PLC nicht den Ursprungsplatz merken, dieser PI Dienst sucht nach einem geeigneten Platz.
- Aus einem Hintergrundmagazin soll ein WZ in das Vordergrundmagazin geholt werden.

### 13.6.3 PLC-Anwenderalarme

#### 13.6.3.1 Anwenderschnittstelle

**Hinweis**

Obwohl im Folgenden die Bezeichnung Anwender-"*Alarme*" Verwendung findet, wird erst beim Eintragen des jeweiligen **Löschkriteriums** in den Bits 6 und 7 des Maschinendatums MD14516[x] festgelegt, ob es sich um eine **Meldung** oder um einen **Alarm** handelt.

Die Anwenderalarme werden im HMI angelegt und zur automatisierten Bearbeitung vorbereitet. Die Anwendernahtstelle des DB1600 gestattet es:

- diese 248 Anwenderalarme der Nummern 700000 bis 700247 zu aktivieren,
- mit einem zusätzlichen numerischen Parameter zu versehen und
- zu deaktivieren und zu quittieren sowie
- die durch sie initiierten Systemreaktionen auszuwerten.

#### Aktivierungsschnittstelle der Anwenderalarme

Jeder Anwenderalarm wird über sein zugeordnetes Aktivierungsbit aktiviert. Diese Bits werden in der **Aktivierungsschnittstelle** (DB1600 ab Offset 0) gesetzt:

Ein neuer Anwenderalarm wird mit einer 0/1 Flanke des jeweiligen Bits aktiviert.

DB1600		Aktivierung Alarm [r/w]						
Datenbaustein	Nahtstelle PLC -> HMI							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Aktivierung Alarm-Nr.							
<b>DBB 0</b>	700007	700006	700005	700004	700003	700002	700001	700000
	Aktivierung Alarm-Nr.							
<b>DBB 1</b>	700015	700014	700013	700012	700011	700010	700009	700008
	Aktivierung Alarm-Nr.							
<b>DBB 2</b>	700023	700022	700021	700020	700019	700018	700017	700016
	Aktivierung Alarm-Nr.							
<b>DBB 3</b>	700031	700030	700029	700028	700027	700026	700025	700024
	Aktivierung Alarm-Nr.							
<b>DBB 4</b>	700039	700038	700037	700036	700035	700034	700033	700032
	Aktivierung Alarm-Nr.							
<b>DBB 5</b>	700047	700046	700045	700044	700043	700042	700041	700040
	....							
...								
	Aktivierung Alarm-Nr.							
<b>DBB 30</b>	700247	700246	700245	700244	700243	700242	700241	700240

### Variablenschnittstelle der Anwenderalarme

Jedem Anwenderalarm kann eine Variable als Parameter mitgegeben werden. In der **Variablenschnittstelle** ab Offset 1000 ist dafür je ein Doppelwort reserviert. Gültige Offsets müssen demzufolge durch 4 teilbar sein.

DB1600	Variable für Alarm [r32/w32]
Datenbaustein	Nahtstelle PLC -> HMI
DBD 1000	Variable für Alarm 700000
DBD 1004	Variable für Alarm 700001
DBD 1008	Variable für Alarm 700002
	...
DBD 1980	Variable für Alarm 700245
DBD 1984	Variable für Alarm 700246
DBD 1988	Variable für Alarm 700247

### Alarmreaktion und Löschkriterium

#### Alarmreaktion und Löschkriterium

Diese beiden Begriffe betreffen die Konzeption und Projektierung von Anwenderalarmen. Für jeden Alarm können folgende Attribute festgelegt werden:

- Alarmreaktion: wie die Steuerung beim Auftreten des Alarms reagiert.
- Löschkriterium: was muss unternommen werden, um den Alarm wieder zu löschen oder zu quittieren. Das Löschkriterium bestimmt gleichzeitig die Priorität des Alarms.

Die Sollwerte werden in folgendem Maschinendatum konfiguriert und ihre Istwerte gemeinsam in einem Byte der Anwendernahtstelle gemeldet: Die Konfiguration der Sollwerte erfolgt bitcodiert in: MD14516[x],  $0 \leq x \leq 247$ ; wobei x = Anwenderalarmnummer - 700000

**Aktive Alarmreaktion und Löschkriterium**

Die momentan **aktiven Alarmreaktionen** (also die IST-Reaktionen) und die **aktiven Löschkriterien** können aus der Schnittstelle global ausgelesen werden:

DB1600	Aktive Alarmreaktion [r]							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 2000	POWER ON	Abbruch mit DB1600 DBX3000.0		PLC-Stop	Not-Halt	Vorschub-sperre aller Achsen	Einlese-sperre	NC-Startsperre
DBB 2001								
DBB 2002								
DBB 2003								

Ein Bit wird gesetzt, wenn mindestens für einen aktiven Alarm die entsprechende Reaktion oder das entsprechende Löschkriterium konfiguriert ist. Es wird gelöscht, wenn für keinen der anstehenden Alarme mehr diese Reaktion/Löschkriterium konfiguriert ist.

Die Codierungen der Löschkriterien und die entsprechenden Prioritäten sind:

Bit 7	Bit 6	Anwenderalarm wird quittiert durch	Typ	Priorität
0	0	Rücksetzen des Aktivierungsbits	Meldung	niedrig
0	1	Quittieren in DB1600.DBX3000.0 (s. u.)	Alarm	mittel
1	0	power on	Alarm	hoch
1	1	reserviert (intern gewertet als {1, 0})		

**Hinweis**

Wird für einen Alarm/eine Meldung in den Maschinendaten keines der Bits 0 bis 4 gesetzt, wird damit festgelegt, dass es sich um eine sogenannte "Anzeigemeldung" ohne Wirkung auf das System handelt. Dies bedeutet insbesondere, dass auch die Bits 6 und 7 des entsprechenden Maschinendatums nicht ausgewertet werden.

### Aktive Alarmreaktionen und Löschkriterien der Anwenderalarme

Die momentan **aktiven Alarmreaktionen** (also die IST-Reaktionen) und die **aktiven Löschkriterien** können aus der Schnittstelle global ausgelesen werden.

DB1600		Aktive Alarmreaktion [r]						
Datenbaustein								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 2000	POWER ON	Abbruch mit DB1600 DBX3000.0		PLC-Stop	Not-Halt	Vorschub-sperre aller Achsen	Einlese-sperre	NC-Startsperre
DBB 2001								
DBB 2002								
DBB 2003								

Ein Bit wird gesetzt, wenn mindestens für einen aktiven Alarm die entsprechende Reaktion oder das entsprechende Löschkriterium konfiguriert ist. Es wird gelöscht, wenn für keinen der anstehenden Alarme mehr diese Reaktion/Löschkriterium konfiguriert ist.

### Quittierungsschnittstelle der Anwenderalarme

#### Voraussetzung

Voraussetzung für das Quittieren eines Anwenderalarms ist das Rücksetzen des entsprechenden Aktivierungsbits.

- Meldungen mit Löschkriterium {0,0} verschwinden dann selbständig aus der Anzeige.
- Alarme mit Löschkriterium {0,1} werden durch das Quittungsbit *Quit* gelöscht.
- Alarme mit Löschkriterium {1,0} bleiben vom Setzen des Quittungsbits unbeeinflusst, sie können nur durch Power On gelöscht werden.

DB1600		Alarmquittung [r/w]						
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB3000								Quit
DBB3001								
DBB3002								
DBB3003								

## Schnittstelle zum HMI

Die PLC kann dem HMI acht Meldungen oder Alarme zur Anzeige übergeben, die in der Reihenfolge ihres Auftretens angezeigt werden.

Beim Auftreten weiterer Meldungen/Alarme bleiben die ersten sieben im HMI erhalten, und die jüngste Meldung bzw. der jüngste Alarm wird nach folgenden Regeln von einem gerade aufgetretenen verdrängt:

- Systemmeldung/-alarm verdrängt Anwendermeldung/-alarm,
- Meldungen/Alarme höherer Priorität verdrängen solche niederer Priorität.

Die ersten sieben Meldungen/Alarme verbleiben deshalb in der Anzeige, weil es sehr wahrscheinlich ist, dass diese die Ursache des Problems benennen und die folgenden lediglich sekundärer Art sind. Werden jedoch eine oder mehrere Meldungen/Alarme quittiert und damit gelöscht, rückt eine entsprechende Anzahl neu hinzugekommener Alarme/Meldungen im HMI nach.

## 13.6.4 PLC-Achssteuerung

### 13.6.4.1 Überblick

#### Ziel

Von der PLC können acht Achsen oder Spindeln über Datenbausteine der Anwendernahtstelle gesteuert werden, dabei wird die Achse/Spindel über die DB-Nummer spezifiziert:

- **DB380x** Nahtstelle PLC → NCK
- **DB390x** Nahtstelle NCK → PLC
- **DB390x** Achsindex:  $0 \leq x \leq 7$ , wobei Achsindex = Achsnummer-1

Folgende Funktionen werden unterstützt:

- Positionierachsen
- Spindel positionieren
- Spindel drehen
- Spindel pendeln
- Teilungsachsen

#### Literatur:

- Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2) und Teilungsachsen (T1)
- Funktionshandbuch Grundfunktionen; Spindeln (S1)



## Voraussetzung

Die zu steuernde Achse muss der PLC zugeordnet sein. Über die Anwendernahtstelle "*Achstausch*" (DB3800.DBB8/DB3900.DBB8) kann eine Achse zwischen NC und PLC getauscht werden.

## Funktionsstart

Jede Funktion wird durch die positive Flanke des entsprechenden Signals "Start" angestoßen. Dieses Signal muss solange auf logisch "1" bleiben, bis die Funktion positiv oder negativ quittiert wurde (z. B. durch *Position erreicht* = "1" oder *Fehler* = "1"). Das Signal "Positionierachse aktiv" = "1" zeigt an, dass die Funktion aktiv ist und die Ausgangssignale gültig sind.

## Abbruch

Das **Abbrechen** der Funktion ist über Rücksetzen des Startsignals **nicht** möglich, sondern nur über andere Nahtstellensignale (etwa über das achsspezifische Signal *Restweg löschen/Spindel-Reset*, DB380x DBX2.2).

Die Achsnahtstelle liefert Statussignale der Achse zurück, die gegebenenfalls auszuwerten sind (z. B. *Genauhalt*, *Fahrbefehl*, → DB390x).

Wenn die Achse/Spindel bei Aufruf der PLC-Achssteuerung infolge Fahrens durch NC-Programm belegt ist (Fahrbefehle stehen an), dann wird die Funktion erst nach Beenden dieser Bewegung gestartet. In dieser Situation wird kein Fehlercode ausgegeben.

## Achssperre

Bei gesetzter **Achssperre** bewegt sich eine über PLC-Achssteuerung gesteuerte Achse nicht. Es wird nur ein simulierter Istwert erzeugt. (Verhalten wie bei NC-Programmierung).

**13.6.4.2 Anwenderschnittstelle: Vorbereiten der NC-Achse als PLC-Achse**

Zunächst muss die Achse/Spindel von der PLC angefordert werden. Dies geschieht über folgende Anwenderschnittstelle:

**Gemeinsame Signale an Achse/Spindel (Auszug)**

Achse oder Spindel anfordern:

DB3800 ... 3807			Signale an Achse/Spindel [r/w]					
Datenbaustein			Nahtstelle PLC → NCK					
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 0008	PLC-Achse/ Spindel anfordern			Aktivier- signal bei Verän- derung dieses Bytes				NC-Achse/ Spindel anfordern

Jede Veränderung eines Anforderungsbits auf der Nahtstelle (DB380x.DBX0008.7 oder Bit DB380x.DBX0008.0) muss der NC über eine 0→1-Flanke des Aktiviersignals (Bit DB380x.DBX0008.4) mitgeteilt werden. Nach einem Zyklus ist dieses Aktiviersignal wieder zurückzusetzen.

**Allgemeine Signale von Achse/Spindel (Auszug)**

Eine Zustandsabfrage ist über die Nahtstelle DB390x.DBB8 möglich. Bei Simulation muss dazu allerdings das Achsmaschinendatum MD30350 MA\_SIMU\_AX\_VDI\_OUTPUT gesetzt sein.

DB3900 ... 3907			Signale von Achse/Spindel [r]					
Datenbaustein			Nahtstelle NCK → PLC					
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 0008	PLC-Achse/ Spindel	neutrale Achse/ Spindel	Achstausch möglich	neuer Typ von PLC gefordert				NC-Achse/ Spindel

PLC-Achse anfordern und abgeben

t	DB380x.DB0008	DB390x.DB0008	Bemerkung
	0 0 0 0 0 0 0 0	0 x 1 0 0 0 0 0 1	PLC erhält Signal "Achstausch möglich", "NC-Achse" und evtl. "neutrale Achse/Spindel" (x).
	1 0 0 1 0 0 0 0	1 0 0 0 0 0 0 1	PLC fordert Achse an.
	1 0 0 0 0 0 0 0	1 0 1 0 0 0 0 1	PLC setzt Aktiviersignal zurück.
..... → Achse ist PLC zugeordnet, kann verwendet werden und wird danach an NC abgegeben:			
	0 0 0 1 0 0 0 1	1 0 0 0 0 0 0 1	PLC gibt Achse an NC zurück.
	0 0 0 0 0 0 0 1	0 0 1 0 0 0 0 1	PLC setzt Aktiviersignal zurück; Status "NC-Achse" und "Achstausch möglich".
..... → Achse ist PLC zugeordnet, kann verwendet werden und wird danach freigegeben:			
	0 0 0 1 0 0 0 0	0 1 0 0 0 0 0 1	PLC gibt Achse frei.
	0 0 0 0 0 0 0 0	0 1 1 0 0 0 0 1	PLC setzt Aktiviersignal zurück; Status "neutrale Achse/Spindel", "NC-Achse" und "Achstausch möglich".

**13.6.4.3 Anwenderschnittstelle: Funktionalität**

Die beiden Tabellen geben einen Überblick über die verfügbaren Nahtstellensignale. Die genaue Beschreibung der Signale und die Erläuterung, welche Signale für die einzelnen Funktionen relevant sind, wird im Folgenden erläutert.

**Signale an PLC-Achse**

DB3800...3807		Signale an PLC-Achse [r/w]						
Datenbaustein		Nahtstelle PLC -> NCK						
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 3000	Start Positionierachse	Start Spindelpositionieren	Start Spindeldrehen	Start Spindelpendeln				
DBB 3001			Stop Spindeldrehen	Stop Spindelpendeln				
DBB 3002	Automatische Getriebeanwahl	konstante Schnittgeschwindigkeit	Drehrichtung wie M4		Handradüberlagerung ein	Verfahrmaß Inch (nicht metrisch)	Wegbedingung kürzester Weg (DC)	Wegbedingung inkrementell (IC)
DBB 3003	Teilungsposition						Wegbedingung abs. pos. Richtung (ACP)	Wegbedingung abs. neg. Richtung (ACN)
DBD 3004	Position (REAL, bei Teilungsachse: DINT)							
DBD 3008	Vorschubgeschwindigkeit (REAL), wenn < 0, wird Wert aus Maschinendatum POS_AX_VELO genommen							

Die Bits der Wegbedingungen und der Drehrichtungsvorgabe geben den jeweiligen Positionier- oder Verfahrmodus vor, es darf nur eines der Bits gesetzt werden:

Bedeutung	Zu setzende Wegbedingung
Positionieren absolut	Kein Mode-Bit gesetzt
Positionieren inkrementell	DBB3002.0 = 1
Positionieren kürzester Weg	DBB3002.1 = 1
Positionieren absolut, positive Anfahrriichtung	DBB3003.1 = 1
Positionieren absolut, negative Anfahrriichtung	DBB3003.0 = 1
Drehrichtung wie M4	DBB3002.5 = 1

Die übrigen Bits dienen dem Spezifizieren und Starten der jeweiligen Funktion, diese Funktionsbits sowie Position und Geschwindigkeit werden bei den einzelnen Funktionen näher erläutert.

## Signale von PLC-Achse

3900...3907		Signale von PLC-Achse [r]						
Datenbaustein		Nahtstelle PLC -> NCK						
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
3000	Positionierachse aktiv	Position erreicht					Fehler während Verfahren	Achse nicht startbar
3001								
3002								
3003	Fehlernummer							

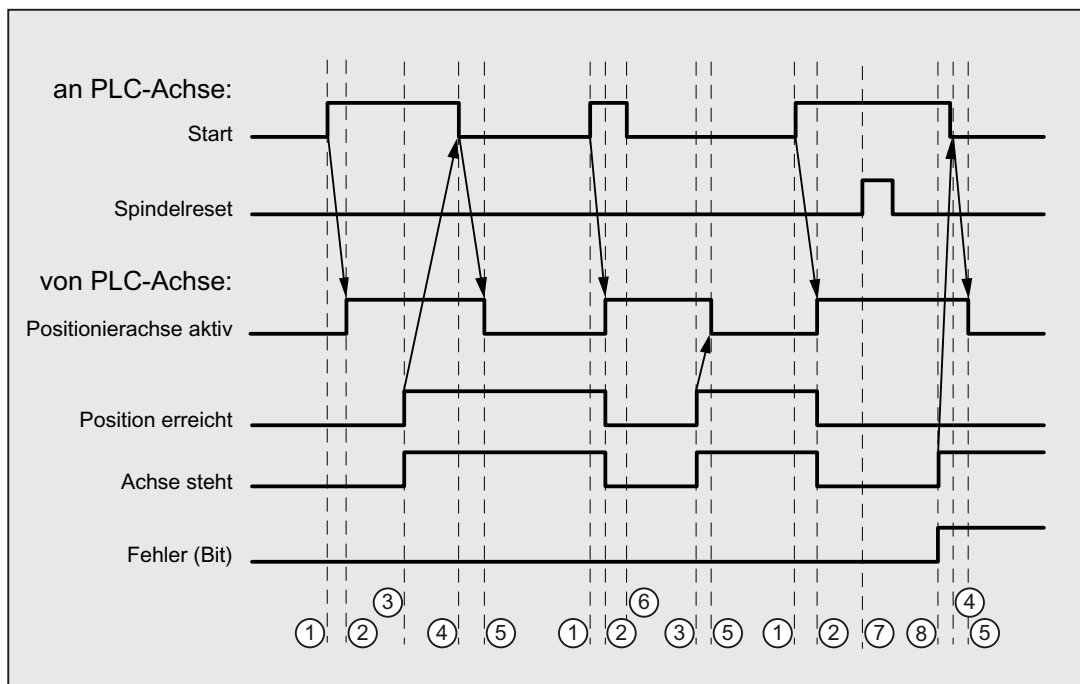
Für die Verwendung der nachfolgend aufgeführten Funktionen müssen folgende Voraussetzungen erfüllt sein:

- Die Achse oder Spindel ist ordnungsgemäß der PLC zugeordnet.
- Regler- und Impulsfreigabe sind gesetzt.
- Nach Setzen aller Steuersignale, ist nur eines der Startsignale in DB380x.DBB3000 gesetzt.

## 13.6.4.4 Spindel positionieren

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0 1	Es darf nur eines der Bits gesetzt werden, sind alle Bits 0, bedeutet dies absolut Positionieren.
DBX3002.1	kürzester Weg	0 1	
DBX3003.0	absolut, negative Richtung	0 1	
DBX3003.1	absolut, positive Richtung	0 1	
DBD3004	Sollposition/Sollweg	REAL	bei "inkrementell": Sollweg
DBD3008	Vorschubgeschwindigkeit	REAL	wenn = 0, wird Wert aus MD35300 \$MA_SPIND_POSCTRL_VELO (Lageregeleinschalt Drehzahl) genommen
DBX3000.6	Start	0 1	Rücksetzen führt nicht zum Stopp!
DBX2.2	Restweg löschen, Spindelreset	0 1	Abbruchsignal, beendet die Funktion

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	Auch dann 1, wenn Override = 0 oder Sollposition erreicht wenn Start = 1
DBX3000.6	Position erreicht	1: Sollposition mit "Genauhalt Fein" erreicht
DBX3000.0	Spindel nicht startbar	
DBX3000.1	Fehler während Verfahren	1: Fehler während des Verfahrens, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	
DBX1.4	Achse/Spindel steht	1: wenn $n < n_{min}$

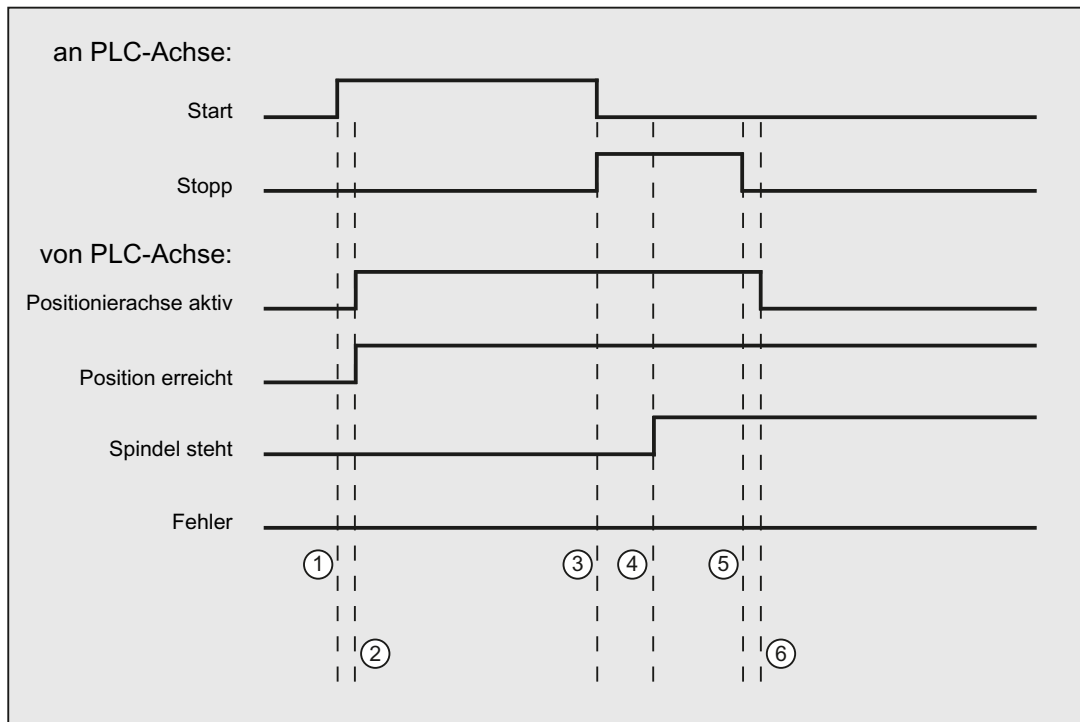


- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.
- ② Meldung *Positionierachse aktiv* zeigt an, dass die Funktion aktiv ist und die Ausgangssignale gültig sind, *Position erreicht* und *Achse steht* werden gegebenenfalls zurückgenommen. Bei Wegvorgabe = 0 werden die Signale nicht zurückgenommen.
- ③ Das Erreichen der Position wird gemeldet (*Position erreicht*), *Spindel steht* wird gesetzt.
- ④ Der Anwender nimmt daraufhin *Start* zurück.
- ⑤ Daraufhin wird das Signal *Positionierachse aktiv* zurückgesetzt.
- ⑥ Der Anwender nimmt das Signal *Start* mit Erhalt des Signals *Positionierachse aktiv* sofort zurück.
- ⑦ Das Positionieren wird über das Setzen von *Spindel Reset* abgebrochen. Dieses Signal muss mindestens einen PLC-Takt anstehen.
- ⑧ Die Spindel kommt zum Stillstand (*Spindel steht*), das Signal *Fehler* wird gesetzt. (In diesem Fall wird die Fehlernummer 115 ausgegeben.)

## 13.6.4.5 Spindel drehen

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0	<i>Drehrichtung wie M4:</i> 1: Drehrichtung durch M4 vorgegeben 0: Drehrichtung durch M3 vorgegeben
DBX3002.1	kürzester Weg	0	
DBX3002.5	Drehrichtung wie M4	0 1	
DBX3003.0	absolut, negative Richtung	0	
DBX3003.1	absolut, positive Richtung	0	
DBD3008	Vorschubgeschwindigkeit	REAL	Spindeldrehzahl
DBX3000.5	Start Spindel drehen	0 1	
DBX3001.5	Stopp Spindel drehen	0 1	

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	1: bei Start oder Stopp == 1
DBX3000.6	Position erreicht	1: Funktion wurde ohne Fehler gestartet
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	
DBX1.4	Achse/Spindel steht	



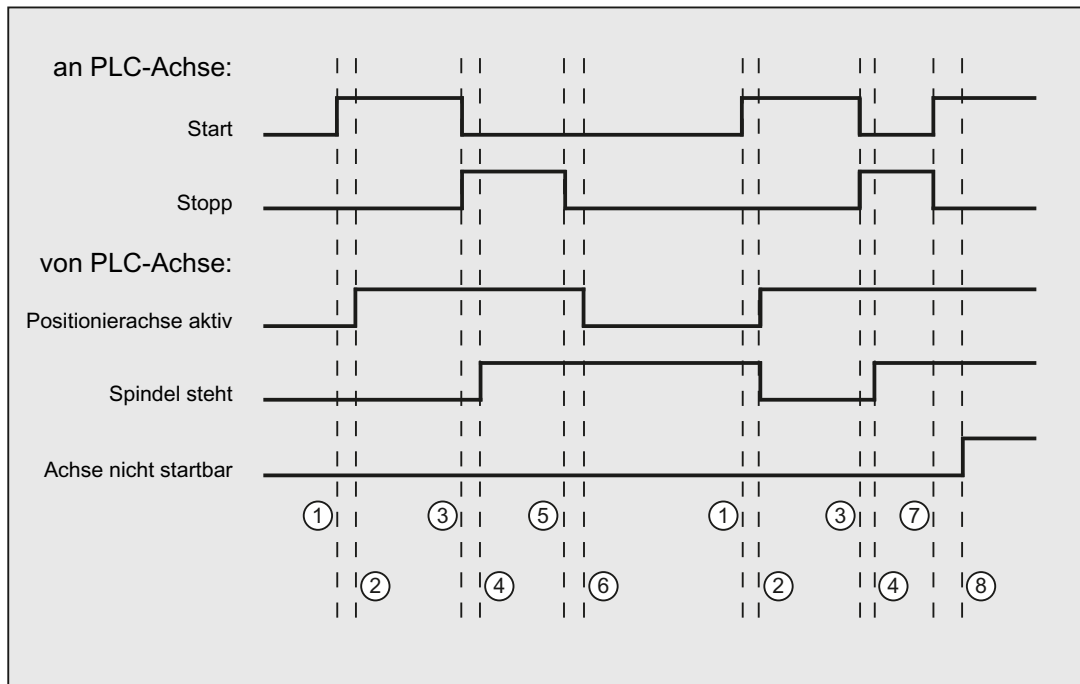
- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.
- ② Meldungen *Positionierachse aktiv* und *Position erreicht* werden zurückgemeldet, *Position erreicht* ist dabei irrelevant.
- ③ Das Drehen der Spindel wird vom Anwender dadurch gestoppt, dass *Start* zurückgesetzt und *Stopp* gesetzt wird.
- ④ Die Spindel hält an und das Signal *Spindel steht* wird gesetzt.
- ⑤ Daraufhin setzt der Anwender *Stopp* zurück.
- ⑥ Das Rücksetzen von *Stopp* bewirkt das Rücksetzen von *Positionierachse aktiv*.



## 13.6.4.6 Spindel pendeln

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0	Es darf keines der Bits gesetzt werden.
DBX3002.1	kürzester Weg	0	
DBX3002.5	Drehrichtung wie M4	0	
DBX3003.0	absolut, negative Richtung	0	
DBX3003.1	absolut, positive Richtung	0	
DBD3004	Sollgetriebestufe: Werte 0, 1, 2, 3, 4, 5	REAL	MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE = 0 0-5: Pendeln MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE = 1 0: Pendeln 1: Pendeln mit Getriebestufenwechsel M41 2: Pendeln mit Getriebestufenwechsel M42 3: Pendeln mit Getriebestufenwechsel M43 4: Pendeln mit Getriebestufenwechsel M44 5: Pendeln mit Getriebestufenwechsel M45
DBD3008	Vorschubgeschwindigkeit	REAL	Beim Pendeln ohne Bedeutung! Die Pendeldrehzahl stammt aus dem Maschinendatum MD35400, \$MA_SPIND_OSCILL_DES_VELO.
DBX3000.5	Start Spindel pendeln	0 1	Start darf nicht unmittelbar auf Stopp folgen. Erst muss Stopp rückgesetzt werden (beide 0).
DBX3001.5	Stopp Spindel pendeln	0 1	

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	1: bei Start oder Stopp == 1
DBX3000.6	Position erreicht	1: nach Start
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBX3000.0	Achse nicht startbar	1: Fehler beim Start, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	
DBX1.4	Achse/Spindel steht	

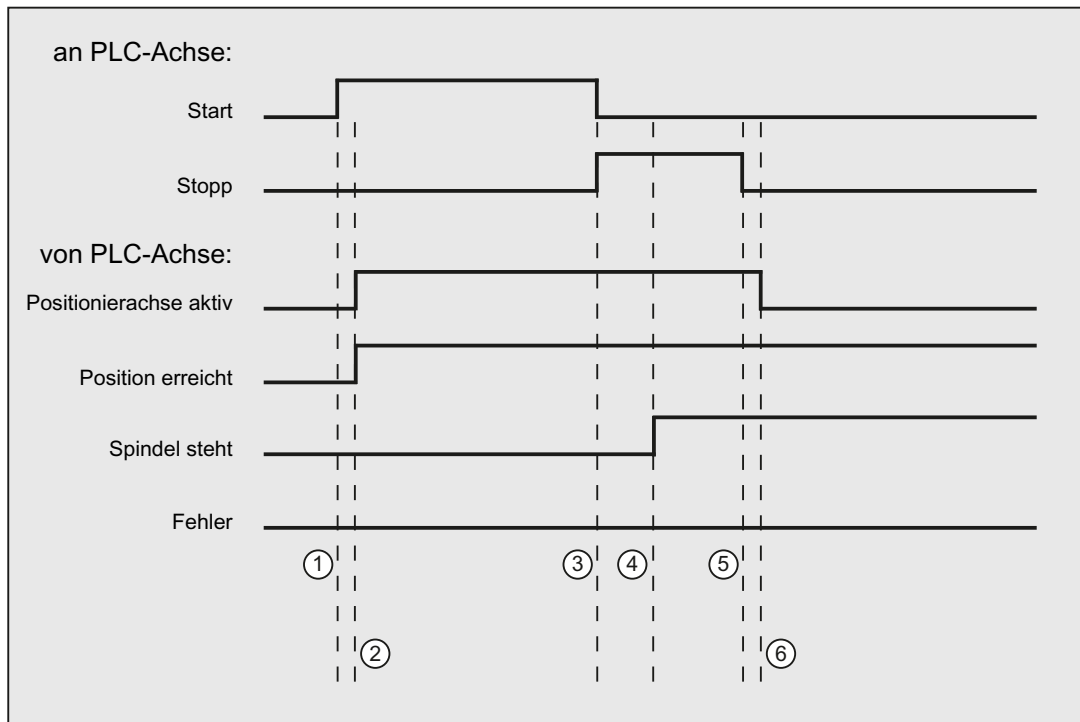


- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.  
**Hinweis:** Dies ist nur bei rückgesetztem Signal *Positionierachse aktiv* möglich!
- ② Meldungen *Positionierachse aktiv* und *Position erreicht* werden zurückgemeldet, *Position erreicht* ist dabei irrelevant und darum nicht eingezeichnet.
- ③ Das Pendeln der Spindel wird vom Anwender dadurch gestoppt, dass *Start* zurückgesetzt und *Stopp* gesetzt wird.
- ④ Die Spindel hält an und das Signal *Spindel steht* wird gesetzt.
- ⑤ Daraufhin setzt der Anwender *Stopp* zurück.
- ⑥ Das Rücksetzen von *Stopp* bewirkt das Rücksetzen von *Positionierachse aktiv*.
- ⑦ Im AWP wird *Stopp* zurückgesetzt und unrichtigerweise im gleichen PLC-Zyklus *Start* auch wieder gesetzt. Dadurch wird *Positionierachse aktiv* nicht rückgesetzt, sondern...
- ⑧ ...es wird das Signal *Achse nicht startbar* gesetzt (Fehlernummer 106).

## 13.6.4.7 Teilungsachse

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0 1	Es darf nur eines der Bits gesetzt werden, sind alle Bits 0, bedeutet dies absolut Positionieren.
DBX3002.1	kürzester Weg	0 1	
DBX3002.5	Drehrichtung wie M4	0	
DBX3003.0	absolut, negative Richtung	0 1	
DBX3003.1	absolut, positive Richtung	0 1	
DBD3003.7	Teilungsposition	1	Teilungsachse EIN
DBD3004	Sollposition/Sollweg	DINT	bei "inkrementell": Sollweg
DBD3008	Vorschubgeschwindigkeit	REAL	wenn = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
DBX3000.7	Start Positionierachse	0 1	Rücksetzen führt nicht zum Stopp!
DBX2.2	Restweg löschen, Spindelreset	0 1	Abbruchsignal, beendet die Funktion

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	Auch dann 1, wenn Override = 0 oder Sollposition erreicht.
DBX3000.6	Position erreicht	1: Sollposition mit "Genauhalt Fein" erreicht.
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	
DBX1.4	Achse/Spindel steht	

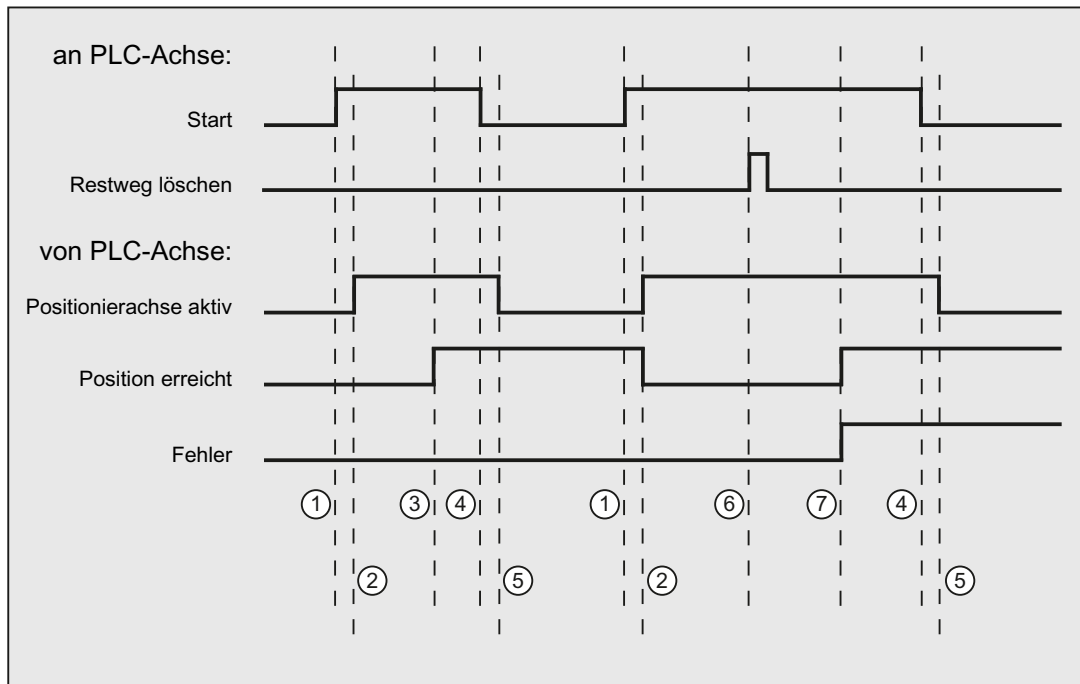


- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.  
**Hinweis:** Dies ist nur bei rückgesetztem Signal *Positionierachse aktiv* möglich!
- ② Meldungen *Positionierachse aktiv* und *Position erreicht* werden zurückgemeldet, *Position erreicht* ist dabei irrelevant.
- ③ Das Pendeln der Spindel wird vom Anwender dadurch gestoppt, dass *Start* zurückgesetzt und *Stopp* gesetzt wird.
- ④ Die Spindel hält an und das Signal *Spindel steht* wird gesetzt.
- ⑤ Daraufhin setzt der Anwender *Stopp* zurück.
- ⑥ Das Rücksetzen von *Stopp* bewirkt das Rücksetzen von *Positionierachse aktiv*.

## 13.6.4.8 Positionierachse metrisch

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0 1	Es darf nur eines der Bits gesetzt werden, sind alle Bits 0, bedeutet dies absolut Positionieren.
DBX3002.1	kürzester Weg	0 1	
DBX3002.5	Drehrichtung wie M4	0	
DBX3003.0	absolut, negative Richtung	0 1	
DBX3003.1	absolut, positive Richtung	0 1	
DBD3002.2	Verfahrmaß Inch	0	Verfahrmaß metrisch
DBD3002.3	Handradüberlagerung	0	Überlagerung AUS
DBD3004	Sollposition/Sollweg	REAL	bei "inkrementell": Sollweg
DBD3008	Vorschubgeschwindigkeit	REAL	wenn = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
DBX3000.7	Start Positionierachse	0 1	Rücksetzen führt nicht zum Stopp!
DBX2.2	Restweg löschen, Spindelreset	0 1	Abbruchsignal, beendet die Funktion

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	Auch dann 1, wenn Override = 0 oder Sollposition erreicht.
DBX3000.6	Position erreicht	1: Achse hat Sollposition erreicht.
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	

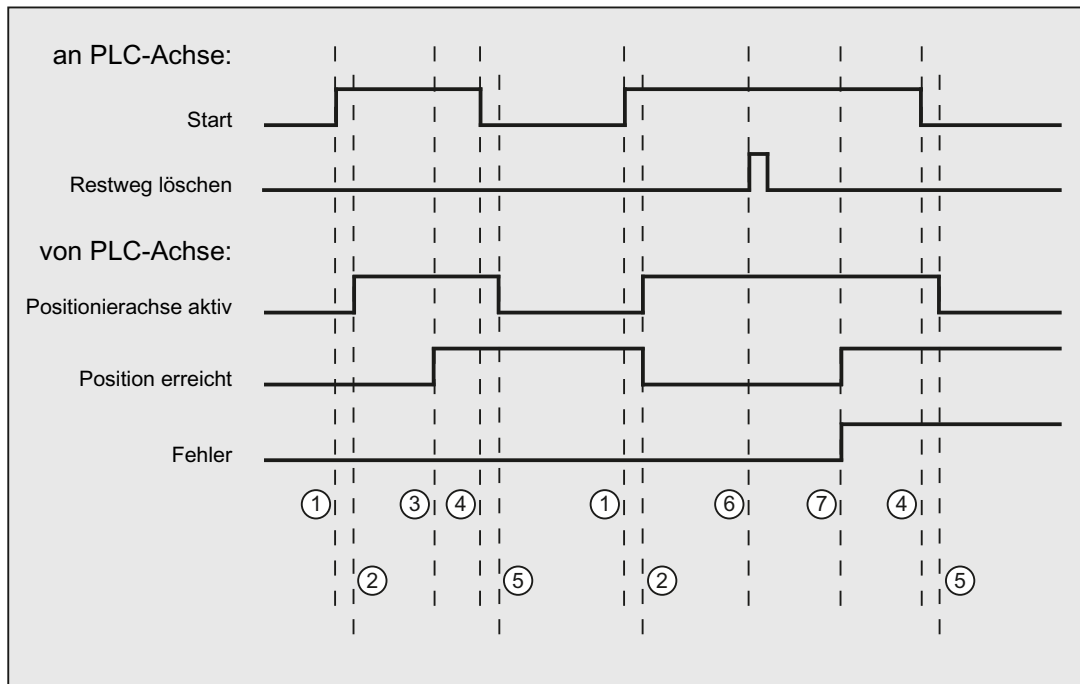


- ① Erster Funktionsanstoß durch positive Flanke von *Start*.
- ② *Positionierachse aktiv* = 1 zeigt an, dass die Funktion aktiv ist und die Ausgangssignale gültig sind, *Position erreicht* und *Achse steht* werden gegebenenfalls zurückgenommen.
- ③ Positive Quittung *Position erreicht* = 1 und *Positionierachse aktiv* = 1
- ④ Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- ⑤ Signalwechsel durch Funktion
- ⑥ Positionierung wird durch Restweg löschen abgebrochen, Signaldauer mind. 1 PLC-Zyklus.
- ⑦ Die Signale *Position erreicht* und *Fehler* werden gesetzt, die *Fehlernummer* (in diesem Fall 30) kann gelesen werden.

## 13.6.4.9 Positionierachse Inch

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0 1	Es darf nur eines der Bits gesetzt werden, sind alle Bits 0, bedeutet dies absolut Positionieren.
DBX3002.1	kürzester Weg	0 1	
DBX3002.5	Drehrichtung wie M4	0	
DBX3003.0	absolut, negative Richtung	0 1	
DBX3003.1	absolut, positive Richtung	0 1	
DBD3002.2	Verfahrmaß Inch	1	Verfahrmaß inch
DBD3002.3	Handradüberlagerung	0	Überlagerung AUS
DBD3004	Sollposition/Sollweg	REAL	bei "inkrementell": Sollweg
DBD3008	Vorschubgeschwindigkeit	REAL	wenn = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
DBX3000.7	Start Positionierachse	0 1	Rücksetzen führt nicht zum Stopp!
DBX2.2	Restweg löschen, Spindelreset	0 1	Abbruchsignal, beendet die Funktion

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	Auch dann 1, wenn Override = 0 oder Sollposition erreicht.
DBX3000.6	Position erreicht	1: Achse hat Sollposition erreicht.
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	



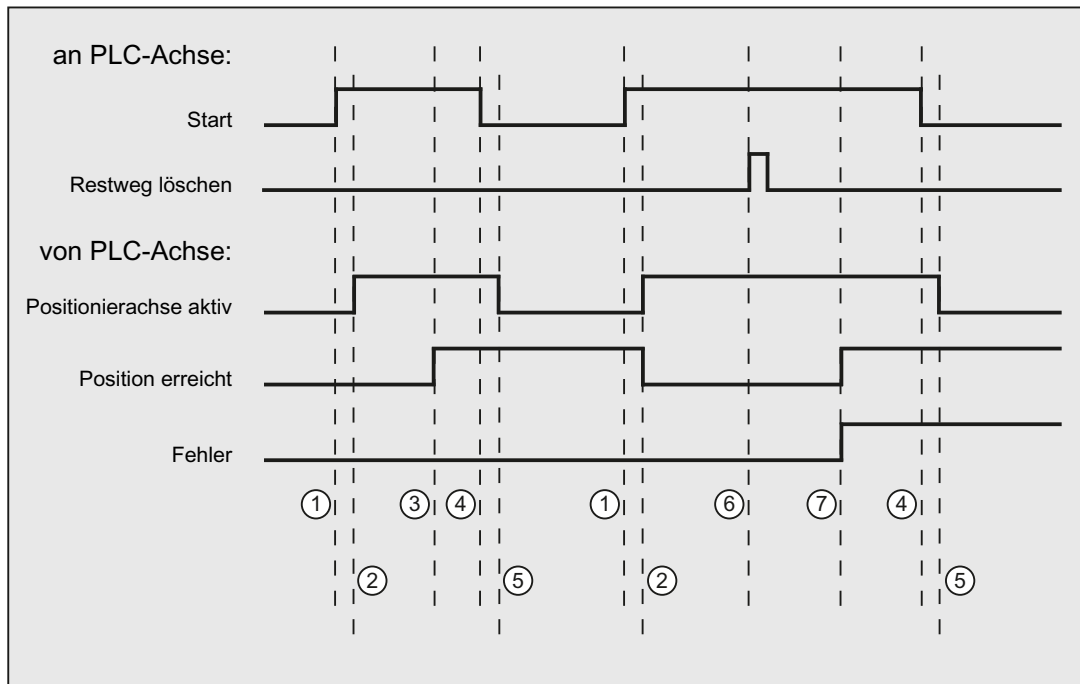
- ① Erster Funktionsanstoß durch positive Flanke von *Start*.
- ② *Positionierachse aktiv* = 1 zeigt an, dass die Funktion aktiv ist und die Ausgangssignale gültig sind, *Position erreicht* und *Achse steht* werden gegebenenfalls zurückgenommen.
- ③ Positive Quittung *Position erreicht* = 1 und *Positionierachse aktiv* = 1
- ④ Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- ⑤ Signalwechsel durch Funktion
- ⑥ Positionierung wird durch Restweg löschen abgebrochen, Signaldauer mind. 1 PLC-Zyklus.
- ⑦ Die Signale *Position erreicht* und *Fehler* werden gesetzt, die *Fehlernummer* (in diesem Fall 30) kann gelesen werden.



## 13.6.4.10 Positionierachse metrisch mit Handradüberlagerung

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0 1	Es darf nur eines der Bits gesetzt werden, sind alle Bits 0, bedeutet dies absolut Positionieren.
DBX3002.1	kürzester Weg	0 1	
DBX3002.5	Drehrichtung wie M4	0	
DBX3003.0	absolut, negative Richtung	0 1	
DBX3003.1	absolut, positive Richtung	0 1	
DBD3002.2	Verfahrmaß Inch	0	Verfahrmaß metrisch
DBD3002.3	Handradüberlagerung	1	Überlagerung EIN
DBD3004	Sollposition/Sollweg	REAL	bei "inkrementell": Sollweg
DBD3008	Vorschubgeschwindigkeit	REAL	wenn = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
DBX3000.7	Start Positionierachse	0 1	Rücksetzen führt nicht zum Stopp!
DBX2.2	Restweg löschen, Spindelreset	0 1	Abbruchsignal, beendet die Funktion

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	Auch dann 1, wenn Override = 0 oder Sollposition erreicht.
DBX3000.6	Position erreicht	1: Achse hat Sollposition erreicht.
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	

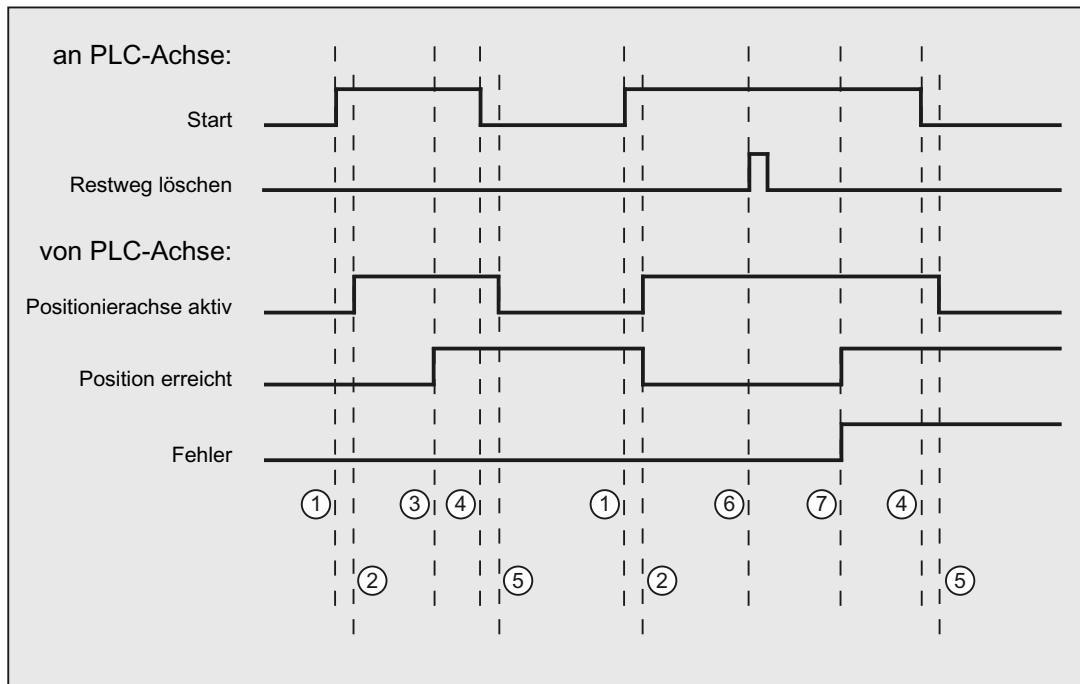


- ① Erster Funktionsanstoß durch positive Flanke von *Start*.
- ② *Positionierachse aktiv* = 1 zeigt an, dass die Funktion aktiv ist und die Ausgangssignale gültig sind, *Position erreicht* und *Achse steht* werden gegebenenfalls zurückgenommen.
- ③ Positive Quittung *Position erreicht* = 1 und *Positionierachse aktiv* = 1
- ④ Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- ⑤ Signalwechsel durch Funktion
- ⑥ Positionierung wird durch Restweg löschen abgebrochen, Signaldauer mind. 1 PLC-Zyklus.
- ⑦ Die Signale *Position erreicht* und *Fehler* werden gesetzt, die *Fehlernummer* (in diesem Fall 30) kann gelesen werden.

## 13.6.4.11 Positionierachse Inch mit Handradüberlagerung

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0 1	Es darf nur eines der Bits gesetzt werden, sind alle Bits 0, bedeutet dies absolut Positionieren.
DBX3002.1	kürzester Weg	0 1	
DBX3002.5	Drehrichtung wie M4	0	
DBX3003.0	absolut, negative Richtung	0 1	
DBX3003.1	absolut, positive Richtung	0 1	
DBD3002.2	Verfahrmaß Inch	1	Verfahrmaß inch
DBD3002.3	Handradüberlagerung	1	Überlagerung EIN
DBD3004	Sollposition/Sollweg	REAL	bei "inkrementell": Sollweg
DBD3008	Vorschubgeschwindigkeit	REAL	wenn = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
DBX3000.7	Start Positionierachse	0 1	Rücksetzen führt nicht zum Stopp!
DBX2.2	Restweg löschen, Spindelreset	0 1	Abbruchsignal, beendet die Funktion

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	Auch dann 1, wenn Override = 0 oder Sollposition erreicht.
DBX3000.6	Position erreicht	1: Achse hat Sollposition erreicht.
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	

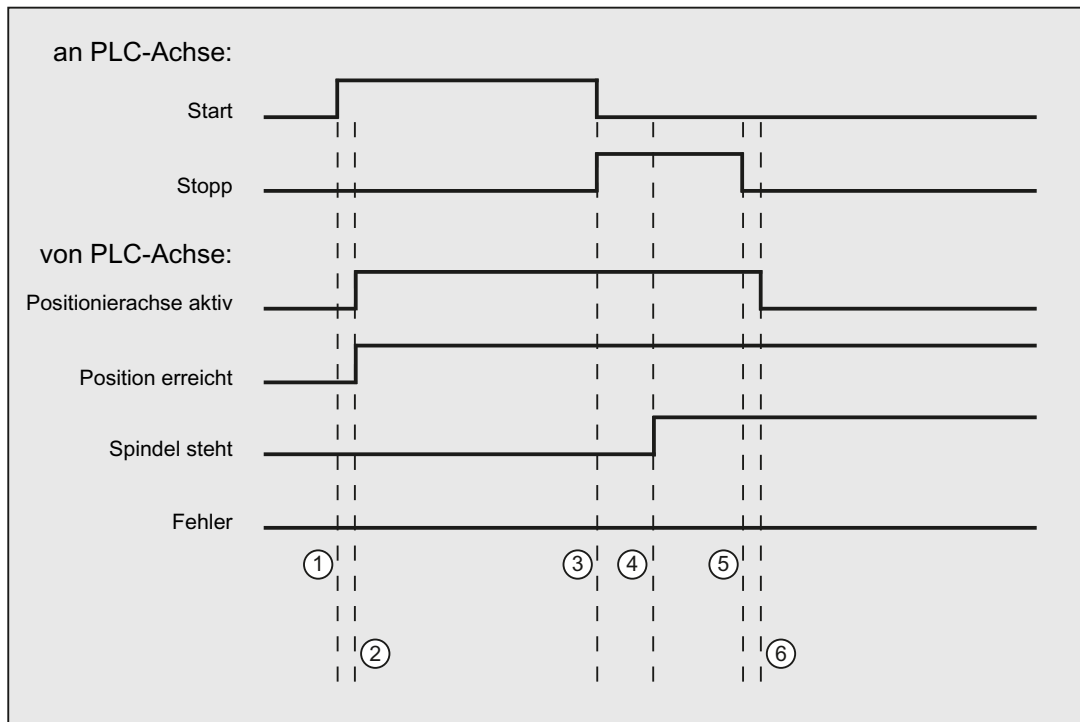


- ① Erster Funktionsanstoß durch positive Flanke von *Start*.
- ② *Positionierachse aktiv* = 1 zeigt an, dass die Funktion aktiv ist und die Ausgangssignale gültig sind, *Position erreicht* und *Achse steht* werden gegebenenfalls zurückgenommen.
- ③ Positive Quittung *Position erreicht* = 1 und *Positionierachse aktiv* = 1
- ④ Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- ⑤ Signalwechsel durch Funktion
- ⑥ Positionierung wird durch Restweg löschen abgebrochen, Signaldauer mind. 1 PLC-Zyklus.
- ⑦ Die Signale *Position erreicht* und *Fehler* werden gesetzt, die *Fehlernummer* (in diesem Fall 30) kann gelesen werden.

## 13.6.4.12 Spindel drehen mit automatischer Getriebestufenwahl

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0	<i>Drehrichtung wie M4:</i> 1: Drehrichtung durch M4 vorgegeben 0: Drehrichtung durch M3 vorgegeben
DBX3002.1	kürzester Weg	0	
DBX3002.5	Drehrichtung wie M4	0 1	
DBX3003.0	absolut, negative Richtung	0	
DBX3003.1	absolut, positive Richtung	0	
DBD3002.7	automatische Getriebestufenwahl	1	automatische Getriebestufenwahl EIN
DBD3008	Vorschubgeschwindigkeit	REAL	Spindeldrehzahl
DBD3000.5	Start Spindel drehen	0 1	
DBX3001.5	Stopp Spindel drehen	0 1	

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	1: bei Start oder Stopp == 1,
DBX3000.6	Position erreicht	1: Solldrehzahl wird ausgegeben
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	
DBX1.4	Achse/Spindel steht	

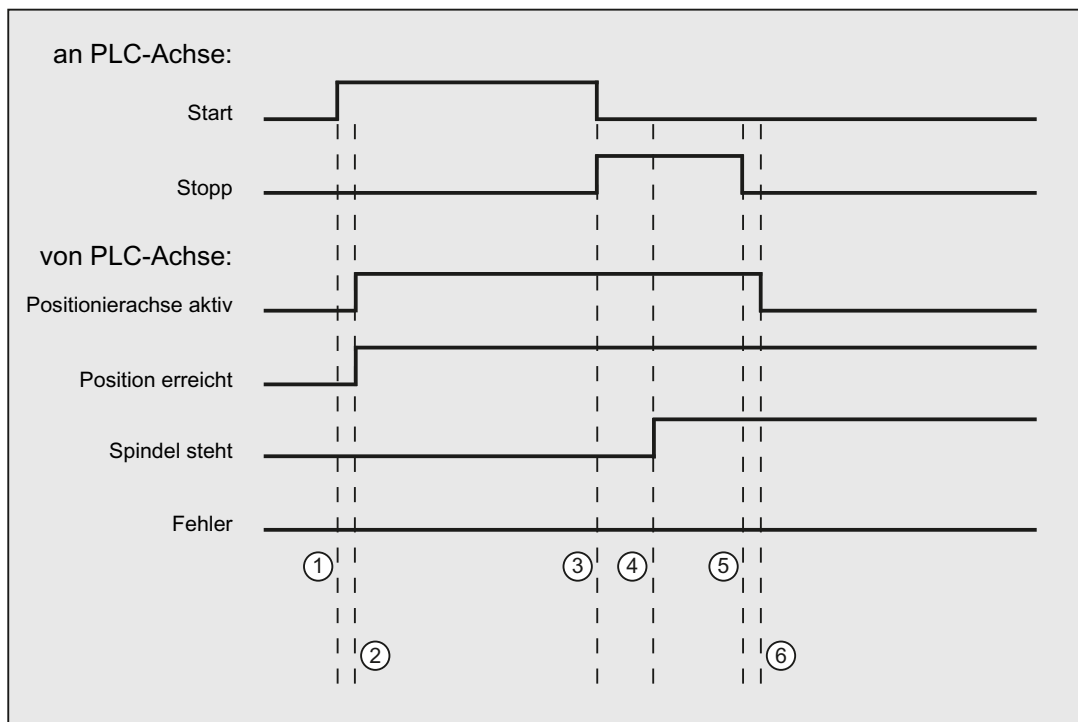


- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.
- ② Meldungen *Positionierachse aktiv* und *Position erreicht* werden zurückgemeldet, *Position erreicht* ist dabei irrelevant.
- ③ Das Drehen der Spindel wird vom Anwender dadurch gestoppt, dass *Start* zurückgesetzt und *Stopp* gesetzt wird.
- ④ Die Spindel hält an und das Signal *Spindel steht* wird gesetzt.
- ⑤ Daraufhin setzt der Anwender *Stopp* zurück.
- ⑥ Das Rücksetzen von *Stopp* bewirkt das Rücksetzen von *Positionierachse aktiv*.

## 13.6.4.13 Spindel drehen mit konstanter Schnittgeschwindigkeit [m/min]

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0	<i>Drehrichtung wie M4:</i> 1: Drehrichtung durch M4 vorgegeben 0: Drehrichtung durch M3 vorgegeben
DBX3002.1	kürzester Weg	0	
DBX3002.5	Drehrichtung wie M4	0 1	
DBX3003.0	absolut, negative Richtung	0	
DBX3003.1	absolut, positive Richtung	0	
DBD3002.2	Verfahrmaß Inch	0	Verfahrmaß metrisch
DBD3002.6	konst. Schnittgeschwindigkeit	1	konstante Schnittgeschwindigkeit EIN
DBD3008	Vorschubgeschwindigkeit	REAL	Spindeldrehzahl
DBD3000.5	Start Spindel drehen	0 1	
DBX3001.5	Stopp Spindel drehen	0 1	

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	1: bei Start oder Stopp == 1,
DBX3000.6	Position erreicht	1: Sollzahl wird ausgegeben
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	
DBX1.4	Achse/Spindel steht	



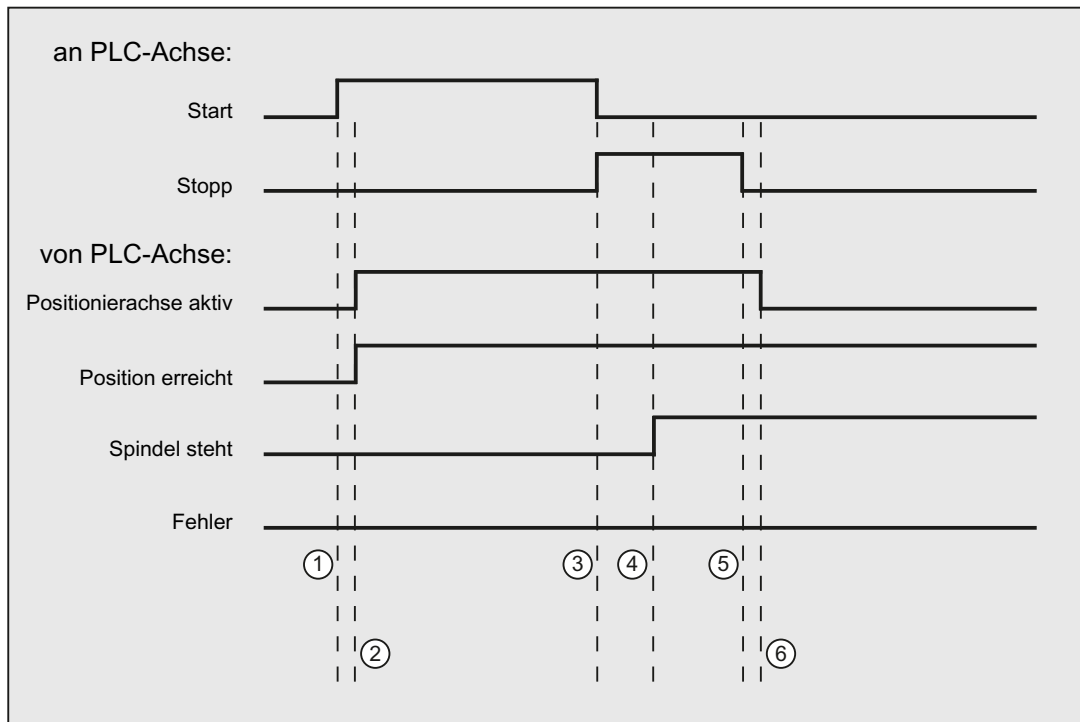
- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.
- ② Meldungen *Positionierachse aktiv* und *Position erreicht* werden zurückgemeldet, *Position erreicht* ist dabei irrelevant.
- ③ Das Drehen der Spindel wird vom Anwender dadurch gestoppt, dass *Start* zurückgesetzt und *Stopp* gesetzt wird.
- ④ Die Spindel hält an und das Signal *Spindel steht* wird gesetzt.
- ⑤ Daraufhin setzt der Anwender *Stopp* zurück.
- ⑥ Das Rücksetzen von *Stopp* bewirkt das Rücksetzen von *Positionierachse aktiv*.



## 13.6.4.14 Spindel drehen mit konstanter Schnittgeschwindigkeit [feet/min]

DB380x	Steuersignale PLC → NCK	Gültige Werte	Bemerkung
DBX3002.0	inkrementell	0	<i>Drehrichtung wie M4:</i> 1: Drehrichtung durch M4 vorgegeben 0: Drehrichtung durch M3 vorgegeben
DBX3002.1	kürzester Weg	0	
DBX3002.5	Drehrichtung wie M4	0 1	
DBX3003.0	absolut, negative Richtung	0	
DBX3003.1	absolut, positive Richtung	0	
DBD3002.2	Verfahrmaß Inch	1	Verfahrmaß inch
DBD3002.6	konst. Schnittgeschwindigkeit	1	konstante Schnittgeschwindigkeit EIN
DBD3008	Vorschubgeschwindigkeit	REAL	Spindeldrehzahl
DBD3000.5	Start Spindel drehen	0 1	
DBX3001.5	Stopp Spindel drehen	0 1	

DB390x	Statussignale NCK → PLC	Bemerkung
DBX3000.7	Positionierachse aktiv	1: bei Start oder Stopp == 1
DBX3000.6	Position erreicht	1: Sollzahl wird ausgegeben
DBX3000.1	Fehler	1: Fehler beim Verfahren, Fehlernummer in DBB3003 auswerten!
DBB3003	Fehlernummer	
DBX1.4	Achse/Spindel steht	



- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.
- ② Meldungen *Positionierachse aktiv* und *Position erreicht* werden zurückgemeldet, *Position erreicht* ist dabei irrelevant.
- ③ Das Drehen der Spindel wird vom Anwender dadurch gestoppt, dass *Start* zurückgesetzt und *Stopp* gesetzt wird.
- ④ Die Spindel hält an und das Signal *Spindel steht* wird gesetzt.
- ⑤ Daraufhin setzt der Anwender *Stopp* zurück.
- ⑥ Das Rücksetzen von *Stopp* bewirkt das Rücksetzen von *Positionierachse aktiv*.

### 13.6.4.15 Fehlermeldungen

Konnte eine Funktion nicht ausgeführt werden, wird dies am Signal *Error* (DB390x.DBX3000.1 oder DB390x.DBX3000.0) mit 'logisch 1' angezeigt. Die Fehlerursache ist als Fehlernummer kodiert:

Status	Bedeutung	
<b>Fehler, die durch PLC-Handling auftreten:</b>		
1	16#01	Es wurden mehrere Funktionen der Achse/Spindel gleichzeitig aktiviert
20	16#14	Es wurde eine Funktion gestartet, ohne dass die Position erreicht wurde
30	16#1e	Die Achse/Spindel wurde vor Ende der Bewegung an die NC abgegeben
40	16#28	Die Achse ist durch NC Programm programmiert, NCK interner Fehler
50	16#32	Fest zugeordnete PLC Achse: verfährt (JOG) oder referiert
60	16#3C	Fest zugeordnete PLC Achse: Kanalzustand lässt keinen Start zu
<b>Fehler, die durch Behandlung des NCK auftreten:</b>		
<b>Die Alarmnummern sind im Diagnosehandbuch Alarme SINUMERIK 828D, SINAMICS S120 beschrieben:</b>		
100	16#64	Falsche Position bei Achse/Spindel programmiert (entspricht Alarmnummer 16830)
101	16#65	Programmierte Drehzahl ist zu hoch
102	16#66	Falscher Wertebereich konstante Schnittgeschwindigkeit (entspricht Alarmnummer 14840)
104	16#68	Folgespindel: unerlaubte Programmierung (entspricht Alarmnummer 22030)
105	16#69	Kein Messsystem vorhanden (entspricht Alarmnummer 16770)
106	16#6a	Positioniervorgang der Achse noch aktiv (entspricht Alarmnummer 22052)
107	16#6b	Referenzmarke nicht gefunden (entspricht Alarmnummer 22051)
108	16#6c	Kein Übergang von der Drehzahlregelung in die Lageregelung (entspricht Alarmnummer 22050)
109	16#6d	Referenzmarke nicht gefunden (entspricht Alarmnummer 22051)
110	16#6e	Geschwindigkeit/Drehzahl ist negativ
111	16#6f	Soll-Drehzahl ist Null
112	16#70	Ungültige Getriebestufe
115	16#73	Programmierte Position wurde nicht erreicht
117	16#75	in der NC ist G96/G961 nicht aktiv
118	16#76	in der NC ist G96/G961 noch aktiv
120	16#78	Achse ist keine Teilungsachse (entspricht Alarmnummer 20072)
121	16#79	Teilungsposition- Fehler (entspricht Alarmnummer 17510)
125	16#7d	DC (kürzester Weg) nicht möglich (entspricht Alarmnummer 16800)
126	16#7e	Absolutwert Minus nicht möglich (entspricht Alarmnummer 16820)
127	16#7f	Absolutwert Plus nicht möglich (entspricht Alarmnummer 16810)
128	16#80	Keine Planachse für Durchmesserprogrammierung vorhanden (entspricht Alarmnummer 16510)

Status		Bedeutung
130	16#82	Softwareendschalter Plus (entspricht Alarmnummer 20070)
131	16#83	Softwareendschalter Minus (entspricht Alarmnummer 20070)
132	16#84	Arbeitsfeldbegrenzung Plus (entspricht Alarmnummer 20071)
133	16#85	Arbeitsfeldbegrenzung Minus (entspricht Alarmnummer 20071)
134	16#85	Frame ist unzulässig für Teilungsachse
135	16#87	Teilungsachse mit "Hirth-Verzahnung" ist aktiv (entspricht Alarmnummer 17501)
136	16#88	Teilungsachse mit "Hirth-Verzahnung" aktiv und Achse nicht referiert (entspricht Alarmnummer 17503)
137	16#89	Spindelbetrieb für transformierte Spindel/Achse nicht möglich (entspricht Alarmnummer 22290)
138	16#8A	für die Achse wird die entsprechende wirksame Koordinatensystem-spezifische Arbeitsfeldbegrenzung Plus verletzt (entspricht Alarmnummer 20082)
139	16#8B	für die Achse wird die entsprechende wirksame Koordinatensystem-spezifische Arbeitsfeldbegrenzung Minus verletzt (entspricht Alarmnummer 20082)
<b>System- oder andere schwere Alarme:</b>		
200	16#c8	entspricht System-Alarmnummer 450007

### 13.6.5 Starten von ASUPs

#### 13.6.5.1 Allgemeines

Ein ASUP (asynchrones Unterprogramm) ist ein NC-Programm, das von der PLC zu einem beliebigen Zeitpunkt gestartet werden kann, d.h. es ist ein NC Interruptprogramm, weil das laufende NC-Programm durch das ASUP unterbrochen wird.

Voraussetzung dafür ist die Anwahl und Parametrierung des ASUP durch ein NC-Programm oder durch den PI-Dienst ASUP (Seite 1029). Durch Ausführen des PI-Dienstes ASUP mit dem entsprechenden PI-Index wird jeweils einem der beiden Interrupts INT1 oder INT2 das dafür vorgesehene ASUP zugeordnet.

Es kann dabei zu einem Zeitpunkt nur ein ASUP gestartet werden. Sollten in einem PLC-Zyklus beide Startsignale der unten beschriebenen Funktionsschnittstelle auf logisch 1 gesetzt werden, werden die ASUP in Aufrufreihenfolge gestartet. Das Startsignal muss vom Anwender auf logisch 0 gesetzt werden, wenn in der Schnittstelle für Auftragsergebnis eine Quittung gesetzt wurde.

---

#### Hinweis

In folgenden Fällen, darf kein Startsignal gesetzt werden:

- PI-Dienst "ASUP" wurde noch nicht beendet.
- DB2600.DBX0.1 == 1 (Not-Halt)
- Anforderung von Kanal-Reset durch die PLC ist aktiv.

Der ASUP-Start ist erst wieder möglich, wenn sich der Kanal im Zustand "Reset" befindet UND DB3300.DBX3.7 == 1 (Kanal-Zustand Reset).

---

### 13.6.5.2 Auftragsstart

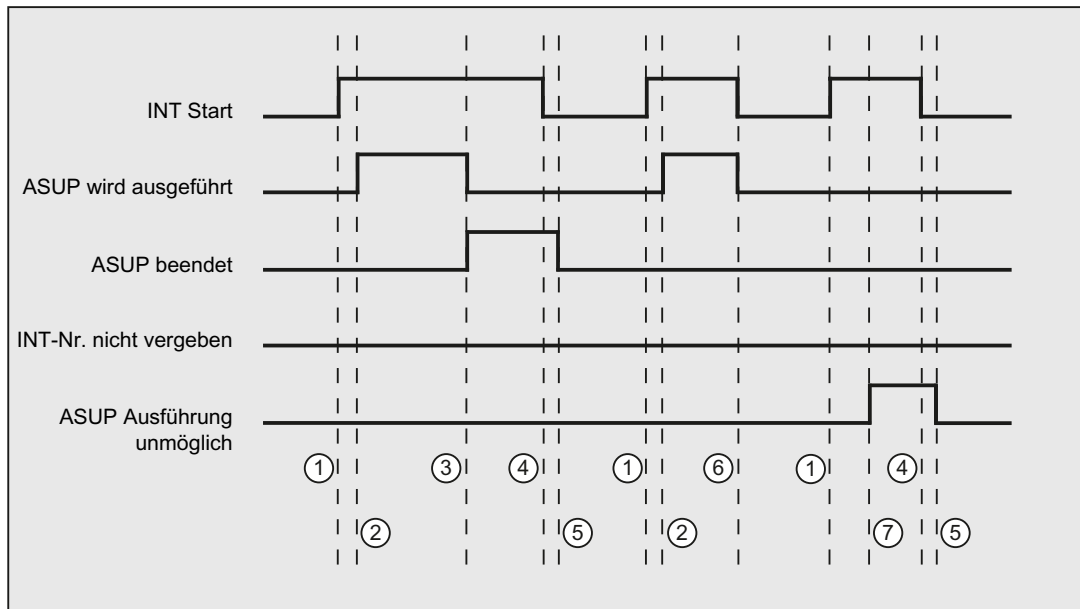
DB3400		ASUP: Auftrag [r/w]						
		Nahtstelle PLC -> NCK						
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 0000								INT1 Start
DBB 0001								INT2 Start
DBB 0002								
DBB 0003								
Start INT1, Start INT2 Bit 0 = 1: Anforderung zum Starten des ASUP, welches dem betreffenden INT zugeordnet ist. Bit 0 = 0: Beenden der ASUP-Anforderung nach Quittierung in der Ergebnisschnittstelle DBB1000, DB1001								

### 13.6.5.3 Auftragsergebnis

DB3400		ASUP: Ergebnis [r]						
		Nahtstelle NCK -> PLC						
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 1000	INT1							
					ASUP Ausführung nicht möglich <sup>1)</sup>	Interrupt-Nr nicht vergeben <sup>2)</sup>	ASUP wird ausgeführt	ASUP beendet <sup>3)</sup>
DBB 1001	INT2							
					ASUP Ausführung nicht möglich <sup>1)</sup>	Interrupt-Nr nicht vergeben <sup>2)</sup>	ASUP wird ausgeführt	ASUP beendet <sup>3)</sup>
DBB 1002								
DBB 2003								
1) Negative Quittung: z. B. bei Not-Halt oder Kanal-Reset-Anforderung 2) Negative Quittung: Die nummernmäßige Zuordnung ist noch nicht erfolgt. Abhilfe: PI-Dienst "ASUP" ausführen 3) Positive Quittung: ASUP erfolgreich beendet. Start-Signal (DB000, DB001) zurücksetzen.								

### 13.6.5.4 Signalfluss

#### Signalfluss



- ① Funktionsanstoß vom Anwender durch positive Flanke von *Start*.
- ② *ASUP wird ausgeführt* wird zurückgemeldet.
- ③ Die Quittung *ASUP beendet* zeigt die erfolgreiche Ausführung an, *ASUP wird ausgeführt* wird zurückgenommen.
- ④ Das Signal zum Funktionsanstoß wird nach Erhalt der Quittung vom Anwender zurückgesetzt.
- ⑤ Signalwechsel durch Firmware
- ⑥ Nicht zulässig! Wird Funktionsanstoß vor Erhalt der Quittung rückgesetzt, werden die Ausgangssignale nicht aktualisiert – ohne Einfluss auf Ablauf der angestoßenen Funktion.
- ⑦ *ASUP Ausführung nicht möglich*: negative Quittung, Fehler aufgetreten.

Bild 13-7 Beispiel: Signalfluss

## R1: Referenzieren

### 14.1 Kurzbeschreibung

#### Funktion

Die Funktion "Referenzpunktfahren" dient zur Synchronisation des Messsystems einer Maschinenachse auf den Maschinennullpunkt. Dazu wird die Maschinenachse auf den Maschinennullpunkt gefahren und das Messsystem dann auf Null gesetzt.

Kann der Maschinennullpunkt nicht direkt angefahren werden, bedient man sich eines im Verfahrbereich der Maschinenachse festgelegten Referenzpunktes, dessen Lage zum Maschinennullpunkt exakt bestimmt ist.

Nach dem Anfahren des Referenzpunktes wird das Messsystem der Maschinenachse dann nicht auf Null, sondern auf den entsprechenden Referenzpunktwert gesetzt.

#### Messsysteme und Referenzierarten

Mit der Funktion "Referenzpunktfahren" können Maschinenachsen mit folgenden Messsystemen und Referenzierarten referenziert werden:

- Messsysteme
  - Inkrementelles rotatorisches Messsystem mit mindestens einer Nullmarke
  - Inkrementelles lineares Messsystem (Längenmesssystem)
  - Rotatorisches Messsystem mit abstandscodierten Referenzmarken (Fa. Heidenhain)
  - Lineares Messsystem mit abstandscodierten Referenzmarken (Fa. Heidenhain)
  - Absolutes rotatorisches Messsystem
  - Absolutes lineares Messsystem (Längenmesssystem)
- Referenzierarten
  - Referenzieren bei inkrementellen Messsystemen mit BERO und Ein- und Zwei-Flankenerkennung
  - Referenzieren bei inkrementellen Messsystemen mit Ersatz des Referenznockens durch BERO
  - Referenzieren bei inkrementellen Messsystemen mit BERO mit projektierte Anfahrgeschwindigkeit bei Spindelapplikationen
  - Referenzieren bei Messsystemen mit abstandscodierten Referenzmarken durch Überfahren von 2 bzw. 4 Nullmarken
  - Referenzieren von passiven Messsystemen mittels Messsystemabgleich
  - Referenzieren im Nachführbetrieb
  - Referenzieren mit Nockenschalter am Antrieb

## Start

Das Referenzpunktfahren einer Maschinenachse kann manuell oder über das Teileprogramm gestartet werden:

- Manuell: Betriebsart JOG und MDA, Maschinenfunktion REF
- Teileprogramm: Teileprogrammbefehl <sup>G74</sup>



## 14.2 Achsspezifisches Referenzieren

Beim achsspezifischen Referenzpunktfahren muss das Referenzpunktfahren einzeln für jede zu referenzierende Maschinenachse angestoßen werden.

### Anwahl der Betriebsart und Maschinenfunktion

Vor dem Referenzpunktfahren der Maschinenachsen ist zuerst die entsprechende BAG in die Betriebsart JOG oder MDA zu versetzen:

DB11, ... DBX0.2 (Betriebsart JOG)

DB11, ... DBX0.1 (Betriebsart MDA)

Anschließend ist die Maschinenfunktion REF (Referenzpunktfahren) anzuwählen:

DB11, ... DBX1.2 (Maschinenfunktion REF)

### Start des Referenzpunktfahrens

Beim achsspezifischen Referenzpunktfahren muss der Start für jede Maschinenachse erfolgen.

Der Start des Referenzpunktfahrens erfolgt über die achsspezifischen Verfahrtasten:

DB31, ... DBX4.6 (Verfahrtaste minus)

DB31, ... DBX4.7 (Verfahrtaste plus)

### Richtungsfreigabe

Zur Vermeidung von Fehlbedienungen muss die Richtungsfreigabe parametrierbar werden:

MD34010 \$MA\_REFP\_CAM\_DIR\_IS\_MINUS (Referenzpunkt anfahren in Minusrichtung)

Über die Richtungsfreigabe wird festgelegt, durch welche Verfahrtaste das Referenzpunktfahren gestartet wird:

Wert	Bedeutung
0	Referenzpunktanfahren in Plusrichtung
1	Referenzpunktanfahren in Minusrichtung

### Tippbetrieb

Ob das Referenzpunktfahren nach dem einmaligen Betätigen der Richtungstaste vollständig ausgeführt wird oder der Maschinenbediener aus Sicherheitsgründen die Richtungstaste kontinuierlich gedrückt halten muss (Tippbetrieb), ist über folgendes Maschinendatum zu parametrieren:

MD11300 \$MN\_JOG\_INC\_MODE\_LEVELTRIGGRD (INC und REF im Tippbetrieb)

Lässt der Maschinenbediener die Richtungstaste im Tippbetrieb los, wird die Maschinenachse bis zum Stillstand abgebremst. Das Referenzpunktfahren wird nicht abgebrochen. Mit dem nächsten Drücken der Richtungstaste wird das Referenzpunktfahren fortgesetzt.

### Referenzierstatus

Mit dem Start des Referenzpunktfahrens wird der Referenzierstatus der Maschinenachse zurückgesetzt:

DB31, ... DBX60.4 (Referiert / Synchronisiert 1)

DB31, ... DBX60.5 (Referiert / Synchronisiert 2)

DB21, ... DBX36.2 (alle referenzpunktspflichtigen Achsen sind referenziert)

### Abstandscodierte Messsysteme

Bei abstandscodierten Messsystemen kann das Referenzpunktfahren mit einer beliebigen Verfahrtaste gestartet werden.

### Reihenfolge

Die Einhaltung einer bestimmten Reihenfolge, in der Maschinenachsen zu referenzieren sind, ist vom Maschinenbediener oder vom Maschinenhersteller per PLC-Anwenderprogramm sicherzustellen:

- Maschinenbediener

Die Maschinenachsen sind vom Maschinenbediener in der erforderlichen Reihenfolge zu starten.

- Maschinenhersteller

Das PLC-Anwenderprogramm des Maschinenherstellers ermöglicht das Starten nur in der entsprechenden Reihenfolge.

### Gleichzeitiges Referenzpunktfahren mehrerer Maschinenachsen

Es können steuerungsabhängig mehrere Maschinenachsen gleichzeitig referenziert werden:

SINUMERIK 840D:	max. 8 Maschinenachsen
-----------------	------------------------

### **Abschluss des Referenzpunktfahrens**

Ist das Referenzpunktfahren einer Maschinenachse erfolgreich abgeschlossen, erfolgt die entsprechende Rückmeldung durch Setzen des Referenzierstatus:

DB31, ... DBX60.4 (Referiert / Synchronisiert 1)

DB31, ... DBX60.5 (Referiert / Synchronisiert 2)

### **Abbruch des Referenzpunktfahrens**

Beim achsspezifischen Referenzpunktfahren wird die Maschinenachse in dem Kanal verfahren, der als Master-Kanal der Maschinenachse parametrierung wurde:

MD30550 \$MA\_AXCONF\_ASSIGN\_MASTER\_CHAN

Zum Abbruch des Referenzpunktfahrens muss daher entweder BAG-Reset oder Kanal-Reset für den Master-Kanal der Maschinenachse ausgelöst werden:

DB11, ... DBX0.7 (BAG-Reset)

DB21, ... DBX7.7 (Kanal-Reset)

Alle Maschinenachsen, die zum Zeitpunkt des Abbruchs das Referenzpunktfahren noch nicht erfolgreich abgeschlossen haben, bleiben im Status "Nicht referenziert":

DB31, ... DBX60.4 (Referiert / Synchronisiert 1)

DB31, ... DBX60.5 (Referiert / Synchronisiert 2)

## 14.3 Kanalspezifisches Referenzieren

Beim kanalspezifischen Referenzpunktfahren werden nach dem Anstoß des Referenzpunktfahrens alle Maschinenachsen des Kanals in der parametrisierten Reihenfolge referenziert.

### Anwahl der Betriebsart und Maschinenfunktion

Vor dem Referenzpunktfahren der Maschinenachsen ist zuerst die entsprechenden BAG in die Betriebsart JOG oder MDA zu versetzen:

DB11, ... DBX0.2 (Betriebsart JOG)

DB11, ... DBX0.1 (Betriebsart MDA)

Anschließend ist die Maschinenfunktion REF (Referenzpunktfahren) anzuwählen:

DB11, ... DBX1.2 (Maschinenfunktion REF)

### Parametrieren der Achsreihenfolge

Die Reihenfolge, in der die Maschinenachsen des Kanals referenziert werden, wird über folgendes Maschinendatum parametrisiert:

MD34110 \$MA\_REFP\_CYCLE\_NR = *Nummer*

Nummer	Bedeutung
-1	Für NC-START im Kanal muss die Maschinenachse nicht referenziert sein.
0	Die Maschinenachse nimmt nicht am kanalspezifischen Referenzpunktfahren teil.
1 - 15	Nummer der Reihenfolge beim kanalspezifischen Referenzpunktfahren.

Die Maschinenachsen werden in aufsteigender Reihenfolge der Nummern referenziert.

Maschinenachsen mit gleicher Nummer werden gleichzeitig referenziert.

### Gleichzeitiges Referenzpunktfahren mehrerer Maschinenachsen

Es können steuerungsabhängig mehrere Maschinenachsen gleichzeitig referenziert werden:

SINUMERIK 840D:	Max. 8 Maschinenachsen
SINUMERIK 810D:	Max. 5 Maschinenachsen

### Start des Referenzpunktfahrens

Der Start des kanalspezifischen Referenzpunktfahrens erfolgt über:

DB21, ... DBX1.0 (Referieren aktivieren)

Den Status des kanalspezifischen Referenzpunktfahrens meldet der Kanal über:

DB21, ... DBX33.0 (Referieren aktiv)

### Referenzierstatus

Mit dem Start des Referenzpunktfahrens wird der Referenzierstatus der Maschinenachse zurückgesetzt:

DB31, ... DBX60.4 (Referiert / Synchronisiert 1)

DB31, ... DBX60.5 (Referiert / Synchronisiert 2)

### Abschluss des Referenzpunktfahrens

Ist das kanalspezifische Referenzpunktfahren für alle beteiligten Maschinenachsen erfolgreich abgeschlossen, erfolgt die entsprechende Rückmeldung über:

DB21, ... DBX36.2 (alle referenzpunktpflichtigen Achsen sind referenziert)

### Abbruch des Referenzpunktfahrens

Beim kanalspezifischen Referenzpunktfahren wird die Maschinenachse in dem Kanal verfahren, dem sie als Kanalachse aktuell zugeordnet ist.

Zum Abbruch des Referenzpunktfahrens muss entweder BAG-Reset oder Kanal-Reset für den entsprechenden Kanal ausgelöst werden:

DB11, ... DBX0.7 (BAG-Reset)

DB21, ... DBX7.7 (Kanal-Reset)

Alle Maschinenachsen, für die zum Zeitpunkt des Abbruchs das Referenzpunktfahren noch nicht erfolgreich abgeschlossen wurde, bleiben im Status "Nicht referenziert":

DB31, ... DBX60.4 (Referiert / Synchronisiert 1)

DB31, ... DBX60.5 (Referiert / Synchronisiert 2)

## 14.4 Referenzpunktfahren aus dem Teileprogramm (G74)

Maschinenachsen können aus einem Teileprogramm heraus erstmalig oder erneut referenziert werden.

Ein erneutes Referenzieren ist z. B. erforderlich nach:

- Umsetzen des Istwertes der Maschinenachse: Funktion `PRESETON`
- Parken der Maschinenachse:  
DB31, ... DBX1.5 (Lagemesssystem 1) = 0  
DB31, ... DBX1.6 (Lagemesssystem 2) = 0
- DB31, ... DBX2.1 (Reglerfreigabe) = 0
- Überschreitung der Gebergrenzfrequenz des Lagemesssystems

### Programmierung

#### Syntax

`G74 Maschinenachse { Maschinenachse }`

#### Funktion

Maschinenachsen können mit der Teileprogrammanweisung `G74` aus einem Teileprogramm heraus referenziert werden.

Parameter: *Maschinenachse*

Es ist der Name der Maschinenachse anzugeben. Die Maschinenachse muss eine Kanalachse des Kanals sein, in dem das Teileprogramm abgearbeitet wird.

Wirksamkeit:

`G74` ist satzweise wirksam.

#### Besonderheiten

`G74` muss in einem eigenen Teileprogrammsatz programmiert werden.

### Reset-Verhalten

Durch BAG-Reset oder Kanal-Reset wird das Referenzpunktfahren für alle programmierten Maschinenachsen abgebrochen:

DB11, ... DBX0.7 (BAG-Reset)

DB21, ... DBX7.7 (Kanal-Reset)

Alle Maschinenachsen, für die zum Zeitpunkt des Abbruchs das Referenzpunktfahren noch nicht erfolgreich abgeschlossen wurde, bleiben im Status "Nicht referenziert":

DB31, ... DBX60.4 (Referiert / Synchronisiert 1)

DB31, ... DBX60.5 (Referiert / Synchronisiert 2)

## 14.5 Referenzieren bei inkrementellen Messsystemen

### 14.5.1 Hardware-Signale

Abhängig vom Maschinenaufbau und den Eigenschaften des verwendeten inkrementellen Messsystems, müssen verschiedene Hardware-Signale angeschlossen werden.

#### Referenznocken

- Anschluss  
Das Referenznockensignal kann an einem digitalen Eingang einer externen PLC-Peripheriebaugruppe oder einem schnellen Eingang der NCU-Schnittstelle X142 angeschlossen werden.
- NC/PLC-Nahtstellensignal  
Das Referenznockensignal muss vom PLC-Anwenderprogramm in die axiale NC/PLC-Nahtstelle übertragen werden:  
DB31, ... DBX12.7 (Verzögerung Referenzpunktfahren)

#### Nullmarkenauswahl

Werden während des Referenzpunktfahres der Achse oder Spindel mehrere Nullmarken des Messsystems erkannt (z.B. Messgetriebe zwischen Motor und Geber), muss mit einem zusätzlichen BERO-Signal die spezifische Nullmarke ausgewählt werden.

- Anschluss  
Der BERO muss an einem schnellen digitalen Eingang der NCU-Schnittstellen X122 oder X132 angeschlossen werden.
- Aktivierung  
Damit das BERO-Signal ausgewertet wird, muss für die Achse / Spindel im Antriebsparameter p0493 der digitale Eingang ausgewählt werden, an dem der BERO angeschlossen ist.

#### Nullmarkenersatz

Liefert das verwendete Messsystem kein Nullmarkensignal, kann über ein BERO-Signal ein Nullmarkenersatz erzeugt werden.

- Anschluss  
Der BERO muss an einem schnellen digitalen Eingang der NCU-Schnittstellen X122 oder X132 angeschlossen werden.
- Aktivierung  
Damit das BERO-Signal ausgewertet wird, muss für die Achse / Spindel im Antriebsparameter p0494 oder p0495 der digitale Eingang ausgewählt werden, an dem der BERO angeschlossen ist.

Übersicht

Signal	Anschluss: dig. Eingang über	Einstellen
Referenznocken	ext. PLC-Peripheriebaugruppe oder NCU: X142	PLC-Anwenderprogramm: DB31, ... DBX12.7
Nullmarkenauswahl	NCU: X122 oder X132	Antriebsparameter: p0493
Externe Nullmarke bzw. Nullmarkenersatz	NCU: X122 oder X132	Antriebsparameter: p0494 oder p0495

Literatur

- NCU-Schnittstellen: SINUMERIK 840D sl Gerätehandbuch NCU7x0.3 PN, Kapitel "Anschliessen" > "Digitale Ein-/Ausgänge"
- Antriebsparameter: SINAMICS S120/S150 Listenhandbuch

14.5.2 Nullmarkenauswahl mit BERO

Funktion

Das Referenzieren von inkrementellen Messsystemen basiert auf der eindeutigen Lage der Gebernullmarke bezogen auf den gesamten Verfahrbereich der Maschinenachse. Werden aufgrund von maschinenspezifischen Gegebenheiten mehrere Gebernullmarken im Verfahrbereich der Maschinenachse erkannt (Beispiele siehe folgendes Bild), muss zur eindeutigen Bestimmung des Referenzpunktes ein BERO an der Maschine angebracht werden. Die Lage des Referenzpunktes ergibt sich dann aus der Kombination von BERO-Signal und Gebernullmarke.

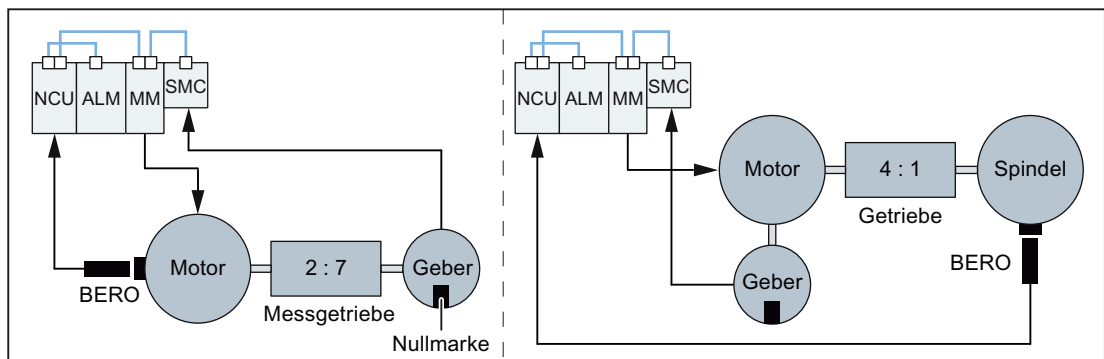


Bild 14-1 Messgetriebe zwischen Motor und Geber bzw. Untersetzungsgetriebe zwischen Motor und Spindel



## Parametrierung

### NC: Referenziermodus

Als Referenziermodus ist "Referenzieren von inkrementellen, rotatorischen oder linearen Messsystemen: Nullimpuls auf der Geberspur" zu parametrieren:

MD34200 \$MA\_ENC\_REFP\_MODE[<Achse>] = 1

### Antrieb: Nullmarkenauswahl

Im Parameter p0493 ist der digitale Eingang der NCU-Schnittstelle einzustellen, an dem der BERO angeschlossen ist.

---

### Hinweis

#### BERO-Signal: Nullmarkenauswahl

Die Verarbeitung des BERO-Signals erfolgt ausschließlich im Antrieb. Anschluss und Parametrierung siehe Kapitel "Hardware-Signale (Seite 1327)".

---

### 14.5.3 Zeitlicher Ablauf

Der zeitliche Ablauf des Referenzpunktfahrens bei inkrementellen Messsystemen unterteilt sich in drei Phasen:

- Phase 1: "Phase 1: Fahren auf den Referenznocken (Seite 1331)"
- Phase 2: "Phase 2: Synchronisation mit der Nullmarke (Seite 1334)"
- Phase 3: "Phase 3: Fahren auf den Referenzpunkt (Seite 1339)"

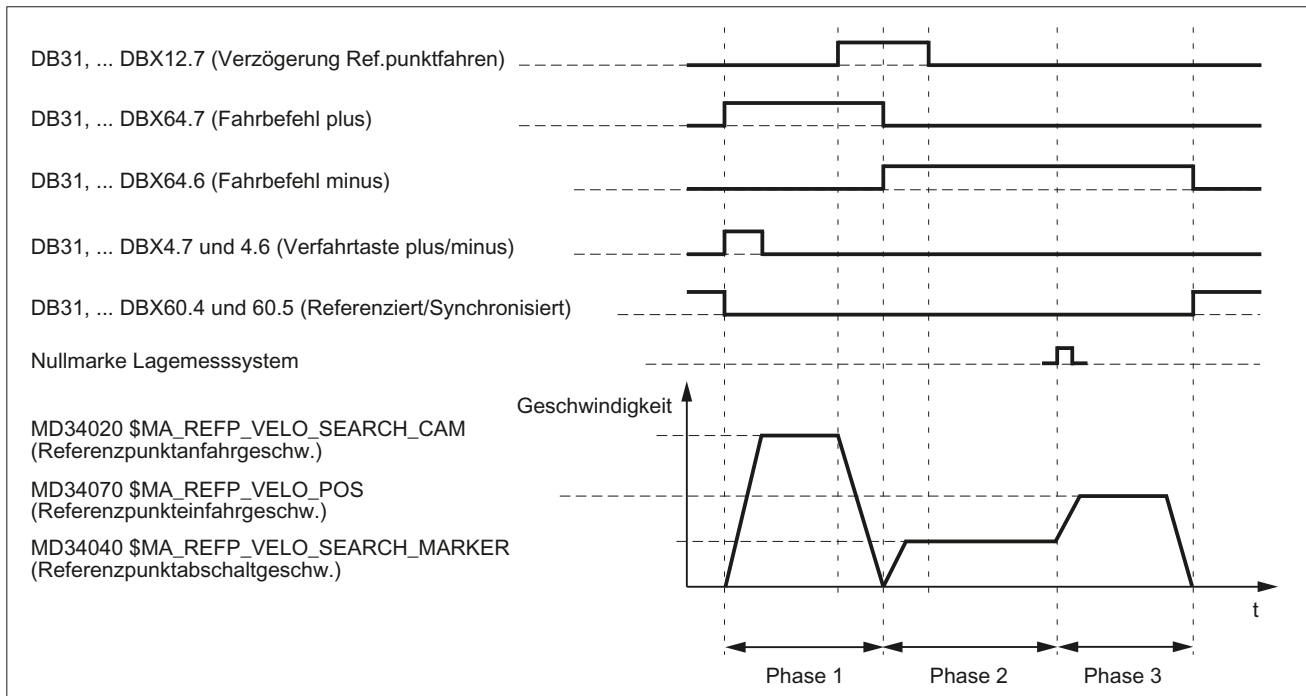


Bild 14-2 Zeitlicher Ablauf beim Referenzieren mit inkrementellen Messsystemen (Beispiel)

## 14.5.4 Phase 1: Fahren auf den Referenznocken

### Phase 1: Grafische Darstellung

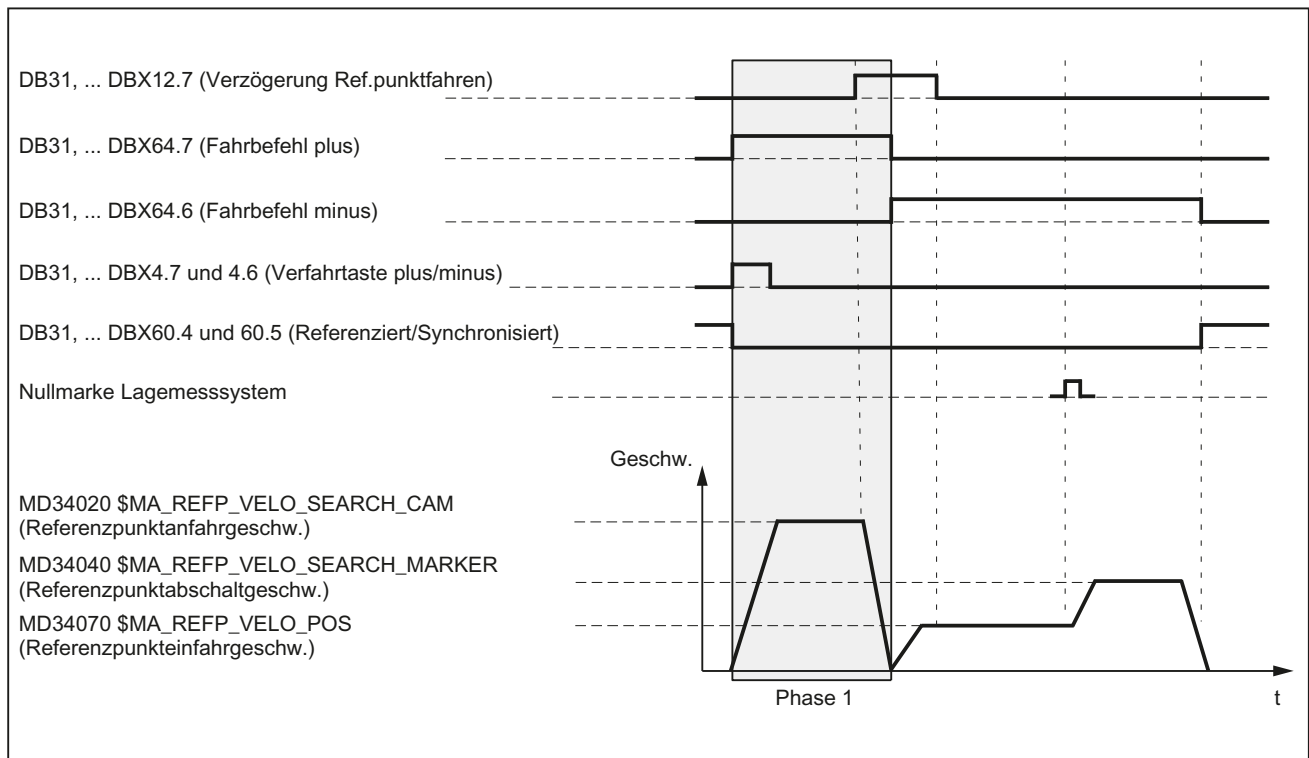


Bild 14-3 Phase 1: Fahren auf den Referenznocken

### Phase 1: Start

Zum Starten des Referenzpunktfahrens siehe Kapitel "Achsspezifisches Referenzieren (Seite 1321)" und "Achsspezifisches Referenzieren (Seite 1321)".

### Phase 1: Ablauf

Für Phase 1 sind, je nach Lage der Maschinenachse bezüglich des Referenznockens, drei Fälle zu unterscheiden:

1. Die Maschinenachse steht vor dem Referenznocken
2. Die Maschinenachse steht auf dem Referenznocken
3. Die Maschinenachse hat keinen Referenznocken

**Fall 1: Die Maschinenachse steht vor dem Referenznocken**

Nach dem Start des Referenzpunktfahrens wird die Maschinenachse in die parametrierte Richtung und auf die parametrierte Referenzpunktfahrgeschwindigkeit beschleunigt:

- MD34010 \$MA\_REFP\_CAM\_DIR\_IS\_MINUS (Referenzpunkt anfahren in Minusrichtung)
- MD34020 \$MA\_REFP\_VELO\_SEARCH\_CAM (Referenzpunktfahrgeschwindigkeit)

Das Erreichen des Referenznockens muss durch Abfrage eines digitalen EIngangs im PLC-Anwenderprogramm erkannt und der NC über folgendes Nahtstellensignal mitgeteilt werden:

DB31, ... DBX12.7 = 1 (Verzögerung Referenzpunktfahren)

Mit Erkennen des NC/PLC-Nahtstellensignals wird die Maschinenachse bis zum Stillstand abgebremst. Dabei wird mindestens noch die Wegstrecke  $s_{min}$  verfahren. Dadurch wird sichergestellt, dass die Maschinenachse den Referenznocken in Phase 2 mit der parametrierten Referenzpunktabschaltgeschwindigkeit verlässt.

$$s_{min} = \frac{(MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER)^2}{2 * MD32300 \$MA\_MAX\_AX\_ACCEL}$$

Phase 1 ist damit beendet. Das Referenzpunktfahren wird mit Phase 2 fortgesetzt.

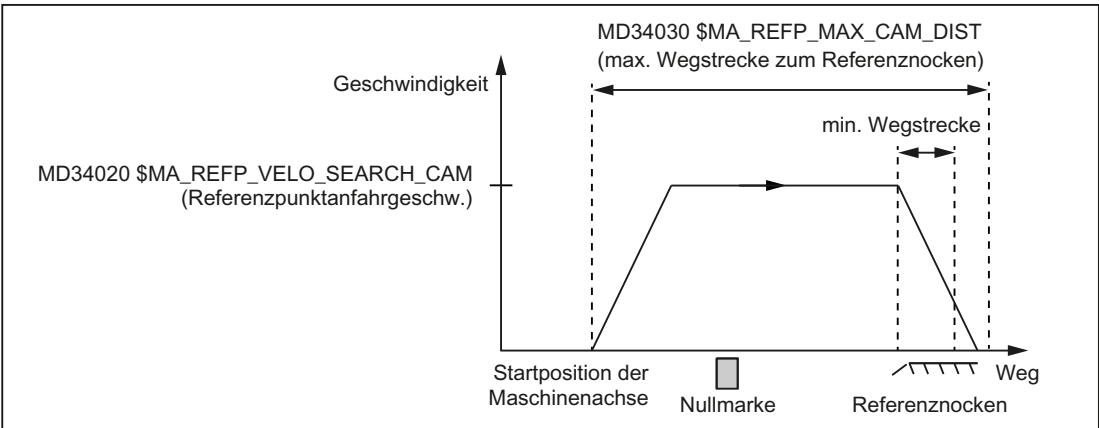


Bild 14-4 Minimale Wegstrecke für Verzögerung

**Fall 2: Die Maschinenachse steht auf dem Referenznocken**

Die Maschinenachse bleibt auf ihrer Ausgangsposition stehen.

Die Phase 1 ist damit beendet. Das Referenzpunktfahren wird mit Phase 2 fortgesetzt.

**Fall 3: Die Maschinenachse hat keinen Referenznocken**

Maschinenachsen ohne Referenznocken bleiben auf ihrer Ausgangsposition stehen.

Dazu gehören z. B.:

- Maschinenachsen, die über ihren gesamten Verfahrbereich nur eine Nullmarke haben
- Rundachsen, die nur eine Nullmarke pro Umdrehung haben

Für Maschinenachsen ohne Referenznocken ist im folgenden Maschinendatum Null einzutragen:

MD34000 \$MA\_REFP\_CAM\_IS\_ACTIVE = 0 (Achse mit Referenznocken)

Die Phase 1 ist damit beendet. Das Referenzpunktfahren wird mit Phase 2 fortgesetzt.

**Phase 1: Eigenschaften**

- Vorschubkorrektur ist wirksam.
- Vorschubhalt (kanal- und achsspezifisch) ist wirksam.
- NC-Stop und NC-Start sind wirksam.
- Die Maschinenachse wird angehalten, wenn der Referenznocken nicht innerhalb der parametrisierten maximalen Wegstrecke erreicht wird:

MD34030 \$MA\_REFP\_MAX\_CAM\_DIST (max. Wegstrecke zum Referenznocken)

**Siehe auch**

Kanalspezifisches Referenzieren (Seite 1324)

### 14.5.5 Phase 2: Synchronisation mit der Nullmarke

#### Phase 2: Grafische Darstellung

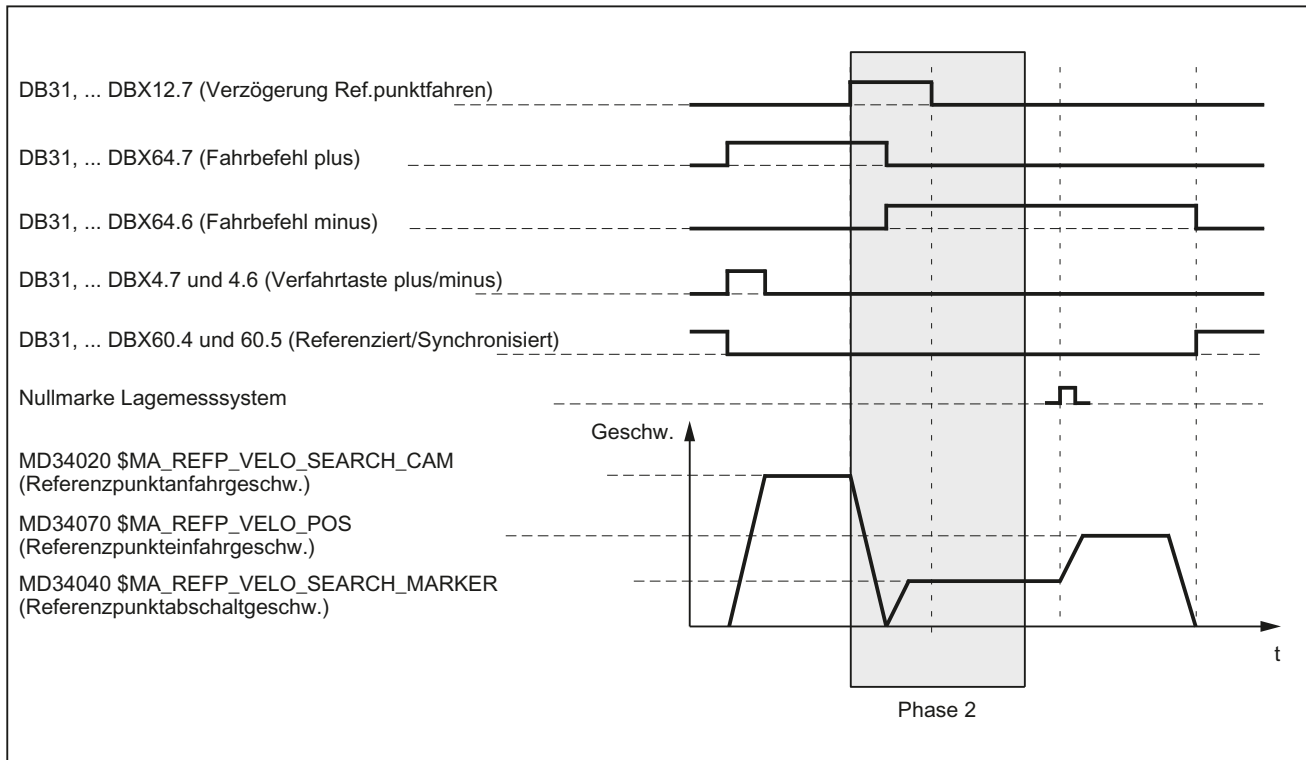


Bild 14-5 Phase 2: Synchronisation mit der Nullmarke

#### Phase 2: Start

Phase 2 wird automatisch gestartet, sobald Phase 1 ohne Alarm beendet wurde.

##### Ausgangssituation:

Die Maschinenachse steht auf dem Referenznocken.

##### Nullmarkensuchrichtung:

Die Richtung der Nullmarkensuche ergibt sich aus den Einstellungen in den Maschinendaten:

- MD34010 \$MA\_REFP\_CAM\_DIR\_IS\_MINUS (Referenzpunkt anfahren in Minusrichtung)
- MD34050 \$MA\_REFP\_SEARCH\_MARKER\_REVERSE (Richtungsumkehr auf Referenznocken)

## Phase 2: Ablauf

Die Synchronisation in Phase 2 kann über die fallende oder steigende Referenznockenflanke erfolgen. Die Parametrierung erfolgt über:

MD34050 \$MA\_REFP\_SEARCH\_MARKER\_REVERSE[<Achse>] = <Wert>

Wert	Bedeutung
0	Synchronisation mit fallender Referenznockenflanke
1	Synchronisation mit steigender Referenznockenflanke

### Hinweis

Hat die Istgeschwindigkeit der Maschinenachse beim Anfahren des Referenznockens noch nicht die Sollgeschwindigkeit der Phase 2 innerhalb der parametrierten Toleranzgrenzen erreicht, wird noch einmal mit Phase 1 gestartet. Das ist z. B. der Fall, wenn die Maschinenachse beim Starten des Referenzpunktfahrens auf dem Referenznocken steht.

MD35150 \$MA\_SPIND\_DES\_VELO\_TOL (Spindeldrehzahltoleranz)

### Fall 1: Synchronisation mit fallender Referenznockenflanke

Bei der Synchronisation mit fallender Referenznockenflanke beschleunigt die Maschinenachse auf die parametrierte Referenzpunktabschaltgeschwindigkeit entgegen der parametrierten Referenzpunktanfahrriichtung (Verfahrriichtung der Phase 1).

Nach dem Verlassen des Referenznockens wird auf die nächste Gebernulmarke gewartet: DB31, ... DBX12.7 == 0

Mit dem Erkennen der Gebernulmarke ist die Phase 2 beendet. Die Maschinenachse fährt mit konstanter Geschwindigkeit weiter und das Referenzpunktfahren wird mit Phase 3 fortgesetzt.

- MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER (Referenzpunktabschaltgeschwindigkeit)

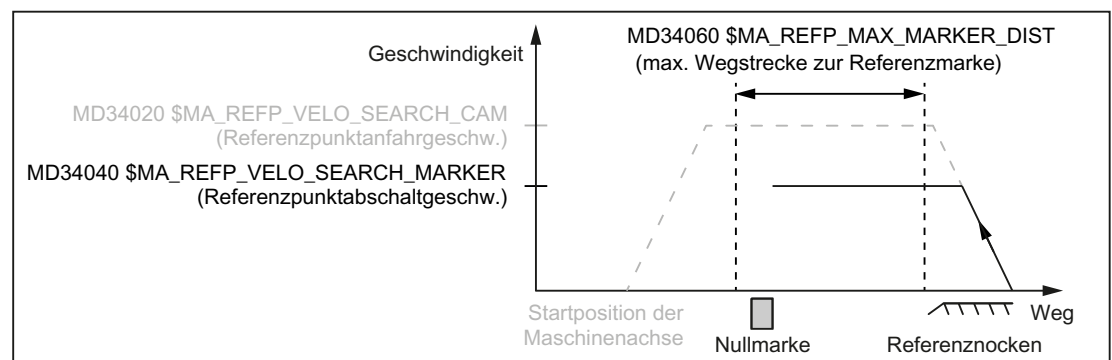


Bild 14-6 Synchronisation mit fallender Referenznockenflanke

**Fall 2: Synchronisation mit steigender Referenznockenflanke**

Bei der Synchronisation mit steigender Referenznockenflanke beschleunigt die Maschinenachse auf die parametrierte Referenzpunktanfahrgeschwindigkeit entgegen der parametrisierten Referenzpunktanfahrrichtung (Verfahrrichtung der Phase 1):

- MD34020 \$MA\_REFP\_VELO\_SEARCH\_CAM (Referenzpunktanfahrgeschwindigkeit)
- MD34010 \$MA\_REFP\_CAM\_DIR\_IS\_MINUS (Referenzpunkt anfahren in Minusrichtung).

Nach dem Verlassen des Referenznockens wird die Maschinenachse auf Stillstand abgebremst:

DB31, ... DBX12.7 == 0

Anschließend verfährt die Maschinenachse mit der parametrisierten Referenzpunktabschaltgeschwindigkeit zurück auf den Referenznocken:

MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER (Referenzpunktabschaltgeschwindigkeit)

Mit Erreichen des Referenznockens (DB31, ... DBX12.7 = 1) wird auf die nächste Gebernullmarke gewartet.

Mit dem Erkennen der Gebernullmarke ist die Phase 2 beendet. Die Maschinenachse fährt mit konstanter Geschwindigkeit weiter und das Referenzpunktfahren wird mit Phase 3 fortgesetzt.

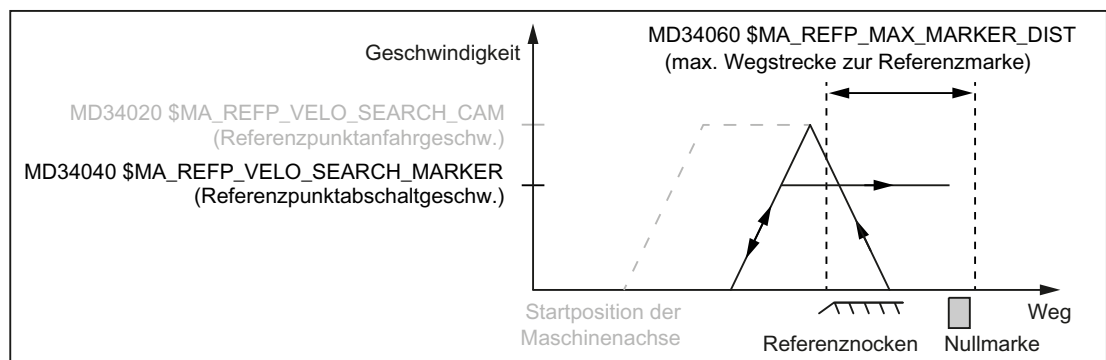


Bild 14-7 Synchronisation mit steigender Referenznockenflanke

**Elektronische Referenznockenverschiebung**

Die elektronische Referenznockenverschiebung dient der Kompensation temperaturbedingter Längenausdehnungen des Referenznockens, so dass sicher immer auf dieselbe Gebernullmarke synchronisiert wird:

MD34092 \$MA\_REFP\_CAM\_SHIFT (Elektronische Referenznockenverschiebung für inkrementelle Messsysteme mit äquidistanten Nullmarken)

Durch die elektronische Referenznockenverschiebung wird nach dem Erkennen der Referenznockenflanke nicht sofort auf die nächste Gebernullmarke synchronisiert, sondern erst nach dem zusätzlichen Zurücklegen des parametrisierten Verschiebungsweges.

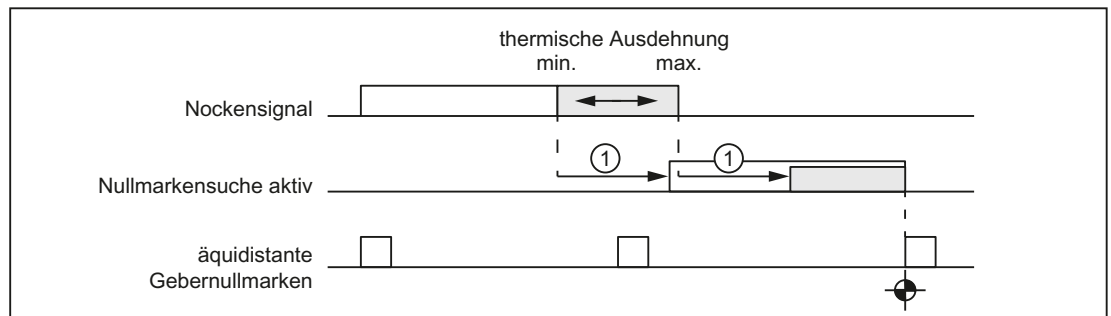


Aufgrund der Ermittlung des, seit dem Erkennen der Referenznockenflanke zurückgelegte Verfahrensweges im Interpolationstakt, ergibt sich der wirksame Verschiebungsweg  $s_{\text{shift}}$  zu:

$s_{\text{shift\_min}} = \text{MD34092 } \$\text{MA\_REFP\_CAM\_SHIFT}$

$s_{\text{shift\_max}} = \text{MD34092 } \$\text{MA\_REFP\_CAM\_SHIFT} +$   
 $\text{MD34040 } \$\text{MA\_REFP\_VELO\_SEARCH\_MARKER} * \text{Interpolationstakt}$

Die elektronische Referenznockenverschiebung wirkt immer in Suchrichtung der Nullmarke.



① Referenznockenverschiebung

Bild 14-8 Elektronische Referenznockenverschiebung

### Voraussetzung

Die elektronische Referenznockenverschiebung wirkt nur bei Maschinenachsen mit Referenznocken:

$\text{MD34000 } \$\text{MA\_REFP\_CAM\_IS\_ACTIVE} == 1$

## Referenznockenjustage

### Geber mit äquidistanten Nullmarken

Bei Gebern, die in äquidistanten Abständen Nullmarken liefern, ist auf exakte Justage des Referenznockens zu achten, damit beim Referenzpunktfahren immer die korrekte Nullmarke erkannt wird.

### Zeitverhalten

Folgende Faktoren beeinflussen das zeitliche Verhalten vom Erreichen des Referenznockens an der Maschine bis zum Erkennen des vom PLC-Anwenderprogramm an die NC übertragenen Referenznockensignals:

- Schaltgenauigkeit des Referenznockenschalters
- Zeitverzögerung des Referenznockenschalters (Öffner)
- Zeitverzögerung am PLC-Eingang
- PLC-Zykluszeit
- Zykluszeit zur Aktualisierung der NC/PLC-Nahtstelle
- Interpolationstakt
- Lageregeltakt

### Einstellhinweise

- Referenznocken

In der Praxis hat sich die Justierung der Flanke des Referenznockens in der Mitte zwischen zwei Nullmarken bewährt.

- Elektronische Referenznockenverschiebung

Als Anhaltspunkt zur Parametrierung der elektronischen Referenznockenverschiebung dient das nur lesbare Maschinendatum:

MD34093 \$MA\_REFP\_CAM\_MARKER\_DIST (Abstand Referenznocken/Referenzmarke)

Der angezeigte Wert entspricht der Distanz zwischen dem Verlassen des Referenznockens und dem Auftreten der Referenzmarke. Bei zu kleinen Werten besteht die Gefahr, dass die Ermittlung des Referenzpunktes aufgrund von Temperatureinflüssen oder einer schwankenden Laufzeit des Nockensignals nicht deterministisch ist.

 <b>WARNUNG</b>
--

Bei fehlerhafter oder ungenauer Justage des Referenznockens kann eine falsche Nullmarke ausgewertet werden. Die Steuerung ermittelt dadurch einen falschen Maschinennullpunkt. Demzufolge fährt die Maschinenachse auf falsche Positionen. Softwareendschalter, Schutzbereiche und Arbeitsfeldbegrenzungen wirken auf falsche Positionen und können so die Maschine nicht schützen. Die Wegdifferenz entspricht +/- dem von der Maschinenachse zurückgelegten Weg zwischen 2 Nullmarken.
--

## Phase 2: Eigenschaften

- Vorschubkorrektur ist **nicht** wirksam.  
Intern wird mit Vorschubkorrektur = 100% verfahren.  
Bei einer vorgegebenen Vorschubkorrektur von 0% erfolgt Abbruch.
- Vorschubhalt (kanal- und achsspezifisch) ist wirksam.
- NC-Stop und NC-Start sind **nicht** wirksam.
- Erreicht die Maschinenachse in Phase 2 nicht innerhalb der parametrisierten Wegstrecke die Referenzmarke (Gebernulldmarke), wird die Maschinenachse angehalten:  
MD34060 \$MA\_REFP\_MAX\_MARKER\_DIST (max. Wegstrecke zur Referenzmarke)

### 14.5.6 Phase 3: Fahren auf den Referenzpunkt

#### Phase 3: Grafische Darstellung

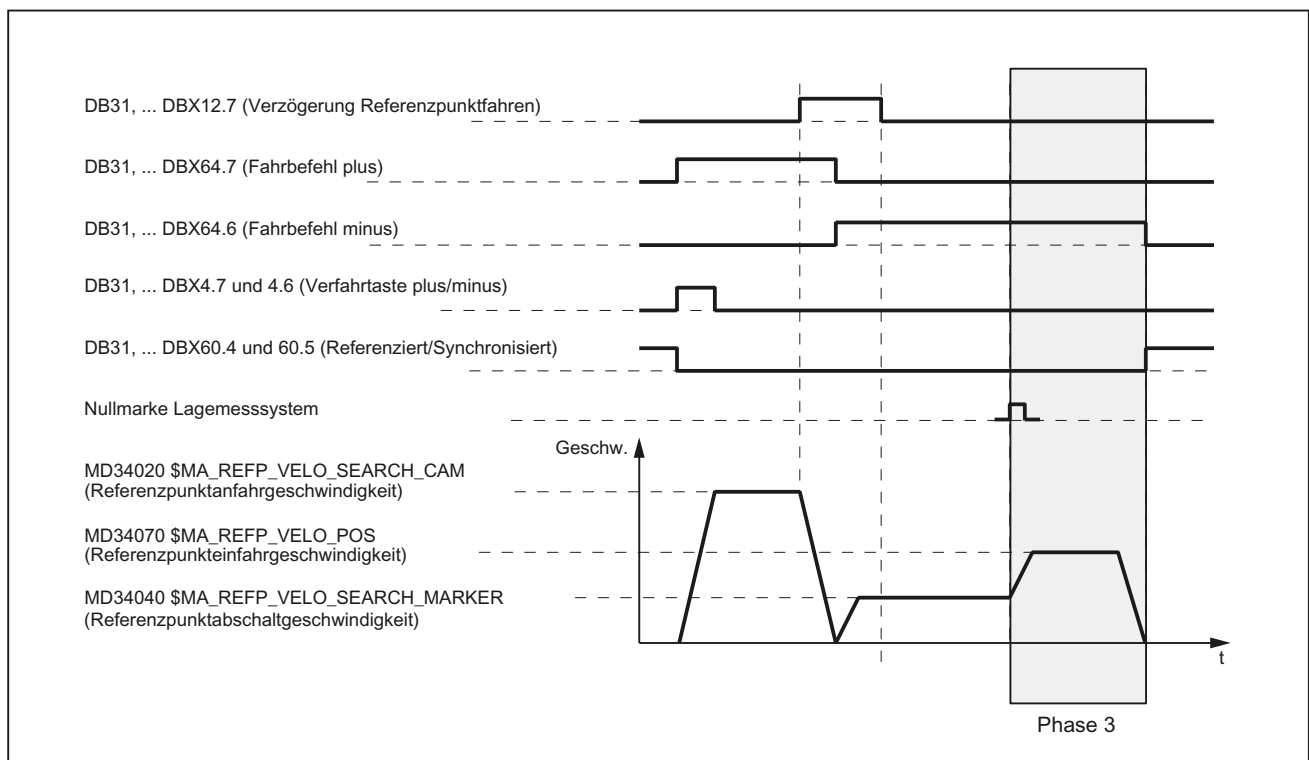


Bild 14-9 Phase 3: Fahren auf den Referenzpunkt

**Phase 3: Start**

Die Maschinenachse fährt am Ende der Phase 2 mit Referenzpunktabschaltgeschwindigkeit. Sobald Phase 2 erfolgreich ohne Alarm beendet wurde, wird daher ohne Anzuhalten in Phase 3 gewechselt.

**Ausgangssituation**

Die Gebernullmarke wurde erkannt.

**Phase 3: Ablauf**

Die Maschinenachse verfährt mit der parametrierten Referenzpunkteinfahrtgeschwindigkeit: MD34070 \$MA\_REFP\_VELO\_POS (Referenzpunkteinfahrtgeschwindigkeit) von der in Phase 2 erkannten Gebernullmarke bis zum Referenzpunkt.

Der dabei zu verfahrenende Weg  $s_{ref}$  ergibt sich aus der Summe von Referenzpunktabstand und Referenzpunktverschiebung:

MD34080 \$MA\_REFP\_MOVE\_DIST (Referenzpunktabstand)

MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR (Referenzpunktverschiebung)

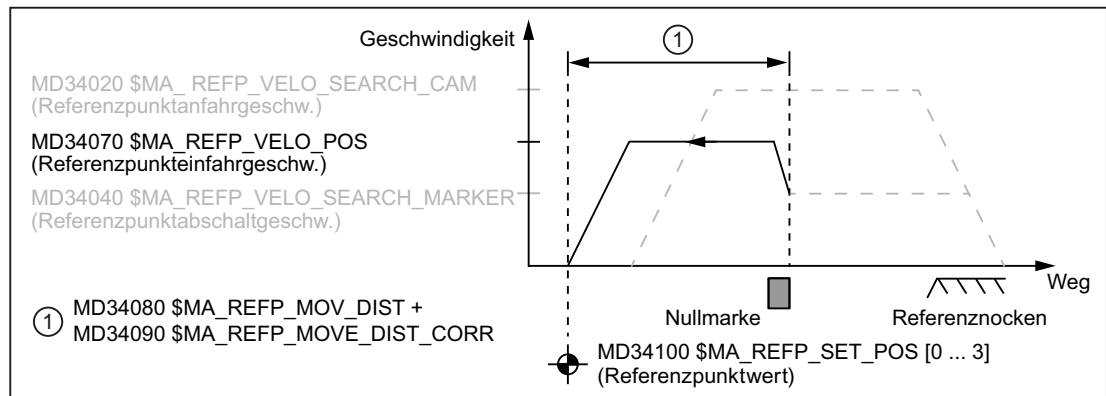


Bild 14-10 Referenzpunktposition

Mit Erreichen des Referenzpunktes, wird die Maschinenachse angehalten und das Istwertsystem auf den über die NC/PLC-Nahtstelle vorgegeben Referenzpunktwert n synchronisiert:

MD34100 \$MA\_REFP\_SET\_POS[<n>] (Referenzpunktwert)

Die Auswahl des Referenzpunktwertse erfolgt über die NC/PLC-Nahtstelle:

DB31, ... DBX2.4 ... 7 (Referenzpunktwert 1 ... 4)

Das Istwertsystem wird auf den Referenzpunktwert synchronisiert, der zum Zeitpunkt des Erreichens des Referenznocksens in Phase 1 (DB31, ... DBX12.7 == 1) angewählt war.

Die Maschinenachse ist jetzt referenziert. Als Rückmeldung an das PLC-Anwenderprogramm wird, abhängig vom aktiven Messsystem, das Nahtstellensignal gesetzt:

DB31, ... DBX60.4 / .5 (Referenziert/Synchronisiert 1 / 2) = 1

### Eigenschaften der Phase 3

- Vorschubkorrektur ist wirksam.
- Vorschubhalt (kanal- und achsspezifisch) ist wirksam.
- NC-Stop und NC-Start sind wirksam.

### Besonderheiten der Phase 3

In folgenden Fällen hält die Maschinenachse nach dem Erkennen der Nullmarke zuerst an, um dann auf den Referenzpunkt zurückzufahren:

- Die Summe aus Referenzpunktabstand und Referenzpunktverschiebung ist kleiner dem, aufgrund der Referenzpunkteinfahrtsgeschwindigkeit, benötigten Bremsweg:  
 $MD34080 + MD34090 < \text{"Benötigter Bremsweg aufgrund MD34070"}$
- Der Referenzpunkt liegt, entgegen der aktuellen Verfahrrichtung, "hinter" dem Referenznocken.

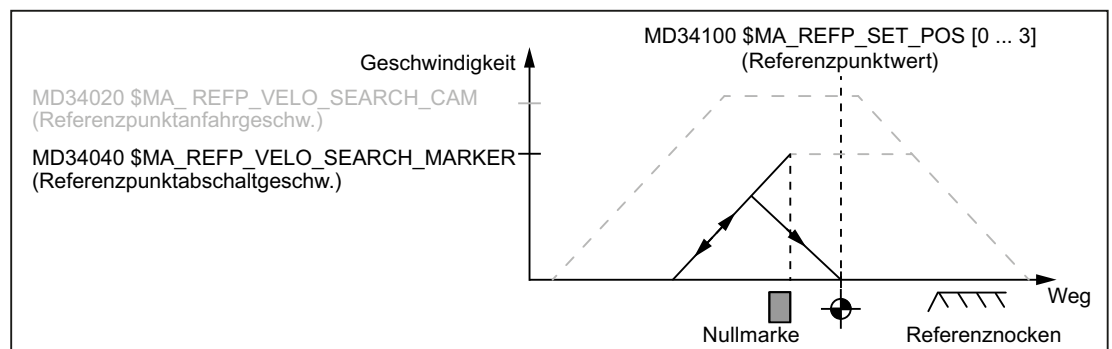


Bild 14-11 Referenzpunktabstand plus Referenzpunktverschiebung kleiner Bremsweg

## 14.6 Referenzieren bei abstandscodierten Referenzmarken

### 14.6.1 Allgemeiner Überblick

#### Abstandscodierte Referenzmarken

Messsysteme mit abstandscodierten Referenzmarken bestehen aus zwei parallel verlaufenden Maßstabsspuren:

- Inkrementales Strichgitter
- Referenzmarkenspur

Der Abstand zwischen jeweils zwei aufeinander folgenden Referenzmarken ist definiert. Somit lässt sich nach dem Überfahren zweier aufeinander folgender Referenzmarken die absolute Position der Maschinenachse bestimmen. Bei einem Referenzmarken-Abstand von z. B. ca. 10 mm wird zum Referenzieren der Maschinenachse nur ein Verfahrensweg von ca. 20 mm benötigt.

Das Referenzieren kann von jeder beliebigen Achsposition aus in positiver wie negativer Richtung erfolgen (Ausnahme: Verfahrbereichsenden).

## 14.6.2 Grundlegende Parametrierung

### Lineare Messsysteme

Zur Parametrierung von linearen Messsystemen sind folgende Daten einzustellen:

- Die absolute Verschiebung zwischen dem Maschinennullpunkt und der Position der ersten Referenzmarke des Längenmesssystems:

MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR (Referenzpunkt-/Absolutverschiebung)

Siehe dazu unten: Ermittlung der Absolutverschiebung

- Die Orientierung des Längenmesssystems (gleich- oder gegensinnig) gegenüber dem Maschinensystemkoordinatensystem:

MD34320 \$MA\_ENC\_INVERS (Längenmesssystem gegensinnig zum Maschinensystem)

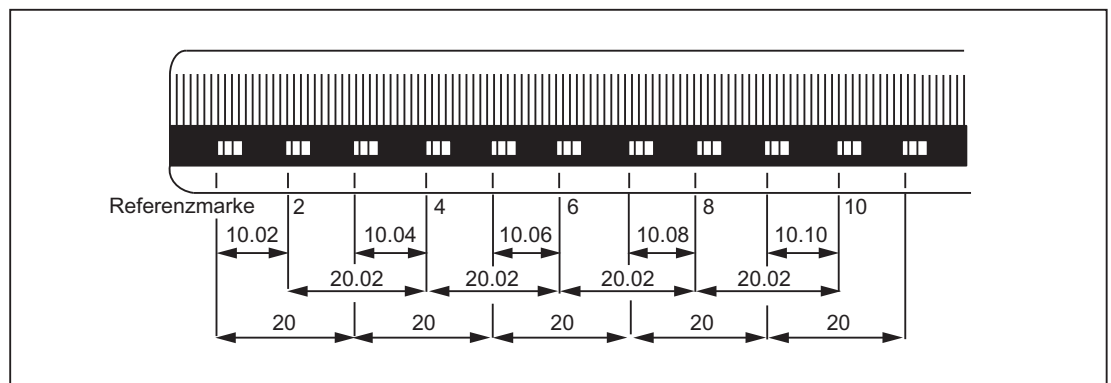


Bild 14-12 DIADUR-Glasmaßstab mit abstandscodierten Referenzmarken  
(Maße in mm für 20 mm Teilungsperiode)

### Rotatorisches Messsystem

Für rotatorische Messsysteme gilt sinngemäß das Gleiche wie für lineare Messsysteme (s. o.).

### Ermittlung der Absolutverschiebung

Zur Ermittlung der Absolutverschiebung zwischen dem Maschinennullpunkt und der Position der ersten Referenzmarke einer Maschinenachse wird folgendes Vorgehen empfohlen:

1. Für die Absolutverschiebung den Wert Null eingeben:

MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR = 0

2. Referenzpunktfahren durchführen.

**Hinweis:** Das Referenzpunktfahren sollte an einer Stelle in der Maschine durchgeführt werden, an der z. B. mittels eines Laserinterferometers die genaue Position der Maschinenachse bezüglich des Maschinennullpunkts einfach ermittelt werden kann.

3. Über die Bedienoberfläche die Istposition der Maschinenachse ermitteln.
4. Aktuelle Position der Maschinenachse bezüglich des Maschinennullpunkts messen.
5. Absolutverschiebung berechnen und im MD34090 eintragen.

Die Absolutverschiebung berechnet sich, abhängig von der Orientierung des Messsystems (gleich- oder gegensinnig) gegenüber dem Maschinensystemkoordinatensystem, zu:

Orientierung des Messsystems	Absolutverschiebung
gleichsinnig	Gemessene Position + Angezeigte Istposition
gegensinnig	Gemessene Position - Angezeigte Istposition

 **WARNUNG**

Nach der Ermittlung der Absolutverschiebung und dem Eintrag im MD34090 muss das Referenzpunktfahren für die Maschinenachse erneut durchgeführt werden.

### Referenziermodi

Für das Referenzieren mit abstandscodierten Referenzmarken stehen zwei Modi zur Verfügung:

- Auswertung von **zwei** aufeinander folgenden Referenzmarken:

MD34200 \$MA\_ENC\_REFP\_MODE (Referenzier-Modus) = 3

Vorteil:

- Kurzer Verfahrenweg

- Auswertung von **vier** aufeinander folgenden Referenzmarken:

MD34200 \$MA\_ENC\_REFP\_MODE = 8

Vorteil:

- Plausibilitätsprüfung durch die NC möglich
- Erhöhung der Sicherheit des Referenzierergebnisses



### 14.6.3 Zeitlicher Ablauf

#### Zeitlicher Ablauf

Der zeitliche Ablauf beim Referenzieren mit abstandscodierten Referenzmarken unterteilt sich in 2 Phasen:

- Phase 1: Überfahren der Referenzmarken mit Synchronisation
- Phase 2: Fahren auf einen festen Zielpunkt

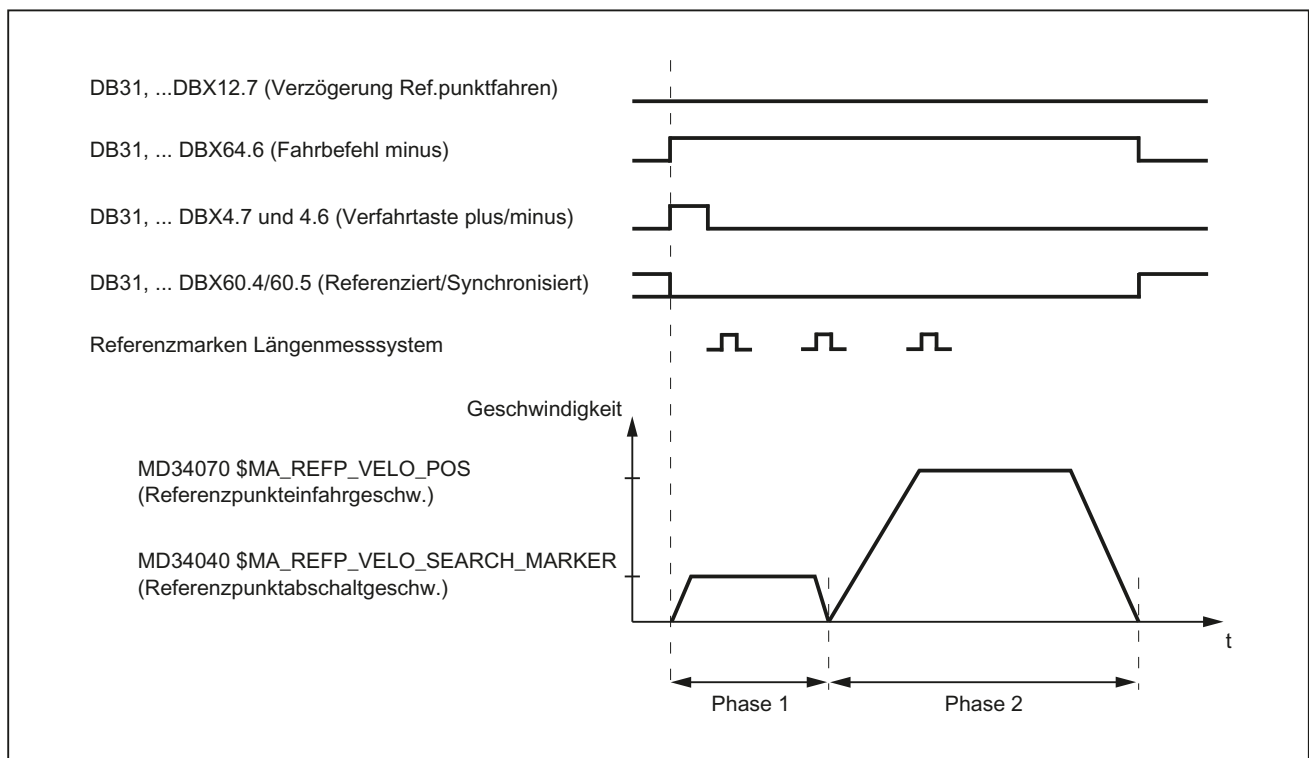


Bild 14-13 Abstandscodierte Referenzmarken

## 14.6.4 Phase 1: Überfahren der Referenzmarken mit Synchronisation

### Phase 1: Start

Zum Starten des Referenzpunktfahrens siehe Kapitel "Achsspezifisches Referenzieren (Seite 1321)" und "Kanalspezifisches Referenzieren (Seite 1324)".

### Referenznocken

Bei Messsystemen mit abstandscodierten Referenzmarken ist für das eigentliche Referenzieren kein Referenznocken erforderlich. Aus funktionalen Gründen wird aber für das kanalspezifische Referenzpunktfahren und beim Referenzpunktfahren aus dem Teileprogramm (<sup>G74</sup>) vor einem Verfahrbereichsende der Maschinenachse ein Referenznocken benötigt.

### Phase 1: Ablauf

#### Ablauf ohne Berührung eines Referenznockens

Nach dem Start des Referenzpunktfahrens beschleunigt die Maschinenachse auf die parametrierte Referenzpunktabschaltgeschwindigkeit:

MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER (Referenzpunktabschaltgeschwindigkeit)

Nach dem Überfahren der parametrierten Anzahl von Referenzmarken wird die Maschinenachse wieder angehalten und das Istwertsystem der Maschinenachse auf die von der NC ermittelte Absolutposition synchronisiert.

#### Ablauf bei Start auf dem Referenznocken

Steht die Maschinenachse beim Start des Referenzpunktfahrens auf dem Referenznocken, beschleunigt sie auf die parametrierte Referenzpunktabschaltgeschwindigkeit entgegen der parametrierten Referenzpunktanfahrrichtung:

MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER (Referenzpunktabschaltgeschwindigkeit)

MD34010 \$MA\_CAM\_DIR\_IS\_MINUS (Referenzpunktanfahren in Minusrichtung)

Somit ist sichergestellt, dass die Maschinenachse nicht vor dem Überfahren der parametrierten Anzahl von Referenzmarken an die Verfahrbereichsgrenze stößt.

Nach dem Überfahren der parametrierten Anzahl von Referenzmarken wird die Maschinenachse wieder angehalten und das Istwertsystem der Maschinenachse auf die von der NC ermittelte Absolutposition synchronisiert.

**Ablauf bei Berührung des Referenznockens während des Referenzierens**

Nach dem Start des Referenzpunktfahrens beschleunigt die Maschinenachse auf die parametrierte Referenzpunktabschaltgeschwindigkeit:

MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER (Referenzpunktabschaltgeschwindigkeit)

Vor dem Überfahren der parametrierten Anzahl von Referenzmarken berührt die Maschinenachse den Referenznocken. Daraufhin wird sie reversiert und die Referenzmarkensuche in entgegengesetzter Richtung erneut gestartet.

Nach dem Überfahren der parametrierten Anzahl von Referenzmarken wird die Maschinenachse wieder angehalten und das Istwertsystem der Maschinenachse auf die von der NC ermittelte Absolutposition synchronisiert.

**Plausibilitätsprüfung des Referenzmarkenabstands**

Ein Fehler liegt vor, wenn die NC beim Referenzpunktfahren für zwei aufeinander folgende Referenzmarken einen größeren Abstand als dem Doppelten des parametrierten Referenzmarkenabstandes ermittelt:

MD34300 \$MA\_ENC\_REFP\_MARKER\_DIST (Referenzmarkenabstand)

Die Maschinenachse wird daraufhin mit der halben parametrierten Referenzpunktabschaltgeschwindigkeit (MD34040) in entgegengesetzter Richtung verfahren und die Referenzmarkensuche erneut gestartet.

Wird erneut ein fehlerhafter Referenzmarkenabstand ermittelt, wird die Maschinenachse angehalten und das Referenzpunktfahren abgebrochen (Alarm 20003 "Fehler im Messsystem").

**Abbruchkriterium**

Wird nicht innerhalb der parametrierten Wegstrecke die parametrierte Anzahl von Referenzmarken erkannt, wird die Maschinenachse angehalten und das Referenzpunktfahren abgebrochen:

MD34060 \$MA\_REFP\_MAX\_MARKER\_DIST (max. Wegstrecke zur Referenzmarke)

**Eigenschaften der Phase 1**

Nach erfolgreichem Abschluss der Phase 1 ist das Istwertsystem der Maschinenachse synchronisiert.

### 14.6.5 Phase 2: Fahren auf Zielpunkt

#### Phase 2: Start

Phase 2 wird automatisch gestartet, sobald Phase 1 ohne Alarm beendet wurde.

#### Ausgangssituation:

- Die Maschinenachse steht unmittelbar nach der letzten der parametrisierten Anzahl von Referenzmarken.
- Das Istwertsystem der Maschinenachse ist synchronisiert.

#### Phase 2: Ablauf

In Phase 2 wird die Maschinenachse zum Abschluss des Referenzpunktfahrens auf eine definierte Zielposition (Referenzpunkt) gefahren. Dieses Verhalten kann zur Abkürzung des Referenzpunktfahrens auch unterdrückt werden:

MD34330 \$MA\_STOP\_AT\_ABS\_MARKER = <Wert>

Wert	Bedeutung
0	Fahren auf Zielposition
1	Kein Fahren auf Zielposition

#### Fahren auf Zielposition (Normalfall)

Die Maschinenachse beschleunigt auf die parametrisierte Referenzpunkteinfahrtgeschwindigkeit und fährt auf den parametrisierten Zielpunkt (Referenzpunkt):

MD34070 \$MA\_REFP\_VELO\_POS (Referenzpunkteinfahrtgeschwindigkeit)

MD34100 \$MA\_REFP\_SET\_POS (Referenzpunktwert)

Die Maschinenachse ist referenziert. Als Kennung setzt die NC, abhängig vom aktiven Messsystem, das entsprechende Nahtstellensignal:

DB31, ... DBX60.4 / 60.5 (Referenziert/Synchronisiert 1/2) = 1

#### Kein Fahren auf Zielposition

Die Maschinenachse ist jetzt referenziert. Als Kennung setzt die NC, abhängig vom aktiven Messsystem, das entsprechende Nahtstellensignal:

DB31, ... DBX60.4 / 60.5 (Referenziert/Synchronisiert 1/2) = 1

## Eigenschaften der Phase 2

Abhängig davon, ob für die Maschinenachse ein Referenznocken parametrier ist, weist die Phase 2 unterschiedliche Eigenschaften auf.

### Maschinenachse ohne Referenznocken

MD34000 \$MA\_REFP\_CAM\_IS\_ACTIVE (Achse mit Referenznocken) = 0

Eigenschaften:

- Vorschubkorrektur ist wirksam.
- Vorschubhalt (kanalspezifisch und achsspezifisch) ist wirksam.
- NC-Stop und NC-Start sind wirksam.

### Maschinenachse mit Referenznocken

MD34000 \$MA\_REFP\_CAM\_IS\_ACTIVE (Achse mit Referenznocken) = 1

Eigenschaften:

- Vorschubkorrektur ist **nicht** wirksam.  
Intern wird mit Vorschubkorrektur = 100% verfahren.  
Bei einer vorgegebenen Vorschubkorrektur von 0% erfolgt Abbruch.
- Vorschubhalt (kanalspezifisch und achsspezifisch) ist wirksam.
- NC-Stop und NC-Start sind **nicht** wirksam.
- Wird nach dem Verlassen des Referenznockens nicht innerhalb der parametrieren Wegstrecke die parametrieren Anzahl von Referenzmarken erkannt, wird die Maschinenachse angehalten:  
MD34060 \$MA\_REFP\_MAX\_MARKER\_DIST (max. Wegstrecke zur Referenzmarke)

## Besonderheiten bei rotatorischen Messsystemen

Bei rotatorischen, abstandscodierten Messsystemen ist die absolute Position nur innerhalb einer Umdrehung eindeutig bestimmbar. Abhängig vom mechanischen Anbau des Gebers stimmt der Hardware-seitige Überlauf der Absolutposition nicht immer mit dem Verfahrbereich der Rundachse überein.

### Besonderheiten bei Modulo-Rundachsen

Bei Modulo-Rundachsen wird die Referenzpunktposition auf den parametrisierten Modulo-Bereich abgebildet:

MD30330 \$MA\_MODULO\_RANGE (Größe des Modulobereichs)

MD30340 \$MA\_MODULO\_RANGE\_START (Startposition des Modulobereichs)

---

#### Hinweis

Die Abbildung der Referenzpunktposition auf den parametrisierten (fiktiven) Modulo-Bereich erfolgt auch bei Achsfunktion "Bestimmung der Referenzpunktposition rotatorischer, abstandscodierter Geber innerhalb des projektierten Modulobereiches":

MD30455 \$MA\_MISC\_FUNCTION\_MASK (Achsfunktionen), BIT1 = 1

---

## 14.7 Referenzieren über Istwertabgleich

### 14.7.1 Istwertabgleich auf das referenzierende Messsystem

#### Funktion

Bei Istwertabgleich auf das referenzierende Messsystem wird nach dem erfolgreichen Referenzieren eines Messsystems einer Maschinenachse die ermittelte absolute Istposition direkt auf alle anderen Messsysteme der Maschinenachse übertragen und diese als referenziert gekennzeichnet:

DB31, ... DBB60.4 / 60.5 (Referenziert/Synchronisiert 1/2) = 1

#### Vorteil

Die Maschinenachse kann beim Umschalten vom explizit referenzierten Messsystem auf das per Istwertabgleich referenzierte Messsystem kontinuierlich in Regelung bleiben (Reglerfreigabe aktiv), da durch die abgeglichene Istposition kein Istwertsprung stattfindet.

---

#### Hinweis

Zur Erhöhung der Positioniergenauigkeit durch Ermittlung der Messsystem-spezifischen Geberfeininformation wird empfohlen, das per Istwertabgleich referenzierte Messsystem nach dem Umschalten explizit neu zu referenzieren.

---

#### Aktivierung

Die Aktivierung des Istwertabgleichs auf das referenzierende Messsystem erfolgt maschinenachsspezifisch über:

MD34102 \$MA\_REFP\_SYNC\_ENCS = 1

## 14.7.2 Istwertabgleich bei Messsystemen mit abstandscodierten Referenzmarken

### Funktion

Zur Erhöhung der Positioniergenauigkeit durch Ermittlung der Messsystem-spezifischen Geberfeininformation wird empfohlen, das per Istwertabgleich referenzierte Messsystem nach einer Messsystemumschaltung explizit neu zu referenzieren.

Wird für das passive Messsystem ein Geber mit abstandscodierten Referenzmarken verwendet, kann unter folgenden Voraussetzungen das Referenzieren vermieden werden:

1. Aktives Messsystem: indirektes Messsystem (Motormesssystem) mit z. B. Absolutwertgeber
2. Passives Messsystem: direktes Messsystem mit abstandscodierten Referenzmarken
3. Verfahrensbewegung der Maschinenachse mit dem referenzierten indirekten Messsystem vor der Messsystemumschaltung so, dass die zum Referenzieren erforderliche Anzahl von Referenzmarken überfahren wird. Dabei wird das passive direkte Messsystem automatisch referenziert.

### Parametrierung

Neben den spezifischen Maschinendaten zum Referenzieren der einzelnen Messsysteme sind folgende Maschinendaten einzustellen:

- Freigabe des Istwertabgleichs:  
MD34102 \$MA\_REFP\_SYNC\_ENCS = 1
- Direktes Messsystem mit abstandscodierten Referenzmarken:
  - MD34200 \$MA\_ENC\_REFP\_MODE[*Messsystem*] = 3  
Abstandscodierte Referenzmarken
  - MD30242 \$MA\_ENC\_IS\_INDEPENDENT[*Messsystem*] = 2

Beim Istwertabgleich wird das passive direkte Messsystem auf die Istposition des aktiven indirekten Messsystems abgeglichen, jedoch nicht als referenziert gekennzeichnet. Nach dem Überfahren der parametrierten Anzahl von Referenzmarken wird das passive direkte Messsystem automatisch referenziert. Das Referenzieren erfolgt in jeder Betriebsart.



## Ablauf

1. Ausgangssituation: Beide Messsysteme sind nicht referenziert:  
DB31, ... DBX60.4 = 0 (Referiert / Synchronisiert 1)  
DB31, ... DBX60.5 = 0 (Referiert / Synchronisiert 2)
2. Referenzieren des indirekten Messsystems entsprechend Messsystemtyp:  
DB31, ... DBX60.4 = 1 (Referiert / Synchronisiert 1)  
DB31, ... DBX60.5 = 0 (Referiert / Synchronisiert 2)
3. Verfahren der Maschinenachse über die parametrisierte Anzahl von Referenzmarken.  
Dadurch automatisches Referenzieren des direkten Messsystems:  
DB31, ... DBX60.4 = 1 (Referiert / Synchronisiert 1)  
DB31, ... DBX60.5 = 1 (Referiert / Synchronisiert 2)

## 14.8 Referenzieren im Nachführbetrieb

### Funktion

Inkrementelle Messsysteme und Messsysteme mit abstandscodierten Referenzmarken können auch während Nachführbetrieb der Maschinenachse referenziert werden. Voraussetzung dafür ist die korrekte Parametrierung des Referenzpunktfahrens entsprechend dem verwendeten Messsystem (siehe Kapitel "Referenzieren bei inkrementellen Messsystemen (Seite 1327)" und "Referenzieren bei abstandscodierten Referenzmarken (Seite 1342)").

Beim Referenzieren im Nachführbetrieb wird die Maschinenachse nicht von der NC, sondern über eine externe Verfahrbewegung über die Gebernullmarke bzw. die parametrierte Anzahl von abstandscodierten Referenzmarken bewegt. Mit dem Erkennen der Gebernullmarke bzw. der parametrierten Anzahl von abstandscodierten Referenzmarken ist das Messsystem referenziert.

---

### Hinweis

#### Reproduzierbarkeit des Referenzierergebnisses

Beim NC-geführten Referenzpunktfahren ist die Reproduzierbarkeit des Referenzierergebnisses durch die Einhaltung der parametrierten Verfahrgeschwindigkeiten während des Referenziervorgangs gewährleistet. Beim Referenzieren im Nachführbetrieb liegt die Verantwortung beim Maschinenhersteller / Anwender, durch geeignete Maßnahmen die Reproduzierbarkeit der Referenzierergebnisse zu erzielen.

---

### Eindeutigkeit der Nullmarke

Das Referenzieren eines inkrementellen Messsystems basiert auf der eindeutigen Lage der Gebernullmarke bezogen auf den gesamten Verfahrbereich der Maschinenachse.

Da beim Referenzieren im Nachführbetrieb das Referenznocken-Signal von der NC nicht ausgewertet wird, ist beim Referenzieren im Nachführbetrieb ohne weitere Maßnahmen eine eindeutige Ermittlung des Referenzpunktes nur gegeben bei:

- nur einer Gebernullmarke im Verfahrbereich der Maschinenachse.
- linearen Messsystemen mit abstandscodierten Referenzmarken.
- Modulorundachsen (Absolutposition innerhalb einer Umdrehung).

### Nullmarkenauswahl bei mehreren Nullmarkensignalen

Werden aufgrund von maschinenspezifischen Gegebenheiten, z. B. Untersetzungsgetriebe zwischen Geber und Last, mehrere Gebernullmarken im Verfahrbereich der Maschinenachse erkannt, muss zur eindeutigen Bestimmung des Referenzpunktes ein BERO an der Maschine angebracht und über einen digitalen Eingang der NCU-Schnittstelle angeschlossen werden.

---

#### Hinweis

##### **BERO-Signal: Nullmarkenauswahl**

Die Verarbeitung des BERO-Signals erfolgt ausschließlich im Antrieb. Anschluss und Parametrierung siehe Kapitel "Hardware-Signale (Seite 1327)".

---

### Freigabe

Die Freigabe der Funktion "Referenzieren im Nachführbetrieb" erfolgt über:

MD34104 \$MA\_REFP\_PERMITTED\_IN\_FOLLOWUP = TRUE

### Start des Referenziervorgangs

Befindet sich die Maschinenachse bei Start des Referenzpunktfahrens im Nachführbetrieb (DB31, ... DBX61.3 == TRUE) wird das Messsystem im Nachführbetrieb referenziert.

Befindet sich die Maschinenachse bei Start des Referenzpunktfahrens nicht im Nachführbetrieb, wird das "normale" von der NC gesteuerte Referenzpunktfahren ausgeführt

Das Referenzieren im Nachführen kann in folgenden Betriebsarten gestartet werden:

- JOG-REF: Verfahrtasten
- AUTOMATIK: Teileprogrammanweisung <sup>G74</sup>

### Ablauf des Referenziervorgangs (Betriebsart JOG-REF)

1. Nachführbetrieb der Maschinenachse aktivieren:  
DB31, ... DBX1.4 (Nachführbetrieb) = 0  
DB31, ... DBX2.1 (Reglerfreigabe) = 0
2. Aktivierung des Nachführbetriebs berücksichtigen:  
DB31, ... DBX61.3 (Nachführen aktiv) = 1
3. Wechsel in die Betriebsart JOG-REF.
4. Externes Verfahren der Maschinenachse über die Gebernullmarke bzw. parametrisierte Anzahl von abstandscodierten Referenzmarken. Der Referenziervorgang wird NC-intern gestartet, sobald die Maschinenachse bewegt wird:  
DB31, ... DBX61.4 (Achse/Spindel steht) = 0
5. Nach erfolgreichem Erkennen der Gebernullmarke bzw. parametrisierten Anzahl von abstandscodierten Referenzmarken ist das Messsystem referenziert:  
DB31, ... DBX60.4/60.5 (Referiert/Synchronisiert 1/2) = 1

### Abbruch des Referenziervorgangs

Ein aktiver Referenziervorgang wird abgebrochen durch:

- Abwahl des Nachführbetriebs
- NCK-Reset

### Verhalten bei bereits referenzierten Messsystemen

Ein bereits referenziertes Messsystem kann nur in der Betriebsart AUTOMATIK über die Teileprogrammmanweisung  $G74$  erneut referenziert werden.

**Ablauf des Referenzvorgangs (Betriebsart AUTOMATIK)**

1. Wechsel in die Betriebsart AUTOMATIK.
2. Start des Teileprogramms.
3. Nachführbetrieb der Maschinenachse aktivieren:  
DB31, ... DBX1.4 (Nachführbetrieb) = 0  
DB31, ... DBX2.1 (Reglerfreigabe) = 0
4. Aktivierung des Nachführbetriebs berücksichtigen:  
DB31, ... DBX61.3 (Nachführen aktiv) = 1
5. Der Referenzvorgang wird NC-intern gestartet, sobald die Teileprogrammanweisung G74 abgearbeitet wird.
6. Externes Verfahren der Maschinenachse über die Gebernullmarke bzw. parametrierte Anzahl von abstandscodierten Referenzmarken.
7. Nach erfolgreichem Erkennen der Gebernullmarke bzw. parametrierten Anzahl von abstandscodierten Referenzmarken ist das Messsystem referenziert:  
DB31, ... DBX60.4 / 60.5 (Referiert/Synchronisiert 1/2) = 1
8. Der Satzwechsel erfolgt nach dem erfolgreichen Abschluss des Referenzvorgangs.

**Abbruch des Referenzvorgangs**

Ein aktiver Referenzvorgang wird abgebrochen durch:

- Abwahl des Nachführbetriebs
- NCK-Reset

**Verhalten bei bereits referenzierten Messsystemen**

Ein bereits referenziertes Messsystem kann erneut referenziert werden.

## 14.9 Referenzieren bei Absolutwertgebern

### 14.9.1 Informationen zur Justage

#### Maschinenachsen mit Absolutwertgebern

Der Vorteil von Maschinenachsen mit Absolutwertgebern ist, dass nach einem einmaligen Justage-Vorgang das bei inkrementellen Messsystemen notwendige Referenzpunktfahren (z. B. Hochlauf der Steuerung, Abwahl von "Parken" der Maschinenachse, etc.) entfallen kann und das Istwertsystem der Maschinenachse sofort auf die ermittelte Absolutposition synchronisiert wird.

#### Justage

Bei der Justage eines Absolutwertgebers wird der Istwert des Gebers einmalig mit dem Maschinennullpunkt abgeglichen und anschließend gültig gesetzt.

Der aktuelle Justage-Status eines Absolutwertgebers wird in dem folgenden achsspezifischen Maschinendatum der Maschinenachse angezeigt, an die er angeschlossen ist:

MD34210 \$MA\_ENC\_REFP\_STATE (Status Absolutwertgeber)

Wert	Bedeutung
0	Geber nicht justiert
1	Geberjustage freigegeben
2	Geber ist justiert

#### Justage-Arten

Folgende Justage-Arten werden unterstützt:

- Justage durch Eingabe der Referenzpunktverschiebung
- Justage durch Eingabe der Referenzpunktwertes
- Automatische Justage mittels Messtaster
- Justage mittels BERO

## Neujustage

Eine erneute Justage des Absolutwertgebers ist erforderlich nach:

- Getriebeumschaltung zwischen Last und Absolutwertgeber
- Ab-/Anbau des Absolutwertgebers
- Ab-/Anbau des Motors mit Absolutwertgeber
- Datenverlust im statischen NC-Speicher
- Batteriespannungsausfall
- Istwertsetzen (PRESETON)

### ACHTUNG

Dass eine Neujustage des Absolutwertgebers erforderlich ist, wird von der Steuerung nur bei folgenden Ereignissen erkannt:

- Getriebeumschaltung mit Wechsel des Übersetzungsverhältnisses
- Ansprechen der Nullmarkenüberwachung
- Neue Geber-Seriennummer nach einem Tausch des Absolutwertgebers

Die Steuerung setzt daraufhin den Status des Absolutwertgebers auf den Wert "0":

MD34210 \$MA\_ENC\_REFP\_STATE = 0 (Geber nicht justiert)

Es wird folgender Alarm angezeigt:

Alarm 25022 "Achse <Achsbezeichner> Geber <Nummer> Warnung 0"

Bei Ansprechen der Nullmarkenüberwachung wird zusätzlich folgender Alarm angezeigt:

Alarm 25020 "Achse <Achsbezeichner> Nullmarkenueberwachung aktiver Geber"

In allen anderen Fällen (z. B. PRESETON) liegt es in der alleinigen Verantwortung des Anwenders die Dejustage des Absolutwertgebers durch **manuelles** Setzen des Status auf den Wert "0" anzuzeigen und eine Neujustage durchzuführen.

### WARNUNG

#### Datensicherung

Beim Sichern von Maschinendaten einer Maschine A wird der Geberstatus (MD34210) der Maschinenachsen mitgesichert.

Beim Laden dieses Datensatzes in eine Maschine B des gleichen Typs, z. B. im Rahmen einer Serieninbetriebnahme oder nach einem Servicefall, werden die betroffenen Maschinenachsen von der NC automatisch als justiert / referenziert betrachtet. Es liegt in der besonderen Verantwortung des Maschinenherstellers / Anwenders, in solchen Fällen eine Neujustage vorzunehmen.

Siehe dazu auch die Erläuterungen zum Maschinendatum:

MD30250 \$MA\_ACT\_POS\_ABS (Absolute Geberposition zum Ausschaltzeitpunkt)

## 14.9.2 Justage durch Eingabe der Referenzpunktverschiebung

### Funktion

Bei der Justage durch Eingabe der Referenzpunktverschiebung wird die Differenz zwischen der auf der Bedienoberfläche angezeigten und der tatsächlichen Istposition in der Maschine ermittelt und der NC als Referenzpunktverschiebung bekannt gemacht.

### Vorgehensweise

1. Position der Maschinenachse bezogen auf den Maschinennullpunkt ermitteln, z. B. durch:
  - Messen der Position (z. B. Laserinterferometer)
  - Fahren der Maschinenachse auf eine bekannte Position (z. B. Festanschlag)
2. Auf der Bedienoberfläche angezeigte Istposition der Maschinenachse ablesen.
3. Referenzpunktverschiebung berechnen (Differenz der unter Punkt 1 und 2 ermittelten Istpositionen) und im Maschinendatum eintragen:

MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR (Referenzpunktverschiebung)

4. Absolutwertgeber als justiert kennzeichnen:

MD34210 \$MA\_ENC\_REFP\_STATE = 2

---

#### Hinweis

Die Geberjustage wird erst mit der nächsten Geberaktivierung (z. B. Hochlauf der Steuerung) wirksam.

---

5. POWER ON - Reset auslösen.
  6. Auf der Bedienoberfläche angezeigte Position der Maschinenachse kontrollieren.
- 

#### Hinweis

##### Losekompensation

Wenn für ein Messsystem mit Absolutwertgeber eine Losekompensation parametrierung ist, muss auf Folgendes geachtet werden:

Beim Verfahren der Maschinenachse auf die Justage-Maschinenposition darf keine Lose entstehen!

##### Referenzpunktverschiebung dauerhaft aktivieren

Erst nach POWER ON - Reset wird die eingegebene Referenzpunktverschiebung (MD34090) dauerhaft wirksam. Wird die Maschinenachse nach der Justage des Absolutwertgebers ohne zwischenzeitlichen POWER ON - Reset verfahren, kann die im Maschinendatum eingegebene Referenzpunktverschiebung z. B. im Rahmen interner Überlaufkorrekturen überschrieben werden.

##### Überprüfen der Istposition

Es wird nach der Justage des Absolutwertgebers empfohlen, nach dem nächsten Hochlauf der Steuerung (POWER ON) die Istposition der Maschinenachse zu verifizieren.

---



### 14.9.3 Justage durch Eingabe des Referenzpunktwertes

#### Funktion

Bei der Justage durch Eingabe des Referenzpunktwertes wird die absolute Position der Maschinenachse bezogen auf den Maschinennullpunkt ermittelt, z. B. durch:

- Messen der Position (z. B. Laserinterferometer)
- Fahren der Maschinenachse auf eine bekannte Position (z. B. Festanschlag)

Der so ermittelte Positionswert wird der NC als Referenzpunktwert bekannt gemacht. Die NC berechnet dann aus der Differenz von Geberabsolutwert und Referenzpunktwert die entsprechende Referenzpunktverschiebung.

#### Vorgehensweise

1. Referenziermodus auf "Übernahme des Referenzpunktwertes" einstellen:  
MD34200 \$MA\_ENC\_REFP\_MODE = 0
2. Maschinenachse in der Betriebsart JOG auf die zu messende (z. B. Laserinterferometer) bzw. bereits bekannte Position (z. B. Festanschlag) fahren.

---

#### Hinweis

Die Maschinenachse kann über die Verfahrtasten nur in die zum Referenzpunktfahren freigegebene Richtung verfahren werden:

MD34010 \$MA\_REFP\_CAM\_DIR\_IS\_MINUS (Referenzpunkt anfahren in Minusrichtung)

Um Verfälschungen der Position aufgrund von Lose im Antriebsstrang zu vermeiden, muss das Anfahren der bekannten Position mit geringer Geschwindigkeit erfolgen.

---

3. Die auf den Maschinennullpunkt bezogene Position der Maschinenachse als Referenzpunktwert der NC bekannt machen:  
MD34100 \$MA\_REFP\_SET\_POS = *Position*
4. Geberjustage freigeben:  
MD34210 \$MA\_ENC\_REFP\_STATE = 1
5. Zur Übernahme der eingegebenen Maschinendatenwerte NCK-Reset auslösen.
6. Wechsel in die Betriebsart JOG-REF.

7. Betätigen der bereits in Schritt 2 verwendeten, zum Referenzpunktfahren freigegebenen Verfahraste.

Beim Betätigen der Verfahraste bewegt sich die Maschinenachse nicht!

Die NC berechnet die Referenzpunktverschiebung aus der vom Absolutwertgeber gemeldeten und dem eingegebenen Referenzpunktwert. Das Ergebnis wird eingetragen in das Maschinendatum:

MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR (Referenzpunktverschiebung)

Der Status des Absolutgebers wird auf "Geber ist justiert" gesetzt:

MD34210 \$MA\_ENC\_REFP\_STATE = 2

Das Istwertsystem der Maschinenachse wird synchronisiert.

Die Maschinenachse ist jetzt referenziert. Als Kennung setzt die NC, abhängig vom aktiven Messsystem, das entsprechende Nahtstellensignal:

DB31, ... DBB60.4 / 60.5 (Referenziert/Synchronisiert 1/2) = 1

8. POWER ON - Reset auslösen.

---

#### **Hinweis**

#### **Referenzpunktverschiebung dauerhaft aktivieren**

Erst nach POWER ON - Reset wird die eingegebene Referenzpunktverschiebung (MD34090) dauerhaft wirksam.

Wird die Maschinenachse nach der Justage des Absolutwertgebers ohne zwischenzeitlichen POWER ON - Reset verfahren, kann die im Maschinendatum eingegebene Referenzpunktverschiebung z. B im Rahmen interner Überlaufkorrekturen überschrieben werden.

#### **Überprüfen der Istposition**

Es wird nach der Justage des Absolutwertgebers empfohlen, nach dem nächsten Hochlauf der Steuerung (POWER ON) die Istposition der Maschinenachse zu verifizieren.

---

## 14.9.4 Automatische Justage mittels Messtaster

### Funktion

Bei der automatischen Justage mittels Messtaster wird mit der Maschinenachse aus einem Teileprogramm heraus eine bekannte Position in der Maschine angefahren. Der Positionswert ist in der NC als Referenzpunktwert hinterlegt. Sobald der Messtaster schaltet, ist die Position erreicht und die NC berechnet aus der Differenz von Geberwert und Referenzpunktwert die entsprechende Referenzpunktverschiebung.

---

### Hinweis

#### Teileprogramm zur automatischen Justage

Das Teileprogramm zur automatischen Justage mittels Messtaster muss vom Maschinenhersteller / Anwender im Hinblick auf die jeweiligen Maschinenspezifika selbst erstellt werden.

#### Kollisionsfreiheit

Da für die zu referenzierenden Maschinenachsen keine Istwert-bezogenen Überwachungen aktiv sind, ist durch den Maschinenbediener während des Verfahrens der Maschinenachsen besonders auf die Kollisionsfreiheit in der Maschine zu achten!

---

### Teileprogramm

Das Teileprogramm zur automatischen Justage von Absolutwertgebern mittels Messtaster muss für jede zu referenzierende Maschinenachse die nachfolgend aufgeführten Punkte in dieser Reihenfolge realisieren:

1. Anfahren der Justageposition der Maschinenachse, die über das Ansprechen des Messtasters erkannt wird.

Das Anfahren sollte mehrfach aus der gleichen Richtung mit stufenweise reduzierter Geschwindigkeit erfolgen, um eine möglichst hohe Genauigkeit des Messwertes zu erhalten. Der Messwert wird in der Systemvariablen \$AA\_IM abgelegt.

2. Berechnen und Schreiben der Referenzpunktverschiebung:

```
MD34090 $MA_REFP_MOVE_DIST_CORR = MD34100 $MA_REFP_SET_POS - $AA_IM
```

3. Setzen des Absolutgeberstatus auf "Geber ist justiert":

```
MD34210 $MA_ENC_REFP_STATE = 2
```

## Ablauf

Bei der automatischen Justage mittels Messtaster ist folgendermaßen vorzugehen:

1. Teileprogrammstart auch bei nicht-referenzierten Maschinenachsen freigeben:  
MD20700 \$MC\_REFP\_NC\_START\_LOCK = 0
2. Für alle relevanten Maschinenachsen die Maschinenachspannung bezogen auf den Maschinennullpunkt bei Schalten des Messtasters als Referenzpunkt eingeben:  
MD34100 \$MA\_REFP\_SET\_POS = Referenzpunkt
3. Zur Übernahme der eingegebenen Maschinendatenwerte NCK-Reset auslösen.
4. Teileprogramm starten.
5. Nach Ablauf des Teileprogramms Teileprogrammstart bei nicht-referenzierten Maschinenachsen wieder sperren:  
MD20700 \$MC\_REFP\_NC\_START\_LOCK = 1
6. POWER ON - Reset auslösen, damit die vom Teileprogramm geschriebene Referenzpunktverschiebung dauerhaft wirksam wird:  
MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR (Referenzpunktverschiebung)

---

### Hinweis

#### Referenzpunktverschiebung dauerhaft aktivieren

Erst nach POWER ON - Reset wird die eingegebene Referenzpunktverschiebung (MD34090) dauerhaft wirksam.

Wird die Maschinenachse nach der Justage des Absolutwertgebers ohne zwischenzeitlichen POWER ON - Reset verfahren, kann die im Maschinendatum eingegebene Referenzpunktverschiebung z. B. im Rahmen interner Überlaufkorrekturen überschrieben werden.

#### Überprüfen der Istposition

Es wird nach der Justage des Absolutwertgebers empfohlen, nach dem nächsten Hochlauf der Steuerung (POWER ON) die Istposition der Maschinenachse zu verifizieren.

---

## 14.9.5 Justage mittels BERO

### Funktion

Bei der Justage mittels BERO wird, wie bei inkrementellen Messsystemen, ein Referenzpunktfahren auf eine definierte Maschinenposition durchgeführt. Der BERO ersetzt dabei die bei Absolutwertgebern nicht vorhandene Gebernullmarke. Nach erfolgreichem Abschluss des Referenzpunktfahrens berechnet die NC automatisch die Referenzpunktverschiebung aus der Differenz von Geberabsolutwert und parametriertem Referenzpunktwert.

### Parametrierung

#### NC: Referenziermodus

Der Referenziermodus ist auf "Referenzieren von inkrementellen, rotatorischen oder linearen Messsystemen: Nullimpuls auf der Geberspur" einzustellen:

MD34200 \$MA\_ENC\_REFP\_MODE[<Achse>] = 1

#### NC: Referenzpunktwert

Die Parametrierung des Referenzpunktwertes erfolgt über:

MD34100 \$MA\_REFP\_SET\_POS[<Achse>] = <Referenzpunktwert>

#### Antrieb: Nullmarkenersatz

Im Parameter p0494 oder p0495 ist der digitale Eingang der NCU-Schnittstelle einzustellen, an dem der BERO angeschlossen ist.

### Durchführung

Das Referenzpunktfahren kann in der Betriebsart JOG-REF manuell oder in den Betriebsarten AUTOMATIK oder MDA über ein Teileprogramm (<sup>G74</sup>) gestartet werden.

Nach erfolgreichem Abschluss des Referenzpunktfahrens ist der Absolutwertgeber justiert und das Istwertsystem der Maschinenachse synchronisiert.

Als Rückmeldung für das PLC-Anwenderprogramm setzt die NC, abhängig vom aktiven Messsystem, für die Maschinenachse das NC/PLC-Nahtstellensignal:

DB31, ... DBB60.4 / 60.5 (Referenziert/Synchronisiert 1 / 2) = 1

---

#### Hinweis

Wird nach der Justage des Absolutwertgebers der BERO entfernt, muss der Referenziermodus auf "Referenzieren mit Absolutwertgeber" umparametriert werden:

MD34200 \$MA\_ENC\_REFP\_MODE[<Achse>] = 0

---

### 14.9.6 Referenzpunktfahren bei Absolutwertgebern

#### Verfahrbewegungsfreigabe

Wird für eine Maschinenachse mit justiertem Absolutwertgeber als aktivem Messsystem Referenzpunktfahren ausgelöst (manuell in der Betriebsart JOG-REF oder automatisch per Teileprogrammanweisung G74), fährt die Maschinenachse abhängig von der parametrisierten Verfahrbewegungsfreigabe:

MD34330 \$MA\_REFP\_STOP\_AT\_ABS\_MARKER = <Wert>

Wert	Bedeutung
0	Verfahrbewegung ist freigegeben. Nach dem Auslösen des Referenzpunktfahrens fährt die Maschinenachse auf die Referenzpunktposition. Mit Erreichen der Referenzpunktposition ist das Referenzpunktfahren abgeschlossen.
1	Verfahrbewegung ist <b>nicht</b> freigegeben. Nach dem Auslösen des Referenzpunktfahrens fährt die Maschinenachse nicht und das Referenzpunktfahren ist sofort abgeschlossen.

### 14.9.7 Referenzpunktfahren bei rotatorischen Absolutwertgebern mit Nullmarkenersatz

#### Funktion

Um das Referenzpunktfahren mit einer Nullmarke, wie bei inkrementellen Gebern üblich (siehe Kapitel "Referenzieren bei inkrementellen Messsystemen (Seite 1327)"), auch bei Absolutwertgebern verwenden zu können, wird die fehlende HW-Nullmarke softwaretechnisch einmal pro Geberumdrehung, immer an gleicher Position innerhalb der Umdrehung, erzeugt.

## Unterschiede zum Referenzieren mit inkrementellen Gebern

Ein Absolutwertgeber mit Ersatznullmarke ist nicht als vollständiger Ersatz für einen inkrementellen Geber zu sehen. Alle Eigenschaften des Absolutwertgebers werden beibehalten. In der folgenden Tabelle sind die unterschiedlichen Eigenschaften von inkrementellen und absoluten Gebern aufgelistet:

Tabelle 14- 1 Eigenschaften von inkrementellen und absoluten Gebern

Eigenschaft	Inkrementeller Geber	Absolutgeber
Gebertyp	MD30240 \$MA_ENC_TYPE	
	= 1	= 4
Interne Geberposition	MD30250 \$MA_ACT_POS_ABS	
	Wert wird nur bei MD34210 ≥ 1 aktualisiert	Wert wird nur bei MD30270 = 0 aktualisiert
Verfahrenbereichserweiterung	MD30270 \$MA_ENC_ABS_BUFFERING	
	Keine Auswirkung	= 0 (Standard): aktiv
Referenzpunktverschiebung	MD34090 \$MA_REFP_MOVE_DIST_CORR	
	Werteingabe zulässig	Wert wird ausschließlich durch Steuerung aktualisiert
Unterstützte Referenzierarten	MD34200 \$MA_ENC_REFP_MODE	
	= 1, 2, 3, 4, 5, 6, 7	= 0, 1, 2
Justagestatus	MD34210 \$MA_ENC_REFP_STATE = 0, 1, 2	
	Automatische Dejustage des Gebers beim Ausschalten in der Bewegung.	Automatische Dejustage des Gebers beim Parametersatzwechsel mit Positionssprung bzw. bei Seriennummeränderung.
Modulobereich Absolutposition	MD34220 \$MA_ENC_ABS_TURNS_MODULO	
	= 0	= 1 - 4096
Geberseriennummer	MD34230 \$MA_ENC_SERIAL_NUMBER	
	= 0	Wert muss bei jedem Geberwechsel von der PLC aus aktualisiert werden, sonst Verlust der Justage plus Alarm.
Übernahme von Serien-IBN Files	Uneingeschränkt möglich.	Wegen Gebereigenschaften MD30250, MD30270, MD34090, MD34210, MD34220, MD34230 nur eingeschränkt möglich.
Einschaltzeit	0 Sekunden	mehrere Sekunden
Nullmarke	1 pro Geberumdrehung	keine
Nullmarkenüberwachung	Hardware	Software
Position nach POWER ON ohne Istwert-Pufferung	0.0	Letzte Position innerhalb von MD34220.
	MD34210 = 0	MD30270 = 1
Position nach POWER ON mit Istwert-Pufferung	Letzte Stillstandsposition vor dem Ausschalten.	Letzte Position inklusive kleiner Bewegungen während POWER OFF.
	MD34210 = 1	MD30270 = 0
Referenziert nach POWER ON	abhängig vom Justagestatus	

## Voraussetzung

Die Funktion ist nur mit rotatorischen Absolutgebern einsetzbar:

- MD31000 \$MA\_ENC\_IS\_LINEAR = 0
- MD30240 \$MA\_ENC\_TYPE = 4

## Parametrierung

- Referenzpunktfahren mit Nullmarke:  
MD34200 \$MA\_ENC\_REFP\_MODE = 1
- Eine Referenzpunktverschiebung darf in folgendem MD nicht eingegeben werden:  
MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR  
Dieses MD beschreibt in Verbindung mit Absolutwertgebern die Verschiebung zwischen Maschinen- und Absolutwertgebernulldpunkt und hat damit eine andere Bedeutung.
- Die lastseitige Nullmarkensuchgeschwindigkeit  
MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER  
sollte die Grenzfrequenz der Absolutspur des Gebers  
MD36302 \$MA\_ENC\_FREQ\_LIMIT\_LOW  
nicht überschreiten.  
Bei zu großer Geschwindigkeit kann keine Absolutinformation mehr gelesen und damit keine Ersatz-Nullmarke generiert werden.
- Wird keine Nullmarke innerhalb von:  
MD34060 \$MA\_REFP\_MAX\_MARKER\_DIST  
gefunden, wird ein Alarm ausgelöst.
- Ein Start der Nullmarkensuche mit dem Überfahren eines BERO (MD34200 = 5) wird nicht unterstützt. Ersatzweise ist MD34200 = 0 zu verwenden.
- Soll der Absolutwertgeber außer der letzten Position auch den Referenziert-Status über POWER OFF beibehalten, muss folgendes MD gesetzt werden:  
MD34210 \$MA\_ENC\_REFP\_STATE = 2

## Datensicherung und Serieninbetriebnahme

Einige Eigenschaften eines Absolutwertgebers schränken die Übertragbarkeit von Serien-IBN Files auf andere Maschinen ein. Die folgenden Maschinendaten müssen nach Einspielen der Serien-IBN kontrolliert und ggf. korrigiert werden:

- MD30250 \$MA\_ACT\_POS\_ABS (Interne Geberposition)
- MD30270 \$MA\_ENC\_ABS\_BUFFERING (Verfahrbereichserweiterung)
- MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR (Absolutverschiebung)
- MD34210 \$MA\_ENC\_REFP\_STATE (Justagestatus)
- MD34220 \$MA\_ENC\_ABS\_TURNS\_MODULO (Modulobereich)
- MD34230 \$MA\_ENC\_SERIAL\_NUMBER (Geber-Seriennummer)



## 14.9.8 Automatische Gebertausch-Erkennung

### Funktion

Eine automatische Gebertausch-Erkennung ist bei Absolutwertgebern erforderlich, um zu erkennen, ob der Geber getauscht wurde und demzufolge neu justiert werden muss.

Die NC liest dazu in jedem Hochlauf der Steuerung vom Antrieb die Geber-spezifische Seriennummer des Gebers. Bei einer Änderung der Seriennummern setzt die NC den Geberstatus zurück auf "Geber ist nicht justiert":

MD34210 \$MA\_ENC\_REFP\_STATE = 0

Der Status des Messsystems wird als "Nicht referenziert" angezeigt:

DB31, ... DBB60.4 / 60.5 (Referenziert/Synchronisiert 1/2) = 0

### Anzeige der Seriennummern

Die NC hinterlegt die im Hochlauf gelesenen Seriennummern maschinenachsspezifisch im Maschinendatum:

MD34230 \$MA\_ENC\_SERIAL\_NUMBER (Geber-Seriennummer)

---

#### Hinweis

Aktuell können nur die Seriennummern von Absolutwertgebern mit EnDat-Schnittstelle ermittelt werden. Für alle anderen Geber wird als Seriennummer Null gelesen / angezeigt.

Eine automatische Gebertausch-Erkennung ist daher nur in Verbindung mit den genannten Gebertypen möglich.

---

### Vermeidung von Neujustagen

In einigen Sonderfällen, z. B. nach dem Abbau und Wiederaufbau einer Maschinenachse (Anbau-Rundachsen), ist eine Neujustage nicht erforderlich / erwünscht.

Um die Neujustage zu vermeiden, muss für das Messsystem der entsprechenden Maschinenachse die Null als zu ignorierende Seriennummer parametrisiert werden:

MD34232 \$MA\_EVERY\_ENC\_SERIAL\_NUMBER = 0

Wird jetzt von der NC als Seriennummer Null gelesen, wird der Geberstatus nicht zurückgesetzt und die im Maschinendatum angezeigte Seriennummer bleibt erhalten.

**Beispielhafter Ablauf:**

1. Die NC liest für das Messsystem der entsprechenden Maschinenachse die Seriennummer des Absolutwertgebers und die Seriennummer ist nicht gleich Null.
2. Der Absolutwertgeber wird ordnungsgemäß justiert.
3. Die NC liest ab einem späteren Hochlauf der Steuerung als Seriennummer des Absolutwertgebers "Null".  
Die Null als Seriennummer wird ignoriert und der Geberstatus bleibt unverändert auf "Justiert".
4. Die NC liest im Hochlauf der Steuerung wieder die unter Punkt 1 gelesene und noch im Maschinendatum angezeigte Seriennummer. Der Geberstatus ist weiterhin "Justiert".

---

**Hinweis**

**PROFIBUS-Antriebe**

Da nicht jeder über PROFIBUS-DP angeschlossene Antrieb die Geber-Seriennummer rechtzeitig zum Hochlauf der Steuerung bzw. überhaupt liefern kann, wird zur Vermeidung von unnötigen Neujustagen NC-intern die Reichweite der Geber-Seriennummer bei PROFIBUS-Antrieben fest mit Null vorgegeben:

MD34232 \$MA\_EVERY\_ENC\_SERIAL\_NUMBER = 0

Eine manuelle Parametrierung auf 1 ist wirkungslos.

---

## 14.9.9 Einschalten des Messsystems

In folgenden Situationen wird das Messsystem einer Maschinenachse eingeschaltet:

- Hochlauf der Steuerung (POWER ON)
- Einschalten des Messsystems per Nahtstellensignal (Abwahl von "Parken"):  
DB31, ... DBB1.5 / 1.6 (Lagemesssystem 1/2)  
DB31, ... DBB2.1 (Reglerfreigabe)
- Unterschreiten der parametrierten Gebergrenzfrequenz (Spindeln):  
MD36300 \$MA\_ENC\_FREQ\_LIMIT

Beim Einschalten des Messsystems synchronisiert die NC das Istwertsystem der Maschinenachse auf den aktuellen Absolutwert. Verfahrbewegungen sind bei Achsen während der Synchronisation gesperrt, bei Spindeln nicht.

## Parametrieren der Gebergrenzfrequenz (Spindeln)

Der Absolutwertgeber der Fa. Heidenhain EQN 1325 besitzt eine Inkremental- und eine Absolutspur.

Wird eine Spindel mit Drehzahlen oberhalb der Gebergrenzfrequenz der Inkrementalspur betrieben, muss als Gebergrenzfrequenz die wesentlich kleinere Grenzfrequenz der Absolutspur parametrieren werden:

MD36300 \$MA\_ENC\_FREQ\_LIMIT

Ansonsten würde beim Einschalten des Messsystems durch Unterschreiten der parametrierten Gebergrenzfrequenz eine falsche Absolutposition gelesen werden. Dies hätte einen Positionsversatz im Istwertsystem der Maschinenachse zur Folge.

### Bestimmen der Gebergrenzfrequenz

Die zu parametrierende Gebergrenzfrequenz ergibt sich aus der kleineren der beiden folgenden Grenzdrehzahlen:

- Geber

Die Grenzdrehzahl bzw. die Gebergrenzfrequenz ist dem Datenblatt des Gebers zu entnehmen (z. B. Grenzdrehzahl = 2000 [1/min])

- NC

Bedingt durch das NC-interne Auswerteverfahren ergibt sich die Grenzdrehzahl, bis zu der eine fehlerfreie Ermittlung des Absolutwertes durch die NC möglich ist, zu 4 Geber-Umdrehungen pro Interpolationstakt.

Bei einem Interpolationstakt von z. B. 12 ms: Grenzdrehzahl = 4 / 12 ms = 20000 1/min

Die der Grenzdrehzahl entsprechende Grenzfrequenz berechnet sich zu:

$$MD36300 = \frac{4 * MD31020}{MD10050 * MD10070}$$

MD31020 \$MA\_ENC\_RESOL (Geberstriche pro Umdrehung)

MD10050 \$MN\_SYSCLOCK\_CYCLE\_TIME (Systemgrundtakt)

MD10070 \$MN\_IPO\_SYSCLOCK\_TIME\_RATIO (Faktor für Interpolatortakt)

---

### Hinweis

Die für Spindeln relevante Lageregeleinschaltdrehzahl ist entsprechend der Gebergrenzfrequenz des Absolutwertgebers der Spindel einzustellen:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO (Lageregeleinschaltdrehzahl)

MD36300 \$MA\_ENC\_FREQ\_LIMIT (Gebergrenzfrequenz)

---

### 14.9.10 Nicht unterstützte Referenziervarianten

Folgende Referenziervarianten werden im Zusammenhang mit Absolutwertgebern nicht unterstützt:

- Referenzieren / Justieren mittels Gebernullmarke
- Abstandscodierte Referenzmarken
- BERO mit Zweiflankenauswertung

## 14.10 Automatisches Wiederherstellen des Maschinenbezugs

Ohne definierten Maschinenbezug können beim Verfahren von Maschinenachsen keine positionsabhängigen Funktionen wie Transformationen oder Toolframe, ausgeführt werden. In verschiedenen Maschinsituationen müssen diese Funktionen sofort mit der Geberaktivierung, z.B. nach dem Einschalten der Steuerung oder nach Beenden von "Parken (Seite 69)", zum Verfahren von Achsen zur Verfügung stehen. Dabei sollen bzw. können die Maschinenachsen aber nicht für ein erneutes Referenzieren verfahren werden.

### Absolutwertgeber

Bei Messsystemen mit justierten Absolutwertgebern wird der Maschinenbezug ohne zusätzliche Maßnahmen sofort mit dem Lesen des Geberwertes wiederhergestellt.

### Inkrementalgeber

Bei inkrementellen Messsystemen kann das Wiederherstellen des Maschinenbezugs ohne Verfahren der Achsen durch "Automatisches Referenzieren" oder "Restaurieren der Istposition" wiederhergestellt werden.

### Randbedingungen

#### WARNUNG

Während der Zeit, in der das Messsystemen der Maschinenachse abgeschaltet ist, darf diese mechanisch **nicht** bewegt werden. Andernfalls kommt es zu einem Versatz zwischen der letzten gepufferten Istposition und der tatsächlichen Istposition der Maschinenachse. Dies würde zu einer Fehlsynchronisation des Messsystems mit Gefahr für Mensch und Maschine führen.

Die Unveränderlichkeit der Istposition ist vom Maschinenhersteller durch entsprechende Maßnahmen an der Maschine, z. B. Haltebremsen, zu unterstützen und vom Anwender sicherzustellen. Die Verantwortung dafür liegt ausschließlich beim Maschinenhersteller/Anwender.

Kann das Bewegen der Achse im abgeschalteten Zustand maschinenbaulich nicht verhindert werden, ist entweder ein Absolutgeber zu verwenden oder die Achse muss nach dem Einschalten per Referenzpunktfahren neu referenziert werden.

#### ACHTUNG

##### Sensor Module SMExx

Automatisches Referenzieren bzw. Restaurieren der Istposition nach dem Wiedereinschalten der Steuerung auf die letzte gepufferte Position, ist nur in Verbindung mit Sensor Modules SMExx (Externally Mounted) möglich. Bei Verwendung von Sensor Modules SMCxx (Cabinet) oder SMIxx (Integrated) kann die Istposition nach dem Wiedereinschalten der Steuerung (Power On) **nicht** wiederhergestellt werden. Das Messsystem der Maschinenachse muss neu referenziert werden.

## 14.10.1 Automatisches Referenzieren

### Funktion

Beim automatischen Referenzieren wird nach dem Einschalten der Steuerung die Istposition des aktiven Messsystem der Maschinenachse auf die letzte gepufferte Position und als Geberstatus "referenziert" gesetzt. Dadurch ist das Starten von Programmen in den Betriebsarten AUTOMATIK und MDA direkt nach dem Hochlauf der Steuerung möglich.

### Voraussetzungen

- Das nach dem Einschalten der Steuerung aktive Messsystemen, muss vor dem Ausschalten bereits erstmalig referenziert worden sein.
- Zum Zeitpunkt das Ausschaltens der Steuerung muss sich die Maschinenachse im Stillstand mit "Genauhalt fein" (DB31, ... DBX60.7 == 1) befinden.

---

### Hinweis

Beindet sich die Maschinenachse zum Zeitpunkt das Ausschaltens **nicht** im Stillstand mit "Genauhalt fein", wird nach dem Einschalten die Istposition mit dem Wert "0" initialisiert. Als Geberstatus wird "nicht referenziert" angezeigt.

---

### Parametrierung

Die Freigabe des Automatischen Referenzierens erfolgt durch Einstellen des Geberstatus auf "Automatisches Referenzieren ist freigegeben, aber der Geber ist noch nicht referenziert":

MD34210 \$MA\_ENC\_REFP\_STATE[<Geber>] = 1

Nach dem Referenzieren des Messsystem wird im Geberstatus angezeigt, dass das Automatische Referenzieren bei der nächsten Geberaktivierung ausgeführt wird:

MD34210 \$MA\_ENC\_REFP\_STATE[<Geber>] == 2

### NC/PLC-Nahtstellensignale

Nach dem Automatischen Referenzieren wird für das aktive Messsystem der Geberstatus "referenziert" angezeigt:

DB31, ... DBX60.4 / .5 == 1 (Referiert/Synchronisiert 1 / 2)

## Randbedingungen

### Geberaktivierung bei MD34210 \$MA\_ENC\_REFP\_STATE[<Geber>] == 1

Ein Geberstatus zum Zeitpunkt der Geberaktivierung gleich "1" bedeutet, dass "Automatische Referenzieren" freigegeben ist. Das Messsystem wurde aber entweder noch nicht referenziert oder es wurde nicht bei Stillstand der Maschinenachse im Zustand "Genauhalt fein" abgeschaltet. Für die Maschinenachse bzw. das aktive Messsystem wird folgendes gesetzt:

- Istposition = 0
- Aktives Messsystem, Geberstatus = "Nicht referenziert":  
DB31, ... DBX60.4 / .5 = 0 (Referenziert/Synchronisiert 1 / 2)

## Literatur

Funktionsbeschreibung Grundfunktionen, Kapitel "R1 Referenzieren" > "Referenzieren bei inkrementellen Messsystemen (Seite 1327)"

## 14.10.2 Restaurieren der Istposition

### Funktion

Beim Restaurieren der Istposition auf die letzte gepufferte Position, wird der Geberstatus des aktiven Messsystems auf "**restauriert**" gesetzt. Die Achse kann nur manuell verfahren werden.

#### Betriebsart AUTOMATIK

Zur Freigabe von NC-Start zum automatischen Abarbeiten von Programmen in der Betriebsart AUTOMATIK, muss das Messsystem der Maschinenachse neu referenziert werden.

#### Betriebsart MDA und Überspeichern

In der Betriebsart MDA und für das Überspeichern, kann die Bearbeitung auch ohne Referenzieren der Achsen, mit restaurierten Positionen erfolgen. Dazu muss kanalspezifisch NC-Start mit restaurierten Positionen explizit freigegeben werden:

MD20700 \$MC\_REFP\_NC\_START\_LOCK = 2

### Voraussetzung

Das nach dem Einschalten der Steuerung aktive Messsystemen, muss vor dem Ausschalten bereits erstmalig referenziert worden sein.

## Parametrierung

### Freigabe: Restaurieren der Istposition

Die Freigabe für das Restaurieren der Istposition erfolgt durch Einstellen den Geberstatus auf "Die letzte vor dem Ausschalten gepufferte Achsposition wird restauriert, kein automatisches Referenzieren":

MD34210 \$MA\_ENC\_REFP\_STATE[<Geber>] = 3

### Freigabe: NC-START für Betriebsarten "MDA" und "Überspeichern"

Die Freigabe von NC-START zum Abarbeiten von Teileprogrammen oder Teileprogrammätzen in den Betriebsarten "MDA" und "Überspeichern", mit Status "Position restauriert" erfolgt über:

MD34110 \$MA\_REFP\_CYCLE\_NR ≠ -1 (Achsenreihenfolge beim kanalspez. Referenzieren)

MD20700 \$MC\_REFP\_NC\_START\_LOCK = 2 (NC-START-Sperre ohne Referenzpunkt)

## NC/PLC-Nahtstellensignale

Die restaurierte Istposition wird als qualitativ nicht gleichwertig mit einer Istposition nach Referenzpunktfahren angesehen. Daher wird für das Messsystem der Maschinenachse der Status "Position restauriert" und nicht "Referiert/Synchronisiert" angezeigt.

### Istposition restauriert:

- DB31, ... DBX60.4 / .5 = 0 (Referiert/Synchronisiert 1 / 2)
- DB31, ... DBX71.4 / .5 = 1 (**Position restauriert**, Geber 1 / 2)

### Messsystem referenziert:

- DB31, ... DBX60.4 / .5 = 0 → 1 (**Referiert/Synchronisiert** 1 / 2)
- DB31, ... DBX71.4 / .5 = 1 → 0 (Position restauriert, Geber 1 / 2)

---

### Hinweis

Im Zustand "Position restauriert" ist die Überwachung der Verfahrbereichsgrenzen (Softwareendschalter, Arbeitsfeldbegrenzung, etc.) bereits aktiv.

---



## Randbedingungen

### Spindeln

Bei Überschreitung der Gebergrenzfrequenz wird eine Spindeln auf den Zustände "Nicht Referiert/Synchronisiert" zurückgesetzt:

- DB31, ... DBX60.4 / .5 = 1 → 0 (Referiert/Synchronisiert 1 / 2)
- DB31, ... DBX71.4 / .5 = 1 → 0 (Position restauriert, Geber 1 / 2)

## Literatur

Funktionsbeschreibung Grundfunktionen, Kapitel "R1 Referenzieren" > "Referenzieren bei inkrementellen Messsystemen (Seite 1327)"

## 14.11 Randbedingungen

### 14.11.1 Große Verfahrbereiche

**Linearachsen mit einem Verfahrbereich > 4096 Geberumdrehungen, rotatorischem Absolutwertgeber EQN 1325 und einem parametrierten Absolutwertgeber-Bereich von MD34220 \$MA\_ENC\_ABS\_TURNS\_MODULO = 4096:**

Der maximale mögliche Verfahrbereich entspricht dem von inkrementellen Gebern.

**Endlos drehende Rundachsen mit Absolutwertgeber:**

- Es sind beliebig ganzzahlige Übersetzungsverhältnisse zulässig.
- Es wird empfohlen, endlos drehende Rundachsen mit Absolutwertgeber als Modulo-Rundachsen (Verfahrbereich 0...360 Grad) zu parametrieren:  
MD30310 \$MA\_ROT\_IS\_MODULO = 1

Ansonsten kann es dazu kommen, dass die Rundachse nach dem Einschalten des Messsystems einen sehr großen Verfahrweg benötigt, um auf absolut Null zu fahren.

**Maschinenachsen mit Absolutwertgeber:**

Damit die Steuerung nach dem Wiedereinschalten des Messsystems die aktuelle Istposition korrekt ermittelt kann, darf bei ausgeschaltetem Messsystem die Maschinenachse nur um weniger als den halben Absolutwertgeber-Bereich bewegt werden:

MD34220 \$MA\_ENC\_ABS\_TURNS\_MODULO

## Hinweise zur Eindeutigkeit von Geberpositionen

---

### Hinweis

#### Lineare Absolutwertgeber

Der Absolutwert von Linearmaßstäben, z. B. Fa. Heidenhain LC181, ist im Rahmen der lieferbaren Maßstabslängen immer eindeutig.

#### Rotatorische Absolutwertgeber

Der Absolutwert von rotatorischen Absolutwertgebern ist nur im Bereich der spezifischen maximalen Geberumdrehungen eindeutig.

Der rotatorische Absolutwertgeber der Fa. Heidenhain EQN 1325 liefert z. B. einen eindeutigen Absolutwert im Bereich von 0 bis 4096 Geberumdrehungen.

Je nach Anbauverhältnissen ergeben sich daraus:

- Rundachse mit Geber an der Last: 4096 Last-Umdrehungen
- Rundachse mit Geber am Motor: 4096 Motor-Umdrehungen
- Linearachse mit Geber am Motor: 4096 Motor-Umdrehungen

Beispiel:

Ein rotatorischer Absolutwertgeber EQN 1325 ist am Motor einer Linearachse angebracht. Bei einer effektiven Spindelsteigung von 10 mm ergibt sich daraus ein eindeutiger Absolutwert im Verfahrbereich von -20,48 m bis +20,48 m.

---

## 14.12 Datenlisten

### 14.12.1 Maschinendaten

#### 14.12.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
11300	JOG_INC_MODE_LEVELTRIGGRD	INC/REF im Tippbetrieb/Dauerbetrieb

#### 14.12.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20700	REFP_NC_START_LOCK	NC-Startsperre ohne Referenzpunkt

#### 14.12.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30200	NUM_ENCS	Anzahl der Geber
30240	ENC_TYP	Istwert Gebertyp
30242	ENC_IS_INDEPENDENT	Geber ist unabhängig
30250	ACT_POS_ABS	Absolute Geberposition zum Ausschaltzeitpunkt
30270	ENC_ABS_BUFFERING	Absolutgeber: Verfahrbereichserweiterung
30300	IS_ROT_AX	Rundachse / Spindel
30310	ROT_IS_MODULO	Modulowandlung für Rundachse / Spindel
30330	MODULO_RANGE	Größe des Modulo-Bereiches
30340	MODULO_RANGE_START	Startposition Modulo-Bereich
30355	MISC_FUNCTION_MASK	Achsfunktionen
31122	BERO_DELAY_TIME_PLUS	BERO-Verzögerungszeit in Plus-Richtung
31123	BERO_DELAY_TIME_MINUS	BERO-Verzögerungszeit in Minus-Richtung
34000	REFP_CAM_IS_ACTIVE	Achse mit Referenznocken
34010	REFP_CAM_DIR_IS_MINUS	Referenzpunktanfahren in Minusrichtung
34020	REFP_VELO_SEARCH_CAM	Referenzpunktanfahrgeschwindigkeit
34030	REFP_MAX_CAM_DIST	max. Wegstrecke zum Referenznocken
34040	REFP_VELO_SEARCH_MARKER	Referenzpunktabschaltgeschwindigkeit
34050	REFP_SEARCH_MARKER_REVERSE	Richtungsumkehr auf Referenznocken
34060	REFP_MAX_MARKER_DIST	max. Wegstrecke zur Referenzmarke; max. Wegstrecke zu 2 Referenzmarken bei abstandscodierten Maßstäben

Nummer	Bezeichner: \$MA_	Beschreibung
34070	REFP_VELO_POS	Referenzpunkteinfahrtgeschwindigkeit
34080	REFP_MOVE_DIST	Referenzpunktabstand/Zielpunkt bei abstandscodiertem System
34090	REFP_MOVE_DIST_CORR	Referenzpunkt-/Absolutverschiebung abstandscodiert
34092	REFP_CAM_SHIFT	Elektronische Referenznockenverschiebung für inkrementelle Messsysteme mit äquidistanten Nullmarken.
34093	REFP_CAM_MARKER_DIST	Abstand Referenznocken/Referenzmarke
34100	REFP_SET_POS	Referenzpunktwert
34102	REFP_SYNC_ENCS	Istwertabgleich auf das referenzierende Messsystem
34104	REFP_PERMITTED_IN_FOLLOWUP	Freigabe Referenzieren im Nachführbetrieb
34110	REFP_CYCLE_NR	Achsreihenfolge beim kanalspez. Referenzieren
34120	REFP_BERO_LOW_ACTIVE	Polaritätswechsel des BERO Nockens
34200	ENC_REFP_MODE	Referenzier-Modus
34210	ENC_REFP_STATE	Status Absolutwertgeber
34220	ENC_ABS_TURNS_MODULO	Bereich Absolutwertgeber bei rotatorischen Gebern
34230	ENC_SERIAL_NUMBER	Geber-Seriennummer
34232	EVERY_ENC_SERIAL_NUMBER	Reichweite der Geber-Seriennummer
34300	ENC_REFP_MARKER_DIST	Grundabstand der Referenzmarken bei abstandscodierten Gebern
34310	ENC_MARKER_INC	Differenzabstand zweier Referenzmarken bei abstandscodierten Maßstäben
34320	ENC_INVERS	Längenmesssystem gegensinnig zum Maschinensystem
34330	REFP_STOP_AT_ABS_MARKER	Abstandscodiertes Längenmesssystem ohne Zielpunkt
35150	SPIND_DES_VELO_TOL	Spindeldrehzahltoleranz
36302	ENC_FREQ_LIMIT_LOW	Gebergrenzfrequenz Neusynchronisation
36310	ENC_ZERO_MONITORING	Nullmarkenüberwachung

## 14.12.2 Signale

### 14.12.2.1 Signale an BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
BAG-Reset	DB11.DBX0.7	DB3000.DBX0.7
Maschinenfunktion REF	DB11.DBX1.2	DB3000.DBX1.2

## 14.12.2.2 Signale von BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Aktive Maschinenfunktion REF	DB11.DBX5.2	DB3100.DBX1.2

## 14.12.2.3 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referieren aktivieren	DB21, ... .DBX1.0	DB3200.DBX1.0
OEM Kanal-Signal: (HMI → PLC) REF	DB21, ... .DBX28.7	-

## 14.12.2.4 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referieren aktiv	DB21, ... .DBX33.0	DB3300.DBX1.0
Reset	DB21, ... .DBX35.7	DB3300.DBX3.7
alle referenzpunktpflichtigen Achsen sind referenziert	DB21, ... .DBX36.2	DB3300.DBX4.2

## 14.12.2.5 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Nachführbetrieb (anfordern)	DB31, ... .DBX1.4	DB380x.DBX1.4
Lagemesssystem 1 / Lagemesssystem 2	DB31, ... .DBX1.5-6	DB380x.DBX1.5-6
Referenzpunktwert 1 bis 4	DB31, ... .DBX2.4-7	DB380x.DBX2.4-7
Verfahrtasten minus / plus	DB31, ... .DBX4.6-7	DB380x.DBX4.6-7
Verzögerung Referenzpunktfahren	DB31, ... .DBX12.7	DB380x.DBX1000.7

## 14.12.2.6 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referenziert, Synchronisiert 1 / 2	DB31, ... .DBX60.4-5	DB390x.DBX0.4-5
Nachführen aktiv	DB31, ... .DBX61.3	DB390x.DBX1.3
Fahrbefehl minus / plus	DB31, ... .DBX64.6-7	DB390x.DBX4.6-7
Restauriert 1 / 2	DB31, ... .DBX71.4-5	DB390x.DBX11.4-5

## S1: Spindeln

### 15.1 Kurzbeschreibung

Spindeln dienen primär dazu, ein Werkzeug oder ein Werkstück in eine Drehbewegung zu versetzen, um dadurch die Voraussetzungen für eine spanabhebende Bearbeitung zu schaffen.

Dazu muss die Spindel, je nach Typ der Maschine, folgende Funktionen unterstützen:

- Vorgabe einer Spindeldrehrichtung (M3, M4)
- Vorgabe einer Spindeldrehzahl (S, SVC)
- Spindel Halt ohne Orientierung (M5)
- Spindel Halt mit Orientierung / Spindelpositionierung (SPOS, M19 und SPOSA)
- Getriebestufenumschaltung (M40 bis M45)
- Spindel-Achsfunktionalität (Spindel wird zur Rundachse und umgekehrt)
- Gewindeschneiden (G33, G34, G35)
- Gewindebohren ohne Ausgleichsfutter (G331, G332)
- Gewindebohren mit Ausgleichsfutter (G63)
- Umdrehungsvorschub (G95)
- konstante Schnittgeschwindigkeit (G96, G961, G97, G971)
- programmierbare Spindeldrehzahlbegrenzungen (G25, G26, LIMS)
- Lagemessgeber an der Spindel oder am Spindelmotor montierbar
- Spindelüberwachungen auf min. und max. Drehzahl sowie max. Gebergrenzfrequenz und Zielpunktüberwachung der Spindel
- Ein-, Ausschalten der Lageregelung (SPCON, SPCOF, M70)
- Programmierung von Spindelfunktionen:
  - aus den Teileprogramm
  - über Synchronaktionen
  - über PLC mit FC18 oder über spezielle Spindelschnittstelle für einfache Spindelansteuerungen

## 15.2 Betriebsarten

### 15.2.1 Übersicht

#### Spindelbetriebsarten

Die Spindel kann folgende Spindelbetriebsarten besitzen:

- Steuerbetrieb
- Pendelbetrieb
- Positionierbetrieb
- Synchronbetrieb Synchronspindel

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Synchronspindel (S3)

- Gewindebohren ohne Ausgleichsfutter

**Literatur:**

Programmierhandbuch Grundlagen; Kapitel: Wegbefehle

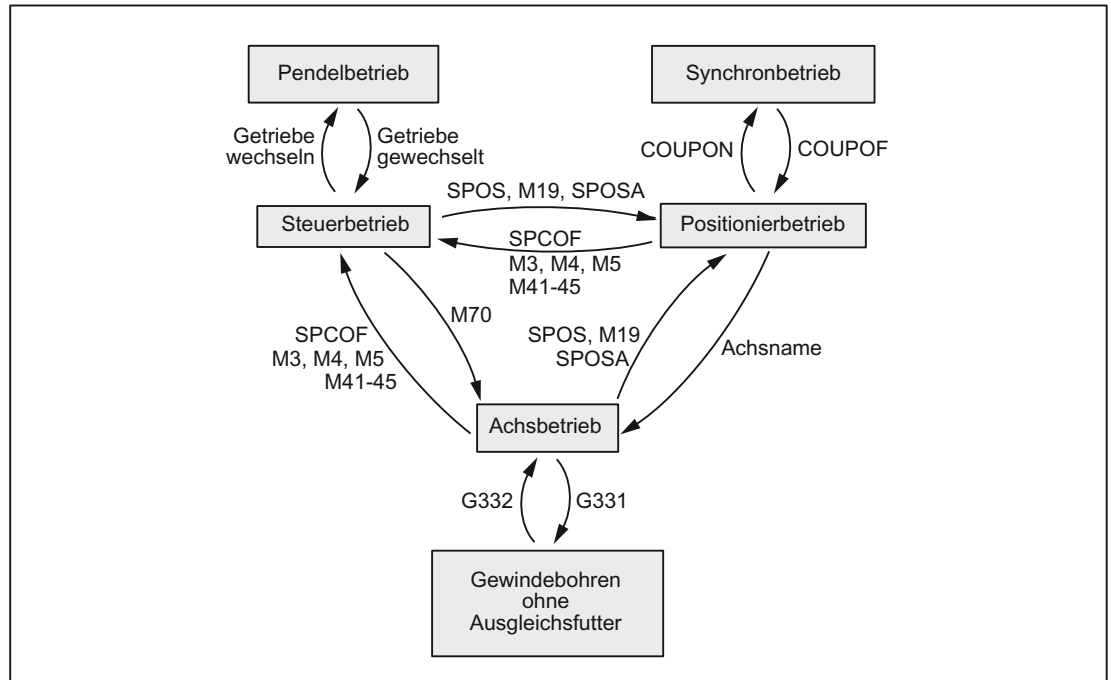
#### Achsbetrieb

Die Spindel kann vom Spindelbetrieb in den Achsbetrieb (Rundachse) geschaltet werden, wenn für Spindel- und Achsbetrieb ein gemeinsamer Motor verwendet wird.



## 15.2.2 Betriebsartenwechsel

Zwischen den Spindelbetriebsarten und dem Achsbetrieb kann wie folgt gewechselt werden:



- Steuerbetrieb → Pendelbetrieb

Die Spindel wechselt in den Pendelbetrieb, wenn durch die automatische Getriebestufenwahl (M40) in Verbindung mit einem neuen S-Wert oder durch M41 bis M45 eine neue Getriebestufe vorgegeben wurde. Die Spindel wechselt nur dann in den Pendelbetrieb, wenn die neue Getriebestufe ungleich der aktuellen Istgetriebestufe ist.

- Pendelbetrieb → Steuerbetrieb

Ist die neue Getriebestufe eingelegt, wird das Nahtstellensignal:

DB31, ... DBX84.6 (Pendelbetrieb)

rückgesetzt und mit dem Nahtstellensignal:

DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

in den Steuerbetrieb gewechselt.

Die letzte programmierte Spindeldrehzahl (S-Wert) ist wieder wirksam.

- Steuerbetrieb → Positionierbetrieb

Soll die Spindel aus der Drehung (M3 oder M4) mit Orientierung angehalten oder aus dem Stillstand (M5) neu orientiert werden, wird mit SPOS, M19 oder SPOSA in den Positionierbetrieb gewechselt.

- Positionierbetrieb → Steuerbetrieb

Soll die Orientierung der Spindel beendet werden, wird mit M3, M4 oder M5 in den Steuerbetrieb gewechselt. Die letzte programmierte Spindeldrehzahl (S-Wert) ist wieder wirksam.

- Positionierbetrieb → Pendelbetrieb  
Soll die Orientierung der Spindel beendet werden, kann mit M41 bis M45 in den Pendelbetrieb gewechselt werden. Ist der Getriebestufenwechsel beendet, wird die letzte programmierte Spindeldrehzahl (S-Wert) und M5 (Steuerbetrieb) wieder wirksam.
- Positionierbetrieb → Achsbetrieb  
Wurde eine Spindel mit Orientierung angehalten, wird durch die Programmierung des zugeordneten Achsnamens in den Achsbetrieb gewechselt. Die Getriebestufe bleibt erhalten.
- Steuerbetrieb → Achsbetrieb  
Soll aus dem Steuerbetrieb in den Achsbetrieb gewechselt werden, so kann dies auch durch Programmieren von M70 erfolgen. Dabei wird eine drehende Spindel wie bei M5 abgebremst, die Lageregelung eingeschaltet und der Parametersatz Null angewählt.
- Achsbetrieb → Steuerbetrieb  
Soll der Achsbetrieb beendet werden, kann mit M3, M4 oder M5 in den Steuerbetrieb gewechselt werden. Die letzte programmierte Spindeldrehzahl (S-Wert) ist wieder wirksam.
- Achsbetrieb → Pendelbetrieb  
Soll der Achsbetrieb beendet werden, kann mit M41 bis M45 in den Pendelbetrieb gewechselt werden (nur wenn die programmierte Getriebestufe ungleich der Istgetriebestufe ist). Ist der Getriebestufenwechsel beendet, wird die letzte programmierte Spindeldrehzahl (S-Wert) und M5 (Steuerbetrieb) wieder wirksam.

### 15.2.3 Steuerbetrieb

#### Wann Steuerbetrieb?

Bei folgenden Funktionen befindet sich die Spindel im Steuerbetrieb:

- Konstante Spindeldrehzahl:
  - S... M3/M4/M5 und G93, G94, G95, G97, G971
  - S... M3/M4/M5 und G33, G34, G35
  - S... M3/M4/M5 und G63
- Konstante Schnittgeschwindigkeit:
  - G96/G961 S... M3/M4/M5

Die Spindel muss nicht synchronisiert sein.

## Voraussetzungen

Ein Spindellageistwertgeber ist zwingend erforderlich für M3/M4/M5 in Verbindung mit:

- Umdrehungsvorschub (G95)
- konstante Schnittgeschwindigkeit (G96, G961, G97, G971)
- Gewindeschneiden (G33, G34, G35)
- Gewindebohren ohne Ausgleichsfutter (G331, G332)
- Lageregelung einschalten (SPCON, M70)

Ein Spindellageistwertgeber ist **nicht** erforderlich für M3/M4/M5 in Verbindung mit:

- zeitreziproker Vorschubverschlüsselung (G93)
- Vorschubgeschwindigkeit in mm/min bzw. inch/min (G94)
- Gewindebohren mit Ausgleichsfutter (G63)

## Drehzahlsteuerbetrieb

Der Drehzahlsteuerbetrieb ist besonders dann geeignet, wenn eine konstante Spindeldrehzahl verlangt wird, die Position der Spindel aber ohne Bedeutung ist (z. B. konstante Fräserdrehzahl zur Erzeugung eines gleichmäßigen Bildes der Werkstückoberfläche).

- Der Drehzahlsteuerbetrieb wird im Teileprogramm mit M3, M4, M5 oder mit SPCOF eingeschaltet.
- Das folgende NC/PLC-Nahtstellensignal ist gesetzt:  
DB31, ... DBX84.7 (Steuerbetrieb)
- Das NC/PLC-Nahtstellensignal:  
DB31, ... DBX61.5 (Lageregler aktiv)  
ist rückgesetzt, wenn ohne Lageregelung gearbeitet wird.
- Die Beschleunigung im Drehzahlsteuerbetrieb wird getriebestufenabhängig festgelegt im Maschinendatum:  
MD35200 \$MA\_GEAR\_STEP\_SPEEDCTRL\_ACCEL  
Der Wert sollte den physikalischen Gegebenheiten möglichst entsprechen.

## Lageregelbetrieb

Der Lageregelbetrieb ist besonders dann geeignet, wenn die Position der Spindel über einen längeren Zeitraum genau bekannt sein muss oder wenn Synchronspindelsollwertkopplung eingeschaltet werden soll.

- Der Lageregelbetrieb wird im Teileprogramm eingeschaltet mit: SPCON (<Spindelnummer>)
- Das folgende NC/PLC-Nahtstellensignal ist gesetzt:  
DB31, ... DBX61.5 (Lageregler aktiv)
- Die Beschleunigung im Lageregelbetrieb wird getriebestufenabhängig festgelegt im Maschinendatum:  
MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

## Eigener Spindel-Reset

Das Verhalten der Spindel nach Reset bzw. Programmende (M2, M30) wird eingestellt mit dem Maschinendatum:

MD35040 \$MA\_SPIND\_ACTIVE\_AFTER\_RESET (eigener Spindel-Reset)

Wert	Bedeutung
0	Die Spindel wird mit Reset bzw. Programmende sofort mit der gültigen Beschleunigung auf Stillstand abgebremst. Die letzte programmierte Spindeldrehzahl und Spindeldrehrichtung werden gelöscht.
1	Bei Reset bzw. Programmende bleiben die letzte programmierte Spindeldrehzahl (S-Wert) und die letzte programmierte Spindeldrehrichtung (M3, M4, M5) erhalten. Die Spindel wird nicht abgebremst.

Ist vor Reset bzw. Programmende die konstante Schnittgeschwindigkeit (G96, G961) aktiv, wird die aktuelle Spindeldrehzahl (bezogen auf 100% Spindelkorrektur) intern als letzte programmierte Spindeldrehzahl übernommen.

Die Spindel kann nur stillgesetzt werden mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX2.2 (Restweg löschen / Spindel-Reset)

Bei Alarmen, die für die Spindel Schnellstopp erzeugen, wird die Drehrichtung gelöscht. Die letzte programmierte Spindeldrehzahl (S-Wert) bleibt erhalten. Nach Beseitigung der Alarmursache muss die Spindel neu gestartet werden.

## Spindel-Istdrehzahlanzeige und Spindelverhalten bei G96, G961

DB31, ... DBX61.4 (Achse/Spindel steht)

Die Drehzahl, bei der die Spindel als "stehend" gilt, wird eingestellt mit dem Maschinendatum:

MD36060 \$MA\_STANDSTILL\_VELO\_TOL

Der Wert sollte so bemessen werden, dass im Stillstand das folgende NC/PLC-Nahtstellensignal sicher ansteht:

DB31,... DBX61.4 (Achse/Spindel steht)

Wenn DB31,... DBX61.4 (Achse/Spindel steht) gemeldet wird und für die Spindel keine Lageregelung aktiv ist, dann wird an der Bedienoberfläche die Istdrehzahl mit Null angezeigt und mit der Systemvariablen \$AA\_S[<n>] wird Null gelesen.

### Spindelverhalten bei konstanter Schnittgeschwindigkeit G96, G961

- Zu Beginn der Bearbeitung (Übergang von G0 zu Gx) und nach NC-Stop, G60 (Genauhalt modal) und G9 (Genauhalt satzweise) wird für den Bahnstart abgewartet, bis die Ist Drehzahl den Toleranzbereich der Soll Drehzahl erreicht hat:  
DB31, ... DBX83.5 (nIst = nSoll)
- Die NC/PLC-Nahtstellensignale:  
DB31, ... DBX83.5 (nIst = nSoll)  
und  
DB31, ... DBX83.1 (Soll Drehzahl begrenzt)  
sind auch bei starken Drehzahländerungsvorgaben (Plan-Achse fährt gegen Position 0) definiert gesetzt.
- Beim Unterschreiten der Minimaldrehzahl  
oder beim Erkennen des NC/PLC-Nahtstellensignals:  
DB31, ... DBX61.4 (Achse/Spindel steht)  
wird das NC/PLC-Nahtstellensignal:  
DB31, ... DBX83.5 (nIst = nSoll)  
zurückgesetzt (z. B. für Maschinennotstrategie).
- Eine begonnene Bahnbearbeitung (G64, Überschleifen) wird nicht unterbrochen.

Zusätzlich wird das Spindelverhalten von dem folgenden Maschinendatum beeinflusst:

MD35500 \$MA\_SPIND\_ON\_SPEED\_AT\_IPO\_START (Vorschubfreigabe bei Spindel im Sollbereich)

### Spindelverhalten bei Getriebestufenwechsel-Ende

- Mit dem NC/PLC-Nahtstellensignal:  
DB31, ... DBX16.3 (Getriebe ist umgeschaltet)  
wird der NC mitgeteilt, dass die neue Getriebestufe  
(NC/PLC-Nahtstellensignale DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C))  
gültig ist, und der Pendelbetrieb beendet wird.  
  
Dabei ist es unwesentlich, ob das NC/PLC-Nahtstellensignal:  
DB31, ... DBX18.5 (Pendeldrehzahl)  
noch gesetzt ist.  
  
Die Istgetriebestufe sollte der Sollgetriebestufe entsprechen.  
  
Zur Anwahl des Parametersatzes ist die gemeldete Istgetriebestufe relevant.
- Nach der Quittierung des Getriebestufenwechsel (GSW) durch die PLC (DB31, ... DBX16.3) befindet sich die Spindel im Drehzahlsteuerbetrieb (DB31, ... DBX84.7 = 1).  
  
War vor dem GSW eine Drehrichtung (M3, M4, M5 oder FC18: "Start Spindel drehen")  
oder eine Spindeldrehzahl (S-Wert) programmiert, dann werden nach dem GSW die  
letzte Drehrichtung und Drehzahl wieder aktiv.

## 15.2.4 Pendelbetrieb

Für die Spindel wird während des Getriebestufenwechsels der Pendelbetrieb aktiviert.

Die Wirkungsweise ist ausführlich beschrieben unter dem Thema "Getriebestufenwechsel mit Pendelbetrieb (Seite 1443)".

## 15.2.5 Positionierbetrieb

### 15.2.5.1 Allgemeine Funktionalität

#### Wann Positionierbetrieb?

Beim Positionierbetrieb wird die Spindel an der vorgegebenen Position angehalten. Dabei wird die Lageregelung eingeschaltet und bleibt bis zur Abwahl aktiv.

Bei folgenden Funktionen befindet sich die Spindel im Positionierbetrieb:

- SPOS[<n>]=...
- SPOS[<n>]=ACP(...)
- SPOS[<n>]=ACN(...)
- SPOS[<n>]=AC(...)
- SPOS[<n>]=IC(...)
- SPOS[<n>]=DC(...)
- SPOSA[<n>]=ACP(...)
- SPOSA[<n>]=ACN(...)
- SPOSA[<n>]=AC(...)
- SPOSA[<n>]=IC(...)
- SPOSA[<n>]=DC(...) gleich mit SPOSA[<n>]=...
- M19 bzw. M[<n>]=19

Die Adresserweiterung [<n>] mit <n> = Spindelnummer kann für die Hauptspindel entfallen.

#### SPOS[<n>]=AC(...)

Spindelpositionierung auf eine absolute Position (0 bis 359,999 Grad). Die Positionierrichtung wird entweder durch die momentane Spindeldrehrichtung (Spindel dreht) oder durch den Restweg bestimmt.

#### SPOS[<n>]=IC(...)

Spindelpositionierung auf eine inkrementelle Position (+/- 999999.99 Grad) bezüglich der letzten programmierten Position. Die Positionierrichtung wird durch das Vorzeichen des zu verfahrenen Wegs festgelegt.

### SPOS[<n>]=DC(...)

Spindelpositionierung auf kürzestem Weg auf eine absolute Position (0 bis 359,999 Grad). Die Positionierrichtung wird entweder durch die momentane Spindeldrehrichtung (Spindel dreht) oder automatisch durch die Steuerung bestimmt (Spindel steht).

### SPOS[<n>]=...

Gleicher Funktionsablauf wie SPOS[<n>]=DC(...).

### SPOS[<n>]=ACP(...)

Fährt die Position aus positiver Richtung an.

Beim Positionieren aus negativer Drehrichtung wird bis auf Drehzahl Null gebremst und in die Gegenrichtung beschleunigt, um die positive Anfahrrichtung ausführen zu können.

### SPOS[<n>]=ACN(...)

Fährt die Position aus negativer Richtung an.

Beim Positionieren aus positiver Drehrichtung wird bis auf Drehzahl Null gebremst und in die Gegenrichtung beschleunigt, um die negative Anfahrrichtung ausführen zu können.

## M19 (DIN 66025)

Mit M19 kann die Spindel positioniert werden. Die Position und der Positionsanfahrmodus werden dabei aus folgenden Settingdaten gelesen:

SD43240 \$SA\_M19\_SPOS[<n>] (Spindelposition für Spindelpositionieren mit M19)

SD43250 \$SA\_M19\_SPOSMODE[<n>] (Spindelpositionsanfahrmodus für Spindelpositionieren mit M19)

Die Positionierungsmöglichkeiten von M19 sind identisch mit denen von:

SPOS = <Anfahrmodus> <Position/Weg>

M19 wird als Hilfsfunktion an die NC/PLC-Nahtstelle alternativ zu M3, M4, M5 und M70 ausgegeben. Der M19-Satz bleibt im Interpolator für die Dauer der Positionierung aktiv (wie SPOS).

Teileprogramme, die M19 als Makro (z. B. DEFINE M19 AS SPOS=0) oder als Unterprogramm verwenden, bleiben weiterhin ablauffähig. Aus Gründen der Kompatibilität zu bisherigen Steuerungen kann die interne Behandlung von M19 (NCK positioniert die Spindel) nach folgendem Beispiel ausgeschaltet werden:

MD22000 \$MC\_AUXFU\_ASSIGN\_GROUP[0] = 4 ; Hilfsfunktionsgruppe: 4

MD22010 \$MC\_AUXFU\_ASSIGN\_TYPE[0] = "M" ; Hilfsfunktionsart: "M"

MD22020 \$MC\_AUXFU\_ASSIGN\_EXTENSION[0] = 0 ; Hilfsfunkt. Erweiterung: 0

MD22030 \$MC\_AUXFU\_ASSIGN\_VALUE[0] = 19 ; Hilfsfunktionswert: 19

### Implizit erzeugte Hilfsfunktion M19

Um eine Durchgängigkeit von M19 und SPOS bzw. SPOSA bezüglich des Verhaltens an der NC/PLC-Nahtstelle zu erreichen, kann bei SPOS und SPOSA die Hilfsfunktion M19 an die NC/PLC-Nahtstelle ausgegeben werden.

Für die Aktivierung dieser Funktion gibt es zwei Möglichkeiten:

- Kanalspezifische Aktivierung für alle Spindeln im Kanal über das Maschinendatum:  
MD20850 \$MC\_SPOS\_TO\_VDI (Ausgabe von M19 an die PLC bei SPOS/SPOSA)

Bit	Wert	Bedeutung
0	0	Wenn auch im MD35035 \$MA_SPIND_FUNCTION_MASK das Bit 19 auf "0" gesetzt ist, dann wird bei SPOS und SPOSA keine Hilfsfunktion M19 erzeugt. Damit entfällt auch die Quittierungszeit der Hilfsfunktion.
	1	Bei der Programmierung von SPOS und SPOSA im Teileprogramm wird die Hilfsfunktion M19 erzeugt und an die PLC ausgegeben. Die Adresserweiterung entspricht der Spindelnummer.

- Spindelspezifische und kanalübergreifende Aktivierung über das Maschinendatum:  
MD35035 \$MA\_SPIND\_FUNCTION\_MASK (Spindelfunktionen)

Bit	Wert	Bedeutung
19	0	Wenn auch im MD20850 \$MC_SPOS_TO_VDI das Bit 0 auf "0" gesetzt ist, dann wird bei SPOS und SPOSA keine Hilfsfunktion M19 erzeugt. Damit entfällt auch die Quittierungszeit der Hilfsfunktion.
	1	Bei der Programmierung von SPOS und SPOSA wird die implizite Hilfsfunktion M19 erzeugt und an die PLC ausgegeben. Die Adresserweiterung entspricht der Spindelnummer.

---

#### Hinweis

Die Aktivierung über MD35035 sollte bei Verwendung einer Spindel in mehreren Kanälen (Achs-/Spindeltausch) bevorzugt werden.

---

Die Hilfsfunktion M19 wird implizit erzeugt, wenn eine der beiden MD-Projektierungen = 1 ist.

Mit der Aktivierung erhöht sich die Mindestdauer eines SPOS/SPOSA-Satzes auf die Ausgabe- und Quittierungszeit der Hilfsfunktionen durch die PLC.

Die implizit erzeugte Hilfsfunktionsausgabe M19 hat die Eigenschaften "Quick" und "Ausgabe während der Bewegung". Diese Eigenschaften sind fest eingestellt und unabhängig von der M19-Projektierung in den Hilfsfunktionsspezifischen Maschinendaten (MD...\$M...\_AUXFU...).

Bei Spindelpositionieranweisungen über FC 18 wird **keine** Hilfsfunktion M19 implizit erzeugt.



## Positionierende

Das Positionierende kann programmiert werden mit:

<code>FINEA[S&lt;n&gt;]</code> :	Bewegungsende beim Erreichen von "Genauhalt fein" (DB31, ... DBX60.7)
<code>COARSEA[S&lt;n&gt;]</code> :	Bewegungsende beim Erreichen von "Genauhalt grob" (DB31, ... DBX60.6)
<code>IPOENDA[S&lt;n&gt;]</code> :	Bewegungsende beim Erreichen von "IPO-Stopp"

Zusätzlich kann mit IPOBRKA für Einzelachsinterpolation ein Bewegungsendekriterium für Satzwechsel bereits in der Bremsrampe (100-0%) eingestellt werden.

### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

## Satzwechsel

Sind die Bewegungsendekriterien für alle im Satz programmierten Spindeln bzw. Achsen und außerdem das Satzwechselkriterium für die Bahninterpolation erfüllt, so erfolgt der Satzwechsel. Dies gilt sowohl für Teileprogrammätze als auch für Technologiezyklussätze.

SPOS, M19 und SPOSA haben gleiche Funktionalität, unterscheiden sich aber im Satzwechselverhalten:

- SPOS und M19

Der Satzwechsel erfolgt, wenn alle im Satz programmierten Funktionen ihr Satzendeckriterium erreicht haben (z. B. alle Hilfsfunktionen von der PLC quittiert, alle Achsen Endpunkt erreicht) und die Spindel das Positionierende erreicht hat.

- SPOSA

Der Satzwechsel erfolgt, wenn alle im Satz programmierten Funktionen (außer der Spindel) ihr Satzendeckriterium erreicht haben. Steht `SPOSA` allein im Satz, wird der Satzwechsel sofort eingeleitet. Die Spindelpositionierung kann sich dabei über mehrere Sätze erstrecken (siehe WAITS).

## Koordinierung

Eine Koordinierung im Bewegungsablauf kann erreicht werden durch:

- PLC
- MD-Projektierung
- Programmierung im Teileprogramm

### PLC

Wenn das NC/PLC-Nahtstellensignal:  
DB31, ... DBX83.5 (Spindel im Sollbereich)  
nicht ansteht, kann das kanalspezifische NC/PLC-Nahtstellensignal:  
DB21, ... DBX 6.1 (Einlesesperre)  
gesetzt werden, um eine bestimmte Position einer Spindel abzuwarten.

**MD-Projektierung**

Mit der Einstellung:

MD35500 \$MA\_SPIND\_ON\_SPEED\_AT\_IPO\_START = 1

unter Berücksichtigung der Toleranz:

MD35150 \$MA\_SPIND\_DES\_VELO\_TOL

erfolgt die Bahninterpolation erst dann, wenn die Spindel auf vorgewählte Drehzahl hochgedreht hat.

Mit der Einstellung:

MD35500 \$MA\_SPIND\_ON\_SPEED\_AT\_IPO\_START = 2

werden fahrende Bahnachsen vor Bearbeitungsbeginn angehalten am letzten G0.

Die weitere Bearbeitung erfolgt:

- mit dem nächsten Fahrbefehl.
- wenn die Spindeldrehzahl hochgefahren ist.
- bei MD35510 \$MA\_SPIND\_STOPPED\_AT\_IPO\_START = 1  
(Bahn-Vorschubfreigabe, wenn Spindel steht).

**Programmierung im Teileprogramm**

Koordinierungsmaßnahmen im Teileprogramm haben folgende Vorzüge:

- Der Teileprogramm-Ersteller kann entscheiden, an welcher Programmstelle der Spindelhochlauf abgeschlossen sein muss, um z. B. eine Werkstückbearbeitung zu beginnen.
- Unnötige Wartezeiten werden vermieden.

Die Koordinierung im Teileprogramm erfolgt durch Programmierung der Anweisung `WAITS`:

<code>WAITS:</code>	für Hauptspindel (Masterspindel)
<code>WAITS [&lt;n&gt;]:</code>	für Spindel mit Nummer <n>
<code>WAITS [&lt;n&gt;, &lt;m&gt;, ...]:</code>	für mehrere Spindeln bis zur maximalen Spindelanzahl

 **VORSICHT**

Der Teileprogramm-Ersteller muss dafür sorgen, dass eine der folgenden Wartebedingungen für `WAITS` eintreten wird:

- Position erreicht
- Spindel steht
- Spindel auf programmierte Drehzahl hochgefahren

Bei kanalübergreifender Spindelnutzung muss der Teileprogramm-Ersteller sicherstellen, dass `WAITS` frühestens in der Phase startet, in der sich die Spindel eines anderen Kanals bereits in der Beschleunigungs- bzw. Bremsphase hin zur gewünschten neuen Drehzahl bzw. Drehrichtung befindet.

Mit der Abarbeitung nachfolgender Sätze wird solange gewartet, bis:

- mit `SPOSA` programmierte Position(en) erreicht sind.

- bei M5 Spindelstillstand erreicht ist:

DB31, ..., DBX 61.4 (Spindel steht)

unter Berücksichtigung der Toleranz:

MD36060 \$MA\_STANDSTILL\_VELO\_TOL

Mit dem Erkennen des ersten Auftretens des Signals wird WAITS beendet und der nächste Satz eingewechselt.

- bei M3/M4 (Drehzahlsteuerbetrieb) die Drehzahl im Sollbereich ist:

DB31, ..., DBX83.5 (Spindel im Sollbereich)

unter Berücksichtigung der Toleranz:

MD35150 \$MA\_SPIND\_DES\_VELO\_TOL

Mit dem Erkennen des ersten Auftretens des Signals wird WAITS beendet und der nächste Satz eingewechselt.

Diese Funktion von WAITS wirkt im programmierten Kanal.

Mit WAITS kann auf alle Spindeln gewartet werden, die diesem Kanal bekannt sind, wobei Spindeln auch in anderen Kanälen gestartet worden sein können.

### Sonderfälle

- **Toleranz für Spindelgeschwindigkeit:**

Bei Maschinendatum-Einstellung:  
MD35150 \$MA\_SPIND\_DES\_VELO\_TOL = 0  
wird das NC/PLC-Nahtstellensignal:  
DB31, ... DBX83.5 (Spindel im Sollbereich)  
immer auf 1 gesetzt.

WAITS wird beendet, sobald bei Drehzahl- oder bei Drehrichtungsänderung (M3/M4) die Spindel sollwertseitig die Vorgabe erreicht hat.

- **Fehlende Freigabesignale:**

Wartet WAITS im Drehzahlsteuerbetrieb auf das Signal "Spindel im Sollbereich" und die Spindel stoppt bzw. dreht nicht, weil ein Freigabesignal (axiale Vorschubfreigabe, Regler-, Impulsfreigabe, ...) fehlt, so wird der Satz erst beendet, wenn das Signal "Spindel im Sollbereich" ansteht, nachdem die Freigabesignale wieder gekommen sind.

- **Verhalten bei NC- und BAG-Stop:**

Wird während WAITS ein NC- oder BAG-Stop ausgelöst, so wird nach NC-Start der Wartevorgang mit allen oben genannten Randbedingungen fortgesetzt.

---

#### Hinweis

Insbesondere bei kanalübergreifender Spindelnutzung ist bei der Programmerstellung zu beachten, dass WAITS in einem Kanal nicht zu früh gestartet wird, also zu einem Zeitpunkt, wo die Spindel des anderen Kanals noch mit ihrer "alten" Drehzahl dreht. In diesem Fall steht das Signal "Spindel im Sollbereich" an und WAITS wird zu früh beendet.

Um dies zu vermeiden, empfiehlt es sich, vor WAITS unbedingt ein WAITM zu setzen.

---

### Vorschub

Die Positioniergeschwindigkeit wird projiziert im Maschinendatum:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO (Lageregeleinschaltdrehzahl)

Die projizierte Positioniergeschwindigkeit kann durch Programmierung oder durch Synchronaktionen geändert werden:

FA [S<n>] =<Wert>

mit:     <n>:         Spindelnummer  
          <Wert>:     Positioniergeschwindigkeit in Grad/min

Bei FA [S<n>] =0 wird die projizierte Geschwindigkeit wirksam.

## Beschleunigung

Die Beschleunigungen werden projiziert in den Maschinendaten:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL (Beschleunigung im Lageregelbetrieb)

MD35200 \$MA\_GEAR\_STEP\_SPEEDCTRL\_ACCEL (Beschleunigung im Drehzahlsteuerbetrieb)

Die projizierte Positionierdynamik kann durch Programmierung oder durch Synchronaktionen geändert werden:

ACC[S<n>] =<Wert>

mit: <n>: Spindelnummer

<Wert>: Beschleunigung in Prozent bezogen auf die projizierte Beschleunigung

Bei ACC[S<n>] = 0 wird die projizierte Beschleunigung wirksam.

## Abbruch Positioniervorgang

Der Positioniervorgang wird abgebrochen:

- durch das NC/PLC-Nahtstellensignal:  
DB31, ... DBX2.2 (Restweg löschen / Spindel-Reset).
- mit jedem Reset (z. B. Bedientafelfront-Reset).
- durch NC-Stop.

Das Abbruchverhalten ist unabhängig vom Maschinendatum:

MD35040 \$MA\_SPIND\_ACTIVE\_AFTER\_RESET (eigener Spindel-Reset)

## Besonderheiten

Der Spindelkorrektorschalter ist gültig.

### 15.2.5.2 Positionieren aus der Drehung

#### Ausgangssituation

Die Spindel kann sich zum Zeitpunkt des Positionierstarts (SPOS, M19 oder SPOSA im Programm) im Drehzahlsteuerbetrieb oder Lageregelbetrieb befinden.

Folgende Fälle sind zu unterscheiden:

Fall 1:	Spindel im Drehzahlsteuerbetrieb, Gebergrenzfrequenz überschritten
Fall 2:	Spindel im Drehzahlsteuerbetrieb, Gebergrenzfrequenz nicht überschritten
Fall 3:	Spindel im Lageregelbetrieb
Fall 4:	Spindeldrehzahl < Lageregel-Einschaltdrehzahl

## Ablauf

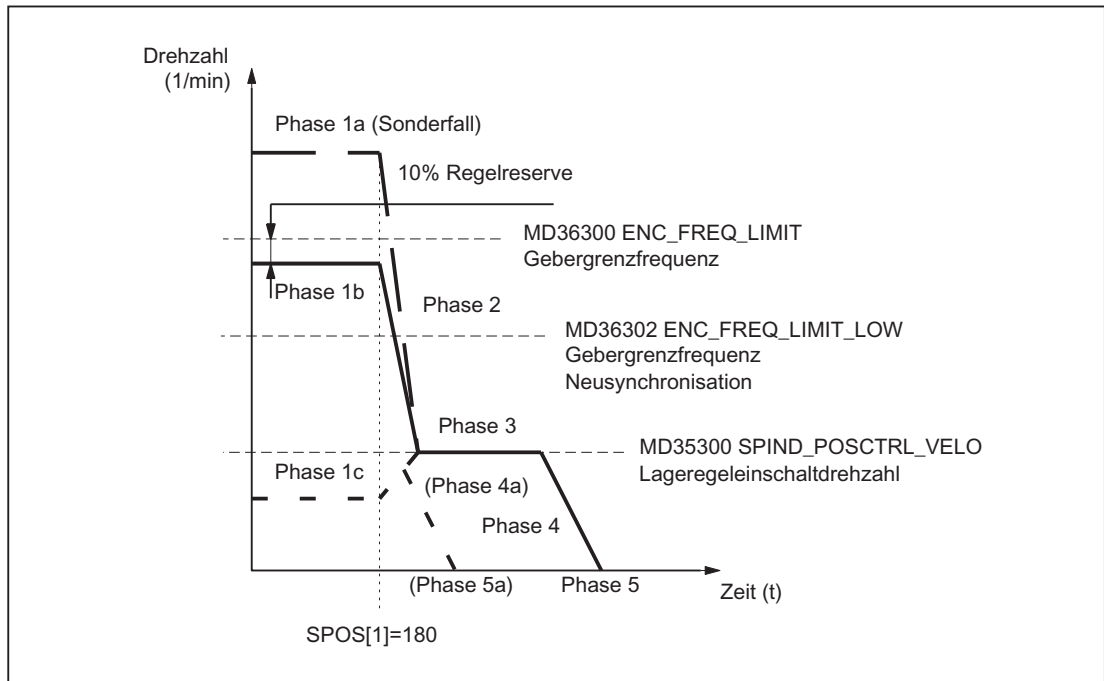


Bild 15-1 Positionieren aus der Drehung

**Hinweis**

Die Drehzahl, die sich aus der Projektierung der Gebergrenzfrequenz für die Geber-Neusynchronisation (MD36302 \$MA\_ENC\_FREQ\_LOW) ergibt, muss größer als die Lageregeleinschalt Drehzahl (MD35300 \$MA\_SPIND\_POSCTRL\_VELO) sein.

**Phase 1****Positionieren aus Phase 1a:**

Spindel dreht mit größerer Drehzahl als Gebergrenzfrequenz. Die Spindel ist nicht synchronisiert.

**Positionieren aus Phase 1b:**

Spindel dreht mit kleinerer Drehzahl als Gebergrenzfrequenz. Die Spindel ist synchronisiert.

**Hinweis**

Ist die Lageregelung eingeschaltet kann die Drehzahl nur 90% der Maximaldrehzahl der Spindel bzw. der Gebergrenzfrequenz betragen (10% Regelreserve notwendig).

### Positionieren aus Phase 1c:

Die Spindel dreht sich mit der programmierten Spindeldrehzahl, wobei die Drehzahl kleiner ist als die projektierte Lageregeleinschaltdrehzahl:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO

Die Spindel ist synchronisiert.

## Phase 2

### Spindeldrehzahl > Lageregeleinschaltdrehzahl

Mit dem Wirksamwerden des Befehles `SPOS, M19` oder `SPOSA` beginnt das Abbremsen der Spindel bis auf die Lageregeleinschaltdrehzahl mit der projektierten Beschleunigung:

MD35200 \$MA\_GEAR\_STEP\_SPEEDCTL\_ACCEL

Mit Unterschreiten der Gebergrenzfrequenz wird die Spindel synchronisiert.

### Spindeldrehzahl < Lageregeleinschaltdrehzahl

Mit der Programmierung von `SPOS, M19` oder `SPOSA` wird die Spindel in den Lageregelbetrieb geschaltet (wenn sie sich nicht schon im Lageregelbetrieb befindet).

Die projektierte Beschleunigung im Lageregelbetrieb wird aktiv:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

Die Berechnung des Verfahrenswegs zur Zielposition wird ausgeführt.

Das Verfahren der Spindel bis zum programmierten Zielpunkt wird zeitoptimal durchgeführt. D. h., der Zielpunkt wird mit höchstmöglicher Geschwindigkeit angefahren (maximal MD35300 \$MA\_SPIND\_POSCTRL\_VELO). Je nach den entsprechenden Randbedingungen werden die Phasen 2 - 3 - 4 - 5, bzw. 2 - 4a - 5a durchlaufen.

## Phase 3

### Spindeldrehzahl > Lageregeleinschaltdrehzahl

Mit Erreichen der projektierten Lageregeleinschaltdrehzahl (MD35300 \$MA\_SPIND\_POSCTRL\_VELO) wird:

- die Lageregelung zugeschaltet (wenn kein Lageregelbetrieb ist)
- der Restweg (zur Zielposition) berechnet
- auf die projektierte Beschleunigung im Lageregelbetrieb umgeschaltet (bzw. diese Beschleunigung beibehalten):

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

### Spindeldrehzahl < Lageregeleinschaltdrehzahl

Um den Zielpunkt zu erreichen, wurde bis auf die projektierte Lageregeleinschaltdrehzahl (MD35300 \$MA\_SPIND\_POSCTRL\_VELO) beschleunigt. Diese wird nicht überschritten.

Die Bremsenansatzpunktberechnung erkennt, wann mit der projektierten Beschleunigung im Lageregelbetrieb (MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL) genau in die programmierte Spindelposition eingefahren werden kann.

#### Phase 4

##### **Spindeldrehzahl > Lageregeleinschalt Drehzahl**

Die Spindel bremst vom errechneten "Bremspunkt" mit Maschinendatum: MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL bis zur Zielposition ab.

##### **Spindeldrehzahl < Lageregeleinschalt Drehzahl**

Zu dem Zeitpunkt, den die Bremseinsatzpunktberechnung in Phase 3 erkannt hat, bremst die Spindel mit der projektierten Beschleunigung im Lageregelbetrieb (MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL) auf Stillstand ab.

##### **Phase 4a:**

Der Zielpunkt liegt bereits bei Wirksamwerden des Befehles `SPOS` so nah, dass die Spindel nicht mehr bis auf die projektierte Lageregeleinschalt Drehzahl (MD35300 \$MA\_SPIND\_POSCTRL\_VELO) beschleunigt werden kann.

Die Spindel wird mit der projektierten Beschleunigung im Lageregelbetrieb (MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL) auf Stillstand abgebremst.

#### Phase 5

##### **Spindeldrehzahl > Lageregeleinschalt Drehzahl**

Die Lageregelung bleibt aktiv und hält die Spindel auf der programmierten Position.

---

##### **Hinweis**

Die max. Gebergrenzfrequenz des Spindellageistwertgebers wird von der Steuerung überwacht (Überschreiten möglich); im Lageregelbetrieb wird dann die Solldrehzahl auf 90% der Messsystem-Grenzdrehzahl verringert.

Dabei wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.1 (Programmierte Drehzahl zu hoch)

Wird nach erfolgter Solldrehzahl-Verringerung immer noch "MS-Grenzfrequenz überschritten" festgestellt, so wird ein Alarm gemeldet.

---



### **Spindeldrehzahl < Lageregeleinschalt Drehzahl (Phase 5, 5a)**

Die Spindel steht und hat die Position erreicht. Die Lageregelung ist aktiv und hält die Spindel auf der programmierten Position.

Wenn der Abstand zwischen der Spindelposition und der programmierten Position (Spindelsollposition) kleiner ist als projektierten Genauhaltgrenzen fein und grob, werden die folgenden NC/PLC-Nahtstellensignale gesetzt:

DB31, ... DBX60.6 (Position mit Genauhalt grob erreicht)

DB31, ... DBX60.7 (Position mit Genauhalt fein erreicht)

Die Genauhaltgrenzen werden festgelegt mit den Maschinendaten:

MD36010 \$MA\_STOP\_LIMIT\_FINE (Genauhalt fein)

MD36000 \$MA\_STOP\_LIMIT\_COARSE (Genauhalt grob)

---

### **Hinweis**

Der Positioniervorgang gilt als beendet, wenn das Positionierendekriterium erreicht ist und gemeldet wird.

Die Voraussetzung ist "Genauhalt fein". Das gilt für *SPOS, M19* oder *SPOSA* aus dem Teileprogramm, Synchronaktionen und Spindelpositionieren durch die PLC mit FC 18.

---

15.2.5.3 Positionieren aus dem Stillstand

Ablauf

Soll die Spindel aus dem Stillstand positioniert werden, sind zwei Fälle zu unterscheiden:

- Fall 1: Die Spindel ist nicht synchronisiert

Das ist dann der Fall, wenn die Spindel nach dem Einschalten der Steuerung und des Antriebs bzw. nach einem Getriebestufenwechsel positioniert werden soll (z. B. für einen Werkzeugwechsel).

MD31040 \$MA\_ENC\_IS\_DIRECT = 0

- Fall 2: Die Spindel ist synchronisiert

Das ist dann der Fall, wenn die Spindel nach dem Einschalten der Steuerung und des Antriebs vor der ersten Positionierung mindestens eine Spindelumdrehung mit M3 oder M4 gedreht und dann mit M5 angehalten wurde (Synchronisation mit der Nullmarke).

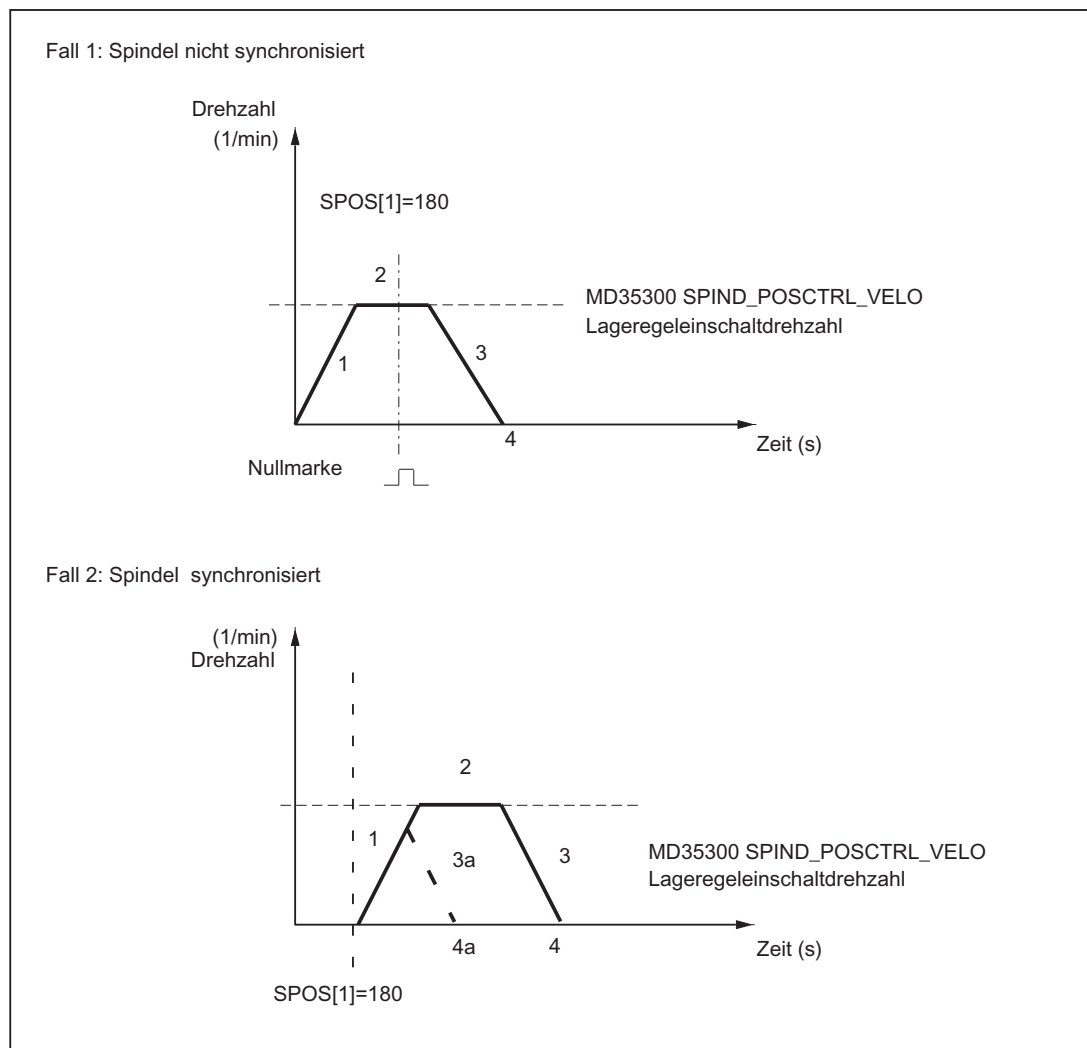


Bild 15-2 Positionieren bei stehender Spindel

## Phase 1

### Fall 1: Spindel nicht synchronisiert

Mit der Programmierung von `SPOS`, `M19` oder `SPOSA` beschleunigt die Spindel mit der Beschleunigung aus Maschinendatum:

MD35200 \$MA\_GEAR\_STEP\_SPEEDCTRL\_ACCEL (Beschleunigung im Drehzahlsteuerbetrieb)

Die Drehrichtung wird festgelegt durch das Maschinendatum:

MD35350 \$MA\_SPIND\_POSITIONING\_DIR (Drehrichtung beim Positionieren aus dem Stillstand)

Ausnahme:

Wird mit `ACN`, `ACP`, `IC` positioniert, dann wird die programmierte Verfahrrichtung aktiviert.

Mit der nächsten Nullmarke des Spindellage-Istwertgebers wird die Spindel synchronisiert und geht in den Lageregelbetrieb.

Es wird überwacht, ob die Nullmarke im hinterlegten Weg gefunden wird (außer bei `IC`):

MD34060 \$MA\_REFP\_MAX\_MARKER\_DIST (Maximale Wegstrecke zur Referenzmarke)

Wird die im Maschinendatum:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO (Positionierdrehzahl)

eingeegebene Drehzahl erreicht, ohne dass die Spindel synchronisiert ist, dreht die Spindel mit der Lagereglereinschaltdrehzahl weiter (keine weitere Beschleunigung mehr).

### Fall 2: Spindel synchronisiert

Mit der Programmierung von `SPOS`, `M19` oder `SPOSA` wird die Spindel in den Lageregelbetrieb geschaltet.

Die Beschleunigung aus dem folgenden Maschinendatum ist aktiv:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL (Beschleunigung im Lageregelbetrieb)

Die Drehrichtung wird durch die programmierte Bewegung (`ACP`, `ACN`, `IC`, `DC`) oder durch den anstehenden Restweg festgelegt.

Die im Maschinendatum:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO (Lageregeleinschaltdrehzahl)

eingeegebene Drehzahl wird nicht überschritten.

Die Berechnung des Verfahrweges zur Zielposition wird ausgeführt.

Das Verfahren der Spindel bis zum programmierten Zielpunkt wird zeitoptimal durchgeführt.

D. h., der Zielpunkt wird mit höchstmöglicher Geschwindigkeit angefahren (maximal

MD35300 \$MA\_SPIND\_POSCTRL\_VELO). Je nach den entsprechenden Randbedingungen werden die Phasen 1 - 2 - 3 - 4, bzw. 1 - 3a - 4a durchlaufen.

## Phase 2

### Fall 1: Spindel nicht synchronisiert

Ist die Spindel synchronisiert, wird die Lageregelung eingeschaltet.

Die Spindel dreht maximal mit der in Maschinendatum:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO

hinterlegten Drehzahl so lange weiter, bis die Bremseinsatzpunktberechnung erkennt, wann mit der festgelegten Beschleunigung genau in die programmierte Spindelposition eingefahren werden kann.

### Fall 2: Spindel synchronisiert

Um den Zielpunkt zu erreichen, wurde bis auf die im Maschinendatum:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO

eingeegebene Drehzahl beschleunigt.

Diese wird nicht überschritten.

Die Bremseinsatzpunktberechnung erkennt, wann mit der festgelegten Beschleunigung im Maschinendatum:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

genau in die programmierte Spindelposition eingefahren werden kann.

Zu dem Zeitpunkt, den die Bremseinsatzpunktberechnung in Phase 1 erkannt hat, bremst die Spindel mit der Beschleunigung aus folgendem Maschinendatum auf Stillstand ab:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

## Phase 3

Zu dem Zeitpunkt, den die Bremseinsatzpunktberechnung in Phase 2 erkannt hat, bremst die Spindel mit der Beschleunigung aus folgendem Maschinendatum auf Stillstand ab:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

### Phase 3a:

Der Zielpunkt liegt bereits bei Wirksamwerden des Befehles `SPOS` so nah, dass die Spindel nicht mehr bis auf das Maschinendatum:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO

beschleunigt werden kann.

Die Spindel wird mit der Beschleunigung aus dem folgenden Maschinendatum auf Stillstand abgebremst:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

#### Phase 4, 4a

Die Spindel steht und hat die Position erreicht. Die Lageregelung ist aktiv und hält die Spindel auf der programmierten Position.

Die NC/PLC-Nahtstellensignale:

DB31, ... DBX60.6 (Position mit Genauhalt grob erreicht)

und

DB31, ... DBX60.7 (Position mit Genauhalt fein erreicht)

werden gesetzt, wenn der Abstand zwischen der Spindelposition und der programmierten Position (Spindelsollposition) kleiner als der Wert Genauhaltgrenze fein und grob ist.

Dies wird festgelegt in den Maschinendaten:

MD36010 \$MA\_STOP\_LIMIT\_FINE

MD36000 \$MA\_STOP\_LIMIT\_COARSE

#### Phase 3:

Zu dem Zeitpunkt, den die Bremseinsatzpunktberechnung in Phase 2 erkannt hat, bremst die Spindel mit der Beschleunigung aus folgendem Maschinendatum auf Stillstand ab:

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

#### Phase 4:

Die Spindel steht und hat die Position erreicht. Die Lageregelung ist aktiv und hält die Spindel auf der programmierten Position.

Die NC/PLC-Nahtstellensignale:

DB31, ... DBX60.6 (Position mit Genauhalt grob erreicht)

und

DB31, ... DBX60.7 (Position mit Genauhalt fein erreicht)

werden gesetzt, wenn der Abstand zwischen der Spindelposition und der programmierten Position (Spindelsollposition) kleiner als der Wert Genauhaltgrenze fein und grob ist.

Dies wird festgelegt in den Maschinendaten:

MD36010 \$MA\_STOP\_LIMIT\_FINE

MD36000 \$MA\_STOP\_LIMIT\_COARSE

#### 15.2.5.4 Signal "Spindel in Position" für Werkzeugwechsel

##### Funktion

Der Bewegungsablauf beim Werkzeugwechsel, vor allem bei Fräsmaschinen, besteht hauptsächlich aus dem Positionieren der Spindel und dem anschließenden (bei Optimierungen auch zeitparallelen) Anfahren der Werkzeugwechselposition mit den Bahnachsen. Als zwingende Voraussetzung gilt das Erreichen der Spindelposition vor dem Anfahren der Werkzeugwechselposition.

Wird der Werkzeugwechselzyklus durch den Maschinenbediener unterbrochen (z. B. mit NC-Stop, NC-Stop Achsen plus Spindeln, BAG-Stop, etc.), dann muss ausgeschlossen werden, dass die Spindel mit falscher Position in den Werkzeugwechsler hineinfährt.

Zur Positionskontrolle wird daher beim Spindelpositionieren beim Erreichen der zuletzt programmierten Spindelposition mit "Genauhalt fein" das folgende NC/PLC-Nahtstellensignal ausgegeben:

DB31, ... DBX85.5 (Spindel in Position)

---

##### Hinweis

Das Signal wird ausschließlich bei der Funktion "Spindelpositionieren" ausgegeben.

Dazu gehören:

- SPOS, SPOSA und M19 im Teileprogramm
  - SPOS und M19 in Synchronaktionen
  - Spindelpositionieren unter Verwendung des FC18
  - Spindelpositionieren über PLC-Schnittstelle (DB31, ... DBX30.4)
- 

##### Setzen des Signals

Voraussetzungen für die Ausgabe des Signals DB31, ... DBX85.5 (Spindel in Position) sind:

- der Referiertzustand der Spindel:

DB31, ... DBX60.4/5 (Referiert/Synchronisiert 1/2) = 1

---

##### Hinweis

Da beim Positionieren der Spindel die Nullmarke selbständig gesucht wird, steht bei einem fehlerfreien Ablauf am Ende der Positionierbewegung das Referiertsignal immer an.

---

- das Erreichen von "Genauhalt fein":

DB31, ... DBX60.7 (Genauhalt fein) = 1

Zusätzlich muss die zuletzt programmierte Spindelposition sollwertseitig erreicht worden sein.

## Löschen des Signals

Mit der Wegnahme des Signals DB31, ... DBX60.7 (Genauhalt fein) wird immer auch das Signal DB31, ... DBX85.5 (Spindel in Position) zurückgesetzt.

## Weitere Eigenschaften

- Steht die Spindel nach einer Positionierung bereits auf der programmierten Position, so bleibt das NC/PLC-Nahtstellensignal DB31, ... DBX85.5 (Spindel in Position) gesetzt.
- Wird die Spindel nach einer Positionierung (Signal "Spindel in Position" wurde ausgegeben) z. B. in der Betriebsart JOG verfahren, dann wird das NC/PLC-Nahtstellensignal DB31, ... DBX85.5 (Spindel in Position) gelöscht.

Wenn die Spindel in dieser Betriebsart auf ihre ursprüngliche Position zurückgefahren wird, dann wird das NC/PLC-Nahtstellensignal DB31, ... DBX85.5 (Spindel in Position) erneut gesetzt. Die letzte Positionsvorgabe bleibt erhalten.

## 15.2.6 Achsbetrieb

### 15.2.6.1 Allgemeine Funktionalität

#### Funktionalität

Ist es für bestimmte Bearbeitungsaufgaben, z. B. an Drehmaschinen mit Stirnflächenbearbeitung, nicht ausreichend, die Spindel ausschließlich drehzahl geregelt über M3, M4, M5 zu verfahren bzw. und mit SPOS, M19 oder SPOSA zu positionieren, kann die Spindel in den lagegeregelten Achsbetrieb umgeschaltet und als Rundachse verfahren werden.

Beispielhafte Rundachsfunktionen:

- Programmierung mit Achsnamen
- Nullpunktverschiebungen (G54, G55, TRANS, ...)
- G90, G91, IC, AC, DC, ACP, ACN
- Kinematischen Transformationen (z. B. TRANSMIT)
- Bahninterpolation
- Verfahren als Positionierachse

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kapitel "Rundachsen (R2)"

### Voraussetzungen

- Der Spindelmotor ist für den Spindelbetrieb und den Achsbetrieb derselbe.
- Das Lagemesssystem kann für den Spindelbetrieb und den Achsbetrieb dasselbe sein oder es können getrennte Lagemesssysteme benutzt werden.
- Für den Achsbetrieb ist zwingend ein Lageistwertgeber erforderlich.
- Die Spindel muss zur Verwendung des Achsbetriebs referenziert sein bzw. z.B. mit G74 referenziert werden.

Beispiel:

Programmcode	Kommentar
M70	; Spindel in den Achsbetrieb umschalten
G74 C1=0 Z100	; Achse referenzieren
G0 C180 X50	; Achse lagegeregelt verfahren

### Projektierbare M-Funktion

Die M-Funktion, mit der die Spindel in den Achsbetrieb geschaltet wird, kann über das folgende Maschinendatum kanalspezifisch projiziert werden:

MD20094 \$MC\_SPIND\_RIGID\_TAPPING\_M\_NR

---

#### Hinweis

Die Steuerung erkennt aufgrund der Programmierabfolge selbständig den Übergang in den Achsbetrieb (siehe "Impliziter Übergang in den Achsbetrieb (Seite 1411)"). Die explizite Programmierung der projizierten M-Funktion zum Schalten der Spindel in den Achsbetrieb im Teileprogramm ist daher nicht explizit notwendig. Die M-Funktion kann jedoch weiterhin programmiert werden, um z. B. die Lesbarkeit des Teileprogramms zu erhöhen.

---



## Besonderheiten

- Der Vorschubkorrekturschalter ist gültig.
- Das NC/PLC-Nahtstellensignal beendet standardmäßig den Achsbetrieb nicht: DB21, ... DBX7.7 (Reset).
- Die NC/PLC-Nahtstellensignale: DB31, ... DBB16 bis DBB19 und DBB82 bis DBB91 sind ohne Bedeutung, wenn: DB31, ... DBX60.0 (Achse / Keine Spindel) = 0
- Der Achsbetrieb kann in jeder Getriebestufe eingeschaltet werden.  
Ist der Lageistwertgeber am Motor angebracht (indirektes Messsystem), können sich zwischen den Getriebestufen unterschiedliche Positionier- und Konturgenauigkeiten ergeben.
- Ist der Achsbetrieb aktiv, kann die Getriebestufe nicht gewechselt werden.  
Dazu muss die Spindel in den Steuerbetrieb geschaltet werden.  
Dies geschieht mit M41 ... M45 bzw. M5, SPCOF.
- Im Achsbetrieb wirkt der erste Parametersatz (Maschinendatenindex = Null).  
**Literatur**  
Funktionshandbuch Grundfunktionen; Kapitel "Geschwindigkeiten, Soll-/Istwertssysteme, Regelung (G2)" > "Regelung" > "Parametersätze des Lagereglers"

## Dynamik

Im Achsbetrieb gelten die dynamischen Grenzwerte der Achse. Z.B.:

- MD32000 \$MA\_MAX\_AX\_VELO[<Achse>] (maximale Achsgeschwindigkeit)
- MD32300 \$MA\_MAX\_AX\_ACCEL[<Achse>] (maximale Achsbeschleunigung)
- MD32431 \$MA\_MAX\_AX\_JERK[<Achse>] (maximaler axialer Ruck bei Bahnbewegung)

### Vorsteuerung

Der für die Achse aktive Vorsteuermodus wird beibehalten.

Eine ausführliche Beschreibung der Funktion "Dynamische Vorsteuerung" findet sich in:

### Literatur

Funktionshandbuch Erweiterungsfunktionen; Kapitel "Kompensationen (K3)" > "Dynamische Vorsteuerung (Schleppfehler-Kompensation)"

**Beispiel: Auflösungsumschaltung bei analogem Steller**

Umschalten in den Achsbetrieb

Programmierung	Kommentar
SPOS=...	
M5	; Reglerfreigabe aus (von PLC) → wird ausgegeben an PLC
M70	; Steller umschalten (von PLC aufgrund von M70) Reglerfreigabe ein (von PLC)
C=...	; NC fährt mit Achsparametersatz

Umschalten in den Spindelbetrieb

Programmierung	Kommentar
C=...	
M71	; → wird ausgegeben an PLC Reglerfreigabe aus (von PLC) Steller umschalten (von PLC) NC-intern wird auf Spindelparametersatz (1-5) umgeschaltet, Reglerfreigabe ein (von PLC)
M3/4/5 oder SPOS=...	; NC fährt mit Spindelparametersatz

**Wechsel in den Spindelbetrieb**

Entsprechend der aktiven Getriebestufe wird der entsprechende Parametersatz 1...5 angewählt.

Die Vorsteuerung wird, außer bei Gewindebohren mit Ausgleichsfutter, eingeschaltet, wenn gilt:

MD32620 \$MA\_FFW\_MODE (Vorsteuerungsart) ≠ 0

Parametersatz	Achsbetrieb	Spindelbetrieb
1	gültig	-
2	-	gültig
3	-	gültig
4	-	gültig
5	-	gültig
6	-	gültig
Spindelbetrieb: Parametersatz entsprechend der Getriebestufe		

## 15.2.6.2 Impliziter Übergang in den Achsbetrieb

### Funktion

Die Steuerung erkennt aufgrund der Programmierabfolge selbständig den Übergang in den Achsbetrieb und erzeugt steuerungsintern den notwendigen M70-Ablauf. Die Schritte werden situationsabhängig behandelt und umfassen maximal:

1. Stillsetzen der Spindel
2. Einschalten der Lageregelung, Behandlung von Vorsteuerung und Parametersatzumschaltung
3. Positionssynchronisation der Satzaufbereitung (interner Vorlaufstopp, wenn notwendig)

Die Funktion ist immer aktiv. Die explizite Programmierung von M70 im Teileprogramm ist damit grundsätzlich nicht notwendig.

### Ablauf

Ablauf des impliziten Übergangs in den Achsbetrieb (keine Programmierung von M70 im Teileprogramm):

- Übergang vom Drehzahlsteuerbetrieb (M3, M4, M5, SPCOF, ...) in den Achsbetrieb:  
Von der Steuerung wird der Übergang intern erkannt und ein Zwischensatz vor dem Satz eingefügt, der Achsbetrieb erfordert. Der erzeugte Satz beinhaltet die M70-Funktionalität. Die Abarbeitungsdauer dieses Satzes entspricht in etwa dem eines programmierten M70-Satzes. Unterschiede können sich bei kurzen Umschaltungen bei stehender Spindel (keine Bremszeit) ergeben, wenn auf die implizite Erzeugung und Ausgabe der Hilfsfunktion M70 an die PLC verzichtet wird (siehe MD35035).
- Übergang vom Positionierbetrieb (M19, SPOS, SPOSA) in den Achsbetrieb:  
Der Übergang wird sofort und ohne die Erzeugung eines Zwischensatzes ausgeführt. Bei entsprechender Projektierung (siehe MD35035) wird die implizit erzeugte Hilfsfunktion M70 mit dem Einwechseln des Satzes, in dem die Spindel Achsbetrieb hat, an die PLC ausgegeben.

### Hilfsfunktionsausgabe an PLC

Der implizite Übergang in den Achsbetrieb kann an die PLC in Form einer Hilfsfunktionsausgabe mitgeteilt werden.

#### Aktivierung / Deaktivierung

Die Aktivierung / Deaktivierung dieser Funktionalität erfolgt über das Maschinendatum:  
MD35035 \$MA\_SPIND\_FUNCTION\_MASK (Spindelfunktionen)

Bit	Wert	Bedeutung
20	0	Keine Hilfsfunktionsausgabe an die PLC bei steuerungsintern erzeugter M70-Funktionalität.
	1	Bei steuerungsintern erzeugter M70-Funktionalität wird implizit die Hilfsfunktion M70 erzeugt und an die PLC ausgegeben. Die Adresserweiterung entspricht der Spindelnummer.

---

#### Hinweis

Eine im Teileprogramm programmierte Hilfsfunktion M70 wird immer an die PLC ausgegeben.

---

#### Eigenschaften

Die implizit erzeugte Hilfsfunktionsausgabe M70 hat die Eigenschaften "Quick" und "Ausgabe während der Bewegung". Diese Eigenschaften sind fest eingestellt und unabhängig von der M70-Projektierung in den Hilfsfunktionsspezifischen Maschinendaten (MD...\_\$M...\_AUXFU...).

M70 wird nur einmal beim Übergang in den Achsbetrieb erzeugt. Bei anschließenden Sätzen, in denen die Spindel als Achse betrieben wird, wird keine weitere M70-Hilfsfunktion erzeugt und ausgegeben. Erst nach dem Verlassen des Achsbetriebs durch z. B. SPOS, M3, M4, M5, SPCOF, etc. und einem erneuten Übergang in den Achsbetrieb wird wieder implizit M70 erzeugt und ausgegeben.

## Randbedingungen

### Synchronaktionen

Bei der Programmierung der Spindel als Achse in Synchronaktionen muss weiterhin applikationsseitig dafür gesorgt werden, dass die Voraussetzungen für den Übergang in den Achsbetrieb bestehen.

Befindet sich die Spindel im Drehzahlsteuerbetrieb, dann muss vor der Programmierung als Achse die Anweisung `M70` oder `SPOS` programmiert werden. Anderenfalls kommt es bei der Achsprogrammierung zur Alarmmeldung.

### FC 18

Wie bei Synchronaktionen muss auch bei FC 18 der Übergang in den Achsbetrieb applikationsseitig vorgenommen werden, z. B. durch eine vorbereitende Positionieranweisung. Anderenfalls wird der FC 18-Aufruf mit einem Fehlerbit im FC 18-Statuswort quittiert.

Bei Übergang in den Achsbetrieb durch Programmierung über FC 18 wird **keine** Hilfsfunktion M70 implizit erzeugt.

## Beispiele

### Beispiel 1:

#### Teilprogramm: Übergang von drehender Spindel in den Achsbetrieb

Projektierung: MD35035 \$MA\_SPIND\_FUNCTION\_MASK, Bit 20 = 1

Programmcode	Kommentar
N05 M3 S1000	
N10 ...	
N15 POS[C]=77	; Vor dem Einwechseln von N15 wird ein M70-Zwischensatz erzeugt, in dem die Spindel still gesetzt und M70 an die PLC ausgegeben wird.
...	

### Beispiel 2:

#### Teilprogramm: Übergang aus dem Positionierbetrieb in den Achsbetrieb

Projektierung: MD35035 \$MA\_SPIND\_FUNCTION\_MASK, Bit 20 = 1

Programmcode	Kommentar
N05 SPOS=0	
N10 ...	
N15 C77	; Ausgabe des impliziten M70 an die PLC, kein Zwischensatz.
...	

**Beispiel 3:****Synchronaktionen: Übergang vom Spindelpositionierbetrieb in den Achsbetrieb**

Projektierung: MD35035 \$MA\_SPIND\_FUNCTION\_MASK, Bit 20 = 1

Programmcode	Kommentar
WHEN COND1==TRUE DO SPOS=180	
WHEN COND2==TRUE DO POS[C]=270	; Ausgabe des impliziten M70 an die PLC.

**Beispiel 4:****Synchronaktionen: Übergang vom Drehzahlsteuerbetrieb in den Achsbetrieb mit M70**

Projektierung: MD35035 \$MA\_SPIND\_FUNCTION\_MASK, Bit 20 = 1

Programmcode	Kommentar
WHEN COND11==TRUE DO M3 S1000	
WHEN COND12==TRUE DO M70	; Ausgabe von M70 an die PLC.
WHEN COND13==TRUE DO POS[C]=270	; Keine Erzeugung eines impliziten M70, da bereits Achsbetrieb vorliegt.

**Beispiel 5:****Synchronaktionen: Unzulässiger Übergang vom Drehzahlsteuerbetrieb in den Achsbetrieb**

Projektierung: MD35035 \$MA\_SPIND\_FUNCTION\_MASK, Bit 20 = 1

Programmcode	Kommentar
WHEN COND21==TRUE DO M3 S1000	
WHEN COND22==TRUE DO POS[C]=270	; Alarm 20141!

## 15.2.7 Spindel-Grundstellung

### Spindel-Grundstellung

Mit dem folgenden Maschinendatum wird eine Spindelbetriebsart als Grundstellung festgelegt:

MD35020 \$MA\_SPIND\_DEFAULT\_MODE

Wert	Spindelgrundstellung
0	Drehzahlsteuerbetrieb, Lageregelung abgewählt
1	Drehzahlsteuerbetrieb, Lageregelung eingeschaltet
2	Positionierbetrieb
3	Achsbetrieb

### Wirkungszeitpunkt der Spindel-Grundstellung

Der Wirkungszeitpunkt für die Spindel-Grundstellung wird eingestellt im Maschinendatum:

MD35030 \$MA\_SPIND\_DEFAULT\_ACT\_MASK

Wert	Wirkungszeitpunkt
0	POWER ON
1	POWER ON und Programmstart
2	POWER ON und RESET (M2 / M30)

## 15.3 Referenzieren / Synchronisieren

### Warum Synchronisieren?

Damit nach dem Einschalten die Steuerung die Position der Spindel exakt kennt, muss die Steuerung mit dem Lagemesssystem der Spindel synchronisiert werden.

Erst mit einer synchronisierten Spindel sind folgende Funktionen möglich:

- Gewindeschneiden
- Gewindebohren ohne Ausgleichfutter
- Achsprogrammierungen

Weitere Erläuterungen zum Synchronisieren der Spindel siehe Kapitel "R1: Referenzieren (Seite 1319)".

### Warum Referenzieren?

Damit nach dem Einschalten die Steuerung den Maschinennullpunkt exakt kennt, muss die Steuerung mit dem Lagemesssystem der Rundachse synchronisiert werden. Diesen Vorgang nennt man Referenzieren. Den zeitlichen Ablauf, der eine Achse referenziert, nennt man Referenzpunktfahren.

Erst eine referenzierte Achse kann eine programmierte Position auf der Maschine genau anfahren.

Weitere Erläuterungen zum Referenzieren der Rundachse siehe Kapitel "R1: Referenzieren (Seite 1319)".

### Montageorte der Lagemesssysteme

Die Lagemesssysteme können wie folgt montiert sein:

- direkt am Motor plus Bero-Schalter an der Spindel als Nullmarkengeber
- am Motor über ein Messgetriebe plus Bero-Schalter an der Spindel als Nullmarkengeber
- direkt an der Spindel
- an der Spindel über ein Messgetriebe plus Bero-Schalter an der Spindel als Nullmarkengeber (nur bei Übersetzungen ungleich 1:1)

Sind zwei Lagemesssysteme vorhanden, können beide Lagemesssysteme am gleichen Montageort oder an getrennten Montageorten angebracht sein.



## Ablauf Synchronisieren

Nach dem Einschalten der Steuerung kann die Spindel wie folgt synchronisiert werden:

- Die Spindel wird mit einer Spindeldrehzahl ( $s$ -Wert) und einer Spindeldrehrichtung ( $M_3$  oder  $M_4$ ) gestartet und synchronisiert sich mit der nächsten Nullmarke des Lagemesssystems oder mit dem nächsten Bero-Signal.
- Die Spindel soll mit  $SPOS$ ,  $M19$  oder  $SPOSA$  aus dem Stillstand positioniert werden. Die Spindel synchronisiert sich mit der nächsten Nullmarke des Lagemesssystems oder mit dem nächsten Bero-Signal. Dann erfolgt die Positionierung auf die programmierte Position.
- Die Spindel kann mit  $SPOS$ ,  $M19$  oder  $SPOSA$  aus der Bewegung (nach  $M_3$  oder  $M_4$ ) synchronisiert werden.

Dabei liegt folgendes Verhalten vor:

- Mit  $SPOS=<Pos>$ ,  $SPOS=DC(<Pos>)$  und  $SPOS=AC(<Pos>)$  wird die Bewegungsrichtung beibehalten und die Position angefahren.
- Mit  $SPOS=ACN(<Pos>)$  bzw.  $SPOS=ACP(<Pos>)$  wird die Position immer mit negativer bzw. positiver Bewegungsrichtung angefahren. Gegebenenfalls wird vor dem Positionieren die Bewegungsrichtung invertiert.
- Überfahren der Nullmarke in der Betriebsart JOG mittels den Richtungstasten im Drehzahlsteuerbetrieb.

---

### Hinweis

Es ist gleichwertig, ob der Anstoß zum Synchronisieren aus dem Teileprogramm, FC 18 oder Synchronaktionen kommt.

---

### Hinweis

Beim Synchronisieren der Spindel wirken alle vier möglichen Referenzpunktwerte des Referenzpunkts je nach angewähltem Messsystem. Die Verschiebung zum Messsystem wirkt gleichermaßen.

Zu beachten sind folgende Maschinendaten:

- MD34080 \$MA\_REFP\_MOVE\_DIST  
(Referenzpunktabstand/Zielpunkt bei abstandskodiertem System)
- MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR  
(Referenzpunktverschiebung/Absolutverschiebung abstandskodiert)
- MD34100 \$MA\_REFP\_SET\_POS  
(Referenzpunktwert, bei abstandskodiertem System ohne Bedeutung)

Wird eine nichtreferenzierte Spindel mit  $SPOS=IC(...)$  und einem Weg  $< 360$  Grad positioniert, dann wird möglicherweise die Nullmarke nicht überfahren und die Spindelposition ist weiterhin nicht mit der Nullmarke synchronisiert. Dies kann vorkommen:

- nach POWER ON
  - durch Setzen der axialen NC/PLC-Nahtstellensignale:  
DB31, ... DBX17.5 (Spindel neu synchronisieren beim Positionieren 2)  
DB31, ... DBX17.4 (Spindel neu synchronisieren beim Positionieren 1)
-

### Besonderheiten beim Synchronisieren mittels BERO

Die durch die BERO-Signalverzögerung hervorgerufene Positionsverfälschung kann NC-intern durch Eintrag einer Signallaufzeitkompensation korrigiert werden.

Die Einstellung der Signallaufzeitkompensation erfolgt über die Maschinendaten:

- MD31122 \$MA\_BERO\_DELAY\_TIME\_PLUS  
(BERO-Verzögerungszeit für positive Bewegungsrichtung)
- MD31123 \$MA\_BERO\_DELAY\_TIME\_MINUS  
(BERO-Verzögerungszeit für negative Bewegungsrichtung)

Die Wirkung ist abhängig von der Einstellung im Maschinendatum:

MD34200 \$MA\_ENC\_REFP\_MODE (Referenzier-Modus)

- MD34200 = 7      Mit der Einstellung MD34200 \$MA\_ENC\_REFP\_MODE = 7 wird die Positionssynchronisation nur bei einer Geschwindigkeit/Drehzahl vorgenommen, die fest vorgegeben ist im Maschinendatum:  
MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER  
(Abschaltgeschwindigkeit)  
Die Nullmarke wird nicht automatisch gesucht, sondern muss explizit angefordert werden mit der 0-1-Flanke des NC/PLC-Nahtstellensignals:  
DB31, ... DBX16.4/5 (Spindel neu synchronisieren 1/2\*)  
\*) 1/2 steht für das angewählte Messsystem  
Die in MD34040 eingestellte Geschwindigkeit wirkt auch beim Referieren in der Betriebsart JOG-REF und durch das Teileprogramm mit G74.
- MD34200 = 2      Mit der Einstellung MD34200 \$MA\_ENC\_REFP\_MODE = 2 wird die Positionssynchronisation ohne Vorgabe einer bestimmten Geschwindigkeit/Drehzahl vorgenommen.

---

#### Hinweis

Die Signallaufzeiten sind im Auslieferungszustand so vorbesetzt, dass in der Regel keine inhaltliche Änderung notwendig ist.

---

### Ablauf Referenzieren

Soll direkt nach dem Steuerungshochlauf die Spindel im Achsbetrieb programmiert werden, muss sichergestellt sein, dass die Achse referenziert ist.

Nach dem Einschalten der Steuerung kann die Spindel referenziert werden (Voraussetzung ist eine Nullmarke pro Umdrehung).

Zum Ablauf des Referenzierens siehe Kapitel "R1: Referenzieren (Seite 1319)".

Mit der Synchronisation der Spindel (siehe Abschnitt "Ablauf Synchronisieren") wird auch die Rundachse gleichzeitig referenziert, wenn das Lagemesssystem der Spindel auch für die Rundachse verwendet wird.

## Lagemesssysteme Spindel

Die Spindel kann vom Spindelbetrieb in den Achsbetrieb (Rundachse) geschaltet werden, wenn für Spindel- und Achsbetrieb ein Motor verwendet wird.

Die Spindel (Spindelbetrieb und Achsbetrieb) kann mit einem oder mit zwei Lagemesssystemen ausgerüstet sein. Bei zwei Lagemesssystemen kann ein Lagemesssystem der Spindel und ein Lagemesssystem der Rundachse zugeordnet sein, oder es können zwei Lagemesssysteme der Spindel zugeordnet sein. Sind zwei Lagemesssysteme vorhanden, werden beide von der Steuerung aktualisiert, aber es kann nur ein Lagemesssystem aktiv sein.

Ausgewählt wird das aktive Lagemesssystem mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX1.5 (Lagemesssystem 1)

bzw.

DB31, ... DBX1.6 (Lagemesssystem 2)

Das aktive Lagemesssystem wird für folgende Funktionen benötigt:

- Lageregelung der Spindel ( $SPCON$ )
- Spindelpositionierung ( $SPPOS$ ,  $M19$  und  $SPOSA$ )
- Gewindegewindeschneiden ( $G33$ ,  $G34$ ,  $G35$ )
- Gewindebohren ohne Ausgleichsfutter ( $G331$ ,  $G332$ )
- Umdrehungsvorschub ( $G95$ )
- Konstante Schnittgeschwindigkeit ( $G96$ ,  $G961$ ,  $G97$ ,  $G971$ )
- Spindelstdrehzahlanzeige
- Achsbetrieb
- Synchronspindelsollwertkopplung

## Lagemesssystem der Spindel neu synchronisieren

In folgenden Fällen muss das Lagemesssystem der Spindel neu synchronisiert werden:

- Der Lagemessgeber ist am Motor, ein Bero ist an der Spindel montiert und es erfolgt ein Getriebestufenwechsel. Die Synchronisation wird intern angestoßen, wenn sich die Spindel in der neuen Getriebestufe dreht (siehe Ablauf Synchronisieren).
- Die Maschine hat eine Umschaltung zwischen vertikaler und horizontaler Spindel. Dabei werden zwei verschiedene Lagemessgeber (einer für die vertikale Spindel und einer für die horizontale Spindel) aber nur ein Istwerteingang an der Steuerung verwendet. Wird zwischen vertikaler und horizontaler Spindel umgeschaltet, muss neu synchronisiert werden.

Diese Synchronisation wird angestoßen mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX16.4 (Spindel neu synchronisieren 1)

oder

DB31, ... DBX16.5 (Spindel neu synchronisieren 2)

Dazu muss sich die Spindel im Steuerbetrieb befinden.

### Positions-Restauration bei POWER ON

Bei Spindeln mit inkrementellen Lagemesssystemen besteht die Möglichkeit, Istwerte über POWER OFF hinweg zu puffern und nach POWER ON die letzte vor dem Ausschalten gepufferte Position zu restaurieren, damit positionsabhängige Funktionen wie z. B. Transformationen wiederhergestellt werden können (siehe Kapitel "Auto-Hotspot"). Eine Anwendung ist z. B. der Werkzeugrückzug nach POWER OFF bei einer Bearbeitung mit Werkzeugorientierung (siehe Kapitel "Werkzeugrückzug nach POWER ON mit Orientierungstransformation (Seite 663)").

Der Zustand des Lagemesssystems nach Positions-Restauration wird durch folgende NC/PLC-Nahtstellensignale angezeigt:

DB31, ... DBX71.4 ("Restauriert 1") für Lagemesssystem 1

DB31, ... DBX71.5 ("Restauriert 2") für Lagemesssystem 2

Nach dem Freifahren des Werkzeugs in der Betriebsart JOG müssen Achsen, deren Positionen restauriert wurden, referiert werden. Dadurch werden die Signale DB31, ... DBX71.4/5 ("Restauriert 1/2") gelöscht und die Signale DB31, ... DBX60.4/5 ("Referiert/Synchronisiert 1/2") gesetzt.

---

#### Hinweis

Wird das Maschinendatum MD20700 \$MC\_REFP\_NC\_START\_LOCK auf den Wert "2" gesetzt, so ist ein NC-Start auch mit "restaurierten" Achspositionen möglich (in der Betriebsart MDA oder beim Überspeichern).

---

## 15.4 Projektierbare Getriebeanpassungen

### 15.4.1 Getriebestufen für Spindeln und Getriebestufenwechsel

#### Warum Getriebestufen?

Getriebestufen bei Spindeln dienen dazu, die Motordrehzahl zu untersetzen und so bei kleinen Spindeldrehzahlen ein hohes Drehmoment zu erzeugen, oder zu übersetzen, um eine hohe Drehzahl zu erhalten.

#### Getriebestufenanzahl

Für jede Spindel sind 5 Getriebestufen projektierbar.

Die Anzahl der verwendeten Getriebestufen wird festgelegt mit dem Maschinendatum:

MD35090 \$MA\_NUM\_GEAR\_STEPS

#### Parametrierung der Getriebestufen

Die Getriebestufen 1 bis 5 sind parametrierbar über folgende Maschinendaten:

Maschinendatum	Bedeutung
MD35012 \$MA_GEAR_STEP_CHANGE_POSITION[<n>]	Getriebestufenwechsel-Position
MD35110 \$MA_GEAR_STEP_MAX_VELO[<n>]	Maximaldrehzahl für den automatischen Getriebestufenwechsel
MD35120 \$MA_GEAR_STEP_MIN_VELO[<n>]	Minimaldrehzahl für den automatischen Getriebestufenwechsel
MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]	Maximaldrehzahl der Getriebestufe
MD35135 \$MA_GEAR_STEP_PC_MAX_VELO_LIMIT[<n>]	Maximaldrehzahl der Getriebestufe bei Lageregelung
MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT[<n>]	Minimaldrehzahl der Getriebestufe
MD35200 \$MA_GEAR_STEP_SPEEDCTRL_ACCEL[<n>]	Beschleunigung im Drehzahlsteuerbetrieb
MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL[<n>]	Beschleunigung im Lageregelbetrieb
MD35300 \$MA_SPIND_POSCTRL_VELO[<n>]	Lageregeleinschaltdrehzahl
MD35310 \$MA_SPIND_POSIT_DELAY_TIME[<n>]	Positionierverzögerungszeit
MD35550 \$MA_DRILL_VELO_LIMIT[<n>]	Maximaldrehzahl für das Gewindebohren ohne Ausgleichsfutter

### Art des Getriebestufenwechsels

Der Art des Getriebestufenwechsels wird eingestellt im Maschinendatum:

MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE

Bit	Wert	Bedeutung
0	0	Der Spindelmotor ist direkt (1:1) oder mit einer nicht veränderbaren Übersetzung an die Spindel angebaut (Grundstellung). Es wirken die Maschinendaten der ersten Getriebestufe.
	1	Spindelmotor mit max. 5 Getriebestufen. Der Getriebestufenwechsel erfolgt: <ul style="list-style-type: none"> <li>• Im Pendelbetrieb</li> <li>• Auf unbestimmter Wechselposition</li> </ul>
1	0	Bedeutung wie bei Bit 0 = 0.
	1	Bedeutung wie bei Bit 0 = 1, jedoch erfolgt der Getriebestufenwechsel auf projektierter Spindelposition. Die Wechselposition wird eingestellt im Maschinendatum: MD35012 \$MA_GEAR_STEP_CHANGE_POSITION Die Position wird in der aktuellen Getriebestufe vor dem Getriebestufenwechsel angefahren. Wenn Bit 1 gesetzt ist, dann wird Bit 0 nicht beachtet!
3	1	Der Getriebestufenwechseldialog zwischen NCK und PLC wird simuliert.
5	1	Beim Gewindebohren mit G331/G332 wird der zweite Getriebestufen-Datensatz verwendet (siehe nachfolgenden Absatz "Zweiter Getriebestufen-Datensatz"). Das Bit muss für die beim Gewindebohren verwendete Masterspindel gesetzt werden.

### Voraussetzung für einen Getriebestufenwechsel

Prinzipiell wird ein Getriebestufenwechsel nur dann ausgeführt, wenn die angeforderte Getriebestufe ungleich der aktiven Getriebestufe ist.

### Parametersatzanwahl beim Getriebestufenwechsel

Mit der Getriebestufe wird auch der Servo-Parametersatz umgeschaltet, wenn:

MD35590 \$MA\_PARAMSET\_CHANGE\_ENABLE = 0 oder 1

Für weitere Informationen siehe Kapitel "Parametersatzanwahl beim Getriebestufenwechsel (Seite 1437)".

## Getriebestufenwechsel anfordern

Ein Getriebestufenwechsel kann angefordert werden:

- im Teileprogramm durch:
  - M40 S . . .  
Automatische Getriebestufenauswahl zur programmierten Drehzahl s . . .
  - M41 ... M45  
Direkte Anwahl der Getriebestufe 1 ... 5
  - M70  
Bei MD35014 \$MA\_GEAR\_STEP\_USED\_IN\_AXISMODE = 1 ... 5  
(siehe "Projektierbare Getriebestufe bei M70 (Seite 1456)")
  - G331 S . . .  
Bei MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE, Bit 5 = 1
- in Synchronaktionen durch:
  - DO M40 S . . .  
Automatische Getriebestufenauswahl zur programmierten Drehzahl s . . .
  - DO M41 . . . M45  
Direkte Anwahl der Getriebestufe 1 ... 5
  - DO M70  
Bei MD35014 \$MA\_GEAR\_STEP\_USED\_IN\_AXISMODE = 1 ... 5
- durch die PLC unter Verwendung des Funktionsbausteins FC18
- im Reset-Zustand durch Beschreiben der NC/PLC-Nahtstelle:  
DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C)  
Insbesondere nach einem POWER ON kann der NC die mechanisch aktive Getriebestufe mitgeteilt werden.

---

### Hinweis

Wenn der Spindelmotor direkt (1:1) oder mit einer nicht veränderbaren Übersetzung an die Spindel angebaut ist (MD35010 = 0), dann haben die Hilfsfunktionen M40 und M41 ... M45 für diese Spindel keine Bedeutung.

---

**Getriebestufenauswahl**

Die Getriebestufenauswahl zwischen zwei Getriebestufen bei Vorgabe einer maximalen Spindeldrehzahl wird in folgendem Beispiel gezeigt:

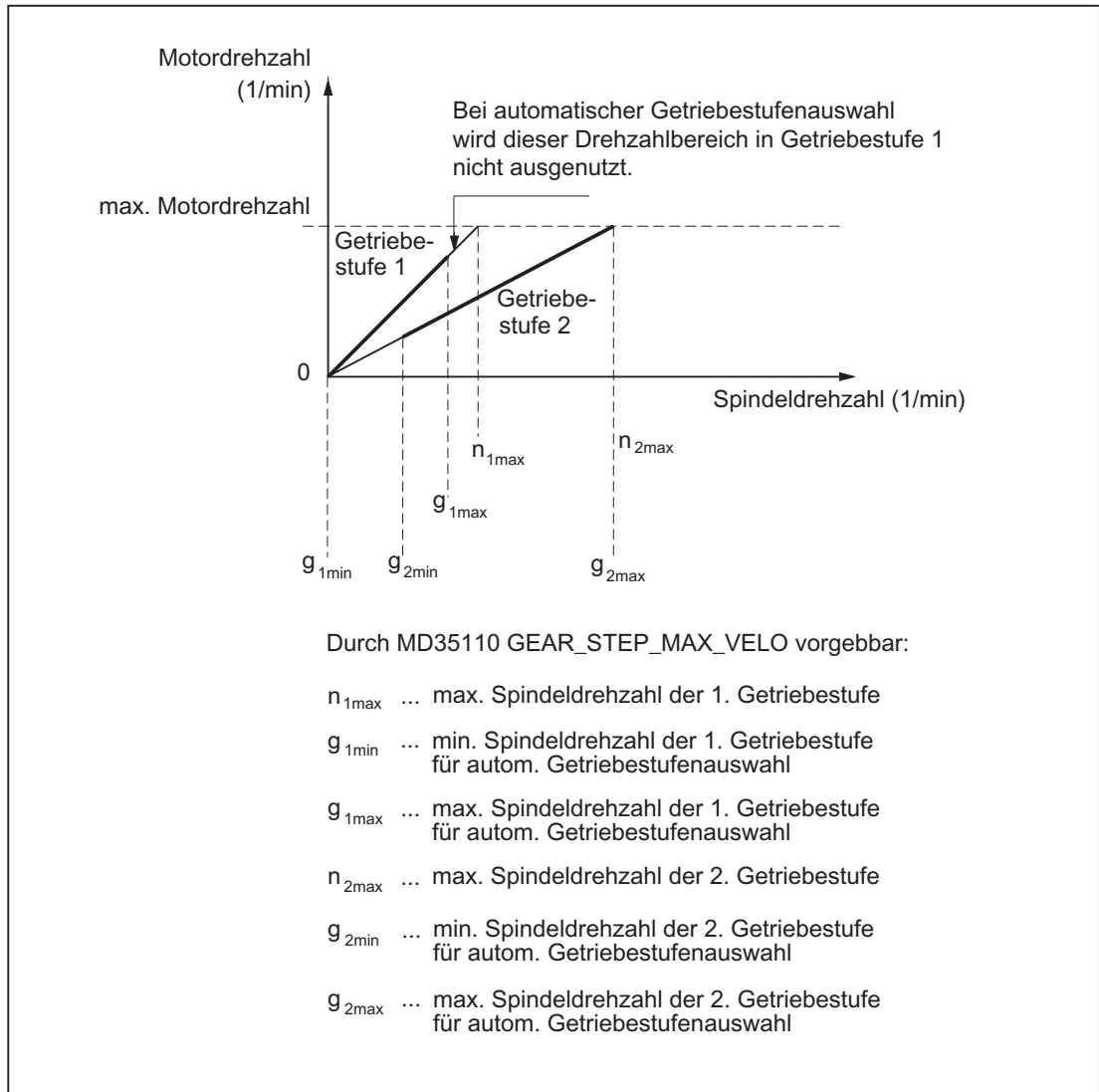


Bild 15-3 Getriebestufenwechsel mit Getriebestufenauswahl bei zwei Getriebestufen



## Ablauf des Getriebestufenwechsels

Ist die neue Getriebestufe vorgewählt, dann findet folgender Ablauf statt:

### 1. Umschaltvorgang

Die beiden folgenden NC/PLC-Nahtstellensignale werden gesetzt:

DB31, ... DBX82.0-82.2 (Sollgetriebestufe A bis C)

DB31, ... DBX82.3 (Getriebe umschalten)

Je nachdem, zu welchem Zeitpunkt das NC/PLC-Nahtstellensignal:

DB31, ... DBX18.5 (Pendeldrehzahl)

gesetzt wird, bremst die Spindel mit der Beschleunigung für Pendeln oder mit der Beschleunigung für Drehzahlregelbetrieb / Lageregelbetrieb auf Stillstand ab.

Spätestens mit dem Stillstand der Spindel:

DB31, ... DBX61.4 (Achse/Spindel steht)

kann mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX18.5 (Pendeldrehzahl)

das Pendeln eingeschaltet werden.

Prinzipiell kann auch ohne Pendelvorgang die neue Getriebestufe eingelegt werden.

Ist die neue Getriebestufe eingelegt, dann werden vom PLC-Programm die beiden folgenden NC/PLC-Nahtstellensignale gesetzt:

DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C)

DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

### 2. Getriebestufenwechsel-Ende

Der Getriebestufenwechsel gilt als beendet (Spindelbetriebsart "Pendelbetrieb" wird abgewählt), wenn das folgende NC/PLC-Nahtstellensignal gesetzt wird:

DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

Im Motorstillstand wird auf den Servo- und Interpolationsparametersatz der neuen Istgetriebestufe umgeschaltet.

Mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

wird der NC mitgeteilt, dass die neue Getriebestufe gültig ist, und der Pendelbetrieb beendet werden kann.

Das NC/PLC-Nahtstellensignal:

DB31, ... DBX82.3 (Getriebe umschalten)

wird durch den NCK rückgesetzt,

worauf das PLC-Programm das NC/PLC-Nahtstellensignal:

DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

rücksetzen muss.

Dabei ist es unwesentlich, ob das NC/PLC-Nahtstellensignal:  
DB31, ... DBX18.5 (Pendeldrehzahl)  
noch gesetzt ist.

Zur Anwahl des Parametersatzes ist die gemeldete Istgetriebestufe relevant,  
die der Sollgetriebestufe entsprechen sollte.

Ist dies nicht der Fall, wird bei:  
MD11410 \$MN\_SUPPRESS\_ALARM\_MASK, Bit 3 = 0  
der Alarm 22010 gemeldet.

Nach der Quittierung des Getriebestufenwechsels durch die PLC  
mit dem NC/PLC-Nahtstellensignal:  
DB31, ... DBX16.3 (Getriebe ist umgeschaltet)  
befindet sich die Spindel im Drehzahlsteuerbetrieb (DB31, ... DBX84.7).

Weitere Hinweise zum Signalaustausch zwischen PLC und NC siehe Kapitel "A2: Diverse  
NC/PLC-Nahtstellensignale und Funktionen (Seite 33)".

### Zweiter Getriebestufen-Datensatz

Der automatische Getriebestufenwechsel M40 kann durch einen zweiten projektierbaren  
Getriebestufen-Datensatz erweitert werden.

Der zweite Getriebestufen-Datensatz wird **ausschließlich** im Zusammenhang mit  
Gewindebohren ohne Ausgleichsfutter (G331, G332) verwendet, um eine effektive  
Anpassung von Spindeldrehzahl und Motormoment zu erreichen.

Die Aktivierung erfolgt, indem für die Masterspindel folgendes Bit gesetzt wird:

MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE, Bit 5 = 1

Die Anzahl der verwendeten Getriebestufen des zweiten Getriebestufen-Datensatzes wird  
festgelegt mit dem Maschinendatum:

MD35092 \$MA\_NUM\_GEAR\_STEPS2

Der zweite Getriebestufen-Datensatz wird deaktiviert, wenn:

MD35092 \$MA\_NUM\_GEAR\_STEPS2 = 0 (Grundstellung)

Bei aktivem M40 erfolgt dann die Auswahl der Getriebestufe durch den ersten  
Getriebestufen-Datensatz.

---

#### Hinweis

Die Anzahl der Getriebestufen im zweiten Datensatz können sich vom ersten unterscheiden.  
Wird für eine programmierte Drehzahl bei M40 keine passende Getriebestufe gefunden,  
dann wird auch kein Getriebestufenwechsel ausgeführt (Ausnahmen siehe "M40:  
Automatische Getriebestufenauswahl bei Drehzahlen außerhalb projektierter  
Schaltschwellen (Seite 1489)").

Weitere Informationen zu einem typischen Programmablauf beim Gewindeschneiden ohne  
Ausgleichsfutter G331/G332 siehe:

#### Literatur:

Programmierhandbuch Grundlagen; Wegbefehle

---

Die Getriebestufen 1 bis 5 des zweiten Getriebestufen-Datensatzes sind parametrierbar über folgende Maschinendaten:

Maschinendatum	Bedeutung
MD35112 \$MA_GEAR_STEP_MAX_VELO2[n]	Maximaldrehzahl für den automatischen Getriebestufenwechsel
MD35122 \$MA_GEAR_STEP_MIN_VELO2[n]	Minimaldrehzahl für den automatischen Getriebestufenwechsel
MD35212 \$MA_GEAR_STEP_POSCTRL_ACCEL2[n]	Beschleunigung im Lageregelbetrieb

#### Hinweis

Die Anzahl der die mechanischen Gegebenheiten betreffenden Servo-Parametersätze bleiben unverändert. Weiterhin können fünf mechanische Getriebestufen für den Spindel- und eine für den Achsbetrieb projiziert werden.

Die Drehzahlbegrenzungen werden unabhängig von den verschiedenen Schaltschwellen nur einmal pro Getriebestufe mit folgenden Maschinendaten projiziert:

Maschinendatum	Bedeutung
MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[n]	Maximaldrehzahl der Getriebestufe
MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT[n]	Minimaldrehzahl der Getriebestufe

Für das Gewindebohren ohne Ausgleichsfutter (G331, G332) kann die Drehzahl zusätzlich auf den linearen Beschleunigungsbereich des Motors begrenzt werden. Dazu wird im folgenden Maschinendatum die maximale Drehzahl des linearen Motorkennlinienbereichs getriebestufenabhängig angegeben:

MD35550 \$MA\_DRILL\_VELO\_LIMIT[n]

## Vorgabe einer Getriebestufe im Teileprogramm

### Automatische Auswahl bei aktiven M40

Die Getriebestufe wird durch die Steuerung automatisch ausgewählt. Dabei wird kontrolliert, in welcher Getriebestufe die programmierte Spindeldrehzahl (s. . .) möglich ist. Wenn daraus eine Getriebestufe resultiert, die ungleich der aktuellen (Ist-)Getriebestufe ist, dann werden folgende NC/PLC-Nahtstellensignale gesetzt:

DB31, ... DBX82.3 (Getriebe umschalten)

DB31, ... DBX82.0-82.2 (Sollgetriebestufe A bis C)

Bei der Ermittlung der passenden Getriebestufe wird nur dann ein Getriebestufenwechsel angefordert, wenn die neue Drehzahl nicht im zulässigen Drehzahlbereich der aktiven Getriebestufe liegt.

Die Drehzahl wird gegebenenfalls auf die Maximaldrehzahl der aktuellen Getriebestufe begrenzt bzw. auf die Minimaldrehzahl der aktuellen Getriebestufe angehoben, und es wird das entsprechende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.1 (Soll-Drehzahl begrenzt)

DB31, ... DBX83.2 (Soll-Drehzahl erhöht)

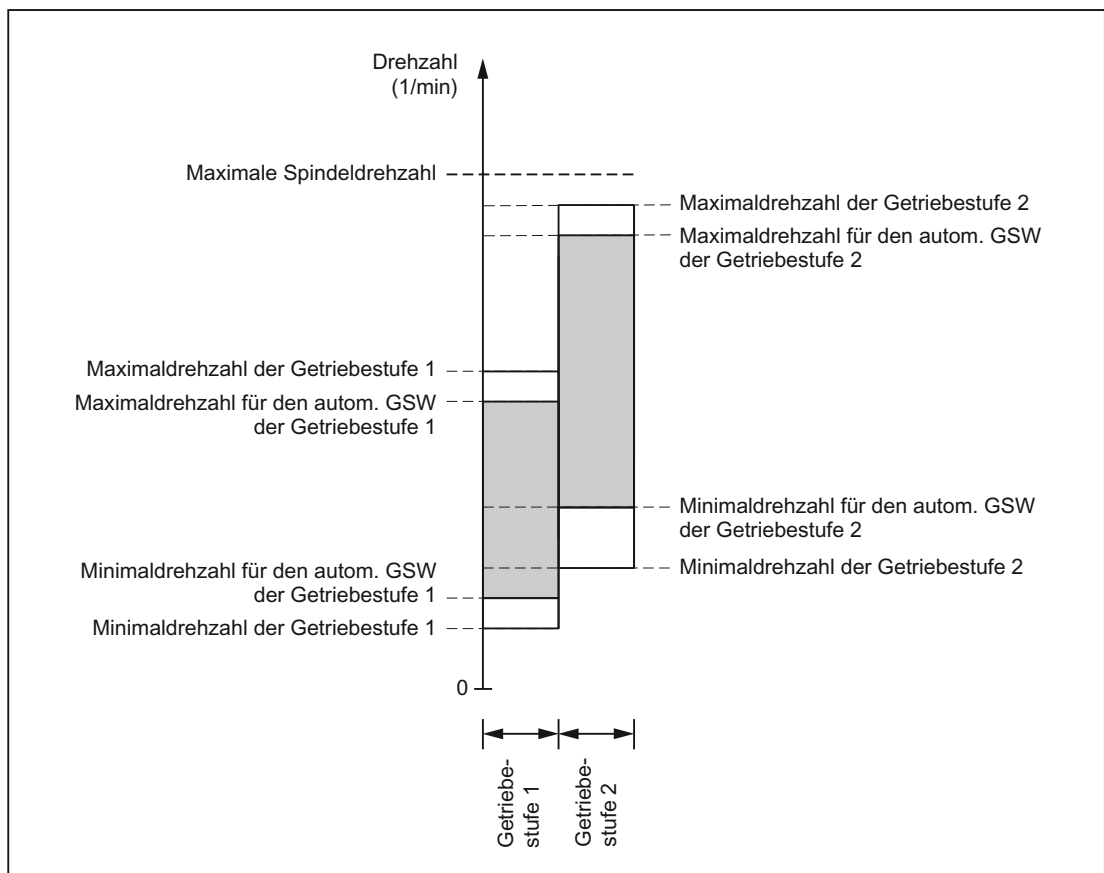


Bild 15-4 Beispiel für zwei Getriebestufen mit überlappenden Drehzahlbereichen bei automatischer Getriebestufenauswahl (M40)

**Hinweis**

Bei M40 muss sich die Spindel zur automatischen Getriebestufenauswahl bei einem s-Wort im Steuerbetrieb befinden. Andernfalls wird der Getriebestufenwechsel abgewiesen und folgender Alarm gesetzt:

Alarm 22000 "Kein Getriebestufenwechsel möglich"

**Hinweis**

Ein aktives Vorsatzgetriebe wird bei der Auswahl für den automatischen Getriebestufenwechsel nicht berücksichtigt.

### Getriebestufe fest vorgeben mit M41 bis M45

Die Getriebestufe kann im Teileprogramm mit M41 bis M45 fest vorgegeben werden.

Wenn durch M41 bis M45 eine Getriebestufe vorgegeben wird, die ungleich der aktuellen (Ist-)Getriebestufe ist, dann werden folgende NC/PLC-Nahtstellensignale gesetzt:

DB31, ... DBX82.3 (Getriebe umschalten)

DB31, ... DBX82.0-82.2 (Sollgetriebestufe A bis C)

Die programmierte Spindeldrehzahl (s. . .) bezieht sich dann auf diese fest vorgegebene Getriebestufe:

- Wenn eine Spindeldrehzahl programmiert wird, die über der Maximaldrehzahl der fest vorgegebenen Getriebestufe (MD35130 \$MA\_GEAR\_STEP\_MAX\_VELO\_LIMIT) liegt, dann wird auf diese Maximaldrehzahl begrenzt und das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.1 (Soll-Drehzahl begrenzt)

- Wenn eine Spindeldrehzahl programmiert wird, die unter der Minimaldrehzahl der fest vorgegebenen Getriebestufe (MD35140 \$MA\_GEAR\_STEP\_MIN\_VELO\_LIMIT) liegt, dann wird auf diese Minimaldrehzahl angehoben und das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.2 (Soll-Drehzahl erhöht)

### Satzwechsel

Bei der Programmierung des Getriebestufenwechsels im Teileprogramm bleibt der Getriebestufenwechselsatz solange aktiv, bis der Getriebestufenwechsel von PLC beendet wurde.

Dies entspricht der Wirkung, als wäre folgendes NC/PLC-Nahtstellensignal gesetzt:

DB21, ... DBX6.1 (Einlesesperre)

### Vorgabe einer Getriebestufe von PLC mit FC18

Der Getriebestufenwechsel ist durch Vorgabe mit dem Funktionsbaustein FC18 während eines Teileprogramms, im Reset-Zustand oder auch bei allen Betriebsarten möglich.

Bei der Drehzahl- und Drehrichtungsvorgabe mit FC18 kann die NC aufgefordert werden, die zur Drehzahl passende Getriebestufe auszuwählen. Dies entspricht einem automatischen Getriebestufenwechsel bei M40.

Die Getriebestufe wird nicht umgeschaltet, wenn:

- die Spindel durch FC18 positioniert wird.
- die Spindel im Achsbetrieb verfahren wird.

Für weitere Hinweise zum Funktionsbaustein FC18 siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)".

### Vorgabe einer Getriebestufe in Synchronaktionen

Der Getriebestufenwechsel kann von Synchronaktionen angefordert werden durch:

- DO M40 S...  
Automatische Getriebestufenauswahl zur programmierten Drehzahl s...
- DO M41... M45  
Direkte Anwahl der Getriebestufe 1 ... 5
- DO M70  
Bei MD35014 \$MA\_GEAR\_STEP\_USED\_IN\_AXISMODE = 1 ... 5  
(siehe "Projektierbare Getriebestufe bei M70 (Seite 1456)")

Die Getriebestufe wird **nicht** umgeschaltet, wenn:

- die Spindel durch Synchronaktionen positioniert wird.
- die Spindel im Achsbetrieb verfahren wird.

---

#### Hinweis

Weitere Einzelheiten siehe Absatz "Vorgabe einer Getriebestufe im Teileprogramm".

Ausnahme:

Durch die Vorgabe einer Getriebestufe in Synchronaktionen wird der Satzwechsel nicht beeinflusst.

---

### Getriebestufe manuell vorgeben

Außerhalb eines laufenden Teileprogramms kann die Getriebestufe auch ohne Aufforderung durch die NC an der Maschine gewechselt werden. Dies ist z. B. der Fall, wenn eine Getriebestufe direkt von Hand ausgewechselt wird.

Zur Auswahl der passenden Parametersätze muss der NC die aktuelle Getriebestufe mitgeteilt werden. Dazu muss sich die Steuerung oder das Teileprogramm im Reset-Zustand befinden.

#### Randbedingungen

Die Übernahme der Getriebestufe in die NC erfolgt mit Änderung der NC/PLC-Nahtstellensignale:  
DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C)

Diese drei Bits müssen während des Betriebs stabil anstehen.

Die vollzogene Übernahme wird mit den NC/PLC-Nahtstellensignalen:  
DB31, ... DBX82.0-82.2 (Sollgetriebestufe A bis C)  
zum PLC quittiert.

Das NC/PLC-Nahtstellensignal:  
DB31, ... DBX16.3 (Getriebe ist umgeschaltet)  
muss nicht gesetzt werden.

Ist zu dem Zeitpunkt der Vorgabe einer neuen Getriebestufe von PLC mit  
DB31, ... DBX16.0-16.2 die Lageregelung aktiv, dann wird diese für die Dauer des  
Umschaltvorgangs ausgeschaltet.

### NC-Stop während Getriebestufenwechsel

Ein Spindelhalt durch das NC/PLC-Nahtstellensignal:  
DB21, ... DBX7.4 (NC-Stop)  
ist nicht möglich, solange:

- die Spindel im Pendelbetrieb für Getriebestufenwechsel noch nicht vorliegt.
- das NC/PLC-Nahtstellensignal:  
DB31, ... DBX16.3 (Getriebe ist umgeschaltet)  
noch nicht vorliegt.

---

#### Hinweis

Abbruchmöglichkeiten:

DB31, ... DBX2.2 (Restweglöschen/Spindelreset)  
oder  
DB31, ... DBX16.3 (Getriebe ist umgeschaltet)  
mit entsprechender Rückmeldung der Istgetriebestufe:  
DB31, ... DBX16.0-16.2 (Istgetriebestufe)

---

### Spindelverhalten nach einem Getriebestufenwechsel

Wie sich die Spindel nach beendetem Getriebestufenwechsel verhält, hängt von folgenden Anfangsbedingungen ab:

- Befand sich die Spindel vor dem Getriebestufenwechsel im Stopp-Zustand (M5, FC18: "Stopp Spindel drehen"), im Positionier- oder Achsbetrieb, dann wird nach Getriebestufenwechsel-Ende M5 (Spindel-Stopp) aktiv.
- War vor dem Getriebestufenwechsel eine Drehrichtung (M3, M4, FC18: "Start Spindel drehen") programmiert, dann werden nach dem Getriebestufenwechsel die letzte Drehrichtung und Drehzahl wieder aktiv. Die Spindel dreht in der neuen Getriebestufe auf die letzte programmierte Spindeldrehzahl (s. . . ) hoch.
- War vor dem Getriebestufenwechsel Lageregelung aktiv (SPCON), dann wird diese nach dem Getriebestufenwechsel wieder aktiviert.

Der nächste Satz im Teileprogramm kann zur Ausführung kommen.

## Besonderheiten

Beim Getriebestufenwechsel sind folgende Punkte zu beachten:

- Der Getriebestufenwechsel wird durch die Anwahl des NC/PLC-Nahtstellensignals:  
DB31, ... DBX20.1 (Hochlaufumschaltung U/F Betrieb) nicht beendet.  
Es wird Sollwert 0 ausgegeben.

Die Quittierung des Getriebestufenwechsels erfolgt wie üblich durch das NC/PLC-Nahtstellensignal:

DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

- Das Signal "Hochlaufgeber Schnellstopp" muss von der PLC zurückgesetzt werden, bevor der Getriebestufenwechsel von der PLC beendet wird.
- Der Ablauf des Getriebestufenwechsels wird bei NC-Reset ohne Alarmausgabe beendet.

Die in den NC/PLC-Nahtstellensignalen:

DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C) gemeldete Getriebestufe wird von der NC übernommen.

## Stern- / Dreieck-Umschaltung mit FC17

Digitale Hauptspindelantriebe können mittels FC17 auch bei laufender Spindel von Stern nach Dreieck in beiden Richtungen umgeschaltet werden. Dieser automatische Umschaltvorgang wird durch eine definierte Umschaltlogik im FC17 gesteuert, die eine vom Anwender parametrierbare Umschaltzeit für die betreffende Spindel zur Verfügung stellt.

Für weitere Hinweise zum Funktionsbaustein FC17 siehe Kapitel "P3: PLC-Grundprogramm für SINUMERIK 840D sl (Seite 907)".



## 15.4.2 Spindelgetriebestufe 0

### Technischer Hintergrund

Bei Maschinen mit umschaltbarem Spindellastgetriebe können Situationen auftreten, bei denen der Getriebezug zwischen Motor und Last (Werkstück/Werkzeug) unterbrochen ist. Dieser Zustand kann z. B. durch Betätigen von Reset oder Not-Halt während der Durchführung eines Getriebestufenwechsels oder bei der Erstinbetriebnahme während der Montage einer Maschine auftreten. Die Steuerung muss diesen Zustand des offenen Getriebezugs erkennen und die nächste Getriebestufenwechselanforderung muss unbedingt ausgeführt werden.

### Funktion

Bei ausgerücktem Getriebe wird von der PLC über die Nahtstellensignalbits DB31, ... DBX16.0-2 (Istgetriebestufe A bis C) der binärcodierte Wert "0" ( $\triangleq$  Getriebestufe 0) an die NC übermittelt:

DB31, ... DBX16.0-2 = 0

Dieser Wert dient der Steuerung als Kennzeichen für den Zustand des offenen Getriebezugs.

### Auswirkungen auf den Getriebestufenwechsel

#### Getriebestufenwechsel im Teileprogramm

Die von der PLC gemeldete Istgetriebestufe wird von der NC beim Start eines Teileprogramms gelesen. Wird für die Istgetriebestufe zu diesem Zeitpunkt der Wert "0" gelesen, dann wird der nächste Getriebestufenwechsel ausgeführt und der Getriebestufenwechseldialog mit der PLC wird durchgeführt. Wenn ein Wert größer "0" gelesen wird, dann erfolgt bereits bei der Programmierung ein Vergleich zwischen angeforderter und aktiver Getriebestufe. Sind beide Getriebestufen gleich, dann wird kein Getriebestufenwechsel durchgeführt und eine eventuell programmierte Bahnbewegung wird nicht unterbrochen.

#### Getriebestufenwechsel in Synchronaktionen, FC18 und DBB30

Die von der PLC gemeldete Istgetriebestufe wird von der NC beim Getriebestufenwechsel immer ausgewertet. Wird der Wert "0" von der NC gelesen, dann wird der Getriebestufenwechsel immer ausgeführt. Beim Lesen eines Werts größer "0" erfolgt ein Vergleich zwischen angeforderter und aktiver Getriebestufe. Nur bei Ungleichheit beider Werte wird der Getriebestufenwechsel mit der PLC durchgeführt und das NC/PLC-Nahtstellensignal DB31, ... DBX82.3 (Getriebe umschalten) wird ausgegeben.

## Randbedingungen

- **Ausgabe von DB31, ... DBX16.0-2 = 0**

Bei ausgerücktem Getriebe muss von der PLC die Getriebestufe 0 in die NC/PLC-Nahtstelle DB31, ... DBX16.0-2 (Istgetriebestufe A bis C) eingetragen werden.

- **Freigabe des Getriebestufenwechsels**

Voraussetzung für einen Getriebestufenwechsel nach Lesen der Getriebestufe 0 ist die generelle Freigabe des Getriebestufenwechsels über die Maschinendaten:

MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE (Getriebestufenwechsel parametrieren)

MD35090 \$MA\_NUM\_GEAR\_STEPS (Anzahl eingerichteter Getriebestufen)

MD35092 \$MA\_NUM\_GEAR\_STEPS2 (2. Getriebestufenwechseldatensatz: Anzahl eingerichteter Getriebestufen), wenn MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE, Bit 5 = 1 (Gewindebohren ohne Ausgleichsfutter)

- **PLC-Anwenderprogramm/ POWER ON-ASUP**

Vom PLC-Anwenderprogramm oder POWER ON-ASUP ist sicherzustellen, dass bei ausgerücktem Getriebe (Getriebestufe 0) vor einer Spindelbewegung eine Getriebestufenwechselanforderung programmiert wird. Das kann z. B. mit M41 im ASUP erfolgen. Spindelbewegungen wie z. B. in JOG oder im Achsbetrieb erzeugen selbst keinen Getriebestufenwechsel.

## Beispiel

Beispiel für den Ablauf zum Schalten in die erste Getriebestufe nach POWER ON

1. POWER ON.
2. Das PLC-Anwenderprogramm ermittelt im mechanischen Umfeld den Zustand "Getriebe ist ausgerückt".
3. Die PLC übermittelt den Zustand "Getriebe ist ausgerückt" an die NC durch Setzen von:  
DB31, ... DBX16.0-2 = 0
4. Teileprogrammstart oder POWER ON-ASUP.
5. N05 (Teileprogramm s. u.) wird abgearbeitet:

Ein Getriebestufenwechsel nach Getriebestufe 1 wird durchgeführt.

Von der NC:

- wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX82.3 (Getriebe umschalten)

- wird die Sollgetriebestufe 1 an die PLC gemeldet:

DB31, ... DBX82.0 = 1

DB31, ... DBX82.1 = 0

DB31, ... DBX82.2 = 0

## 6. Mechanischer Wechsel der Getriebestufe, Quittung

Ist die Getriebestufe eingelegt, dann wird von der PLC:

- das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

- Istgetriebestufe 1 an die NC gemeldet:

DB31, ... DBX16.0 = 1

DB31, ... DBX16.1 = 0

DB31, ... DBX16.2 = 0

## 7. N80 wird abgearbeitet:

Es findet aufgrund der Optimierung der Getriebestufenwechselfrequenz im Teileprogramm kein Getriebestufenwechsel statt.

Teileprogramm:

Programmcode	Kommentar
N05 M41	; Anwahl 1.Getriebestufe
...	
N80 M41	; Kein Getriebestufenwechsel, wenn 1.Getriebestufe eingelegt ist.

Projektierungsdaten für Spindel 1 (AX5):

MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE[AX5] = 1 (Getriebestufenwechsel freigeben)

### 15.4.3 Ermittlung der Spindelgetriebestufe

Die aktuelle Getriebestufe einer Spindel ist über Systemvariablen lesbar:

- Für die Anzeige in der Bedienoberfläche, in Synchronaktionen oder mit Vorlaufstopp im Teileprogramm über die Systemvariablen:

\$VC_SGEAR[<n>]	Aktuell eingelegte Spindelgetriebestufe \$VC_SGEAR liest die von der PLC gemeldete Ist-Getriebestufe. Wertebereich: 0 ... 5
\$AC_SGEAR[<n>]	Aktive Spindelgetriebestufe \$AC_SGEAR liest die Soll-Getriebestufe im Hauptlauf. Wertebereich: 1 ... 5 Entsprechend dieser Getriebestufe wird der Datensatz für die Spindel aktiviert.

---

#### Hinweis

Bei Suchlauf kann sich die Ist-Getriebestufe (\$VC\_SGEAR[<n>]) von der Soll-Getriebestufe (\$AC\_SGEAR[<n>]) unterscheiden, da während des Suchlaufs kein Getriebestufenwechsel stattfindet. Mit Hilfe von \$VC\_SGEAR[<n>] und \$AC\_SGEAR[<n>] kann also abgefragt werden, ob ein Getriebestufenwechsel nach einem Suchlauf erfolgen soll.

---

- Ohne Vorlaufstopp im Teileprogramm über die Systemvariablen:

\$P_SGEAR[<n>]	Soll-Getriebestufe \$P_SGEAR liest die im Teileprogramm programmierte (M41 ... M45), bei M40 ausgewählte oder bei M70 projektierte Getriebestufe.
\$P_SEARCH_SGEAR[<n>]	Suchlauf: Getriebestufen-M-Funktion \$P_SEARCH_SGEAR enthält die bei Satzsuchlauf aufgesammelte zuletzt programmierte Getriebestufen-M-Funktion.

## 15.4.4 Parametersatzanwahl beim Getriebestufenwechsel

### Servo-Parametersätze

Die Servo-Parametersätze 1 bis 6 dienen zur Anpassung des Lagereglers an veränderte Eigenschaften der Maschine bei einer Getriebeumschaltung der Spindel.

### Parametersatzanwahl beim Getriebestufenwechsel

Beim Getriebestufenwechsel werden der Getriebestufen-Parametersatz (Interpolationsparameter) und, abhängig von der Einstellung im folgenden Maschinendatum, auch der Servo-Parametersatz umgeschaltet:

MD35590 \$MA\_PARAMSET\_CHANGE\_ENABLE (Parametersatzwechsel möglich)

Wert	Bedeutung
0	<p>Systeminterne Parametersatzanwahl</p> <p>Die Parametersätze des Servos sind fest zugeordnet.</p> <p>Dabei gilt:</p> <ul style="list-style-type: none"> <li>Bei Achsen und Spindeln im Achsbetrieb wirkt grundsätzlich der erste Parametersatz.</li> </ul> <p>Ausnahme:</p> <p>Bei G33, G34, G35, G331 und G332 wird für die beteiligten Achsen der Parametersatz mit folgender Nummer aktiviert:</p> <p>Masterspindel-Getriebestufe + 1 (entspricht Parametersatznummer 2 ... 6)</p> <ul style="list-style-type: none"> <li>Bei Spindeln im Spindelbetrieb wird der Parametersatz passend zur Getriebestufe eingestellt.</li> </ul>
1	<p>Außer der systeminternen Parametersatzanwahl besteht auch die Möglichkeit einer "externen" Parametersatzanwahl:</p> <ul style="list-style-type: none"> <li>durch die PLC (DB31, ... DBX 9.0 - 9.2)</li> <li>durch Programmierung von <code>SCPARA</code> im Teileprogramm oder in Synchronaktionen</li> </ul> <p>Die systeminterne Parametersatzanwahl hat jedoch Vorrang.</p> <p>Hinweis: Der Wert 1 ist nur für Achsen sinnvoll.</p>
2	<p>Der Servo-Parametersatz wird <b>ausschließlich</b> durch die PLC (DB31, ... DBX 9.0 - 9.2) oder durch Programmierung von <code>SCPARA</code> im Teileprogramm oder in Synchronaktionen vorgegeben (für Achsen und Spindeln).</p> <p>Nach POWER ON ist der 1. Parametersatz angewählt.</p>

**Spindelbetrieb**

MD35590 \$MA\_PARAMSET\_CHANGE\_ENABLE = 0 oder 1

Der Parametersatz wird entsprechend der Getriebestufe + 1 angewählt.

Die aktive Getriebestufe steht in:

DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C)

Der aktive Parametersatz wird ausgegeben in:

DB31, ... DBX69.0-69.2 (Regler-Parametersatz A bis C)

Für jede der 5 Getriebestufen gibt es einen Parametersatz, der von der NC folgendermaßen zugeordnet ist:

Datensatz für ...	NC/PLC-Nahtstelle DBX 69.2 / 69.1 / 69.0	Parametersatz Nummer	Parametersatz Index [n]
Achsbetrieb	letzte aktive Getriebestufe	1	0
Getriebestufe 1	001	2	1
Getriebestufe 2	010	3	2
Getriebestufe 3	011	4	3
Getriebestufe 4	100	5	4
Getriebestufe 5	101 110 111	6	5

**Spindel im Achsbetrieb**

Befindet sich die Spindel im Achsbetrieb, dann wird im Servo der Parametersatz-Index "0" angewählt (MD35590 \$MA\_PARAMSET\_CHANGE\_ENABLE beachten!).

Das Getriebestufenwechsel-Verhalten ist abhängig von der Einstellung im Maschinendatum:

MD35014 \$MA\_GEAR\_STEP\_USED\_IN\_AXISMODE (Getriebestufe für den Achsbetrieb bei M70)

Wenn keine Getriebestufe für den Achsbetrieb projiziert ist (MD35014 = 0), dann findet kein impliziter Getriebestufenwechsel bei M70 statt (Standardeinstellung!). Die letzte Getriebestufe wird intern abgespeichert und wird mit dem dazugehörigen Parametersatz mit der nächsten Spindelprogrammierung wieder aktiv.

Ist dagegen eine Getriebestufe für den Achsbetrieb projiziert (MD35014 = 1 ... 5), dann findet ein Getriebestufenwechsel in die Getriebestufe 1 ... 5 während der Abarbeitung von M70 statt. Beim Wechsel vom Achsbetrieb in den Spindelbetrieb bleibt die mit M70 eingewechselte Getriebestufe weiterhin aktiv. Die im Spindelbetrieb vor M70 aktive Getriebestufe wird nicht automatisch wieder eingewechselt.

Siehe auch "Projektierbare Getriebestufe bei M70 (Seite 1456)".

### Lastgetriebeübersetzung

Es ist möglich, für jede Getriebestufe und im Achsbetrieb positive oder negative **Lastgetriebefaktoren** zu projektieren.

Die Einstellung erfolgt getrennt für Zähler und Nenner über die Maschinendaten:

MD31050 \$MA\_DRIVE\_AX\_RATIO\_DENOM[n] (Nenner Lastgetriebe)

MD31060 \$MA\_DRIVE\_AX\_RATIO\_NUMERA[n] (Zähler Lastgetriebe)

Der Einstellbereich ist für positive und negative Lastgetriebefaktoren gleich groß.

Die Eingabe des Werts "0" ist nicht möglich.

---

#### Hinweis

Wenn ein indirekter Geber projiziert ist und sich die Lastgetriebeübersetzung ändert, dann geht die Referenz verloren und das NC/PLC-Nahtstellensignal:

DB31, ... DBX60.4/60.5 (Referiert / Synchronisiert 1 bzw. 2)

wird für das betreffende Messsystem zurückgesetzt.

---

### Literatur

Weitere Informationen zur Regelung und zum Servo-Parametersatz siehe:

- Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsystem, Regelung (G2)
- Programmierhandbuch Arbeitsvorbereitung; Kap.: Programmierbarer Servo-Parametersatz

### 15.4.5 Vorsatz-Getriebe

#### Anwendung und Funktionalität

Mit einem projektierbaren Vorsatz-Getriebe können verschiedene angetriebene Werkzeuge angepasst werden. Das an der Werkzeugseite befindliche Vorsatz-Getriebe wirkt multiplikativ zum Motor-/Last-Getriebe.

Es wird eingestellt über die Maschinendaten:

MD31066 \$MA\_DRIVE\_AX\_RATIO2\_NUMERA (Zähler Vorsatzgetriebe)

MD31064 \$MA\_DRIVE\_AX\_RATIO2\_DENOM (Nenner Vorsatzgetriebe)

Ein für das Vorsatz-Getriebe vorhandener werkzeugseitiger Geber wird mit dem Maschinendatum:

MD31044 \$MA\_ENC\_IS\_DIRECT2 (Geber am Vorsatzgetriebe) projiziert.

Eine geänderte Parametrierung dieser Maschinendaten kann mit "NewConfig" entweder mit Hilfe der Inbetriebnahme-Software SinuCOM-NC oder über die Bedientafel HMI über einen Softkey aktiviert werden. Die bereits existierenden Motor-/Last-Getriebe sind dagegen über Power-On wirksam.

#### Werkzeugwechsel

Wird beim Werkzeugwechsel das Vorsatzgetriebe mit gewechselt, dann muss vom Anwender das Übersetzungsverhältnis vom Zähler und Nenner über die Maschinendaten des Vorsatz-Getriebes auch umprojiziert werden.

##### Beispiel:

Für ein eingewechseltes Werkzeug mit einem Übersetzungsverhältnis von 2:1 wird ein passendes Vorsatzgetriebe projiziert und im Teileprogramm mit dem Befehl `NEWCONF` sofort wirksam.

**Programmcode**

```
N05 $MA_DRIVE_AX_RATIO2-NUMERA [AX5] = 2
M10 $MA_DRIVE_AX_RATIO2-DENOM [AX5] = 1
N15 NEWCONF
```

 **VORSICHT**

Es bleibt Aufgabe des Anwenders, in der betreffenden Zeitspanne anzuhalten, um in unmittelbarem zeitlichem Zusammenhang eine Änderung der Maschinendaten vorzunehmen und danach ein "NewConfig" zu aktivieren.



## Umschaltung

Die Umschaltung auf die neue Übersetzung wird mittels NewConfig sofort durchgeführt. Technologisch hat primär der zugehörige mechanische Umschaltvorgang eine zeitliche Ausdehnung, da mechanisch ein anderer Vorsatzkopf mit angetriebenen Werkzeug eingewechselt wird.

---

### Hinweis

Im Stillstand ist die Umschaltung ruckfrei. Es liegt daher in der Verantwortung des Anwenders hierfür entsprechende Vorsorgemaßnahmen zu ergreifen.

Anwendungen bei denen in Bewegung umgeschaltet wird und die einen geglätteten oder weichen Drehzahl-Übergang benötigen, sind mit existierenden antriebsseitigen Drehzahlsollwert-Filtern beherrschbar.

---

Weitere Erläuterungen hinsichtlich regelungstechnischer Abhängigkeiten siehe Kapitel "G2: Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (Seite 331)".

## 15.4.6 Nicht quittierter Getriebestufenwechsel

### Betriebsartenwechsel

Ein nicht quittierter Getriebestufenwechsel kann nicht durch einen Betriebsartenwechsel (z.B. Umschaltung nach JOG) unterbrochen werden.

Die Umschaltung wird maximal um die im Maschinendatum:  
MD10192 \$MN\_GEAR\_CHANGE\_WAIT\_TIME  
eingetragene Zeitspanne verzögert.

Wird der Getriebestufenwechsel nicht innerhalb dieser Zeitdauer quittiert, so löst die NC einen Alarm aus.

### Weitere Ereignisse

Ereignisse, die ein Reorganisieren auslösen, warten ebenso das Ende eines Getriebestufenwechsels ab.

Die im Maschinendatum:  
MD10192 \$MN\_GEAR\_CHANGE\_WAIT\_TIME  
eingestellte Zeit bestimmt nun wie lange auf den Getriebestufenwechsel gewartet wird. Läuft diese Zeit ab, ohne dass ein Getriebestufenwechsel beendet wurde, so reagiert die NC mit einem Alarm.

Folgende Ereignisse besitzen analoges Verhalten:

- Anwender-ASUP
- Mode-Wechsel
- Restweg löschen
- Achstausch
- PI-Anwenderdaten wirksam setzen
- PI-Dienst Maschinendaten wirksam setzen
- Ausblenden umschalten, DryRun umschalten
- Editieren in der Betriebsart
- Korrektursatzalarme
- Overstore
- Schnellrückzug bei G33, G34, G35
- Unterprogrammebenenabbruch, Unterprogrammabbruch

#### Verhalten nach Power On

Nach Power On und im RESET-Zustand des NCK kann die an der Maschine aktive Getriebestufe von der PLC Steuerung vorgegeben werden.

Die NCK wählt daraufhin den entsprechenden Parametersatz an und meldet die NC/PLC-Nahtstellensignale:  
DB31, ... DBX82.0-82.2 (Sollgetriebestufe A bis C)  
an die PLC zurück.

## 15.4.7 Getriebestufenwechsel mit Pendelbetrieb

### Was ist Pendeln?

Beim Pendeln dreht sich der Spindelmotor abwechselnd im Uhrzeigersinn und gegen den Uhrzeigersinn. Durch diese Pendelbewegung wird das leichte Einrücken einer neuen Getriebestufe unterstützt.

### Pendelbetrieb

Mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX82.3 (Getriebe umschalten)

wird angezeigt, dass ein Getriebestufenwechsel notwendig ist.

Prinzipiell kann auch ohne Pendelvorgang die neue Getriebestufe eingelegt werden.

1. MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE muss auf 1 gesetzt sein.
2. NC/PLC-Nahtstellensignal DB31, ... DBX84.6 (Pendelbetrieb) wird gesetzt.
3. Die Beschleunigung wird festgelegt im Maschinendatum:

MD35410 \$MA\_SPIND\_OSCILL\_ACCEL

### DB31, ... DBX18.5 (Pendeldrehzahl)

Die Spindel befindet sich im Pendelbetrieb, wenn durch die automatische Getriebestufenauswahl ( $M_{40}$ ) oder durch  $M_{41}$  bis  $M_{45}$  eine neue Getriebestufe vorgegeben wurde (DB31, ... DBX82.3 (Getriebe umschalten) ist gesetzt).

Das NC/PLC-Nahtstellensignal:

DB31, ... DBX82.3 (Getriebe umschalten)

wird nur gesetzt, wenn eine neue Getriebestufe vorgegeben wird, die ungleich der aktuellen Istgetriebestufe ist.

Wird nur das NC/PLC-Nahtstellensignal:

DB31, ... DBX18.5 (Pendeldrehzahl)

von der PLC gesetzt, ohne dass eine neue Getriebestufe durch die NC vorgegeben wurde, erfolgt kein Wechsel in den Pendelbetrieb.

Das Pendeln wird mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX18.5 (Pendeldrehzahl)

gestartet.

Beim Funktionsablauf wird je nach NC/PLC-Nahtstellensignal:

DB31, ... DBX18.4 (Pendeln durch die PLC)

unterschieden in:

- Pendeln durch den NCK
- Pendeln durch die PLC
- Pendeln mit FC 18

#### Literatur:

Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

## Pendelzeit

Für jede Drehrichtung lässt sich beim Pendeln die Pendelzeit in einem Maschinendatum festlegen:

Pendelzeit in M3-Richtung (im Folgenden t1 genannt) in:	MD35440 \$MA_SPIND_OSCILL_TIME_CW
Pendelzeit in M4-Richtung (im Folgenden t2 genannt) in:	MD35450 \$MA_SPIND_OSCILL_TIME_CCW

## Pendeln durch den NCK

### Phase 1:

Mit dem NC/PLC-Nahtstellensignal:  
DB31, ... DBX18.5 (Pendeldrehzahl)  
beschleunigt der Spindelmotor auf die in Maschinendatum:  
MD35400 \$MA\_SPIND\_OSCILL\_DES\_VELO (Pendeldrehzahl)  
festgelegte Geschwindigkeit (mit Pendelbeschleunigung).

Die Startrichtung wird durch das folgende Maschinendatum festgelegt:  
MD35430 \$MA\_SPIND\_OSCILL\_START\_DIR (Startrichtung beim Pendeln)

Die Zeit t1 (oder t2) wird gestartet,  
je nachdem welche Startrichtung im Maschinendatum festgelegt wurde:  
MD35430 \$MA\_SPIND\_OSCILL\_START\_DIR

Entscheidend ist immer die Zeit und nicht das Erreichen der Pendelgeschwindigkeit.

### Phase 2:

Ist die Zeit t1 (t2) abgelaufen, beschleunigt der Spindelmotor  
in die entgegengesetzte Richtung auf die in Maschinendatum:  
MD35400 \$MA\_SPIND\_OSCILL\_DES\_VELO  
festgelegte Geschwindigkeit.

Die Zeit t2 (t1) wird gestartet.

### Phase 3:

Ist die Zeit t2 (t1) abgelaufen, beschleunigt der Spindelmotor in die entgegengesetzte  
Richtung (gleiche Richtung wie in Phase 1) auf die in Maschinendatum:  
MD35400 \$MA\_SPIND\_OSCILL\_DES\_VELO  
festgelegte Geschwindigkeit.

Die Zeit t1 (t2) wird gestartet. Fortsetzung mit Phase 2.

## Pendeln durch die PLC

Mit den NC/PLC-Nahtstellensignalen:  
DB31, ... DBX18.4 (Pendeln durch die PLC))  
und  
DB31, ...DBX18.5 (Pendeldrehzahl)  
beschleunigt der Spindelmotor auf die in Maschinendatum:  
MD35400 \$MA\_SPIND\_OSCILL\_DES\_VELO  
festgelegte Geschwindigkeit (mit Pendelbeschleunigung).

Die Drehrichtung wird durch die NC/PLC-Nahtstellensignale:  
DB31, ... DBX18.7 (Solldrehrichtung links)  
und  
DB31, ... DBX18.6 (Solldrehrichtung rechts)  
festgelegt.

Das Pendeln (die Pendelbewegung) und die zwei Zeiten t1 und t2 (Zeit für Drehrichtung im und entgegen dem Uhrzeigersinn) müssen in der PLC nachgebildet werden.

## Besonderheiten

Setzen/Rücksetzen von NC/PLC-Nahtstellensignalen und Maschinendaten bei Pendelbetrieb:

- Zum Abbremsen der Spindel braucht der PLC-Anwender das NC/PLC-Nahtstellensignal: DB31, ... DBX4.3 (Spindel-Halt) nicht zu setzen.

Die Spindel wird steuerungsintern stillgesetzt, wenn ein Getriebestufenwechsel angefordert wird.

- Der Getriebestufenwechsel sollte immer mit dem NC/PLC-Nahtstellensignal: DB31, ... DBX16.3 (Getriebe ist umgeschaltet) beendet werden.
- Das NC/PLC-Nahtstellensignal: DB31, ... DBX18.5 (Pendeldrehzahl) soll das mechanische Einrasten des Getriebes unterstützen.

Es hat steuerungsintern keinen Einfluss auf den Ablauf des Getriebestufenwechsels und braucht deshalb nur bei Bedarf gesetzt werden.

- Wird das NC/PLC-Nahtstellensignal:  
DB31, ... DBX18.5 (Pendeldrehzahl)  
rückgesetzt, stoppt die Pendelbewegung.  
Die Spindelbetriebsart "Pendelbetrieb" wird aber nicht verlassen.
- Die Beschleunigung wird im folgenden Maschinendatum festgelegt:  
MD35410 \$MA\_SPIND\_OSCILL\_ACCEL
- Bei indirektem Messsystem (Motorgeber) geht die Synchronisation verloren.  
Ist die Maschinendatum-Einstellung:  
MD31050 \$MA\_ENC\_IS\_DIRECT = 0,  
wird automatisch das NC/PLC-Nahtstellensignal:  
DB31, ... DBX60.4/5 = 0 (Referenziert/synchronisiert)  
gelöscht.  
Anschließend wird beim nächsten Überfahren der Nullmarke synchronisiert.

### Ende des Pendelbetriebes

Die Spindel befindet sich nach Beendigung des Pendelbetriebes wieder im Steuerbetrieb und geht automatisch in den durch `SPCON` bzw. `SPCOF` festgelegten Betrieb.

Alle getriebespezifischen Grenzwerte (min./max. Drehzahl der Getriebestufe, etc.) entsprechen den vorgegebenen Werten der Istgetriebestufe.

### Funktionalität

Werkzeugmaschinen herkömmlicher Bauart erfordern einen Getriebestufenwechsel der Spindel im Pendelbetrieb.

Bei der Maschinendatum-Projektierung:

**MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE = 1**

ist folgender Ablauf implementiert:

- Abbremsen der Spindel.  
Der Bremsvorgang entspricht dabei einer M5-Bewegung.
- Ausgabe der VDI-Nahtstellensignale:  
DB31, ... DBX84.6 (Pendelbetrieb)  
DB31, ... DBX82.3 (Getriebe umschalten)  
DB31, ... DBX82.0-82.2 (Sollgetriebestufe A bis C)  
Die Lageregelung wird, falls diese aktiv war, ausgeschaltet:  
DB31, ...DBX61.5 = 0.
- Das Lastgetriebe kann jetzt "ausrücken".

- Von der PLC kann das NC/PLC-Nahtstellensignal: DB31, ... DBX18.5 (Pendelfreigabe) gesetzt werden.

Daraufhin führt der Spindelmotor eine Pendelbewegung mit projektierten Werten aus. Die Pendelbewegung soll den Einrastvorgang der Getrieberäder ermöglichen und beschleunigen.

- Beschreiben der NC/PLC-Nahtstellensignale: DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C) durch die PLC.
- Nach der Meldung von der PLC an den NCK: DB31, ... DBX16.3 (Getriebe ist umgeschaltet) wird die zuletzt aktive Bewegung, falls vorhanden, fortgesetzt.

Für indirekte Geber (Motorgeber) wird der Referenziertstatus gelöscht: DB31, ... DBX60.4/5 = 0.

## Satzwechsel

Wurde die Spindel mit dem NC/PLC-Nahtstellensignal: DB31, ... DBX82.3 (Getriebe umschalten) in den Pendelbetrieb geschaltet, bleibt die Teileprogrammbearbeitung angehalten. Ein neuer Satz wird nicht bearbeitet.

Wird der Pendelbetrieb mit dem NC/PLC-Nahtstellensignal: DB31, ... DBX16.3 (Getriebe ist umgeschaltet) beendet, wird die Teileprogrammbearbeitung fortgesetzt. Ein neuer Satz wird bearbeitet.

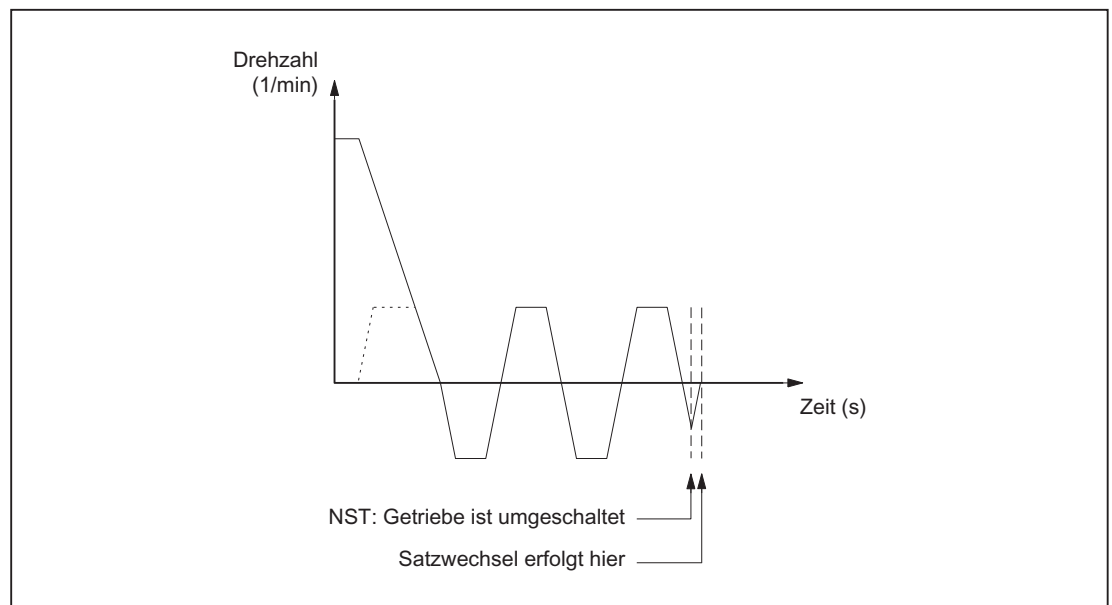
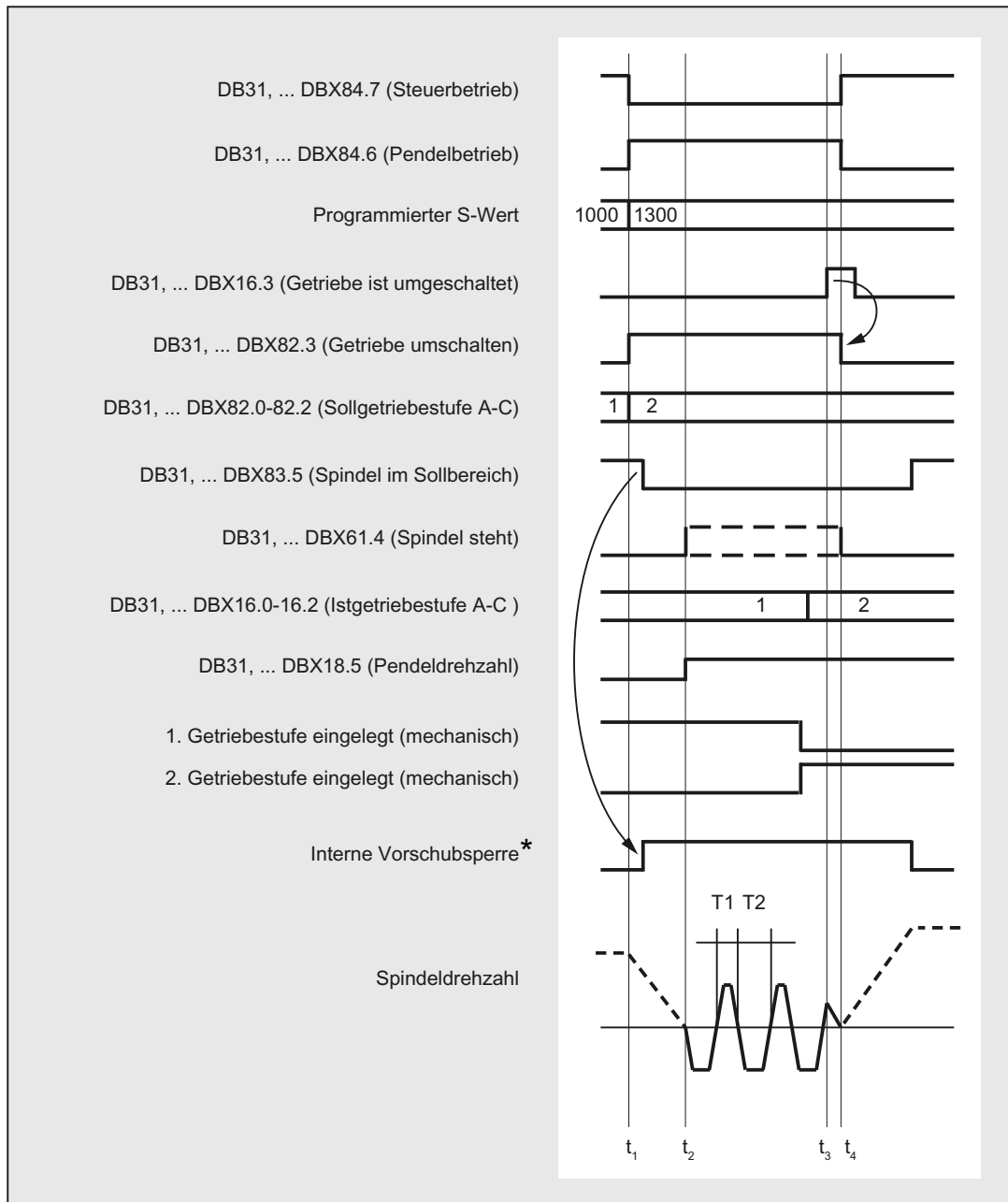


Bild 15-5 Satzwechsel nach Pendelbetrieb

**Pendelbetrieb**

Typischer zeitlicher Ablauf für den Getriebestufenwechsel bei einer Spindel:





- t<sub>1</sub>: NCK erkennt durch Programmierung von S1300 eine neue Getriebestufe (2. Getriebestufe), setzt das NST DB31, ... DBX82.3 (Getriebe umschalten) und sperrt die Bearbeitung für den nächsten Teileprogrammsatz (= interne Vorschubsperrung\*).
- t<sub>2</sub>: Die Spindel steht und es wird mit dem Pendeln begonnen (Pendeln durch die NCK). Das NST DB31, ... DBX18.5 (Pendeldrehzahl) muss spätestens zum Zeitpunkt t<sub>2</sub> gesetzt werden.
- t<sub>3</sub>: Die neue Getriebestufe ist eingelegt. Der PLC-Anwender übergibt die neue (Ist-)Getriebestufe an die NCK und setzt das NST DB31, ... DBX16.3 (Getriebe ist umgeschaltet).
- t<sub>4</sub>: Die NCK nimmt daraufhin das NST DB31, ... DBX82.3 (Getriebe umschalten) zurück, beendet das Pendeln, gibt den nächsten Teileprogrammsatz zur Bearbeitung frei und beschleunigt die Spindel auf den neuen S-Wert (S1300).
- \* : Die interne Vorschubsperrung wird gesetzt, wenn:
- der Spindel-Getriebestufenwechsel über das Teileprogramm programmiert wurde **und**
  - ein Bearbeitungssatz aktiv ist (d. h. G0 ist nicht aktiv).
- Bei einem Getriebestufenwechsel aus Synchronaktionen oder bei Vorgaben über die PLC mit FC 18 wird die interne Vorschubsperrung **nicht** gesetzt.

Bild 15-6 Getriebestufenwechsel bei stehender Spindel

## 15.4.8 Getriebestufenwechsel auf Festposition

### Anwendung und Vorteile

An Werkzeugmaschinen werden zunehmend standardisierte Spindelgetriebe eingesetzt, um einerseits technologische Totzeit beim Getriebestufenwechsel zu sparen und um andererseits den Kostenvorteil aus der Verwendung standardisierter mechanischer Bauteile zu nutzen.

Die Funktion Getriebestufenwechsel auf Festposition unterstützt den "gerichteten Getriebestufenwechsel" von Lastgetrieben, welche von der NC eine andere Ansteuerung erfordern. Der Getriebestufenwechsel kann dabei nur auf einer definierten Spindelposition ausgeführt werden. Eine Pendelbewegung, wie bei herkömmlichen Lastgetrieben, wird nicht mehr benötigt.

### Ablauf zum GSW auf Festposition

Der Getriebestufenwechsel auf Festposition

Die Maschinendatum-Projektierung:

**MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE = 2**

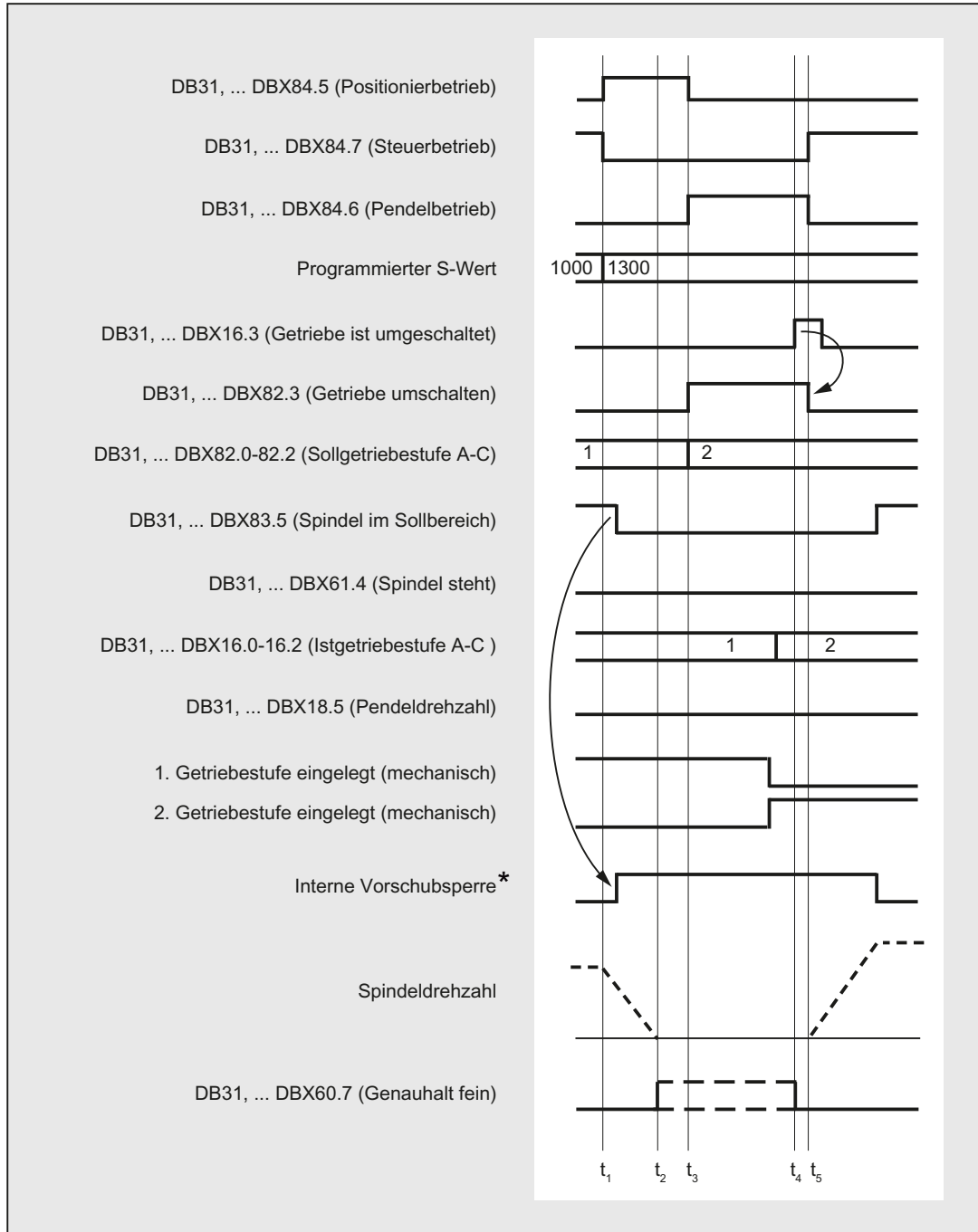
führtfolgenden Ablauf aus:

- Positionierung der Spindel aus dem Stillstand oder einer Bewegung auf die in Maschinendatum:  
MD35012 \$MA\_GEAR\_STEP\_CHANGE\_POSITIONprojektierte Position.  
  
Erfolgt der Getriebestufenwechsel aus der Bewegung, dann wird die aktuelle Drehrichtung beibehalten. Die Spindel befindet sich während des Positioniervorganges im Positionierbetrieb.  
  
Es wird das NC/PLC-Nahtstellensignal:  
DB31, ... DBX84.5 (Positionierbetrieb)  
ausgegeben.  
  
Ist keine Referenz vorhanden:  
DB31, ... DBX60.4/5 = 0  
oder das NC/PLC-Nahtstellensignal:  
DB 31, ... DBX17.4/5 (Beim Positionieren neu Synchronisieren MS 1/2)  
gesetzt, dann verlängert sich der Positioniervorgang um die Nullmarkensuche.
- Nach dem Erreichen der im Maschinendatum:  
MD35012\$MA\_GEAR\_STEP\_CHANGE\_POSITION  
projektierten GSW-Position wird die Zeit aus Maschinendatum:  
MD35310 \$MA\_SPIND\_POSIT\_DELAY\_TIME  
abgewartet, bevor auf Pendelbetrieb umgeschaltet wird,  
und der bekannte GSW-Dialog beginnt.
- Ausgabe der NC/PLC-Nahtstellensignale:  
DB31, ... DBX84.6 (Pendelbetrieb)  
DB31, ... DBX82.3 (Getriebe umschalten)  
DB31, ... DBX82.0-82.2 (Sollgetriebestufe A bis C)
- Die Lageregelung wird bei aktivem Messsystem mit indirektem Geber (Motorgeber) nicht ausgeschaltet:  
  
MD31040 \$MA\_ENC\_IS\_DIRECT = 0  
  
Bei aktivem Messsystem mit direktem Geber (Lastgeber) wird die Lageregelung ausgeschaltet:  
DB31, ... DBX61.5 = 0,  
da der Kraftfluss zur Last unterbrochen wird und eine Positionsregelung dadurch nicht mehr möglich ist.
- Ist der lageregelte Betrieb nicht möglich, dann kann dieser durch Rücksetzen der "Reglerfreigabe":  
DB31, ... DBX2.1 = 0  
abgeschaltet werden.

- Mechanisches Umschalten der Getriebestufe an der Maschine.  
Dazu ist keine Pendelbewegung des Motors erforderlich.  
Die NC/PLC-Nahtstellensignale:  
DB31, ... DBX18.5 (Pendelfreigabe)  
und  
DB31, ... DBX18.4 (Pendeln durch die PLC)  
sollen **nicht** gesetzt werden.  
Die Pendelbewegung ist jetzt noch prinzipiell möglich.
- Beschreiben der NC/PLC-Nahtstellensignale:  
DB31, ... DBX16.0-16.2 (Istgetriebestufe A bis C)  
durch die PLC.
- Nach der Meldung:  
DB31, ... DBX16.3 (Getriebe ist umgeschaltet)  
wird die zuletzt aktive Bewegung, falls vorhanden, fortgesetzt.  
Für indirekte Geber (Motorgeber) wird der Referenzierstatus gelöscht:  
DB31, ... DBX60.4/5 = 0.  
Die Spindel befindet sich im Drehzahlsteuerbetrieb und das NC/PLC-Nahtstellensignal:  
DB31, ... DBX84.7 (Steuerbetrieb)  
wird ausgegeben.

**GSW Festposition**

Typischer zeitlicher Ablauf für den Getriebestufenwechsel auf Festposition:



- t<sub>1</sub>: NCK erkennt durch Programmierung von S1300 eine neue Getriebestufe (2. Getriebestufe), setzt das NST DB31, ... DBX84.5 (Positionierbetrieb) und sperrt die Bearbeitung für den nächsten Teileprogrammsatz (= interne Vorschubsperrung\*).
- t<sub>2</sub>: Die Spindel steht und es wird Genauhalt gemeldet.
- t<sub>3</sub>: Getriebestufenwechsel - Wartezeit
- t<sub>4</sub>: Die neue Getriebestufe ist eingelegt. Der PLC-Anwender übergibt die neue (Ist-)Getriebestufe an die NCK und setzt das NST DB31, ... DBX16.3 (Getriebe ist umgeschaltet).
- t<sub>5</sub>: Die NCK nimmt daraufhin das NST DB31, ... DBX82.3 (Getriebe umschalten) zurück, gibt den nächsten Teileprogrammsatz zur Bearbeitung frei und beschleunigt die Spindel auf den neuen S-Wert (S1300).
- \*: Die interne Vorschubsperrung wird gesetzt, wenn:
- der Spindel-Getriebestufenwechsel über das Teileprogramm programmiert wurde **und**
  - ein Bearbeitungssatz aktiv ist (d. h. GO ist nicht aktiv).
- Bei einem Getriebestufenwechsel aus Synchronaktionen oder bei Vorgaben über die PLC mit FC 18 wird die interne Vorschubsperrung **nicht** gesetzt.

Bild 15-7 Getriebestufenwechsel bei stehender Spindel

### GSW-Position MD35012

Die Getriebestufenwechselposition wird im Maschinendatum:  
MD35012 \$MA\_GEAR\_STEP\_CHANGE\_POSITION  
für jede Getriebestufe projiziert.

### GSW-Wartezeit MD35310

Nach dem Positioniervorgang wird um die im Maschinendatum:  
MD35310 \$MA\_SPIND\_POSIT\_DELAY\_TIME  
projizierte Zeit gewartet, bis die Getriebeumschaltanforderung:  
DB31, ... DBX84.6 (Pendelbetrieb)  
DB31, ... DBX82.3 (Getriebe umschalten)  
und  
DB31, ... DBX82.0-82.2 (Sollgetriebebestufe A bis C)  
ausgegeben wird.

### Positionierkennwerte / Position

Die Position wird immer auf kürzesten Weg (entspricht DC) angefahren.

Ist keine Referenz vorhanden und befindet sich die Spindel im Stillstand (z.B. nach Power On), dann wird die Anfahrrichtung durch das folgende Maschinendatum bestimmt:

MD35350 \$MA\_SPIND\_POSITIONING\_DIR

Wird eine veränderliche GSW-Position benötigt, dann kann dies durch Beschreiben des Maschinendatums mit anschließendem "NewConfig" erreicht werden.

Die Änderung des MD-Wertes kann über das Teileprogramm oder HMI erfolgen.

Sollte die projektierte Position nicht erreicht werden, dann wird der Alarm 22020 gemeldet und der GSW-Dialog zwischen NCK und PLC findet in diesem Fall nicht statt, um das Getriebe nicht zu zerstören. Da der Alarm schwerwiegend ist, kann das Teileprogramm nicht fortgesetzt werden und die Ursache muss unbedingt beseitigt werden. Die Ursache für den Abbruch der Positionierung dürfte erfahrungsgemäß auf falsche MD-Einstellungen oder unverträgliche PLC-Signale zurückführbar sein.

### Geschwindigkeit

Die Positioniergeschwindigkeit wird entnommen aus dem getriebestufenabhängig projektierten Maschinendatum:

MD35300 \$MA\_SPIND\_POSCTRL\_VELO

Die NC/PLC-Nahtstellensignale "Spindelkorrektur"/"Vorschubkorrektur" bei:

DB31, ... DBX17.0=0: DB31, ... DBB19

sowie bei:

DB31, ... DBX17.0=1: DB31, ... DBB0

sind wie beim Positionieren üblich wirksam.

Die Positioniergeschwindigkeit kann durch die Programmanweisung  $OVRA_{[Sn]}$  prozentual verändert werden.

---

#### Hinweis

$OVRA_{[Sn]}$  ist modal gültig. Nach dem Getriebestufenwechsel sollte wieder ein für die Bearbeitung sinnvoller Wert eingestellt werden.

---

Die Teileprogrammanweisung  $FA_{[Sn]}$  ändert die Positioniergeschwindigkeit beim Getriebestufenwechsel nicht.

## Beschleunigung

Die Beschleunigungswerte werden bestimmt durch die getriebestufenabhängig projektierten Maschinendaten:

MD35200 \$MA\_GEAR\_STEP\_SPEEDCTRL\_ACCEL

und

MD35210 \$MA\_GEAR\_STEP\_POSCTRL\_ACCEL

Die Beschleunigung kann durch Programmierung von ACC [Sn] prozentual verändert werden.

---

### Hinweis

ACC [Sn] ist modal gültig. Nach dem Getriebestufenwechsel sollte wieder ein für die Bearbeitung sinnvoller Wert eingestellt werden.

---

## Drehzahlabhängige Beschleunigung

Die "geknickte Beschleunigungskennlinie" ist wie beim Positionieren mit SPOS oder FC18 wirksam.

## Ruck

Gegenwärtig ist die Begrenzung der Beschleunigungsänderung nicht möglich.

## Positionierende

Der Übergang zwischen dem Ende des Positioniervorganges (DB31, ... DBX84.5) und dem Beginn des Pendelbetriebes (DB31, ... DBX84.6) wird bestimmt durch das Erreichen von "Genauhalt fein" (DB31, ... DB60.7) und der Größe der in Maschinendatum: MD3510 \$MA\_SPIND\_POSIT\_DELAY\_TIME eingetragenen Zeit.

Die Festlegung der Übergangsbedingung wirkt sich einerseits auf die GSW-Zeit und andererseits auf die Genauigkeit des Erreichens der projektierten GSW-Position aus.

## Satzwechsel

Der Satzwechsel wird angehalten und die Bearbeitungssätze werden bis zur Beendigung des Getriebestufenwechsel durch die PLC (DB31, ... DBX16.3) nicht begonnen.

## GSW-Ende

Die Spindel befindet sich nach Beendigung des Getriebestufenwechsels wieder im Steuerbetrieb und geht automatisch in den durch SPCON bzw. SPCOF festgelegten Reglermodus.

Alle getriebespezifischen Grenzwerte (min./max. Drehzahl der Getriebestufe, etc.) entsprechen der zurückgemeldeten Istgetriebestufe.

### Randbedingungen

- Die Spindel muss mindestens ein Messsystem besitzen.
- Der lagegeregelte Betrieb muss möglich und in Betrieb genommen worden sein.
- Grundsätzlich muss `SP05` aus dem Teileprogramm, Synchronaktion oder über FC18: "Start Spindel positionieren" fehlerfrei ausführbar sein.

Können nicht alle Voraussetzungen erfüllt werden, dann kann die beschriebene Funktion nicht erfolgreich angewendet werden.

### Aktivierung

Die Funktion des Getriebestufenwechsel auf Festposition wird durch die Projektierung aktiviert:  
`MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 2`

## 15.4.9 Projektierbare Getriebestufe bei M70

### Technischer Hintergrund

Bei manchen Maschinen ist es erforderlich, dass sich die Spindel während des Achsbetriebs in einer bestimmten Getriebestufe befindet.

Gründe dafür können sein:

- Im Servo-Parametersatz für den Achsbetrieb (Index 0) kann nur eine Optimierung ( $K_v$ , Vorsteuerung, Filter) passend zu einer Getriebestufe gefunden werden. Die Maschinendaten dieses Parametersatzes sollen nicht umgeschrieben werden.
- Es gibt nur eine mechanische Getriebeübersetzung, die im Gegensatz zu den restlichen eine geringe oder keine Umkehrlose besitzt. Nur in dieser Getriebestufe kann die Spindel zusammen mit anderen Achsen einer Bahnbewegung oder Transformationen (z. B. TRANSMIT) folgen.

### Funktion

Bei aktiver Funktion kommt es beim Übergang in den Achsbetrieb zum automatischen Einwechseln einer vorbestimmten Getriebestufe.

Der Getriebestufenwechsel ist in den M70-Ablauf integriert und findet nach dem Abbremsen der Spindel und vor dem Einwechseln des Servo-Parametersatzes mit Index 0 statt (`MD35590 $MA_PARAMSET_CHANGE_ENABLE` beachten!).

Der beim Getriebestufenwechsel typische Dialog zwischen NC und PLC wird gleichermaßen wie bei einem programmierten Getriebestufenwechsel (M41 ... M45) ausgeführt.



## Voraussetzungen

Voraussetzung für einen Getriebestufenwechsel beim Übergang in den Achsbetrieb ist die generelle Freigabe des Getriebestufenwechsels über die Maschinendaten:

MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE (Getriebestufenwechsel parametrieren)

MD35090 \$MA\_NUM\_GEAR\_STEPS (Anzahl eingerichteter Getriebestufen)

## Aktivierung / Deaktivierung

Die Funktion wird aktiviert / deaktiviert über das Maschinendatum:

MD35014 \$MA\_GEAR\_STEP\_USED\_IN\_AXISMODE (Getriebestufe für den Achsbetrieb bei M70)

Wert	Bedeutung
0	Es findet kein impliziter Getriebestufenwechsel bei M70 statt. Die aktuelle Getriebestufe wird beibehalten (Standardeinstellung!).
1 ... 5	Es findet ein Getriebestufenwechsel in die Getriebestufe 1 ... 5 während der Abarbeitung von M70 statt.

## Randbedingungen

### Getriebestufenwechsel auf Festposition (MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE = 2)

Die Funktion "Getriebestufenwechsel auf Festposition" wird unterstützt. Der Ablauf bei M70 verlängert sich dann um die Spindelpositionierungsdauer. Die Position wird dabei mit der aktuellen Getriebestufe angefahren.

### Übergang in den Achsbetrieb ohne Programmierung von M70

Die Steuerung erkennt aufgrund der Programmierabfolge selbständig den Übergang in den Achsbetrieb (siehe "Impliziter Übergang in den Achsbetrieb (Seite 1411)") und erzeugt steuerungsintern den notwendigen M70-Ablauf inklusive des Getriebestufenwechsels.

### Übergang in den Achsbetrieb mit FC 18

Bei dem Übergang in den Achsbetrieb mit dem FC 18 ("Start Achse") wird der implizite Getriebestufenwechsel nicht unterstützt. Hier muss von der PLC-Applikation vor der Umschaltung in den Achsbetrieb die passende Getriebestufe eingelegt werden. Der Getriebestufenwechsel ist auch mit dem FC 18 möglich ("Start Getriebestufenwechsel").

### Wechsel vom Achsbetrieb in den Spindelbetrieb

Beim Wechsel vom Achsbetrieb in den Spindelbetrieb bleibt die mit M70 eingewechselte Getriebestufe weiterhin aktiv. Die im Spindelbetrieb vor M70 aktive Getriebestufe wird nicht automatisch wieder eingewechselt. Der Servo-Parametersatz wird von Parametersatz 1 (Index 0) auf den zu der Getriebestufe passenden Parametersatz 2 ... 6 (Index 1 ... 5) gewechselt (bei MD35590 \$MA\_PARAMSET\_CHANGE\_ENABLE < 2).

**Beispiel**

Beim Übergang der Spindel in den Achsbetrieb soll die Getriebestufe 4 eingewechselt werden.

Projektierung: MD35014

\$MA\_GEAR\_STEP\_USED\_IN\_AXISMODE[<Spindelbezeichner>] = 4

Programmcode	Kommentar
N05 M3 S1000	
N10 G1 X100 F1000	
N15 M70	; Getriebestufe 4 wird eingewechselt.
N20 POS[C]=77	
N25 ...	

**Hinweis**

MD35014 ist NewConfig-änderbar. Damit kann die einzuwechselnde Getriebestufe im Teileprogramm vor dem Übergang in den Achsbetrieb bei Bedarf noch verändert werden.

**15.4.10 Unterdrückung des Getriebestufenwechsels bei DryRun, Programmtest und SERUPRO**

**Funktion**

Bei Probelaufvorschub (DryRun), Programmtest und SERUPRO ist normalerweise kein Getriebestufenwechsel notwendig. Er kann deshalb für diese Funktionen unterdrückt werden. Die entsprechende Projektierung erfolgt mit den Bits 0 ... 2 im Maschinendatum:

MD35035 \$MA\_SPIND\_FUNCTION\_MASK

Probelaufvorschub (DryRun)	
Bit 0 = 0	Getriebestufen werden bei aktivierter Funktion DryRun für Teileprogrammsätze mit M40, M41 bis M45, Programmierung über FC18 und Synchronaktionen eingewechselt.
Bit 0 = 1	Getriebestufenwechsel werden bei aktivierter Funktion DryRun für Teileprogrammsätze mit M40, M41 bis M45, Programmierung über FC18 und Synchronaktionen unterdrückt.
Programmtest und SERUPRO	
Bit 1 = 0	Getriebestufen werden bei aktivierter Funktion Programmtest / SERUPRO für Teileprogrammsätze mit M40, M41 bis M45, Programmierung über FC18 und Synchronaktionen eingewechselt.
Bit 1 = 1	Getriebestufenwechsel werden bei aktivierter Funktion Programmtest / SERUPRO für Teileprogrammsätze mit M40, M41 bis M45, Programmierung über FC18 und Synchronaktionen unterdrückt.

DryRun, Programmtest und SERUPRO	
Bit 2 = 0	Getriebestufenwechsel für programmierte Getriebestufe wird nach Abwahl der Funktionen DryRun, Programmtest und SERUPRO bei REPOS <b>nicht</b> nachgeholt.
Bit 2 = 1	Getriebestufenwechsel für programmierte Getriebestufe wird nach Abwahl der Funktionen DryRun und SERUPRO wenn möglich nachgeholt.

## Ablauf

Wird ein Getriebestufenwechsel unterdrückt, so wird die programmierte Spindeldrehzahl gegebenenfalls vom Interpolator auf den zulässigen Drehzahlbereich der aktiven Getriebestufe begrenzt.

Die bei einer solchen Begrenzung erzeugten NC/PLC-Nahtstellensignale DB31, ... DBX83.2 (Soll Drehzahl erhöht) und DB31, ... DBX83.1 (Soll Drehzahl begrenzt) werden unterdrückt.

Überwachungen vom PLC-Programm sind während DryRun und im Probelaufvorschub überflüssig.

Bei der Unterdrückung des Getriebestufenwechsels wird keine neue Sollgetriebestufe (DB31,... DBX82.0-82.2) an die PLC ausgegeben.

Auch die GSW-Aufforderung DB31, ... DBX82.3 (Getriebe umschalten) wird unterdrückt.

Damit wird sichergestellt, dass keine GSW-Information vom PLC-Programm verarbeitet werden muss.

## Ermittlung der zuletzt aktiven Getriebestufe

Die Systemvariable \$P\_GEAR gibt die durch das Teileprogramm programmierte (und gegebenenfalls nicht an die PLC ausgegeben) Getriebestufe zurück.

Mit der Systemvariablen \$AC\_SGEAR kann die zuletzt aktive Getriebestufe vom Teileprogramm, Synchronaktionen und auf der Bedienoberfläche gelesen werden.

## Verhalten nach Abwahl

Die Funktion DryRun kann innerhalb eines laufenden Teileprogramms abgewählt werden. Nach dieser Abwahl muss die durch das Teileprogramm geforderte korrekte Getriebestufe ermittelt und eingewechselt werden.

Erst nach korrekter aktiver Getriebestufe ist die fehlerfreie Abarbeitung des restlichen Teileprogramms sichergestellt. Ein eventuell notwendiger Getriebestufenwechsel wird in dem bei der Abwahl gestarteten System-REPOS ausgeführt, wenn sich die Spindel im Drehzahlsteuerbetrieb befindet. Dabei findet ein vollständiger GSW-Dialog mit der PLC statt und die zuletzt programmierte Getriebestufe wird eingewechselt.

Stimmen bei REPOS die programmierte Getriebestufe des Teileprogramms und die durch die NC/PLC-Schnittstelle gelieferte aktuelle Getriebestufe überein, wird kein Getriebestufenwechsel ausgeführt.

Gleiches gilt für die Funktion SERUPRO.

Weitere Erläuterungen zum Satzsuchlauf SERUPRO siehe:

**Literatur:**

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

## Randbedingungen

Bei unterdrücktem Getriebestufenwechsel bewegt sich die ausgegebene Spindeldrehzahl innerhalb des durch die aktuelle Getriebestufe vorgegebenen Drehzahlbereichs.

Einschränkungen für das Nachholen eines GSW bei REPOS sind:

- Der GSW wird nicht nachgeholt, wenn die Spindel im Abwahl- bzw. Zielsatz eine Kommandospindel (Synchronaktion) oder PLC-Spindel (FC18) ist.
- Wenn die Getriebestufe nicht eingelegt werden kann, weil sich die Spindel im Positions- oder Achsbetrieb befindet oder eine Kopplung aktiv ist, wird der Alarm 22011"Kanal%1 Satz%3 Spindel%2% Wechsel in programmierte Getriebestufe nicht möglich" gemeldet.

## Beispiel

### Getriebestufenwechsel bei Probelaufvorschub (DryRun)

```

; 1. Getriebestufe (GS) für Ausgangszustand aktivieren
N00 M3 S1000 M41                ; 1.GS wird eingewechselt
M0                               ; Teileprogramm halt

; PI-Dienst: Probelaufvorschub (DryRun) aktivieren
                                ; (Projektierung)
N10 M42                          ; 2.GS gefordert, es findet kein GSW statt
N11 G0 X0 Y0 Z0                  ; Achsen positionieren
N12 M0                           ; Teileprogramm halt

; PI-Dienst: Probelaufvorschub (DryRun) deaktivieren
                                ; REORG und REPOS werden durchgeführt
                                ; jetzt findet GSW auf die 2. Getriebestufe
                                ; statt
N20 G1 Z100 F1000
...
N99 M30                          ; Teileprogramm Ende

```

## 15.5 Weitere projektierbare Anpassungen der Spindelfunktionalität

Spindelspezifische Funktionen werden eingestellt mit dem Maschinendatum:

MD35035 \$MA\_SPIND\_FUNCTION\_MASK (Spindel-Funktionen)

MD35035 ist bitcodiert:

Bit	Bedeutung	
0 ... 2	Getriebestufenwechselverhalten bei Probelaufvorschub (DryRun), Programmtest und SERUPRO Siehe "Unterdrückung des Getriebestufenwechsels bei DryRun, Programmtest und SERUPRO (Seite 1458)".	
4	Die programmierte Drehzahl s. . . einschließlich Drehzahlvorgaben über FC18 und Synchronaktionen kann in das Settingdatum SD43200 \$SA_SPIND_S (Drehzahl für Spindelstart über PLC-Schnittstelle) übernommen werden. Siehe "Spezielle Spindelbewegungen über PLC-Schnittstelle (Seite 1472)".	
5	Bei Bit 5 = 1 wirkt der Inhalt des Settingdatums SD43200 \$SA_SPIND_S als Solldrehzahl bei JOG. Über die JOG-Tasten kann die Spindel mit der Drehzahl aus SD43200 verfahren werden. Ist der Inhalt Null, dann werden andere JOG-Drehzahlvorgaben aktiv (siehe SD41200 \$SN_JOGSPIND_SET_VELO).	
8	Die programmierte Schnittgeschwindigkeit s. . . einschließlich Vorgaben über FC18 und Synchronaktionen kann in das Settingdatum SD43202 \$SA_SPIND_CONSTCUT_S (Schnittgeschwindigkeit für Spindelstart über PLC-Schnittstelle) übernommen werden. Siehe "Spezielle Spindelbewegungen über PLC-Schnittstelle (Seite 1472)".	
10	Für die Masterspindel kann der Wert der 15. G-Gruppe (Vorschubtyp) in das Settingdatum SD43206 \$SA_SPIND_SPEED_TYPE (Spindeldrehzahltyp für Spindelstart über PLC-Schnittstelle) übernommen werden. Siehe "Spezielle Spindelbewegungen über PLC-Schnittstelle (Seite 1472)".	
12	Bit 12 = 0	Spindeloverride wirkt <b>nicht</b> bei der Nullmarkensuche bei M19, SPOS bzw. SPOSA = 0.
	Bit 12 = 1	Spindeloverride wirkt bei der Nullmarkensuche bei M19, SPOS bzw. SPOSA = 0.

Bit	Bedeutung	
19	Bei der Programmierung von SPOS und SPOSA kann die Hilfsfunktion M19 implizit erzeugt und an die PLC ausgegeben werden. Siehe "Positionierbetrieb (Seite 1390)".	
20	Bei steuerungsintern erzeugter M70-Funktionalität kann die Hilfsfunktion M70 implizit erzeugt und an die PLC ausgegeben werden. Siehe "Impliziter Übergang in den Achsbetrieb (Seite 1411)".	
22	Bit 22 = 0	Das NC/PLC-Nahtstellensignal DB31, ... DBX17.6 (M3/M4 invertieren) wirkt auch auf die Funktion "Interpolatorisches Gewindebohren (G331/G332)". <b>Hinweis:</b> Bei dieser Einstellung muss auf folgendes geachtet werden: <ul style="list-style-type: none"> <li>• Bei Teileprogrammen mit G331 und G332 ist es notwendig, das NC/PLC-Nahtstellensignal DB31, ... DBX17.6 (M3/M4 invertieren) <b>vor Teileprogrammstart</b> auf einen stabilen Wert zu setzen.</li> <li>• Vorhandene applikative Lösungen, wie sie z. B. in Gewindebohr-Zyklen eingesetzt werden, müssen ggf. angepasst werden. Bekannt ist, dass z. B. in Gewindebohr-Zyklen die Invertierung der Spindeldrehrichtung durch MIRROR in Abhängigkeit eines Zyklen-spezifischen Settingdatums erreicht wird.</li> </ul>
	Bit 22 = 1	Das NC/PLC-Nahtstellensignal DB31, ... DBX17.6 (M3/M4 invertieren) wirkt <b>nicht</b> auf die Funktion "Interpolatorisches Gewindebohren (G331/G332)".

Änderungen von MD35035 werden nach NC-Reset wirksam.

## 15.6 Umschaltbare Spindeln

### Funktion

Mit der Funktion "Umschaltbare Spindeln" ist es möglich, Teileprogramme bezüglich der verwendeten Spindeln ("Kanalspindel, Logische Spindel") unabhängig von der aktuellen Zuordnung von projektierten Spindeln ("Physikalische Spindel") zu einem Kanal zu schreiben.

Die durch "Achstausch" eingewechselten oder abgegebenen physikalischen Spindeln müssen im Teileprogramm nicht mehr explizit angegeben werden.

### Spindelnummernumsetzer (SD42800)

Jede Spindel wird eindeutig durch eine projektierbare Nummer auf eine Maschinenachse abgebildet:

```
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX...] = <Spindelnummer>
```

Diese Nummer gilt für eine Spindel immer, wobei es gleichgültig ist, in welchem Kanal die Spindel aktiv behandelt wird.

Die Möglichkeit der Umschaltung der Kanalspindeln wird durch eine Zwischenebene zwischen den in dem Teileprogramm verwendeten logischen Spindelnummern und den im Kanal vorhandenen physikalischen Spindeln erreicht.

Dazu wird in einer aus Settingdaten bestehenden Tabelle jeder im Teileprogramm verwendeten logischen Spindel eine physikalische Spindel zugeordnet:

```
SD42800 $SC_SPIND_ASSIGN_TAB[<n>] (Spindelnummernumsetzer) = ...
```

Der Index <n> entspricht der programmierten Spindelnummer bzw. der programmierten Adresserweiterung. Der Inhalt des jeweiligen SD ist die physikalische, tatsächlich vorhandene Spindel.

Der Spindelnummernumsetzer wirkt bei Spindelprogrammierung durch:

- das Teileprogramm
- Synchronaktionen

Der Spindelnummernumsetzer wirkt nicht bei PLC-Vorgaben unter Verwendung des Funktionsbausteins FC18. Dort muss immer die betreffende physikalische Spindel im axialen Kontext angesprochen werden.

Die logischen Spindeln können durch Ändern von SD42800 umgeschaltet werden. Die Umschaltung kann aus dem Teileprogramm, von der PLC und/oder HMI erfolgen.

---

#### Hinweis

Im Settingdatum SD42800 \$SC\_SPIND\_ASSIGN\_TAB[0] ist die logische Masterspindel enthalten. Es dient nur zur Anzeige.

Dieses Settingdatum wird im Teileprogramm durch SETMS (logische Spindel) beschrieben.

Für nicht benutzte Spindeln ist in SD42800 der Wert 0 einzutragen.

Systemvariablen, die von der Spindelumsetzung betroffen sind:

\$P\_S, \$P\_SDIR, \$P\_SMODE, \$P\_GWPS, \$AC\_SDIR, \$AC\_SMODE, \$AC\_MSNUM, \$AA\_S

---

#### Literatur:

Programmierhandbuch Arbeitsvorbereitung

Als Adresserweiterung bei der Hilfsfunktionsausgabe wird immer die umgesetzte, physikalische Spindelnummer ausgegeben.

### Randbedingungen

- Die umschaltbaren Kanalspindeln ersetzen **nicht** die Funktion "Achstausch".
- Es lassen sich nur Spindeln umschalten, die dem Kanal durch Projektierung zugeordnet wurden.
- Werden Spindeln zur Umschaltung angegeben, die gerade in einem anderen Kanal aktiv sind, dann wird je nach Projektierungsvariante entweder die Funktion "Auto-Get" für die physikalische Spindel angestoßen oder es wird Alarm 16105 "Zugewiesene Spindel nicht vorhanden" gemeldet.
- Wird SD42800 \$SC\_SPIND\_ASSIGN\_TAB[<n>] von der PLC oder von HMI beschrieben, dann sollte sich der Kanal, dessen Tabelle geändert wird, im Resetzustand befinden bzw. die umzuschaltende Spindel sollte nicht in dem laufenden Teileprogramm verwendet werden.

Synchrones Verhalten kann über einen Vorlaufstopp STOPPRE erreicht werden.

- Die mehrfache Abbildung logischer auf physikalische Spindel wird in der NC nicht verriegelt. Bei der Anzeige logischer Spindel an der Bedienoberfläche ergeben sich jedoch entsprechend der Umsetztabelle Mehrdeutigkeiten.
- Der Spindelumsetzer wirkt auf Spindeln über FC18

### Aktivierung

Die Freigabe von SD42800 \$SC\_SPIND\_ASSIGN\_TAB[<n>] erfolgt durch Setzen der folgenden Maschinendatum-Einstellung:

MD20092 \$MC\_SPIND\_ASSIGN\_TAB\_ENABLE=1



## Grundstellung SD42800

Nach dem Einschalten der NC in Inbetriebnahmeschalterstellung 1 (SRAM löschen) befindet sich SD42800 \$SC\_SPIND\_ASSIGN\_TAB[<n>] in der Grundstellung.

Die Nummern der logischen und physikalischen Spindeln sind gleich.

SD42800 \$SC\_SPIND\_ASSIGN\_TAB[1] = 1

SD42800 \$SC\_SPIND\_ASSIGN\_TAB[2] = 2

SD42800 \$SC\_SPIND\_ASSIGN\_TAB[3] = 3

SD42800 \$SC\_SPIND\_ASSIGN\_TAB[4] = 4

SD42800 \$SC\_SPIND\_ASSIGN\_TAB[5] = 5

...

## Beispiel

Spindelprojektierungen:

- Zuordnung von Spindelnummer und Maschinenachse:

MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX [AX4] = 1

MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX [AX5] = 2

MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX [AX6] = 3

MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX [AX7] = 5

- Maschinenachse in Kanal aufnehmen:

MD20070 \$MC\_AXCONF\_MACHAX\_USED[0] = 4

MD20070 \$MC\_AXCONF\_MACHAX\_USED[1] = 5

MD20070 \$MC\_AXCONF\_MACHAX\_USED[2] = 6

MD20070 \$MC\_AXCONF\_MACHAX\_USED[3] = 7

- Festlegung der Masterspindel:

MD20090 \$MC\_SPIND\_DEF\_MASTER\_SPIND = 1

## Spindelnummerumsetzer:

```

MD20092 $MC_SPIND_ASSIGN_TAB_ENABLE=1          ; Spindelnummerumsetzer aktivieren

SD42800 $SC_SPIND_ASSIGN_TAB[0]=1              ; Masterspindel wie projiziert

SD42800 $SC_SPIND_ASSIGN_TAB[1]=1              ; Grundstellung der Tabelle

SD42800 $SC_SPIND_ASSIGN_TAB[2]=2
SD42800 $SC_SPIND_ASSIGN_TAB[3]=3
SD42800 $SC_SPIND_ASSIGN_TAB[4]=0              ; logische Spindel nicht zugewiesen

M3 S1000          ; Adresserweiterung=1, ausgegeben wird M1=3 S1=1000
                  Die Spindel mit der projizierten Nr."1" (Nr. der
                  physikalischen Masterspindel) dreht.
...
...
SD42800 $SC_SPIND_ASSIGN_TAB[1]=5              ; Zuweisung der logischen Spindel 1 zur
                  physikalischen Spindel 5
SD42800 $SC_SPIND_ASSIGN_TAB[2]=3              ; Zuweisung der logischen Spindel 2 zur
                  physikalischen Spindel 3.
Achtung: die physikalische Spindel 3
ist jetzt zweimal vergeben worden. Bei
Programmierung der logischen Spindel 2
und 3 wird immer die physikalische
Spindel 3 angesprochen. In den
Maschinengrundbildern drehen sich beide
Spindeln.

SETMS(2)          ; SD42800 $SC_SPIND_ASSIGN_TAB[0] = 2 wird intern von NCK
                  beschrieben.
...
M5                ; Masterspindel = Adresserweiterung=2, ausgegeben wird ungesetzte
                  Spindelnr. M3=5
                  Die physikalische Spindel mit der projizierten Nummer "3"
                  stoppt.
...
GET(S4)           ; Alarm 16105, da die logische Spindel "4" nicht umgesetzt werden
                  kann.
...
RELEASE(S1)       ; Kanalspindel "1" = physik. Spindel "5" wird freigegeben.
...
M30

```

## 15.7 Programmierung

### 15.7.1 Programmierung aus dem Teileprogramm

#### Programmieranweisungen

Anweisung	Beschreibung
SETMS:	Masterspindel ist die im folgenden Maschinendatum angegebene Spindel: MD20090 \$MC_SPIND_DEF_MASTER_SPIND (Löschstellung Masterspindel im Kanal)
SETMS (<n>):	Spindel mit der Nummer <n> ist die Masterspindel (kann abweichend von der Löschstellung sein: MD20090 \$MC_SPIND_DEF_MASTER_SPIND) Die Masterspindel muss für folgende Funktionen festgelegt werden: <ul style="list-style-type: none"> <li>• G95: Umdrehungsvorschub</li> <li>• G96 S.../G961 S...: konstante Schnittgeschwindigkeit in m/min bzw. ft/min</li> <li>• G97/G971: G96/G961 aufheben und letzte Spindeldrehzahl einfrieren</li> <li>• G63: Gewindebohren mit Ausgleichsfutter</li> <li>• G33/G34/G35: Gewindeschneiden</li> <li>• G331/G332: Gewindebohren ohne Ausgleichsfutter</li> <li>• G4 S...: Verweilzeit in Spindelumdrehungen</li> <li>• Programmierung von M3, M4, M5, S, SVC, SPOS, M19, SPOSA, M40, M41 bis M45 und WAITS ohne Angabe der Spindelnummer.</li> </ul> Die aktuelle Einstellung der Masterspindel kann über RESET/Teileprogrammende und Teileprogrammstart erhalten bleiben. Die Einstellung erfolgt über die Maschinendaten: <ul style="list-style-type: none"> <li>• MD20110 \$MC_RESET_MODE_MASK</li> <li>• MD20112 \$MC_START_MODE_MASK</li> </ul>
M3: M<n>=3:	Spindeldrehrichtung rechts für die Masterspindel Spindeldrehrichtung rechts für die Spindel mit der Nummer <n>
M4: M<n>=4:	Spindeldrehrichtung links für die Masterspindel Spindeldrehrichtung links für die Spindel mit der Nummer <n>
M5: M<n>=5:	Spindel Halt ohne Orientierung für die Masterspindel Spindel Halt ohne Orientierung für die Spindel mit der Nummer <n>
S...: S<n>=...:	Spindeldrehzahl in 1/min für die Masterspindel Spindeldrehzahl in 1/min für die Spindel mit der Nummer <n>
SVC=...: SVC [<n>]=...:	Schnittgeschwindigkeit in m/min bzw. ft/min für die Masterspindel Schnittgeschwindigkeit in m/min bzw. ft/min für die Spindel mit der Nummer <n>
SPOS=...: SPOS [<n>]=...:	Spindelpositionierung für die Masterspindel Spindelpositionierung für die Spindel mit der Nummer <n> Der Satzwechsel erfolgt erst, wenn die Spindel in Position ist.

Anweisung	Beschreibung
SPOSA=...: SPOSA [<n>]=...:	Spindelpositionierung für die Masterspindel Spindelpositionierung für die Spindel mit der Nummer <n> Der Satzwechsel erfolgt sofort. Die Spindel positioniert unabhängig von der weiteren Teileprogrammabarbeitung so lange, bis sie die Position erreicht hat.
SPOS=DC(...): SPOS [<n>]=DC(...): SPOSA=DC(...): SPOSA [<n>]=DC(...):	Die Bewegungsrichtung wird beim Positionieren aus der Bewegung beibehalten und die Position angefahren. Beim Positionieren aus dem Stillstand wird auf kürzestem Weg die Position angefahren.
SPOS=ACN(...): SPOS [<n>]=ACN(...): SPOSA=ACN(...): SPOSA [<n>]=ACN(...):	Die Position wird immer mit negativer Bewegungsrichtung angefahren. Gegebenfalls wird vor dem Positionieren die Bewegungsrichtung invertiert.
SPOS=ACP(...): SPOS [<n>]=ACP(...): SPOSA=ACP(...): SPOSA [<n>]=ACP(...):	Die Position wird immer mit positiver Bewegungsrichtung angefahren. Gegebenfalls wird vor dem Positionieren die Bewegungsrichtung invertiert.
SPOS=IC(...): SPOS [<n>]=IC(...): SPOSA=IC(...): SPOSA [<n>]=IC(...):	Es wird der Verfahrweg vorgegeben. Die Verfahrrichtung ergibt sich aus dem Vorzeichen des Verfahrwegs. Ist die Spindel bereits in Bewegung, dann wird gegebenenfalls die Verfahrrichtung invertiert, um in die programmierte Richtung verfahren zu können. Wird während der Verfahrbewegung die Nullmarke überfahren, dann wird die Spindel automatisch mit der Nullmarke synchronisiert, falls noch keine Referenz vorhanden ist oder diese per NST-Signal neu angefordert wurde.
M19: M [<n>]=19:	Spindelpositionierung für die Masterspindel auf die Position in SD43240 Spindelpositionierung für die Spindel mit der Nummer <n> auf die Position in SD43240 Der Satzwechsel erfolgt erst, wenn die Spindel in Position ist.
M70: M<n>=70:	Spindel stillsetzen und Lageregelung einschalten, Parametersatz Null anwählen, Achsbetrieb aktivieren für die Masterspindel für die Spindel mit der Nummer <n>
SPCON: SPCON (<n>): SPCON (<n>, <m>):	Spindellageregelung EIN für die Masterspindel für die Spindel mit der Nummer <n> für die Spindeln mit den Nummern <n> und <m>
PCOF: SPCOF (<n>): SPCOF (<n>, <m>):	Spindellageregelung AUS, Drehzahlsteuerbetrieb aktivieren für die Masterspindel für die Spindel mit der Nummer <n> für die Spindeln mit den Nummern <n> und <m>
FPRAON (S<n>):	Umdrehungsvorschub für Spindel <n> EIN, abgeleitet von der Masterspindel
FPRAON (S<n>, S<m>):	Umdrehungsvorschub für Spindel <n> EIN, abgeleitet von der Spindel <m> Der Umdrehungsvorschubwert muss mit FA [S<m>] vorgegeben werden.
FPRAOF (S<n>):	Umdrehungsvorschub für Spindel <n> AUS
C30 G90 G1 F3600	Rundachse C (Spindel im Achsbetrieb) fährt auf die Position 30 Grad mit einer Geschwindigkeit von 3600 Grad/min = 10 U/min

Anweisung	Beschreibung
G25 S...: G25 S<n>:	Programmierbare minimale Spindeldrehzahlbegrenzung für die Masterspindel für die Spindel mit der Nummer <n>
G26 S...: G26 S<n>:	Programmierbare maximale Spindeldrehzahlbegrenzung für die Masterspindel für die Spindel mit der Nummer <n>
LIMS=...: LIMS [<n>]=...:	Programmierbare maximale Spindeldrehzahlbegrenzung bei G96, G961, G97 für die Masterspindel für die Spindel mit der Nummer <n>
VELOLIM [<Spindel>]=...:	<p>Programmierbare Begrenzung der projizierten getriebestufenabhängigen Maximaldrehzahl</p> <p>Über Maschinendatum (MD30455 \$MA_MISC_FUNCTION_MASK, Bit 6) kann für die Programmierung im Teileprogramm eingestellt werden, ob VELOLIM unabhängig von der aktuellen Verwendung als Spindel oder Achse wirkt (Bit 6 = 1) oder getrennt für jede Betriebsart programmierbar sein soll (Bit 6 = 0). Ist eine getrennte Wirkung projiziert, dann wird die Auswahl über den Bezeichner bei der Programmierung getroffen:</p> <ul style="list-style-type: none"> <li>• Spindelbezeichner S&lt;n&gt; für Spindelbetriebsarten</li> <li>• Achsbezeichner, z. B. "C", für den Achsbetrieb</li> </ul> <p>Der Korrekturwert bezieht sich:</p> <ul style="list-style-type: none"> <li>• Bei Spindeln im Achsbetrieb (wenn MD30455 Bit 6 = 0): auf die projizierte maximale Achsgeschwindigkeit (MD32000 \$MA_MAX_AX_VELO).</li> <li>• Bei Spindeln im Spindel- oder Achsbetrieb (wenn MD30455 Bit 6 = 1): auf die Maximaldrehzahl der aktiven Getriebestufe (MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT [&lt;n&gt;])</li> </ul> <p>Weitere Erläuterungen zur Programmierung von VELOLIM siehe: <b>Literatur:</b> Programmierhandbuch Arbeitsvorbereitung</p>
WAITS:	<p>Synchronisationsbefehl für Masterspindel</p> <p>Mit der Abarbeitung nachfolgender Sätze wird solange gewartet, bis die in einem früheren NC-Satz mit SPOSA programmierte Spindel ihre Position erreicht hat (mit Genauhalt fein).</p> <p>WAITS nach M5: Warten, bis die Spindel steht.</p> <p>WAITS nach M3/M4: Warten, bis die Spindel ihre Solldrehzahl erreicht hat.</p>
WAITS (<n>): WAITS (<n>, <m>):	<p>Synchronisationsbefehl für die Spindel mit der Nummer &lt;n&gt;</p> <p>Synchronisationsbefehl für die Spindeln mit den Nummern &lt;n&gt; und &lt;m&gt;</p>
FA [S<n>]:	<p>Programmierung der Positioniergeschwindigkeit (axialer Vorschub) für die Spindel &lt;n&gt; in [Grad/min]</p> <p>Mit FA [S&lt;n&gt;] = 0 wird der projizierte Wert wieder gültig: MD35300 \$MA_SPIND_POSCTRL_VELO</p>
OVRA [S<n>]:	Programmierung des axialen Override-Werts für die Spindel <n> in [%]
ACC [S<n>]:	Programmierung des axialen Beschleunigungsvermögens der Spindel <n> in [%]

Anweisung	Beschreibung
SPI (<n>):	<p>Mit SPI (&lt;n&gt;) wird eine Spindelnummer entsprechend dem Maschinendatum: MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[ ] in den Datentyp AXIS konvertiert.</p> <p>SPI wird verwendet, wenn mit der Spindel Achsfunktionen programmiert werden sollen. Folgende Anweisungen sind mit SPI möglich:</p> <ul style="list-style-type: none"> <li>• Frameanweisungen: <ul style="list-style-type: none"> <li>- CTRANS ( )</li> <li>- CFINE ( )</li> <li>- CMIRROR ( )</li> <li>- CSCALE ( )</li> </ul> </li> <li>• Geschwindigkeits- und Beschleunigungswerte für Folgespindeln: <ul style="list-style-type: none"> <li>- FA [SPI (&lt;n&gt;)]</li> <li>- ACC [SPI (&lt;n&gt;)]</li> <li>- OVRA [SPI (&lt;n&gt;)]</li> </ul> </li> <li>• Systemvariablen: <ul style="list-style-type: none"> <li>- \$P_PFRAME [SPI (&lt;n&gt;), TR] =&lt;Wert&gt;</li> <li>- \$P_PFRAME= <p style="margin-left: 20px;">CTrans (X, &lt;Achswert&gt;, Y, &lt;Achswert&gt;, SPI (&lt;n&gt;), &lt;Achswert&gt;)</p> </li> <li>- \$P_PFRAME= <p style="margin-left: 20px;">CSCALE (X, &lt;Maßstab&gt;, Y, &lt;Maßstab&gt;, SPI (&lt;n&gt;), &lt;Maßstab&gt;)</p> </li> <li>- \$P_PFRAME=CMIRROR (S&lt;n&gt;, Y, Z)</li> <li>- \$P_UBFR=CTrans (A, 10) : CFINE (19, 0.1)</li> </ul> </li> </ul> <p>Weitere Erläuterungen zur Programmierung von SPI siehe:  <b>Literatur:</b>  Programmierhandbuch Arbeitsvorbereitung</p>
M40: M<n>=40:	<p>Automatische Getriebestufenauswahl für die Masterspindel</p> <p>Automatische Getriebestufenauswahl für die Spindel mit der Nummer &lt;n&gt;</p>
M41 bis M45: M<n>=41 bis M<n>=45:	<p>Getriebestufe 1 bis 5 auswählen für die Masterspindel</p> <p>Getriebestufe 1 bis 5 auswählen für die Spindel mit der Nummer &lt;n&gt;</p>

### Hinweis

Die M-Funktionen M3, M4, M5 und M70 werden nicht im DB21, ... DBB194 und DBB202 ausgegeben, wenn in einem Kanal eine Spindel konfiguriert ist. Diese M-Funktionen werden als erweiterte M-Funktionen im DB21, ... DBB68 ff. und im jeweiligen Achs-DB, DB31, ... DBB86 ff., angeboten.

### Literatur

Genauere Erläuterungen zur Programmierung der Spindel siehe:

- Programmierhandbuch Grundlagen

## 15.7.2 Programmierung über Synchronaktionen

Die M-Funktionen M40 bis M45 sind auch in Synchronaktionen programmierbar.

Dabei ist Folgendes zu beachten:

- Die Programmierung von M40 ... M45 im Teileprogramm hat keine Auswirkungen auf den aktuellen Zustand des automatischen Getriebestufenwechsels der Synchronaktionen und umgekehrt.

- Der automatische Getriebestufenwechsel wirkt bei S-Wert-Programmierung mit M40 für Synchronaktionen und Teileprogramm getrennt.

- Nach POWER ON ist M40 ausgeschaltet.

Es wird keine Getriebestufenanpassung bei einer S-Wert-Vorgabe aus Synchronaktionen vorgenommen.

- Ein durch Synchronaktionen programmiertes M40 bleibt für Synchronaktionen immer aktiv (modal wirksam) und wird bei Reset nicht zurückgesetzt.

- M41 ... M45 wählt entsprechend der Programmierung im Teileprogramm die erste bis fünfte Getriebestufe an.

Zur Ausführung der Funktion wird ein Achstausch notwendig.

Nach Ausführung des Getriebestufenwechsels befindet sich die Spindel im neutralen Zustand (analoges Verhalten zur M3-, M4-, M5-Programmierung).

### Literatur

Weitere Erläuterungen zur Programmierung der Spindel sowie Spindelbewegungen aus Synchronaktionen siehe:

- Programmierhandbuch Arbeitsvorbereitung
- Funktionshandbuch Synchronaktionen

## 15.7.3 Programmierung von Spindelsteuerungen über PLC mit FC18 - nur 840D sl

Bei der Vorgabe von Drehrichtung und Drehzahl durch die PLC mittels FC18 kann die zur Drehzahl passende Getriebestufe vom NCK ermittelt und eingelegt werden. Dies entspricht der M40-Funktionalität bei der Programmierung über das Teileprogramm.

Zur Aktivierung der Getriebestufenauswahl muss beim Aufruf des FC18 bei PLC-Anwenderprogrammen der entsprechende Startcode gesetzt werden.

### Literatur

Genauere Erläuterungen zur Programmierung von Spindelsteuerungen über PLC mit FC18 siehe:

- Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

## 15.7.4 Spezielle Spindelbewegungen über PLC-Schnittstelle

---

### Hinweis

Die Funktion steht nur bei Verwendung von SINUMERIK Operate zur Verfügung!

---

### Warum eine spezielle Spindelschnittstelle?

Die Funktion bietet die Möglichkeit, über eine axiale PLC-Schnittstelle die Spindel alternativ zum FC18 zu programmieren. Der Einfachheit der Vorgaben steht eine etwas verringerte Funktionalität gegenüber. Diese Funktionalität kann bevorzugt für einfache Steuerungs-Applikationen eingesetzt werden.

### Funktionalität

Über die interne DBB30-Spindelschnittstelle können Spindeln außerhalb eines laufenden Teileprogramms gestartet und gestoppt werden:

- DB31, ... DBX30.0 (Spindel-Stopp)
- DB31, ... DBX30.1 (Spindel-Start Rechtslauf)
- DB31, ... DBX30.2 (Spindel-Start Linkslauf)
- DB31, ... DBX30.3 (Getriebestufe auswählen)

Zusatzsignal zum Signal "Spindel-Start Rechts-/Linkslauf"; ermittelt zur Drehzahl die passende Getriebestufe analog zu M40 im Teileprogramm.

- DB31, ... DBX30.4 (Spindel-Start Positionieren)

Um einen Spindelauftrag starten zu können, muss sich der spindelbehandelnde Kanal im Akzeptanzzustand befinden. Ein Spindelauftrag wird immer mit der Low-High-Flanke eines internen DBB30-Signals gestartet.

Prinzipiell sind die internen DBB30-Startsignale im statischen Zustand bedeutungslos und verhindern nicht die Programmierung der Spindel durch FC18, Synchronaktionen, Teileprogramm oder JOG-Verfahrbewegungen (z. B. wenn das Stopp-Signal statisch auf "1" steht).

### Voraussetzungen

Damit Spindelaufträge über die DBB30-Spindelschnittstelle akzeptiert werden, müssen folgende Bedingungen erfüllt sein (**Akzeptanzzustand**):

- Der Kanalzustand muss sich im Modus "unterbrochen" oder "Reset" befinden:
  - DB21, ... DBX35.6 = 1 (Kanalzustand "unterbrochen")
  - DB21, ... DBX35.7 = 1 (Kanalzustand "Reset")
- Der Programmzustand muss sich im Modus "unterbrochen" oder "abgebrochen" befinden:
  - DB21, ... DBX35.3 = 1 (Programmzustand "unterbrochen")
  - DB21, ... DBX35.4 = 1 (Programmzustand "abgebrochen")



Diese Zustände werden bei Reset und in der Betriebsart JOG eingenommen.

Die betreffende Spindel muss weitere Bedingungen zum Startzeitpunkt aufweisen:

- Sie muss den Zustand "Kanalachse" oder "neutrale Achse" besitzen und darf nicht durch JOG-Tasten bewegt werden.
- Bei der Vorgabe darf keine Positionierbewegung durch FC18 oder Synchronaktionen ausgeführt werden.

---

#### **Hinweis**

##### **Spindelauftrag außerhalb vom Akzeptanzbereich**

Low-High-Flanken außerhalb des Akzeptanzbereichs werden ignoriert. Es erfolgt keine Alarmmeldung durch den NCK. Es ist denkbar, dass durch eine anwenderseitige PLC-Applikation der Akzeptanzbereich dem Bediener bekannt gegeben wird.

Weiterhin können Spindelaufträge außerhalb des Akzeptanzbereichs unter Verwendung der Funktionen FC18 oder ASUP vorgenommen werden.

---

## **Mehrkanaligkeit**

Bei Mehrkanaligkeit wird die von der PLC gestartete Spindel in dem Kanal aktiv, der diese zu dem Startzeitpunkt behandelt.

Dieser Kanal kann PLC-seitig ermittelt werden durch Lesen der NC/PLC-Nahtstellensignale:  
DB31, ... DBX68.0-68.3 (NC-Achse/Spindel in Kanal A bis D)

## Spindelvorgaben

Die Spindelvorgaben bleiben über einen Betriebsarten-Wechsel (z. B. von der Betriebsart JOG nach AUTOMATIK) hinaus erhalten. Mit dem Start eines Teileprogramms werden die Spindelvorgaben in das Teileprogramm übernommen und können durch Teileprogrammanweisungen wieder verändert werden.

Durch folgende Einstellungen im Maschinendatum:  
MD35035 \$MA\_SPIND\_FUNCTION\_MASK (Spindelfunktionen)  
können bestimmte Spindelvorgaben (Drehzahl oder Schnittgeschwindigkeit, Vorschubtyp) aus dem Teileprogramm, Synchronaktionen und FC18 in die entsprechenden Settingdaten übernommen werden:

Bit 4 = 1	<p>Die programmierte <b>Drehzahl</b> s. . . einschließlich Drehzahlvorgaben über FC18 und Synchronaktionen wird in das folgende Settingdatum übernommen: SD43200 \$SA_SPIND_S (Drehzahl für Spindelstart über PLC-Schnittstelle) S-Programmierungen, die keine Drehzahlprogrammierungen sind, werden nicht in das SD übernommen. Dazu gehören z. B.:</p> <ul style="list-style-type: none"> <li>• S-Wert bei konstanter Schnittgeschwindigkeit (G96, G961)</li> <li>• S-Wert bei umdrehungsbezogener Verweilzeit (G4)</li> </ul>
Bit 8 = 1	<p>Die programmierte <b>Schnittgeschwindigkeit</b> s. . . einschließlich Vorgaben über FC18 und Synchronaktionen wird in das folgende Settingdatum übernommen: SD43202 \$SA_SPIND_CONSTCUT_S (Schnittgeschwindigkeit für Spindelstart über PLC-Schnittstelle) S-Programmierungen, die keine Schnittgeschwindigkeitsprogrammierungen sind, werden nicht in das SD übernommen. Dazu gehören z. B.:</p> <ul style="list-style-type: none"> <li>• S-Wert außerhalb konstanter Schnittgeschwindigkeit (G96, G961, G962)</li> <li>• S-Wert bei umdrehungsbezogener Verweilzeit (G4)</li> </ul>
Bit 10 = 1	<p>Für die Masterspindel wird der Wert der 15. G-Gruppe (<b>Vorschubtyp</b>) in das folgende Settingdatum SD43206 \$SA_SPIND_SPEED_TYPE übernommen: SD43206 \$SA_SPIND_SPEED_TYPE (Spindeldrehzahltyp für Spindelstart über PLC-Schnittstelle) Für alle anderen Spindeln bleibt der Wert in SD43206 unverändert.</p>

## Drehzahlvorgabe

Drehzahlvorgaben aus Teileprogramm, FC18 oder Synchronaktionen werden aus allen üblichen Quellen in das folgende Settingdatum geschrieben:

SD43200 \$SA\_SPIND\_S (Drehzahl für Spindelstart über PLC-Schnittstelle)

Das Settingdatum kann wie folgt beschrieben werden:

- Durch Drehzahlprogrammierung
- Durch Direktbeschreiben im Teileprogramm
- Durch HMI-Software

---

### Hinweis

Das direkte Beschreiben des Settingdatums wird sofort wirksam und wirkt asynchron zum Teileprogrammablauf.

---

Für das Beschreiben gelten folgende Bedingungen:

Beschreiben durch:	Bedingungen:
Drehzahlprogrammierung	<ul style="list-style-type: none"> <li>• MD35035 \$MA_SPIND_FUNCTION_MASK Bit 4 = 1 muss gesetzt sein.</li> <li>• Die konstante Schnittgeschwindigkeit G96, G961 darf nicht aktiv sein.</li> <li>• Das folgende NC/PLC-Nahtstellensignal muss gesetzt sein: DB31, ... DBX 84.0 = 0 (Konstante Schnittgeschwindigkeit)</li> </ul>
Direktes Beschreiben im Teileprogramm	Ein programmierter S-Wert und der Wert des direkt beschriebenen SD können sich zeitlich überholen. Dann sollte nach dem Beschreiben des SD mit der Anweisung <code>STOPRE</code> gearbeitet werden.
HMI	Nur positive Werte einschließlich Null werden akzeptiert. Anderenfalls wird eine entsprechende Meldung erzeugt.

### GSW und Einfluss auf Drehzahl

Gegenwärtig wird kein Getriebestufenwechsel (GSW) ausgelöst, wenn die vorgegebene Drehzahl außerhalb des Drehzahlbereichs der Getriebestufe liegt (Ausnahmen siehe "M40: Automatische Getriebestufenauswahl bei Drehzahlen außerhalb projektierter Schaltschwellen (Seite 1489)"). Es wirken die üblichen Drehzahlbegrenzungen und die Drehzahlanhebung auf die vorgegebene Drehzahl.

### Solldrehzahl bei JOG

Bei folgender MD-Projektierung:

MD35035 \$MA\_SPIND\_FUNCTION\_MASK Bit 5 =1

wirkt der Inhalt aus SD43200 \$SA\_SPIND\_S als Solldrehzahl bei JOG.

Über die JOG-Tasten kann die Spindel mit der Drehzahl aus SD43200 verfahren werden.

Ist der Inhalt Null, dann werden andere JOG-Drehzahlvorgaben aktiv (siehe SD41200 \$SN\_JOGSPIND\_SET\_VELO).

### Vorgabe der konstanten Schnittgeschwindigkeit

Vorgaben der konstanten Schnittgeschwindigkeit aus Teileprogramm, FC18 oder Synchronaktionen werden aus allen üblichen Quellen in das folgende Settingdatum geschrieben:

SD43202 \$SA\_SPIND\_CONSTCUT\_S (Schnittgeschwindigkeit für Spindelstart über PLC-Schnittstelle)

#### Voraussetzungen

Voraussetzung für das Wirken der Vorgabe einer konstanten Schnittgeschwindigkeit ist:

- Die betreffende Spindel muss Masterspindel im spindelbehandelnden Kanal sein.

Diese Bedingung ist erfüllt, wenn das folgende NC/PLC-Nahtstellensignal gesetzt ist:

DB31, ... DBX84.0 = 1 (Konstante Schnittgeschwindigkeit aktiv)

#### Beschreiben aus Teileprogramm

Der Wert der konstanten Schnittgeschwindigkeit wird beim Beschreiben aus dem Teileprogramm wie folgt interpretiert:

- Wenn in der 12. G-Gruppe G710 aktiv ist: metrisch
- Wenn in der 12. G-Gruppe G700 eingestellt ist: inch als [feet/min]

Bei G70, G71 und beim Beschreiben von extern (HMI) entscheidet die Einstellung im Maschinendatum:

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC  
über die Interpretation des geschriebenen Werts.

Für weitere Erläuterungen zum Maßsystem (Metrisch / Inch) siehe Kapitel "G2: Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (Seite 331)".

#### Vorgabe über FC18

Bei der Vorgabe der konstanten Schnittgeschwindigkeit über FC18 wird die Interpretation des Drehzahlwerts (Byte 8 ... 11) durch Setzen des Bit 6 im Byte 2 vom Bereich "Signale an konkurrierende Positionierachsen" entschieden.

#### Vorgabe über Synchronaktionen

Bei der Vorgabe über Synchronaktionen entscheidet analog zum Teileprogramm der Vorschubtyp über die Interpretation des S-Werts.

#### Lesen aus Teileprogramm und Synchronaktionen

Der programmierte Schnittgeschwindigkeitswert kann im Teileprogramm und in Synchronaktionen durch Lesen folgender Systemvariablen ermittelt werden:

- \$P\_CONSTCUT\_S[<n>] (zuletzt programmierte konstante Schnittgeschwindigkeit)
- \$AC\_CONSTCUT\_S[<n>] (aktuelle konstante Schnittgeschwindigkeit)

Festgelegter Wertebereich der beiden neuen Systemvariablen: WB = {0, DBL\_Max}

Auch das Lesen des programmierten Schnittgeschwindigkeitswerts über die BTSS-Schnittstelle ist möglich.

### Vorgabe des Spindeldrehzahltyps für die Masterspindel

Vorgaben des Spindeldrehzahltyps für die **Masterspindel** aus Teileprogramm, FC18 oder Synchronaktionen werden aus allen üblichen Quellen in das folgende Settingdatum geschrieben:

SD43206 \$SA\_SPIND\_SPEED\_TYPE (Spindeldrehzahltyp für Spindelstart über PLC-Schnittstelle)

Der Wertebereich und die Funktionalität entsprechen der 15. G-Gruppe (Vorschubtyp).

Zulässige Werte sind die G-Werte: 93, 94, 95, 96, 961, 97, und 971.

Je nach Einstellung wird bei DB31, ... DBX30.1/2 (Spindel-Start Rechts-/Linkslauf) entweder die Drehzahl aus SD43200 \$SA\_SPIND\_S oder die Schnittgeschwindigkeit aus SD43202 \$SA\_SPIND\_CONSTCUT\_S wirksam:

93, 94, 95, 97 und 971:	Die Masterspindel wird mit der Drehzahl aus SD43200 gestartet.
96 und 961:	Die Drehzahl der Masterspindel ergibt sich aus der vorgegebenen Schnittgeschwindigkeit (SD43202) und dem Radius der Planachse.

### Vorgaben für das Spindelpositionieren

Die Vorgaben für das Spindelpositionieren durch DB31, ... DBX30.4 (Spindel-Start Positionieren) werden aus folgenden Settingdaten gelesen:

SD43240 \$SA\_M19\_SPOS (Spindelposition für Spindelpositionieren mit M19)

SD43250 \$SA\_M19\_SPOSMODE (Spindelpositionanfahrmodus für Spindelpositionieren mit M19)

## 15.7.5 Programmierung von extern (PLC, HMI)

### SD43300 und SD42600

Über das axiale Settingdatum:

SD43300 \$SA\_ASSIGN\_FEED\_PER\_REV\_SOURCE (Umdrehungsvorschub für Spindeln)  
kann in der Betriebsart JOG über das kanalspezifische Settingdatum:

SD42600 \$SC\_JOG\_FEED\_PER\_REV\_SOURCE (Steuerung Umdrehungsvorschub in JOG)  
das Verfahren mit Umdrehungsvorschub von extern ausgewählt werden.

Über das Settingdatum sind folgende Einstellungen möglich:

>0:	die Maschinenachsnnummer der Rundachse/Spindel, von der Umdrehungsvorschub abgeleitet werden soll.
-1:	der Umdrehungsvorschub wird abgeleitet von der Masterspindel des Kanals, in dem die Achse/Spindel jeweils aktiv ist.
0:	Funktion ist abgewählt.

### FPRAON (S2)

Umdrehungsvorschub für Spindel S2 ein, abgeleitet von der Masterspindel.

### FPRAON (S2, A)

Umdrehungsvorschub für Spindel S2 ein, abgeleitet von Achse A.

Der Umdrehungsvorschubwert muss mit  $FA[S_n]$  vorgegeben werden.

### FPRAOF (S2)

Umdrehungsvorschub für Spindel S2 aus.

### SPI(n)

Es ist auch die Programmierung von  $SPI(n)$  anstelle von  $SPI(S_n)$  zulässig.

## 15.8 Spindelüberwachungen

### 15.8.1 Zulässige Drehzahlbereiche

Aus den parametrisierten bzw. programmierten Drehzahlgrenzwerten sowie der aktiven Spindelfunktion (G94, G95, G96, G961, G97, G971, G33, G34, G35, G331, G332, etc.) ergibt sich der zulässige Drehzahlbereich einer Spindel.

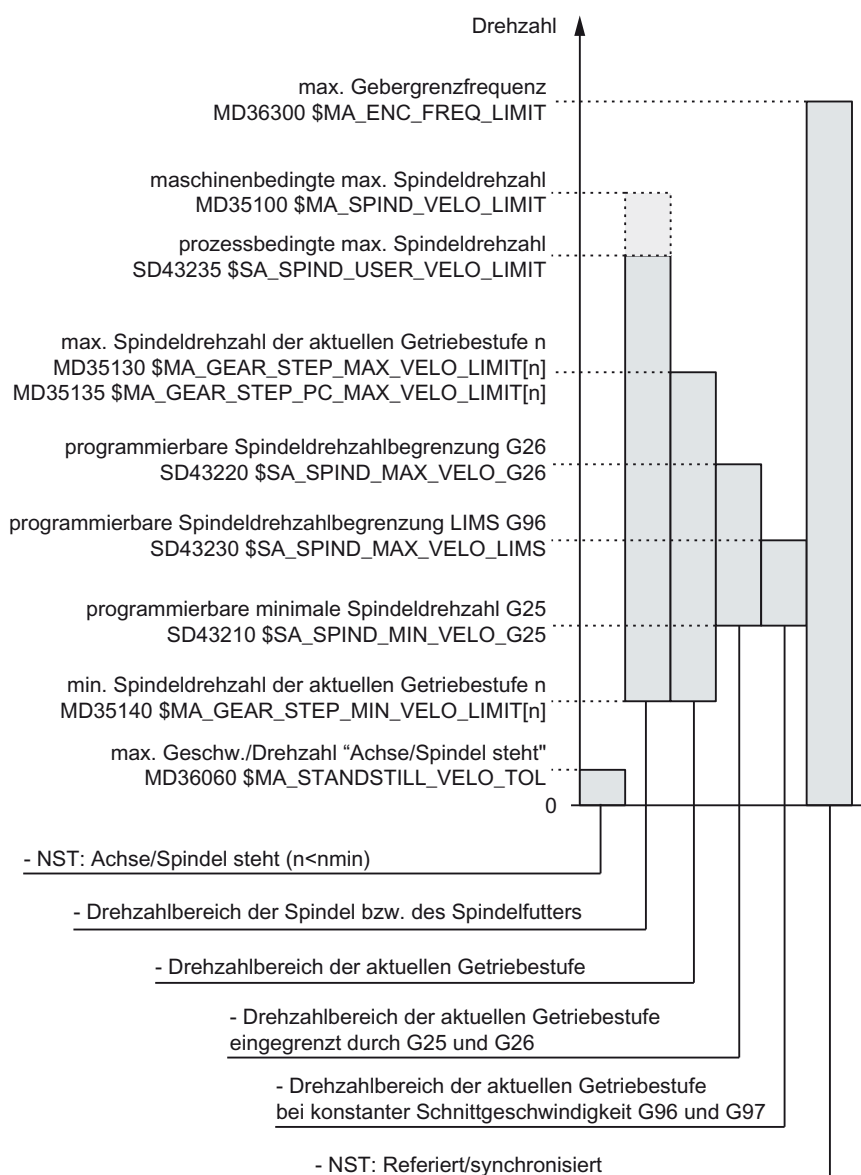


Bild 15-8 Bereiche der Spindelüberwachungen / Drehzahlen

## 15.8.2 Achse/Spindel steht

Erst wenn die Spindel steht, sind an der Maschine Funktionen wie Werkzeugwechsel, Maschinentüre öffnen, Bahnvorschub freigeben, etc. möglich.

### Funktion

Der Zustand "Achse/Spindel steht" wird erreicht, wenn keine Sollwerte mehr erzeugt werden und die Spindel-Istdrehzahl den projektierten Schwellwert für "Achse/Spindel steht" unterschreitet:

MD36060 \$MA\_STANDSTILL\_VELO\_TOL (max. Geschwindigkeit/Drehzahl "Achse/Spindel steht")

Ist die Spindel zum Stillstand gekommen, wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX61.4 (Achse/Spindel steht)

### Wirksamkeit

Die Überwachung des Spindelstillstandes wirkt in allen Spindelbetriebsarten und im Achsbetrieb.

### Bahnvorschub sperren

Wird eine Spindel im Steuerbetrieb gestoppt (M5), dann wird der Bahnvorschub gesperrt, wenn das folgende Maschinendatum gesetzt ist:

MD35510 \$MA\_SPIND\_STOPPED\_AT\_IPO\_START (Vorschubfreigabe bei Spindel steht)

Ist die Spindel zum Stillstand gekommen, wird der Bahnvorschub wieder freigegeben.

## 15.8.3 Spindel im Sollbereich

### Funktion

Die Spindelüberwachung "Spindel im Sollbereich" kontrolliert, ob:

- die programmierte Spindeldrehzahl erreicht ist.
- die Spindel steht:  
DB31, ... DBX61.4 (Achse/Spindel steht) = 1
- sich die Spindel noch in der Beschleunigungs- oder Bremsphase befindet.



In der Spindelbetriebsart Steuerbetrieb wird die Soll Drehzahl mit der Ist Drehzahl verglichen. Weicht die Ist Drehzahl um mehr als die über MD vorgebbare Spindeldrehzahltoleranz (s. u.) ab, wird:

- das folgende axiale NC/PLC-Nahtstellensignal auf "0" gesetzt:  
DB31, ... DBX83.5 (Spindel im Sollbereich) = 0
- der nächste Bearbeitungssatz nicht freigegeben (abhängig von der Einstellung in MD35500 \$MA\_SPIND\_ON\_SPEED\_AT\_IPO\_START, siehe "Achse/Spindel steht (Seite 1480)").

### Spindelsolldrehzahl

Die Spindelsolldrehzahl ergibt sich aus der programmierten Drehzahl unter Berücksichtigung der Spindelkorrektur und der aktiven Begrenzungen.

Eine Begrenzung oder Anhebung der programmierten Drehzahl wird durch DB31, ... DBX83.1 (Soll-Drehzahl begrenzt) oder DB31, ... DBX83.2 (Soll-Drehzahl erhöht) angezeigt (siehe auch "Minimale/Maximale Drehzahl der Getriebestufe (Seite 1482)"). Das Erreichen des Toleranzbereichs der Solldrehzahl wird dadurch **nicht** verhindert.

### Toleranzbereich der Solldrehzahl

Der Toleranzbereich der Solldrehzahl wird festgelegt durch den Spindeldrehzahl-Toleranzfaktor:

MD35150 \$MA\_SPIND\_DES\_VELO\_TOL

Beispiel:

MD35150 \$MA\_SPIND\_DES\_VELO\_TOL = 0,1

⇒ Die Spindel-Ist Drehzahl darf  $\pm 10\%$  von der Solldrehzahl abweichen.

Liegt die Spindel-Ist Drehzahl innerhalb des Toleranzbereichs, wird das folgende NC/PLC-Nahtstellensignal auf "1" gesetzt:

DB31, ... DBX83.5 (Spindel im Sollbereich) = 1

#### Sonderfall:

Wird die Spindeldrehzahltoleranz auf "0" gesetzt, dann wird DB31, ... DBX83.5 (Spindel im Sollbereich) permanent auf "1" gesetzt und es findet keine Bahnbeeinflussung statt.

### Drehzahländerung

Die Bahnbeeinflussung findet nur zu Beginn des Verfahrsatzes und nur bei programmierter Drehzahländerung statt. Wird der Drehzahltoleranzbereich z. B. durch Überlastung verlassen, dann führt dies nicht automatisch zum Stillstand der Bahnbewegung.

## 15.8.4 Minimale/Maximale Drehzahl der Getriebestufe

### Minimale Drehzahl

Die minimale Drehzahl der Getriebestufe einer Spindel wird projiziert im Maschinendatum:

MD35140 \$MA\_GEAR\_STEP\_MIN\_VELO\_LIMIT[<n>]

Die unter Berücksichtigung des Overrides erzeugten Drehzahlsollwerte unterschreiten die minimale Drehzahl nicht.

Wird ein S-Wert programmiert, der kleiner als die minimale Drehzahl ist, so wird die Soll-drehzahl auf die Minimaldrehzahl angehoben und das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.2 (Soll-Drehzahl erhöht)

Die minimale Drehzahl der Getriebestufe wirkt nur im Drehzahlbetrieb und kann nur unterschritten werden durch:

- Spindelkorrektur 0%
- M5
- S0
- DB31, ... DBX4.3 (Spindel Halt)
- DB31, ... DBX2.1 (Reglerfreigabe) wegnehmen
- DB21, ... DBX7.7 (Reset)
- DB31, ... DBX2.2 (Restweglöschen/Spindel-Reset)
- DB31, ... DBX18.5 (Pendeldrehzahl)
- DB21, ... DBX7.4 (NC-Stop Achsen plus Spindeln)
- DB31, ... DBX1.3 (Achs-/Spindelsperre)
- DB31, ... DBX16.7 (S-Wert löschen)

### Maximale Drehzahl

Die maximale Drehzahl der Getriebestufe einer Spindel wird projiziert im Maschinendatum:

MD35130 \$MA\_GEAR\_STEP\_MAX\_VELO\_LIMIT[<n>]

Die unter Berücksichtigung des Overrides erzeugten Drehzahlsollwerte werden auf diese Drehzahl begrenzt.

Im Falle der Begrenzung der Drehzahl wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.1 (Soll-Drehzahl begrenzt)

## 15.8.5 Diagnose von Spindeldrehzahlbegrenzungen

### Funktion

Die Begrenzung bzw. Anhebung der Spindeldrehzahl wird durch die Ausgabe folgender NC/PLC-Nahtstellensignale gemeldet:

- DB31, ... DBX83.1 (Soll-Drehzahl begrenzt)
- DB31, ... DBX83.2 (Soll-Drehzahl erhöht)

Zur Diagnose der wirksamen/begrenzenden Spindelparameter kann auf die wichtigsten Einflussgrößen der Spindelbewegung lesend über Systemvariablen zugegriffen werden. Die Systemvariablen werden mit der Spindelnummer indexiert und geben nur Werte, die im Drehzahlsteuer- und Spindelpositionierbetrieb relevant sind, zurück.

Folgende Systemvariablen sind für den Spindelbetrieb verfügbar:

Systemvariable	Bedeutung
\$AC_SMAXVELO[<n>]	Maximal mögliche Spindeldrehzahl [U/min], resultierend aus den aktiven Begrenzungsdaten.
\$AC_SMAXVELO_INFO[<n>]	Angabe des drehzahlbegrenzenden Datums als Zahlenwert. *)
\$AC_SMINVELO[<n>]	Minimal mögliche Spindeldrehzahl [U/min], entspricht der Mindestdrehzahl im Drehzahlsteuerbetrieb.
\$AC_SMINVELO_INFO[<n>]	Angabe des drehzalanhebenden Datums als Zahlenwert. *)
\$AC_SMAXACC[<n>]	Beschleunigungswert der Spindel [U/s <sup>2</sup> ].
\$AC_SMAXACC_INFO[<n>]	Beschleunigungsbegrenzende Ursache in Form eines Zahlenwertes. *)
\$AC_SPIND_STATE[<n>]	Statusbits der Spindel.
<n>: Spindelnummer (n= 0: die Variablen beziehen sich auf die aktuelle Masterspindel)	

\*) Der Text zum Zahlenwert ist der Beschreibung zur Systemvariablen zu entnehmen.

Auswertung der Diagnosedaten:

Im **NCK** können die Systemvariablen für jede Spindel über Synchronaktionen und im Teileprogramm unter Berücksichtigung des Vorlaufstopps gelesen werden.

## Randbedingungen

Die von den Systemvariablen gelieferten Werte sind von der Spindelbetriebsart abhängig:

- Drehzahlsteuerbetrieb:

Alle Systemvariablen liefern aktuelle Werte.

- Positionierbetrieb:

Die Systemvariablen \$AC\_SMAXVELO, \$AC\_SMAXACC und \$AC\_SPIND\_STATE liefern gültige Werte. Die Systemvariablen \$AC\_SMINVELO und \$AC\_SMINVELO\_INFO liefern die Daten, die beim Wechsel in den Drehzahlsteuerbetrieb wirksam werden.

- Achsbetrieb (z. B. wenn die Spindel von einer Transformation TRANSMIT, TRACYL,... verwendet wird oder als Zusatzachse einer Bahnbewegung folgt):

Die Systemvariable \$AC\_SPIND\_STATE kann auch im Achsbetrieb verwendet werden. Für die Dynamikdaten stehen im Achsbetrieb eigene Systemvariable zur Verfügung:

\$AA\_VMAXM, \$AA\_VMAXB und \$AA\_VLFCT.

Bei Satzsuchlauf Typ SERUPRO ergibt sich folgendes Steuerungsverhalten:

- Die Systemvariable \$AC\_SMAXVELO / \$AC\_SMAXACC liefert die maximal darstellbare Geschwindigkeit / Beschleunigung.
- \$AC\_SMAXVELO\_INFO und \$AC\_SMAXACC\_INFO liefern den WERT "0" (keine Limitierungen aktiv).
- \$AC\_SMINVELO und \$AC\_SMINVELO\_INFO liefern Daten wie bei der normalen Teileprogrammabarbeitung.
- \$AC\_SPIND\_STATE gibt die Zustände zurück wie sich diese bei SERUPRO einstellen.

## Beispiel

Beispiel zur Visualisierung des Inhalts der Systemvariablen für die Spindel 1. Die Variablen werden zyklisch in R-Parameter geschrieben. Diese können am HMI im Bereich R-Parameter angezeigt werden.

### Programmcode

```
N05 IDS=1 WHENEVER TRUE DO $R10=$AC_SMAXVELO [1]
N10 IDS=2 WHENEVER TRUE DO $R11=$AC_SMAXVELO_INFO [1]
N15 IDS=3 WHENEVER TRUE DO $R12=$AC_SMINVELO [1]
N20 IDS=4 WHENEVER TRUE DO $R13=$AC_SMINVELO_INFO [1]
N25 IDS=5 WHENEVER TRUE DO $R14=$AC_SPIND_STATE [1]
```

## Siehe auch

Spindel im Sollbereich (Seite 1480)

## 15.8.6 Maximale Drehzahl der Spindel

### Maschinenbedingte parametrierbare maximale Spindeldrehzahl

Die maschinenbedingte maximale Spindeldrehzahl wird über folgendes Maschinendatum parametrierbar:

MD35100 \$MA\_SPIND\_VELO\_LIMIT (maximale Spindeldrehzahl)

Der Drehzahlgrenzwert wird von der NC istwertseitig, d.h. unter Berücksichtigung der aktuellen Getriebestufe, überwacht.

---

#### Hinweis

##### Maschinenhersteller

Es wird empfohlen, eine Reduzierung der maximalen Spindeldrehzahl im Maschinendatum MD35100 nur bei stillgesetzter Spindel vorzunehmen. Dies ist insbesondere bei Spindeln zu beachten, die über NC-Reset hinaus aktiv sind (siehe Maschinendatenbeschreibung MD35040 SPIND\_ACTIVE\_AFTER\_RESET). Andernfalls kann es zum Auftreten des Alarms 22100 kommen.

#### Literatur

- Herstellerdokumentation: Listenhandbuch "Ausführliche Maschinendaten Beschreibung"
  - Anwenderdokumentation: Diagnosehandbuch
- 

### Reaktionen bei Überschreitung

Überschreitet die Istdrehzahl der Spindel die parametrierbare maximale Spindeldrehzahl um mehr als den Toleranzwert (MD35150 \$MA\_SPIND\_DES\_VELO\_TOL), erfolgen NC-seitig folgende Reaktionen:

- DB31, ... DBX83.0 = 1 (Drehzahlgrenze überschritten)
- Alarm 22100 "Futterdrehzahl überschritten" wird angezeigt
- Alle Achsen und Spindeln des Kanals werden stillgesetzt

### Prozessbedingte einstellbare maximale Spindeldrehzahl

Über das folgende Settingdatum besteht die Möglichkeit, prozessbedingt, eine im Verhältnis zum Maschinendatum MD35100 niedrigere maximale Spindeldrehzahl einzustellen:

SD43235 \$SA\_SPIND\_USER\_VELO\_LIMIT (Maximale Spindeldrehzahl)

Eine Änderung kann vorgenommen werden über:

- Programmierung im Teileprogramm
- Vom Maschinenbediener über die Bedienoberfläche

Die Änderung wird sofort wirksam.

### Begrenzung der Solldrehzahl

Die Steuerung begrenzt die Spindelsolldrehzahl auf den im Settingdatum angegebenen Wert.

Eine Begrenzung der Spindelsolldrehzahl aufgrund des Settingdatums SD43235 wird über folgende Systemvariable mit dem Wert "21" angezeigt:

\$AC\_SMAXVELO\_INFO[<Spindelnummer>] == 21

## 15.8.7 Maximale Gebergrenzfrequenz

### VORSICHT

Die max. Gebergrenzfrequenz des Spindellageistwertgebers wird von der Steuerung überwacht (Überschreiten möglich). Der Werkzeugmaschinenhersteller muss durch die Auslegung der Komponenten Spindelmotor, Getriebe, Messgetriebe und Geber und der dazugehörigen Maschinendaten sicherstellen, dass die max. Drehzahl (mechanische Grenzdrehzahl) des Spindellageistwertgebers nicht überschritten werden kann.

### Max. Geberfrequenz überschritten

Erreicht die Spindel in der Spindelbetriebsart Steuerbetrieb eine Drehzahl (großer S-Wert programmiert), die über der max. Gebergrenzfrequenz liegt (die max. Drehzahl des Gebers darf dabei nicht überschritten sein), geht die Synchronisation verloren. Die Spindel dreht weiter, jedoch mit verminderter Funktionalität.

Bei folgenden Funktionen wird die Spindeldrehzahl reduziert, bis das aktive Messsystem wieder unterhalb der Gebergrenzfrequenz arbeitet:

- Gewindeschneiden (G33, G34, G35)
- Gewindebohren ohne Ausgleichsfutter (G331, G332)
- Umdrehungsvorschub (G95)
- Konstante Schnittgeschwindigkeit (G96, G961, G97, G971)
- SPCON (Lage geregelter Betrieb der Spindel)

Bei Überschreitung der Gebergrenzfrequenz werden für das betreffende Messsystem die NC/PLC-Nahtstellensignale:

DB31, ... DBX60.4 (Referenziert / Synchronisiert 1)

bzw.

DB31, ... DBX60.5 (Referenziert / Synchronisiert 2)

zurückgesetzt und die NC/PLC-Nahtstellensignale:

DB31, ... DBX60.2 (Gebergrenzfrequenz 1 überschritten)

bzw.

DB31, ... DBX60.3 (Gebergrenzfrequenz 2 überschritten)

gesetzt.

Befindet sich die Spindel im Achsbetrieb, darf die max. Gebergrenzfrequenz nicht überschritten werden.

Die max. Geschwindigkeit (MD32000 \$MA\_MAX\_AX\_VELO) muss unterhalb der max. Gebergrenzfrequenz liegen, ansonsten wird der Alarm 21610 ausgegeben und die Achse stillgesetzt.

### Maximale Gebergrenzfrequenz unterschritten

Wurde die max. Gebergrenzfrequenz überschritten und wird anschließend wieder eine Drehzahl erreicht, die unterhalb der max. Gebergrenzfrequenz liegt (kleineren S-Wert programmiert, Spindelkorrektorschalter verändert, etc.), synchronisiert sich die Spindel automatisch mit der nächsten Nullmarke bzw. dem nächsten Bero-Signal. Die neue Synchronisation läuft immer für das aktive Lagemesssystem, das seine Synchronisation verloren hat und dessen max. Gebergrenzfrequenz momentan unterschritten ist.

### Besonderheiten

Sind folgende Funktionen aktiv, kann die max. Gebergrenzfrequenz nicht überschritten werden:

- Spindelbetriebsart Positionierbetrieb, Achsbetrieb
- Gewindeschneiden (G33, G34, G35)
- Gewindebohren ohne Ausgleichsfutter G331, G332 (gilt nicht für G63)
- Umdrehungsvorschub (G95)
- Konstante Schnittgeschwindigkeit (G96, G961, G97, G971)
- SPCON

### 15.8.8 Zielpunktüberwachung

#### Zielpunktüberwachung

Beim Positionieren (Spindel befindet sich in der Spindelbetriebsart Positionierbetrieb) wird überwacht, wie weit die Spindel (mit ihrer Istposition) von der programmierten Spindelsollposition (Zielpunkt) entfernt ist.

Dazu können in den Maschinendaten:

MD36000 \$MA\_STOP\_LIMIT\_COARSE (Genauhaltgrenze grob)

und

MD36010 \$MA\_STOP\_LIMIT\_FINE (Genauhaltgrenze fein)

zwei Grenzwerte als inkrementeller Weg von der Spindelsollposition ausgehend vorgegeben werden.

Die Genauigkeit der Spindelpositionierung ist unabhängig von den zwei Grenzwerten immer so gut, wie durch den angeschlossenen Spindelmessgeber, die Lose, die Getriebeübersetzung etc. vorgegeben wird.

#### Genauhaltfenster parametersatzabhängig

Parametersatzabhängig unterschiedliche Genauhaltfenster können projiziert werden. Damit ist es möglich, beim Achsbetrieb und beim Spindelpositionieren mit unterschiedlichen Genauigkeiten zu arbeiten. Für das Spindelpositionieren kann die Projektierung der Genauhaltfenster für jede Getriebestufe getrennt vorgenommen werden.

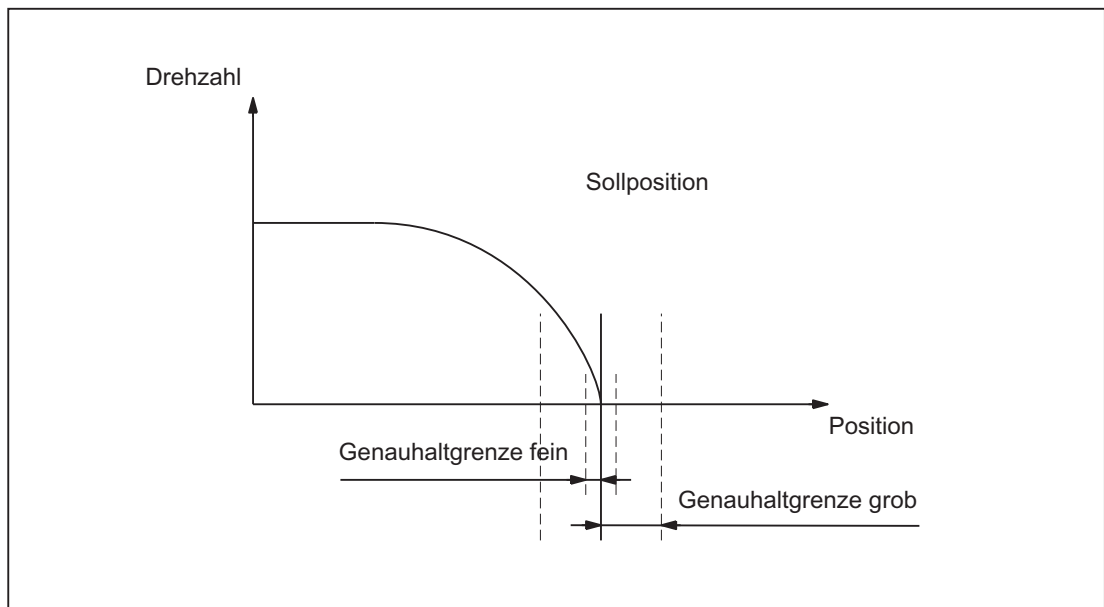


Bild 15-9 Genauhalt-Zonen einer Spindel



**DB31, ... DBX60.7 und DB31, ...DBX60.6 (Position erreicht mit Genauhalt grob/fein)**

Die zwei durch die Maschinendaten:  
MD36000 \$MA\_STOP\_LIMIT\_COARSE (Genauhaltgrenze grob)  
und  
MD36010 \$MA\_STOP\_LIMIT\_FINE (Genauhaltgrenze fein)  
festgelegten Grenzwerte werden mit den NC/PLC-Nahtstellensignalen:  
DB31, ... DBX60.7 (Position erreicht mit Genauhalt grob)  
und  
DB31, ... DBX60.6 (Position erreicht mit Genauhalt fein)  
an die PLC ausgegeben.

**Satzwechsel bei SPOS und M19**

Beim Positionieren der Spindel mit `SPOS` oder `M19` erfolgt der Satzwechsel abhängig von der Zielpunktüberwachung mit dem NC/PLC-Nahtstellensignal: DB31, ... DBX60.6 (Position erreicht mit Genauhalt fein).

Dabei müssen auch alle anderen im Satz programmierten Funktionen ihr Satzendekriterium erreicht haben (z.B. alle Hilfsfunktionen von der PLC quittiert).

Bei `SPOSA` ist der Satzwechsel unabhängig von der Zielpunktüberwachung.

**15.8.9 M40: Automatische Getriebestufenauswahl bei Drehzahlen außerhalb projektierter Schaltschwellen****Funktion**

Bei aktivem M40 erfolgt eine automatische Getriebestufenauswahl auch dann, wenn die programmierte Spindeldrehzahl `s...` außerhalb der projektierter Schaltschwellen liegt.

Dabei sind folgende Fälle zu unterscheiden:

- **Programmierte Drehzahl zu groß**

Die programmierte Spindeldrehzahl ist höher als die projektierte Maximaldrehzahl der zahlenmäßig größten Getriebestufe:

S... > MD35110 \$MA\_GEAR\_STEP\_MAX\_VELO[<n>]

In diesem Fall wird in die **höchste Getriebestufe** (gemäß MD35090 \$MA\_NUM\_GEAR\_STEPS) geschaltet.

- **Programmierte Drehzahl zu klein**

Die programmierte Spindeldrehzahl ist kleiner als die projektierte Minimaldrehzahl der ersten Getriebestufe:

S... < MD35120 \$MA\_GEAR\_STEP\_MIN\_VELO[1]

In diesem Fall wird in die **erste Getriebestufe** geschaltet.

- **Programmierte Drehzahl = 0**

Bei der Programmierung der Drehzahl 0 ( $s_0$ ) ist das Verhalten abhängig von der Projektierung der Minimaldrehzahl der ersten Getriebestufe MD35120 \$MA\_GEAR\_STEP\_MIN\_VELO[1]:

- Ist MD35120 \$MA\_GEAR\_STEP\_MIN\_VELO[1] = 0 projektiert, dann wird bei Programmierung von  $s_0$  in die **erste Getriebestufe** geschaltet.
- Ist MD35120 \$MA\_GEAR\_STEP\_MIN\_VELO[1] > 0 projektiert, dann wird bei Programmierung von  $s_0$  **kein Getriebestufenwechsel** durchgeführt und die letzte Getriebestufe bleibt weiterhin aktiv. Damit bleibt es möglich, die Spindel mit  $s_0$  (anstatt von  $m_5$ ) zu stoppen ohne einen Getriebestufenwechsel auszulösen.

## Wirksamkeit

Das Schalten in die höchste Getriebestufe oder in die erste Getriebestufe bei automatischer Getriebestufenauswahl (M40) ist bei Programmierungen von Spindeldrehzahlen  $s_{...}$  über das Teileprogramm, in Synchronaktionen und bei Vorgaben über PLC FC18 wirksam.

Bei Drehzahlprogrammierungen aus dem Teileprogramm für Gewindebohren mit G331 wird das Verhalten auch für den zweiten Datensatz zur Getriebestufenauswahl unterstützt (Voraussetzung: MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE, Bit 5 = 1).

## Randbedingungen

### Freigabe des Getriebestufenwechsels

Voraussetzung für die Funktion ist die generelle Freigabe des Getriebestufenwechsels über die Maschinendaten:

MD35010 \$MA\_GEAR\_STEP\_CHANGE\_ENABLE (Getriebestufenwechsel parametrieren)

MD35090 \$MA\_NUM\_GEAR\_STEPS (Anzahl eingerichteter Getriebestufen)

**Beispiel**

Automatische Getriebestufenauswahl M40 ist Grundstellung nach NC-Reset.

Teileprogramm:

Programmcode	Kommentar
...	
N15 S3500 M3	; S3500 ist größer als MD35110 der 2.Getriebestufe. Es wird in die 2. Getriebestufe geschaltet.
...	
N50 S0 M3	; Spindel wird gestoppt, S0 fordert keinen Getriebestufenwechsel an (Sonderbehandlung S0).
...	
N95 S5 M3	; S5 ist kleiner als MD35120 der 1.Getriebestufe. Es wird in die 1.Getriebestufe geschaltet.
...	

Projektierungsdaten für Spindel 1 (AX5):

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE[AX5] = 1	; Getriebestufenwechsel freigeben
MD35090 \$MA_NUM_GEAR_STEPS[AX5] = 2	; Anzahl vorhandener Getriebestufen
MD35110 \$MA_GEAR_STEP_MAX_VELO[1,AX5] = 500	; Obere Schaltschwelle für Getriebestufe 1
MD35120 \$MA_GEAR_STEP_MIN_VELO[1,AX5] = 10	; Untere Schaltschwelle für Getriebestufe 1
MD35110 \$MA_GEAR_STEP_MAX_VELO[2,AX5] = 2000	; Obere Schaltschwelle für Getriebestufe 2
MD35120 \$MA_GEAR_STEP_MIN_VELO[2,AX5] = 500	; Untere Schaltschwelle für Getriebestufe 2

## 15.9 Spindel mit SMI 24 (Weiss-Spindel)

### 15.9.1 Allgemeine Informationen

Um die Sensordaten der Spindel in der Steuerung verarbeiten zu können, mussten bisher die Sensoren an I/O-Baugruppen verdrahtet und per Feldbus (PROFIBUS DP bzw. PROFINET IO) an die PLC übertragen werden.

Bei einer Spindel mit SMI 24 (Weiss-Spindel) werden die Sensordaten per DRIVE-CLiQ zum Antrieb übertragen und stehen dort in Antriebsparametern zur Verfügung. Bei Verwendung des zyklischen Antriebstelegramms 139 werden die Sensordaten vom Antrieb weiter an die Steuerung übertragen. Dort stehen sie dann in folgenden Systemdaten zur Verfügung:

- Systemvariable
- BTSS-Variable
- NC/PLC-Nahtstellensignale

#### Voraussetzung

- Die Spindel ist über das Sensormodul SMI 24 per DRIVE-CLiQ mit dem Antrieb verbunden.
- Für die Spindel ist Antriebstelegramm 139 projektiert.

---

#### Hinweis

##### Antriebstelegramm 139

Eine Spindel mit Sensormodul SMI 24 kann prinzipiell auch mit einem anderen Antriebstelegramm betrieben werden. Die Sensordaten werden aber nur mit dem Antriebstelegramm 139 zur Steuerung übertragen.

---

## 15.9.2 Sensordaten

### Sensoren des Spindelmotors

Die Sensoren der Spindel liefern Informationen über die Spannvorrichtung und die Winkellage der Motorwelle:

- Analog-Sensor S1: Spannzustand  
Spannungswert im Bereich von 0 - 10 V abhängig von der Lage der Zugstange.
- Digital-Sensor S4: Kolbenendlage
  - 0 = Kolben nicht in Position
  - 1 = Kolben in Position, d. h. Kolben ist frei
- Digital-Sensor S5: Winkellage der Motorwelle
  - 0 = Motorwelle nicht gerichtet
  - 1 = Motorwelle steht auf Position (Voraussetzung: Stillstand der Spindel)

---

#### Hinweis

#### Spindel mit Sensormodul SMI 24 und Achscontainer

Eine Spindel mit Sensormodul SMI 24 und Antriebstelegramm 139 zur Übertragung der Sensordaten an die Steuerung darf nicht Bestandteil eines Achscontainers sein, dessen Achsen per NCU-Link über mehrere NCUs verteilt sind.

---

### Übertragung der Sensordaten

Die Sensordaten werden vom Sensormodul SMI 24 im zyklischen Antriebstelegramm 139 als Prozessdaten 11 - 14 zur Steuerung übertragen. Das Antriebstelegramm 139 basiert auf dem Antriebstelegramm 136, bei dem statt der Daten des 2. Gebers die Sensordaten übertragen werden. Eine detaillierte Beschreibung des Antriebstelegramms 139 findet sich in:

#### Literatur:

SINAMICS S120/S150 Listenhandbuch; Funktionspläne, Kapitel: PROFIdrive

**Systemdaten: Sensordaten**

Die Sensordaten können in der Steuerung über folgende Systemdaten gelesen werden:

Bedeutung	Systemvariable \$VA_	NC/PLC- Nahtstelle DB31, ...	BTSS-Variable	Antriebs- parameter
Sensorkonfiguration	MOT_SENSOR_CONF[<Achse>]	DBB132	vaMotSensorConf	r5000
Spannzustand <sup>1)</sup>	MOT_CLAMPING_STATE[<Achse>]	DBW134	vaMotClampingState	r5001
Messwert Analogsensor S1 <sup>2)</sup>	MOT_SENSOR_ANA[<Achse>]	DBW136	vaMotSensorAna	r5002
Status Digitalsensoren	MOT_SENSOR_DIGI[<Achse>]	DBW138	vaMotSensorDigi	r5003
<Achse>:	Maschinenachsbezeichner: AX1 ... AXn oder Spindelbezeichner: S1 ... Sm			

1) siehe Kapitel "Spannzustand (Seite 1495)"

2) Sensor S1: 0 - 10 V

Analoger Istwert: 0 - 10000 Inkremente, Auflösung 1 mV

Beispiel:

SIMATIC S7-Eingangsbaugruppe: 0 - 27648 Inkremente, Auflösung 0,36 mV

Anpassfaktor bei Umstellung auf eine Spindel mit SMI 24: 2,7648

**Ausführliche Systemdatenbeschreibung**

Für Informationen über NC/PLC-Nahtstellen siehe Kapitel "Signale von Achse/Spindel (DB31, ...) (Seite 1900)".

**Literatur:**

Systemvariable: Listenhandbuch Systemvariable

BTSS-Variable: Listenhandbuch 2; Variablen

Antriebsparameter: SINAMICS S120/S150 Listenhandbuch

### 15.9.3 Spannzustand

Der Sensor S1 liefert einen analogen Spannungswert 0 V - 10 V abhängig von der Lage der Spannvorrichtung. Der Spannungswert steht in den Systemdaten zur anwenderseitigen Auswertung des Spannzustandes zur Verfügung.

---

#### Hinweis

Die nachfolgend beschriebene Auswertung des Sensors S1 zur Erzeugung des Zustandswertes für den Spannzustand und die Begrenzung der Spindeldrehzahl, erfolgen nur, wenn im Antriebsparameter r5000 folgende Zustandswerte angezeigt werden:

- r5000.0 == 1: Sensorik vorhanden
- r5000.1 == 1: Sensor S1 (Spannzustand) vorhanden
- r5000.10 == 1: Zustandswerte werden gebildet, Drehzahlbegrenzungen p5043 aktiv

Siehe auch Kapitel "Sensordaten (Seite 1493)", Absatz: "Systemdaten: Sensordaten"

---

#### Zustandswert

Zur Vereinfachung der Auswertung steht der Spannzustand in den Systemdaten auch als Zustandswert 0 - 11 zu Verfügung.

Ein Spannungsbereich entspricht einem bestimmten Spannzustand. Die Spannungsbereiche können über den Antriebsparameter p5041[0...5] eingestellt werden.

Für die Spannungsbereiche kann über den Antriebsparameter p5040 zusätzlich eine Spannungstoleranz eingestellt werden.

---

#### Hinweis

Die Spannungsbereiche  $\pm$  Spannungstoleranz dürfen sich nicht überschneiden.

---

#### Drehzahlgrenzen

Für die Spannzustände mit den Zustandswerten 3 - 10 können über den Antriebsparameter p5043[0...6] Drehzahlgrenzwerte vorgegeben werden. In den anderen Spannzuständen (Zustandswerte 1, 2 und 11) ist als Drehzahlgrenzwert 0 [1/min] fest vorgegeben.

Die Steuerung begrenzt in den verschiedenen Spannzuständen die Spindeldrehzahl auf den jeweiligen Grenzwert.

**Hinweis**

**Ändern der Drehzahlgrenzen**

Eine Änderung der Drehzahlgrenzen in den Antriebsparametern p5043[0...6] wird in der Steuerung erst wirksam (Begrenzung der Spindeldrehzahl auf die neue Drehzahlgrenze) nach:

- Power On-Reset oder Aus/Einschalten der Steuerung
- Abwahl des Zustandes "Parken" für die Spindel (siehe Kapitel "Ausschalten aller Überwachungen: "Parken" (Seite 123)")

**Zusammenhang: Zustandswert, Spannungsbereich und Drehzahlgrenze**

Zustandswert <sup>1)</sup>	Spannzustand	Spannungsbereich <sup>2)</sup>		Drehzahlgrenze
		Obergrenze	Untergrenze	
0	Sensor S1 nicht vorhanden oder Zustandswerte inaktiv	-	-	-
1	Zustandsinitialisierung läuft	-	-	<sup>3)</sup>
2	Gelöst mit Meldung (Fehlerzustand)	-	p5041[0] + p5040	<sup>3)</sup>
3	Gelöst	p5041[0]	p5041[1]	p5043[0]
4	Spannend mit Werkzeug	-	-	p5043[1]
5	Lösend mit Werkzeug	-	-	p5043[2]
6	Lösend ohne Werkzeug	-	-	p5043[3]
7	Gespannt mit Werkzeug UND S4 == 0	p5041[2]	p5041[3]	p5043[4]
8	Gespannt mit Werkzeug UND S4 == 1			p5043[4]
9	Spannend ohne Werkzeug	-	-	p5043[5]
10	Gespannt ohne Werkzeug	p5041[4]	p5041[5]	p5043[6]
11	Gespannt mit Meldung (Fehlerzustand)	p5041[5] - p5040	-	<sup>3)</sup>

1) Der Zustandswert kann in der Steuerung über folgende Systemdaten gelesen werden:

- Systemvariable: \$VA\_MOT\_CLAMPING\_STATE[<Achse>]
- NC/PLC-Nahtstelle: DB31, ... DBW134
- BTSS-Variable: vaMotClampingState
- Antriebsparameter: r5001

2) p5041[0...5]: Spannungsschwellwerte, p5040: Spannungsschwellwerte Toleranz

3) Drehzahlgrenzwert fest vorgegeben: 0 [1/min]



## 15.9.4 Weitere Antriebsparameter

### **p5042:Transitionszeit**

Im Antriebsparameter p5042 können für die Spannzustandserkennung folgende Zeiten eingestellt werden:

- p5042[0]: Stabilisierungszeit für "Gespannt mit Werkzeug"  
Der Spannzustand "Gespannt mit Werkzeug" muss im Spindelmotor mindesten die eingestellte Stabilisierungszeit anstehen, bevor der Zustand an die Steuerung gemeldet wird.
- p5042[1]: Maximalzeit zum Spannen  
Der Übergang vom Zustand "Gelöst" zum Zustand "Gespannt mit Werkzeug" oder "Gespannt ohne Werkzeug" darf maximal die eingestellte Zeit dauern.

### **r5044: Drehzahlbegrenzung aus Spanncyklus**

Im Antriebsparameter r5044 wird die im Spannzustand "Gespannt ohne Werkzeug" wirkende Drehzahlgrenze aus p5043[6] angezeigt.

Ein Wert von 65535 bedeutet, dass die Drehzahlgrenze nicht wirksam ist.

## 15.10 Randbedingungen

### 15.10.1 Ändern von Regelparametern

Bei Spindeln, die sich nicht im lagegeregelten Betrieb befinden, werden mit dem Befehl NEWCONF Maschinendatenänderungen auch dann wirksam, wenn die Spindel nicht steht.

Bei Änderungen von Regelparametern kann es nach dem Wirksam werden der neuen Werte zu Drehzahlsollwertsprüngen kommen. Regelparameter sind z.B.:

- MD32200 \$MA\_POSCTRL\_GAIN (KV-Faktor)
- MD32210 \$MA\_POSCTRL\_INTEGR\_TIME (Nachstellzeit Lageregler)
- MD32410 \$MA\_AX\_JERK\_TIME (Zeitkonstante für den axialen Ruckfilter)

## 15.11 Beispiele

### 15.11.1 Automatische Getriebestufenanwahl (M40)

#### Beispiel

Veranschaulichung der Inhalte der neuen Satzsuchlaufvariablen:  
Annahmen zur automatischen Getriebestufenanwahl (M40):

S0...500	1. Getriebestufe
S501..1000	2. Getriebestufe
S1001..2000	3. Getriebestufe

Inhalte der Systemvariablen:

\$P_SEARCH_S	; Aufgesammelte S-Wert
\$P_SEARCH_DIR	; Aufgesammelte Drehrichtung
\$P_SEARCH_GEAR	; Aufgesammelte Getriebestufe

Aufgesammelte	S-Wert:	Drehrichtung:	Getriebestufe:
	; 0/letzte Drehzahl	-5	40/letzte GS
N05 G94 M40 M3 S1000	; 1000	3	40
N10 G96 S222	: 222	3	40
N20 G97	; f (PlanAxPosPCS)*	3	40
N30 S1500	; 1500	3	40
N40 SPOS=0**	; 1500	-19	40
N50 M19**	; 1500	-19	40
N60 G94 G331 Z10 S300	; 300	-19	40
N70 M42	; 300	-19	42
N80 M4	; 300	4	42
N90 M70	; 300	70	42
N100 M3 M40	; 300	3	40
N999 M30			

\* f (PlanAxPosPCS): Die Drehzahl ist abhängig von der aktuellen Position der Planachse im Werkstückkoordinatensystem.

\*\* (\$P\_SEARCH\_SPOS und \$P\_SEARCH\_SPOSMODE werden beschrieben)

## 15.12 Datenlisten

### 15.12.1 Maschinendaten

#### 15.12.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10192	GEAR_CHANGE_WAIT_TIME	Wartezeit auf Quittierung eines Getriebestufenwechsel beim Reorganisieren
10714	M_NO_FCT_EOP	M-Funktion für Spindel aktive nach RESET
12060	OVR_SPIND_IS_GRAY_CODE	Spindelkorrektur Gray-codiert
12070	OVR_FACTOR_SPIND_SPEED	Bewertung des Spindelkorrekturschalters
12080	OVR_REFERENCE_IS_PROG_FEED	Override-Bezugsgeschwindigkeit
12082	OVR_REFERENCE_IS_MIN_FEED	Festlegung des Bezugs des Bahn-Override
12090	OVR_FUNCTION_MASK	Auswahl von Override-Spezifikationen

#### 15.12.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20090	SPIND_DEF_MASTER_SPIND	Löschstellung Masterspindel im Kanal
20092	SPIND_ASSIGN_TAB_ENABLE	Freigabe/Sperren des Spindelumsetzers
20850	SPOS_TO_VDI	Ausgabe von M19 an die PLC bei SPOS/SPOSA
22400	S_VALUES_ACTIVE_AFTER_RESET	S-Funktion über RESET wirksam

### 15.12.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30300	IS_ROT_AX	Rundachse
30310	ROT_IS_MODULO	Modulowandlung
31044	ENC_IS_DIRECT2	Geber am Vorsatz-Getriebe
31050	DRIVE_AX_RATIO_DENOM	Nenner Lastgetriebe
31060	DRIVE_AX_RATIO_NUMERA	Zähler Lastgetriebe
31064	DRIVE_AX_RATIO2_DENOM	Nenner Vorsatz-Getriebe
31066	DRIVE_AX_RATIO2_NUMERA	Zähler Vorsatz-Getriebe
31070	DRIVE_ENC_RATIO_DENOM	Nenner Messgetriebe
31080	DRIVE_ENC_RATIO_NUMERA	Zähler Messgetriebe
31122	BERO_DELAY_TIME_PLUS	BERO-Verzögerungszeit in Plus-Richtung
31123	BERO_DELAY_TIME_MINUS	BERO-Verzögerungszeit in Minus-Richtung
32200	POSCTRL_GAIN	Kv-Faktor
32800	EQUIV_CURRCTRL_TIME	Ersatzzeitkonstante Stromregelkreis fuer Vorsteuerung
32810	EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonstante Drehzahlregelkreis fuer Vorsteuerung
32910	DYN_MATCH_TIME	Zeitkonstante der Dynamikanpassung
34040	REFP_VELO_SEARCH_MARKER	Referenzpunktabschaltgeschwindigkeit
34060	REFP_MAX_MARKER_DIST	Überwachung der Nullmarkenstrecke
34080	REFP_MOVE_DIST	Referenzpunktabstand/Zielpunkt bei abstandskodiertem System
34090	REFP_MOVE_DIST_CORR	Referenzpunktverschiebung/Absolutverschiebung abstandskodiert
34100	REFP_SET_POS	Referenzpunktwert
34200	ENC_REFP_MODE	Referenziermodus
35000	SPIND_ASSIGN_TO_MACHAX	Zuordnung Spindel zu Maschinenachse
35010	GEAR_STEP_CHANGE_ENABLE	Art des Getriebestufenwechsels
35012	GEAR_STEP_CHANGE_POSITION	Getriebestufenwechselposition
35014	GEAR_STEP_USED_IN_AXISMODE	Getriebestufe für den Achsbetrieb bei M70
35020	SPIND_DEFAULT_MODE	Grundstellung der Spindel
35030	SPIND_DEFAULT_ACT_MASK	Aktivieren Spindel Grundstellung
35035	SPIND_FUNCTION_MASK	Einstellen von spindelspezifischen Funktionen
35040	SPIND_ACTIVE_AFTER_RESET	Spindel über Reset aktiv
35090	NUM_GEAR_STEPS	Anzahl eingerichteter Getriebestufen
35092	NUM_GEAR_STEPS2	2. Getriebestufen Datensatz: Anzahl eingerichteter Getriebestufen
35100	SPIND_VELO_LIMIT	Maximale Spindeldrehzahl
35110	GEAR_STEP_MAX_VELO[n]	Maximaldrehzahl für den automatischen Getriebestufenwechsel

Nummer	Bezeichner: \$MA_	Beschreibung
35112	GEAR_STEP_MAX_VELO2[n]	2. Getriebestufendatensatz: Maximaldrehzahl für den automatischen Getriebestufenwechsel
35120	GEAR_STEP_MIN_VELO[n]	Minimaldrehzahl für den automatischen Getriebestufenwechsel
35122	GEAR_STEP_MIN_VELO2[n]	2. Getriebestufendatensatz: Minimaldrehzahl für den automatischen Getriebestufenwechsel
35130	GEAR_STEP_MAX_VELO_LIMIT[n]	Maximaldrehzahl der Getriebestufe
35135	GEAR_STEP_PC_MAX_VELO_LIMIT[n]	Maximaldrehzahl der Getriebestufe bei Lageregelung
35140	GEAR_STEP_MIN_VELO_LIMIT[n]	Minimaldrehzahl der Getriebestufe
35150	SPIND_DES_VELO_TOL	Spindeldrehzahltoleranz
35160	SPIND_EXTERN_VELO_LIMIT	Spindeldrehzahlbegrenzung von PLC
35200	GEAR_STEP_SPEEDCTRL_ACCEL[n]	Beschleunigung im Drehzahlregelbetrieb
35210	GEAR_STEP_POSCTRL_ACCEL[n]	Beschleunigung im Lageregelbetrieb
35212	GEAR_STEP_POSCTRL_ACCEL2[n]	2. Getriebestufendatensatz: Beschleunigung im Lageregelbetrieb
35220	ACCEL_REDUCTION_SPEED_POINT	Drehzahlgrenze reduzierte Beschleunigung
35230	ACCEL_REDUCTION_FACTOR	reduzierte Beschleunigung
35300	SPIND_POSCTRL_VELO	Lageregeleinschaltdrehzahl
35350	SPIND_POSITIONING_DIR	Positionierdrehrichtung bei nicht synchronisierter Spindel
35400	SPIND_OSCILL_DES_VELO	Pendeldrehzahl
35410	SPIND_OSCILL_ACCEL	Beschleunigung beim Pendeln
35430	SPIND_OSCILL_START_DIR	Startrichtung beim Pendeln
35440	SPIND_OSCILL_TIME_CW	Pendelzeit für M3-Richtung
35450	SPIND_OSCILL_TIME_CCW	Pendelzeit für M4-Richtung
35500	SPIND_ON_SPEED_AT_IPO_START	Vorschubfreigabe bei Spindel im Sollbereich
35510	SPIND_STOPPED_AT_IPO_START	Vorschubfreigabe bei Spindel steht
35550	DRILL_VELO_LIMIT[n]	Maximaldrehzahlen für das Gewindebohren
35590	PARAMSET_CHANGE_ENABLE	Parametersatzvorgabe durch PLC möglich
36060	STANDSTILL_VELO_TOL	Schwellgeschwindigkeit "Achse/Spindel steht"
36200	AX_VELO_LIMIT	Schwellwert für Geschwindigkeitsüberwachung.

## 15.12.2 Settingdaten

### 15.12.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42600	JOG_FEED_PER_REF_SOURCE	Steuerung Umdrehungsvorschub in JOG
42800	SPIND_ASSIGN_TAB	Spindelnummernumsetzer
42900	MIRROR_TOOL_LENGTH	Werkzeuflängenkorrektur spiegeln
42910	MIRROR_TOOL_WEAR	Verschleißwerte der Werkzeuflängenkorrektur spiegeln
42920	WEAR_SIGN_CUTPOS	Verschleißwerte der Bearbeitungsebene spiegeln
42930	WEAR_SIGN	Vorzeichen aller Verschleißwerte invertieren
42940	TOOL_LENGTH_CONST	Beim Wechsel der Bearbeitungsebene (G17 bis G19) die Zuordnung der Werkzeuflängenkomponenten beibehalten

### 15.12.2.2 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: \$SA_	Beschreibung
43200	SPIND_S	Drehzahl für Spindelstart über PLC-Schnittstelle
43202	SPIND_CONSTCUT_S	Schnittgeschwindigkeit für Spindelstart über PLC-Schnittstelle
43206	SPIND_SPEED_TYPE	Spindeldrehzahltyp für Spindelstart über PLC-Schnittstelle
43210	SPIND_MIN_VELO_G25	progr. Spindeldrehzahlbegrenzung G25
43220	SPIND_MAX_VELO_G26	progr. Spindeldrehzahlbegrenzung G26
43230	SPIND_MAX_VELO_LIMS	progr. Spindeldrehzahlbegrenzung G96/G961
43235	SPIND_USER_VELO_LIMIT	maximale Spindeldrehzahl
43240	M19_SPOS	Spindelposition für Spindelpositionieren mit M19
43250	M19_SPOSMODE	Spindelpositionieranfahrmodus für Spindelpositionieren mit M19
43300	ASSIGN_FEED_PER_REF_SOURCE	Umdrehungsvorschub für Positionierachsen/Spindeln

## 15.12.3 Signale

### 15.12.3.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschubkorrektur A bis H	DB31, ... .DBX0.0-7	DB380x.DBX0.0-7
Achsen-/Spindelsperre	DB31, ... .DBX1.3	DB380x.DBX1.3
Nachführbetrieb	DB31, ... .DBX1.4	DB380x.DBX1.4
Lagemesssystem 1	DB31, ... .DBX1.5	DB380x.DBX1.5
Lagemesssystem 2	DB31, ... .DBX1.6	DB380x.DBX1.6
Korrektur wirksam	DB31, ... .DBX1.7	DB380x.DBX1.7
Reglerfreigabe	DB31, ... .DBX2.1	DB380x.DBX2.1
Spindel-Reset/Restweg löschen	DB31, ... .DBX2.2	DB380x.DBX2.2
Geschwindigkeits-/Spindeldrehzahlbegrenzung	DB31, ... .DBX3.6	DB380x.DBX3.6
Programmtest Achs-/Spindel Freigabe	DB31, ... .DBX3.7	DB380x.DBX3.7
Istgetriebestufe A bis C	DB31, ... .DBX16.0-2	DB380x.DBX2000.0-2
Getriebe ist umgeschaltet	DB31, ... .DBX16.3	DB380x.DBX2000.3
Spindel neu synchronisieren 1	DB31, ... .DBX16.4	DB380x.DBX2000.4
Spindel neu synchronisieren 2	DB31, ... .DBX16.5	DB380x.DBX2000.5
keine n-Überwachung bei Getriebeumschaltung	DB31, ... .DBX16.6	DB380x.DBX2000.6
S-Wert löschen	DB31, ... .DBX16.7	DB380x.DBX2000.7
Vorschubkorrektur für Spindel gültig	DB31, ... .DBX17.0	DB380x.DBX2001.0
Spindel neu synchronisieren beim Positionieren 1	DB31, ... .DBX17.4	DB380x.DBX2001.4
Spindel neu synchronisieren beim Positionieren 2	DB31, ... .DBX17.5	-
M3/M4 invertieren	DB31, ... .DBX17.6	DB380x.DBX2001.6
Pendeln durch die PLC	DB31, ... .DBX18.4	DB380x.DBX2002.4
Pendelfreigabe (Pendeldrehzahl)	DB31, ... .DBX18.5	DB380x.DBX2002.5
Pendeln Drehrichtung rechts (Solldrehrichtung rechts)	DB31, ... .DBX18.6	DB380x.DBX2002.6
Pendeln Drehrichtung links (Solldrehrichtung links)	DB31, ... .DBX18.7	DB380x.DBX2002.7
Spindelkorrektur A bis H	DB31, ... .DBX19.0-7	DB380x.DBX2003.0-7



### 15.12.3.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Spindel/Keine Achse	DB31, ... .DBX60.0	DB390x.DBX0.0
Gebergrenzfrequenz überschritten 1	DB31, ... .DBX60.2	DB390x.DBX0.2
Gebergrenzfrequenz überschritten 2	DB31, ... .DBX60.3	-
Referiert/Synchronisiert 1	DB31, ... .DBX60.4	DB390x.DBX0.4
Referiert/Synchronisiert 2	DB31, ... .DBX60.5	DB390x.DBX0.5
Position erreicht mit Genauhalt grob	DB31, ... .DBX60.6	DB390x.DBX0.6
Position erreicht mit Genauhalt fein	DB31, ... .DBX60.7	DB390x.DBX0.7
Achse/Spindel steht ( $n < n_{min}$ )	DB31, ... .DBX61.4	DB390x.DBX1.4
Lageregler aktiv	DB31, ... .DBX61.5	DB390x.DBX1.5
Drehzahlregler aktiv	DB31, ... .DBX61.6	DB390x.DBX1.6
Stromregler aktiv	DB31, ... .DBX61.7	DB390x.DBX1.7
Restauriert 1	DB31, ... .DBX71.4	-
Restauriert 2	DB31, ... .DBX71.5	-
Sollgetriebestufe A bis C	DB31, ... .DBX82.0-2	DB390x.DBX2000.0-2
Getriebe umschalten	DB31, ... .DBX82.3	DB390x.DBX2000.3
Drehzahlgrenze überschritten	DB31, ... .DBX83.0	DB390x.DBX2001.0
Soll-Drehzahl begrenzt	DB31, ... .DBX83.1	DB390x.DBX2001.1
Soll-Drehzahl erhöht	DB31, ... .DBX83.2	DB390x.DBX2001.2
Spindel im Sollbereich	DB31, ... .DBX83.5	DB390x.DBX2001.5
Ist-Drehrichtung rechts	DB31, ... .DBX83.7	DB390x.DBX2001.7
Gewindebohren ohne Ausgleichsfutter aktiv	DB31, ... .DBX84.3	DB390x.DBX2002.3
aktive Spindelbetriebsart Synchronbetrieb	DB31, ... .DBX84.4	DB390x.DBX2002.4
aktive Spindelbetriebsart Positionierbetrieb	DB31, ... .DBX84.5	DB390x.DBX2002.5
aktive Spindelbetriebsart Pendelbetrieb	DB31, ... .DBX84.6	DB390x.DBX2002.6
aktive Spindelbetriebsart Steuerbetrieb	DB31, ... .DBX84.7	DB390x.DBX2002.7
Spindel in Position wirklich erreicht	DB31, ... .DBX85.5	DB390x.DBX2003.5
M-Funktion für Spindel	DB31, ... .DBB86-87	DB370x.DBX0000
S-Funktion für Spindel	DB31, ... .DBB88-91	DB370x.DBX0004
Sensorik vorhanden	DB31, ... .DBX132.0	DB390x.DBX7000.0
Sensor S1 vorhanden (Spannzustand)	DB31, ... .DBX132.1	DB390x.DBX7000.1
Sensor S4 vorhanden (Kolbenendlage)	DB31, ... .DBX132.4	DB390x.DBX7000.4
Sensor S5 vorhanden (Winkellage der Motorwelle)	DB31, ... .DBX132.5	DB390x.DBX7000.5
Zustandswert wird gebildet, Drehzahlbegrenzung p5043 aktiv	DB31, ... .DBX133.2	DB390x.DBX7001.2
Spannzustand	DB31, ... .DBW134	DB390x.DBW7002
Analogwert: Spannzustand	DB31, ... .DBW136	DB390x.DBW7004
Sensor S4 (Kolbenendlage)	DB31, ... .DBX138.4	DB390x.DBX7006.4
Sensor S5 (Winkellage der Motorwelle)	DB31, ... .DBX138.5	DB390x.DBX7006.5



## V1: Vorschübe

### 16.1 Kurzbeschreibung

#### Vorschubarten

Der Vorschub bestimmt die Bearbeitungsgeschwindigkeit (Achs- bzw. Bahngeschwindigkeit) und wird bei jeder Interpolationsart auch unter Berücksichtigung von Werkzeugkorrekturen auf der Kontur oder auf der Werkzeugmittelpunktbahn (abhängig von G-Funktionen) eingehalten.

Zur optimalen Anpassung an die verschiedenen technologischen Anwendungen (Drehen, Fräsen, Bohren, ...) sind folgende Vorschubarten möglich:

- Eilgangvorschub (G0)
- Zeitreziproker Vorschub (G93)
- Linearvorschub (G94)
- Umdrehungsvorschub (G95)
- Konstante Schnittgeschwindigkeit (G96, G961)
- Konstante Drehzahl (G97, G971)
- Vorschub bei Gewindeschneiden (G33, G34, G35)
- Vorschub bei Gewindebohren ohne Ausgleichsfutter (G331, G332)
- Vorschub bei Gewindebohren mit Ausgleichsfutter (G63)
- Vorschub für Fase/Rundung FRC, FRCM
- Satzweiser Vorschub FB

### Achszuordnung der Vorschübe

Zur Anpassung an die verschiedenen technologischen Anforderungen können Vorschübe den Achsen variabel zugeordnet werden.

Folgende Varianten sind möglich:

- Getrennte Vorschübe für Arbeitsebene und Zustellachse
- Variable Achszuordnung für Bahnvorschub
- Vorschub für Positionierachsen

### Vorschubbeeinflussung

Zur Anpassung an geänderte technologische Gegebenheiten während der Bearbeitung oder für Testzwecke kann der programmierte Vorschub verändert werden:

- Über die Maschinensteuertafel
- Über die Bedientafelfront
- Über die PLC
- Per Programmbefehl

## 16.2 Bahnvorschub F

### Bahnvorschub F

Der Bahnvorschub stellt die geometrische Summe der Geschwindigkeitskomponenten der beteiligten Achsen dar. Er ergibt sich also aus den Einzelbewegungen der miteinander interpolierenden Achsen.

In der Grundeinstellung werden die axialen Geschwindigkeiten der Geometrieachsen, die programmiert sind, verwendet. Mit dem Befehl `FGROUP` kann man erreichen, dass andere Geometrie- und/oder Synchronachsen bei der Berechnung des Bahnvorschubs berücksichtigt werden.

Der Bahnvorschub F bestimmt die Bearbeitungsgeschwindigkeit und wird bei jeder Interpolationsart auch unter Berücksichtigung von Werkzeugkorrekturen eingehalten. Der unter der Adresse F programmierte Wert bleibt in einem Programm erhalten, bis ein neuer F-Wert bzw. eine neue Vorschubart programmiert wird.

### Wertebereich für Bahnvorschub F

Siehe Funktionsbeschreibung G2: "Geschwindigkeiten, Soll-/ Istwertsysteme, Regelung", Kapitel: "Geschwindigkeiten (Seite 332)".

### F-Wert an PLC-Nahtstelle

Der F-Wert des aktuellen Bahnvorschubs wird in die kanalspezifische PLC-Nahtstelle für Hilfsfunktionen (DB21, ... DBB158 bis 193) eingetragen.

Erläuterungen der zugehörigen Nahtstellensignale (Änderungssignal, F-Wert) siehe Funktionsbeschreibung "H2: Hilfsfunktionsausgaben an PLC (Seite 395)".

### Vorschub bei Übergangskreis

**Literatur:**  
Programmierhandbuch Grundlagen

## Vorschub für innen- und außengekrümmte Bahnstücke

Bei Kreissätsen oder gleichsinnig gekrümmten Spline-Sätzen und eingeschalteter Werkzeugradiuskorrektur (G41/G42) kann der programmierte Vorschub an der Mittelpunktsbahn oder an der Kontur (in Abhängigkeit von innen- und außengekrümmten Bahnstücken) wirksam werden.

Dazu existiert eine Gruppe von G-Funktionen:

- CFTCP  
Programmierter Vorschub an der Mittelpunktsbahn wirksam.
- CFC  
Programmierter Vorschub an der Kontur wirksam.
- CFCIN  
Programmierter Vorschub an der Kontur nur bei Innenkrümmung wirksam.

### Literatur:

Programmierhandbuch Grundlagen

## Maximale Bahngeschwindigkeit

Die maximale Bahngeschwindigkeit ergibt sich aus den Maximalgeschwindigkeiten der beteiligten Linear- bzw. Rundachsen (MD32000 \$MA\_MAX\_AX\_VELO), d. h. die Achse mit der niedrigsten Maximalgeschwindigkeit bestimmt die maximale Bahngeschwindigkeit. Diese kann nicht überschritten werden.

Ist G0 programmiert, so wird mit der Bahngeschwindigkeit gefahren, die sich durch die Begrenzung von MD32000 \$MA\_MAX\_AX\_VELO ergibt.

## Grenzgeschwindigkeit für Bahnachsen

Zusätzlich kann für Bahnachsen (Geometrie- und Synchronachsen) mit der Anweisung `FL[<Achse>]` eine Grenzgeschwindigkeit für die jeweilige Achse programmiert werden.

Damit können getrennte Vorschübe für Arbeitsebene und Zustellachse programmiert werden. Das bedeutet, dass sowohl für die bahnbezogene Interpolation als auch für die Zustellachse eine Vorschubgeschwindigkeit vorgegeben wird. Als Zustellachse wird die senkrecht zur angewählten Bearbeitungsebene wirkende Achse bezeichnet. Die zustellachsspezifische Vorschubangabe begrenzt programmierbar die Achsgeschwindigkeit und damit die Bahngeschwindigkeit. Es ist dabei zu beachten, dass keine Koordinatendrehungen durch Frames berücksichtigt werden. Dies bedeutet, dass die Zustellachse eine Achse des Basiskoordinatensystems sein muss. Mit der Funktion kann z. B. berücksichtigt werden, dass ein Fräser auf der Stirnseite eine geringere Schnittleistung aufweist als über den Fräserumfang.

Programmierbeispiel:

Programmcode	Kommentar
... G94 ...	; Anwahl des Vorschubtyps (mm/min)
X30 Y20 F200	; Bahnvorschub = 200 mm/s
FL[Z]=50 Z-30	; Max. Vorschub für Z-Achse: 50 mm/s

## Niedrig auflösende Geber

Bei der Verwendung von niedrig auflösenden Gebern können mit geglätteten Istwerten stetigere Bahn- bzw. Achsbewegungen erzielt werden. Je größer dabei die Zeitkonstante eingestellt wird, desto besser ist die Glättung der Istwerte und umso größer ist der Nachlauf.

MD34990 \$MA\_ENC\_ACTVAL\_SMOOTH\_TIME[<Achse>] (Glättungszeitkonstante für Istwerte)

Geglättete Istwerte werden verwendet bei:

- Gewindeschneiden (G33, G34, G35)
- Umdrehungsvorschub (G95, G96, G97, FPRAON)
- Anzeige von Drehzahl, Istposition und -geschwindigkeit

### 16.2.1 Vorschubart G93, G94, G95

#### Wirksamkeit

Die Vorschubarten G93, G94, G95 wirken bei den G-Funktionen der Gruppe 1 (außer bei G0) in den Automatikbetriebsarten.

Im JOG-Betrieb kann mit G94 oder G95 verfahren werden.

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1)

#### Zeitreziproker Vorschub (G93)

Der zeitreziproke Vorschub wird verwendet, wenn es einfacher ist, anstatt des Vorschubs die Zeitdauer für das Abfahren eines Satzes zu programmieren.

Der zeitreziproke Vorschub errechnet sich nach folgender Formel:

$$F = \frac{v}{s}$$

- mit
- F: zeitreziproker Vorschub in 1/min
  - v: Gewünschte Bahngeschwindigkeit in mm/min bzw. inch/min
  - s: Bahnlänge in mm bzw. inch

Programmierbeispiel:

Programmcode	Kommentar
N10 G1 G93 X100 Y200 F2	; Der programmierte Bahnweg wird in 0,5 min abgefahren.
...	

#### Hinweis

Bei aktivem G41/G42 darf G93 nicht verwendet werden. Ist die Satzlänge von Satz zu Satz sehr unterschiedlich, sollte bei G93 in jedem Satz ein neu bestimmter F-Wert programmiert werden.

### Lineavorschub (G94)

Der Lineavorschub wird in folgenden Einheiten bezogen auf eine Linearachse bzw. Rundachse programmiert:

- [mm/min, Grad/min] bei metrischem Grundsystem
- [inch/min, Grad/min] bei Inch-Grundsystem

### Umdrehungsvorschub (G95)

Der Umdrehungsvorschub wird in folgenden Einheiten bezogen auf eine Masterspindel programmiert:

- [mm/Umdrehung] bei metrischem Grundsystem
- [inch/Umdrehung] bei Inch-Grundsystem
- [Grad/Umdrehung] bei einer Rundachse

Die Bahngeschwindigkeit ergibt sich aus der Ist-Drehzahl der Spindel nach folgender Formel:

$$V = n * F$$

mit

- V: Bahngeschwindigkeit in mm/min bzw. inch/min
- n: Drehzahl der Masterspindel in U/min
- F: programmierter Umdrehungsvorschub in mm/U bzw. inch/U

#### Hinweis

Wird zwischen den Vorschubarten G93, G94, G95 umgeschaltet, so wird der programmierte F-Wert gelöscht.



### Zahnvorschub

Vorrangig für Fräsbearbeitungen kann anstelle des Umdrehungsvorschubwerts  $F$ ... auch der in der Praxis gebräuchlichere Zahnvorschubwert  $FZ$ ... (Vorschubweg pro Zahn) programmiert werden.

Über den Werkzeugparameter \$TC\_DPNT (Anzahl der Zähne pro Umdrehung) des aktiven Werkzeugkorrekturdatensatzes berechnet die Steuerung aus dem programmierten Zahnvorschub für jeden Verfahrssatz den wirksamen Umdrehungsvorschub:

$$F = FZ * \$TC\_DPNT$$

mit  $F$ : Umdrehungsvorschub in mm/U bzw. inch/U  
 $FZ$ : Zahnvorschub in mm/Zahn bzw. inch/Zahn  
 $\$TC\_DPNT$ : Werkzeugparameter: Anzahl der Zähne/U

Beispiel: Fräser mit 5 Zähnen ( $\$TC\_DPNE = 5$ )

Programmcode	Kommentar
N10 G0 X100 Y50	
N20 G1 G95 FZ=0.02	; Zahnvorschub 0,02 mm/Zahn
N30 T3 D1	; Werkzeug einwechseln und Werkzeugkorrekturdatensatz aktivieren.
M40 M3 S200	; Spindeldrehzahl 200 U/min
N50 X20	; Fräsen mit: $FZ = 0,02$ mm/Zahn → wirksamer Umdrehungsvorschub: $F = 0,02$ mm/Zahn * 5 Zähne/U = 0,1 mm/U bzw.: $F = 0,1$ mm/U * 200 U/min = 20 mm/min
...	

### Umdrehungsvorschub in der Betriebsart JOG

In der Betriebsart JOG ist das Verhalten der Achse/Spindel auch von der Einstellung des folgenden Settingdatums abhängig:

SD41100 \$SN\_JOG\_REV\_IS\_ACTIVE (Umdrehungsvorschub bei JOG aktiv)

Ist dieses Settingdatum aktiv, so wird eine Achse/Spindel immer mit Umdrehungsvorschub:

MD32050 \$MA\_JOG\_REV\_VELO (Umdrehungsvorschub bei JOG)

bzw.

MD32040 \$MA\_JOG\_REV\_VELO\_RAPID (Umdrehungsvorschub bei JOG mit Eilgangsüberlagerung)

in Abhängigkeit von der Masterspindel verfahren.

Ist das Settingdatum nicht aktiv, dann:

- ist das Verhalten der Achse/Spindel abhängig vom Settingdatum:  
SD43300 \$SA\_ASSIGN\_FEED\_PER\_REV\_SOURCE (Umdrehungsvorschub für Positionachsen/Spindeln)
- ist das Verhalten einer Geometrieachse, auf die ein Frame mit Rotation wirkt, abhängig vom kanalspezifischen Settingdatum:  
SD42600 \$SC\_JOG\_FEED\_PER\_REV\_SOURCE

#### **DB31, ... DBX62.2 (Umdrehungsvorschub aktiv)**

Ein programmierter, aktiver Umdrehungsvorschub (G95) wird über das folgende NC/PLC-Nahtstellensignal angezeigt:

DB31, ... DBX62.2 (Umdrehungsvorschub aktiv)

## **16.2.2 Vorschubart G96, G961, G962, G97, G971**

### **Konstante Schnittgeschwindigkeit (G96, G961)**

Die konstante Schnittgeschwindigkeit wird bei Drehmaschinen benutzt, um die Schnittbedingungen unabhängig vom Arbeitsdurchmesser des Werkstücks konstant zu halten. Damit kann das Werkzeug immer im optimalen Schnittleistungsbereich gefahren werden und hält deshalb länger.

#### **Anwahl von G96, G961**

Beim Programmieren von G96, G961 wird der zugehörige S-Wert als Schnittgeschwindigkeit in m/min bzw. ft/min in der Planachse interpretiert. Verringert sich bei der Bearbeitung der Werkstückdurchmesser in der Planachse, so wird die Drehzahl solange erhöht, bis die konstante Schnittgeschwindigkeit erreicht wird.

Bei Erstanwahl von G96, G961 im Teileprogramm muss, bei Wiederanwahl kann eine konstante Schnittgeschwindigkeit in m/min bzw. ft/min eingegeben werden.

Bei G96 schaltet die Steuerung automatisch auf Umdrehungsvorschub (wie bei G95) um, d. h. der programmierte Vorschub F wird in mm/Umdrehung bzw. inch/Umdrehung interpretiert.

Bei der Programmierung von G961 wird automatisch Linearvorschub angewählt (wie bei G94). Ein programmierter Vorschub F wird in mm/min bzw. inch/min interpretiert.

### Ermittlung der Spindeldrehzahl

Ausgehend von der programmierten Schnittgeschwindigkeit (entweder  $S_{G96}$  oder  $S_{G961}$ ) und der aktuellen kartesischen Position der Planachse (Radius) am TCP ermittelt die Steuerung die Spindeldrehzahl nach folgender Formel:

$$n = \frac{S_{\text{Speed}}}{2 * \pi * r}$$

n:	Spindeldrehzahl
$S_{\text{Speed}}$ :	Programmierte Schnittgeschwindigkeit
$\pi$	Kreiskonstante
r:	Radius (Abstand Drehmitte zum TCP)

Hierbei wird zur Ermittlung des Radius Folgendes angenommen:

- Die Planachsposition 0 im WKS stellt die Drehmitte dar.
- Positionskorrekturen (wie Online-Werkzeugkorrektur, externe Nullpunktverschiebung, \$AA\_OFF, DRF-Verschiebung und Compile-Zyklen) und Positionsanteile durch Kopplungen (z. B. Folgeachse bei TRAIL) werden bei der Radiusermittlung nicht berücksichtigt.

Frames (z. B. programmierbare Frames wie SCALE, TRANS oder ROT) werden bei der Ermittlung der Spindeldrehzahl berücksichtigt und können eine Drehzahländerung bewirken, sofern sich der wirksame Durchmesser am TCP ändert.

### Durchmesserprogrammierung und Bezugsachse für mehrere Planachsen im Kanal:

Eine oder mehrere Planachsen sind gleichzeitig oder getrennt zugelassen und aktivierbar:

- Programmierung und Anzeige in der Bedienoberfläche im Durchmesser
- Zuweisung der angegebenen Bezugsachse mit `SCC [<Achse>]` für konstante Schnittgeschwindigkeit G96, G961, G962

Weitere Informationen siehe Funktionsbeschreibung "P1: Planachsen (Seite 893)".

### Beispiel

$S_{G96} = 230 \text{ m/min}$

- bei  $r = 0,2 \text{ m} \rightarrow n = 183,12 \text{ U/min}$
- bei  $r = 0,1 \text{ m} \rightarrow n = 366,24 \text{ U/min}$

⇒ Je kleiner der Werkstückdurchmesser, desto höher die Drehzahl.

Für G96, G961 oder G962 muss eine Geometrieachse als Planachse definiert werden.

Die Planachse, deren Position die Drehzahl der Masterspindel beeinflusst, wird bestimmt über das kanalspezifische Maschinendatum:

MD20100 \$MC\_DIAMETER\_AX\_DEF (Geometrieachse mit Planachsfunktion)

Die Funktion G96, G961 oder G962 setzt voraus, dass der Maschinennullpunkt und der Werkstücknullpunkt der Planachse sich in Drehmitte der Spindel befinden.

### Konstante Drehzahl (G97, G971)

Mit G97, G971 wird die Funktion "Konstante Schnittgeschwindigkeit" (G96, G961) ausgeschaltet und die zuletzt berechnete Spindeldrehzahl abgespeichert. Bei G97 wird der Vorschub als Umdrehungsvorschub (wie bei G95) interpretiert.

Bei der Programmierung von G971 wird Linearvorschub (wie bei G94) angewählt. Der Vorschub F wird in mm/min bzw. inch/min interpretiert.

Bei aktivem G97, G971 kann durch ein erneutes Programmieren eines S-Wertes eine neue Spindeldrehzahl vorgegeben werden. Die unter G96, G961 programmierte Schnittgeschwindigkeit wird dadurch nicht verändert.

Mit G97, G971 können Drehzahlveränderungen bei Bewegungen in der Planachse ohne Bearbeitung (z. B. Werkzeugmeißel) vermieden werden.

---

#### Hinweis

G96, G961 ist nur beim Bearbeiten am Werkstück aktiv (G1, G2, G3, Spline-Interpolation usw., wo Vorschub F wirkt).

Das Verhalten der Spindeldrehzahl bei aktivem G96, G961 und G0-Sätzen kann definiert werden in dem kanalspezifischen Maschinendatum:

MD20750 ALLOW\_G0\_IN\_G96 (G0-Logik bei G96, G961)

Bei konstanter Schnittgeschwindigkeit G96, G961 kann kein Getriebestufenwechsel stattfinden.

Der Spindelkorrekturschalter wirkt auf die ermittelte Spindeldrehzahl.

Eine DRF-Verschiebung in der Planachse beeinflusst die Spindeldrehzahlsollwertberechnung nicht.

Zu Beginn der Bearbeitung (nach G0) und nach NC-Stop, G60, G09, ... wird für den Bahnstart "nIst = nSoll" abgewartet.

Die Nahtstellensignale "nIst = nSoll" und "Soll Drehzahl begrenzt" werden nicht durch interne Drehzahlvorgaben verändert.

Bei Unterschreitung der Minimaldrehzahl oder beim Erkennen des Signals "Achse /Spindel steht" wird "nIst = nSoll" zurückgesetzt.

Eine begonnene Bahnbearbeitung (G64, Überschleifen) wird nicht unterbrochen.

---

## Drehzahlbegrenzung bei G96, G961

Für die Funktion "Konstante Schnittgeschwindigkeit" kann eine maximale Spindeldrehzahl vorgegeben werden:

- im Settingdatum:  
SD43230 \$SA\_SPIND\_MAX\_VELO\_LIMS (Spindeldrehzahlbegrenzung bei G96/G961)
- im Teileprogramm (für die Masterspindel) mit dem Programmierbefehl LIMS

Der zuletzt geänderte Wert (LIMS oder SD) ist wirksam.

LIMS wirkt bei G96, G961, G97 und kann auf bis zu vier Drehzahlbegrenzungen im Teileprogramm in einem Satz vorgegeben werden. Hierfür kann bei der Teileprogrammanweisung LIMS[<Sn>] die Spindelnummer <Sn> = 1, 2, 3 oder 4 der jeweils möglichen Maserspindel angegeben werden.

---

### Hinweis

Beim Einwechseln des Satzes in den Hauptlauf werden alle programmierten Werte in die Settingdaten SD43230 \$SA\_SPIND\_MAX\_VELO\_LIMS übernommen.

---

Abhängig von dem Maschinendatum:

MD10710 \$MN\_PROG\_SD\_RESET\_SAVE\_TAB[n] (zu aktualisierende Settingdaten) bleibt die geschriebene Drehzahlbegrenzung mit LIMS nach dem Ausschalten der Steuerung gespeichert.

Mit dem erneuten Aktivieren von G96, G961, G97 wird diese Drehzahlbegrenzung wieder aktiviert.

Die maximal zulässige Spindeldrehzahl, die über G26 bzw. über das Settingdatum:

SD43220 \$SA\_SPIND\_MAX\_VELO\_G26 (Maximale Spindeldrehzahl) vorgegeben wurde, kann nicht überschritten werden.

Bei falscher Programmierung, die zu einer Überschreitung einer der Drehzahlgrenzen (G26 bzw. SD43220 \$SA\_SPIND\_MAX\_VELO\_G26) führen würde, wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.1 (Programmierte Drehzahl zu hoch)

Um einen Rundlauf bei großen Teiledurchmessern sicher zu stellen, darf eine minimale Spindeldrehzahl nicht unterschritten werden.

Diese Drehzahl ist über das Settingdatum:

SD43210 \$SA\_SPIND\_MIN\_VELO\_G25 (Minimale Spindeldrehzahl)

und je Getriebestufe mit dem Maschinendatum:

MD35140 \$MA\_GEAR\_STEP\_MIN\_VELO\_LIMIT (Minimaldrehzahl der Getriebestufe) einstellbar.

Die minimale Spindeldrehzahl kann im Teileprogramm mit G25 verändert werden. Bei Programmierung, die zu einer Unterschreitung einer der Drehzahlgrenzen (G25 bzw. SD43220 \$SA\_SPIND\_MAX\_VELO\_G25) führen würde, wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX83.2 (Solldrehzahl zu niedrig)

Ausführliche Informationen zu den Spindeldrehzahlbegrenzungen und zur Wirkungsweise der Settingdaten siehe Funktionsbeschreibung S1: "Spindeln", Kapitel: "Spindelüberwachungen (Seite 1479)".

### Masterspindel Umschaltung bei G96, G961

Beim Wechsel der Masterspindel während aktivem G96, G961 wird die Drehzahl der ehemaligen Masterspindel beibehalten. Dies entspricht einem Übergang von G96 nach G97. Die mit SETMS neu definierte Masterspindel führt die so erzeugte Funktion "Konstante Schnittgeschwindigkeit" aus.

### Alarmer

#### Konstante Schnittgeschwindigkeit G96, G961, G962

- Wird kein F-Wert programmiert, so wird der Alarm 10860 "Kein Vorschub programmiert" ausgegeben. Der Alarm wird nicht bei G0-Sätzen erzeugt.
- Bei Programmierung einer negativen Bahngeschwindigkeit wird der Alarm 14800 "Programmierte Bahngeschwindigkeit kleiner oder gleich Null" ausgegeben.
- Ist bei aktivem G96, G961 oder G962 keine Planachse im Maschinendatum: MD20100 \$MC\_DIAMETER\_AX\_DEF (Geometrieachsen mit Planachsenfunktion) definiert, so wird der Alarm 10870 "keine Planachse definiert" ausgegeben.
- Wird bei aktivem G96, G961 eine negative maximale Spindeldrehzahl mit dem Programmierbefehl `LIMS` programmiert, so wird der Alarm 14820 "negative maximale Spindeldrehzahl für G96, G961 programmiert" ausgegeben.
- Ist bei erstmaliger Anwahl von G96, G961 keine konstante Schnittgeschwindigkeit programmiert, so wird der Alarm 10900 "kein S-Wert für konstante Schnittgeschwindigkeit programmiert" ausgegeben.

## 16.2.3 Vorschub bei Gewindschneiden (G33, G34, G35)

### G33

Mit der Funktion G33 können Gewinde mit konstanter Steigung bearbeitet werden.

#### Drehzahl S, Vorschub F, Gewindesteigung

Bei G33-Gewinden wirkt ein Umdrehungsvorschub [mm/Umdrehung], der durch die Programmierung der Gewindesteigung [mm/Umdrehung] vorgegeben wird.

Die Geschwindigkeit der Achsen für die Gewindelänge errechnet sich aus der programmierten Spindeldrehzahl S und der Gewindesteigung:

$$\text{Vorschub } F \text{ [mm/min]} = \text{Drehzahl } S \text{ [U/min]} * \text{Gewindesteigung [mm/U]}$$

Am Ende der Beschleunigungsrampe wird die Lagekopplung vom Spindelwert (Spindelsollwert bei SPCON der Masterspindel) mit dem Achssollwert hergestellt. Die Achsposition steht zu diesem Zeitpunkt in Bezug auf die Nullmarke (Nullmarkenverschiebungen berücksichtigt) der Spindel so, als wäre die Achse am Satzanfangspunkt bei Überfahren der Gewindestartposition (Nullmarke plus SF) schlagartig beschleunigt worden. Der Schleppabstand der Achse wird dabei kompensiert.

## Minimale Spindeldrehzahl

Um einen Rundlauf bei geringen Drehzahlen sicher zu stellen, darf eine minimale Spindeldrehzahl nicht unterschritten werden.

Diese Drehzahl ist einstellbar:

- mit dem Settingdatum:  
SD43210 \$SA\_SPIND\_MIN\_VELO\_G25 (Minimale Spindeldrehzahl)
- je Getriebestufe mit dem Maschinendatum:  
MD35140 \$MA\_GEAR\_STEP\_MIN\_VELO\_LIMIT (Minimaldrehzahl für Getriebestufenwechsel)

Die minimale Spindeldrehzahl kann im Teileprogramm mit G25 verändert werden.

## NC-Stop, Einzelsatz

NC-Stop und Einzelsatz (auch an Satzgrenze) wirken erst am Ende einer Gewindekettung. Dabei werden alle aufeinander folgenden G33-Sätze und der erste darauf folgende Nicht-G33-Satz wie ein Satz abgefahren.

## Vorzeitiger zerstörungsfreier Abbruch

Es ist möglich, das Gewindeschneiden vor Erreichen des Endpunkts zerstörungsfrei zu unterbrechen. Dies kann durch die Aktivierung einer Rückzugsbewegung erfolgen.

## Gewindeschneiden mit ROT-Frame

Bei ROT-Frame und G33, G34, G35 wird Alarm 10607 "Gewinde mit Frame nicht ausführbar" gemeldet, wenn durch die Rotation die Gewindelänge und damit die Steigung verändert wird. Rotationen um die Gewindeachse sind zulässig.

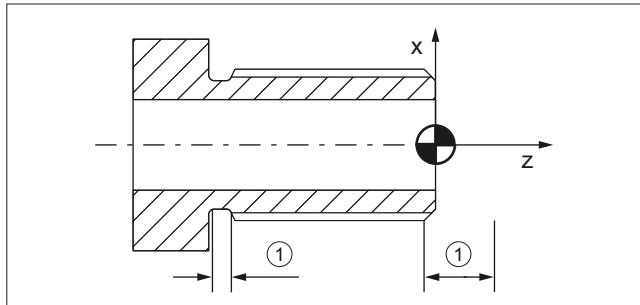
Der Alarm 10607 "Gewinde mit Frame nicht ausführbar" kann durch Setzen von Bit 12 im Maschinendatum MD11410 \$MN\_SUPPRESS\_ALARM\_MASK unterdrückt werden, falls in der Anwendung bewusst die ROT-Anweisung verwendet wird.

Alle weiteren Frames werden von der NC ohne Alarm akzeptiert. Auf die steigungsändernde Wirkung von SCALE wird hingewiesen.

### 16.2.3.1 Programmierbarer Einlauf- und Auslaufweg bei G33, G34 und G35

#### Funktion

Mit den Anweisungen `DITS` und `DITE` kann der Einlauf- bzw. Auslaufweg des Gewindes vorgegeben werden. Die Gewindeachse wird innerhalb des vorgegebenen Wegs beschleunigt bzw. gebremst.



① Ein- bzw. Auslaufweg, je nach Bearbeitungsrichtung

#### Kurzer Einlaufweg

Durch den Bund am Gewindeeinlauf ist wenig Platz für die Werkzeug-Startrampe. Diese muss deshalb über `DITS` kürzer vorgegeben werden.

#### Kurzer Auslaufweg

Durch den Bund am Gewindeauslauf ist wenig Platz für die Bremsrampe, wodurch Kollisionsgefahr zwischen Werkstück und Schneide besteht. Die Bremsrampe kann über `DITE` kürzer vorgegeben werden. Wegen der Trägheit der Mechanik kann es dennoch zur Kollision kommen.

Abhilfe: Gewinde kürzer programmieren, Spindeldrehzahl reduzieren.

---

#### Hinweis

`DITE` wirkt am Gewindeende als Überschleifabstand. Damit wird eine stoßfreie Änderung der Achsbewegung erreicht.

---

#### Auswirkungen

Der programmierte Einlauf- und Auslaufweg wirkt auf die Bahn ausschließlich beschleunigungssteigernd. Wird einer der beiden Wege größer vorgegeben, als er für die Gewindeachse mit aktiver Beschleunigung benötigt wird, wird die Gewindeachse mit maximaler Beschleunigung beschleunigt bzw. gebremst.

---

#### Hinweis

Bei der Vorgabe eines sehr kurzen Wegs kann die Achse überlastet werden.

---



## Aktivierung

Die Anweisungen `DITS` und `DITE` sind beim Gewindeschneiden immer aktiv.

## Beispiel

Programmcode	Kommentar
N...	
N59 G90 G0 Z100 X10 SOFT M3 S500	
N60 G33 Z50 K5 SF=180 DITS=1 DITE=3	; Überschleifbeginn bei Z=53
N61 G0 X20	

### Hinweis

Unter `DITS` und `DITE` werden ausschließlich Wege programmiert, keine Positionen!

Der programmierte Ein-/Auslaufweg wird entsprechend der aktuellen Maßeinstellung (Inch, metrisch) behandelt.

## Settingdatum SD42010 (Beschleunigungsverhalten der Achse beim Gewindeschneiden)

Mit dem Einwechseln eines Satzes mit `DITS` und/oder `DITE` in den Hauptlauf wird der programmierte Ein- bzw. Auslaufweg in das Settingdatum übernommen:

- SD42010 \$SC\_THREAD\_RAMP\_DISP[ 0 ] = programmierter Wert von `DITS`
- SD42010 \$SC\_THREAD\_RAMP\_DISP[ 1 ] = programmierter Wert von `DITE`

Wird vor oder im ersten Gewindegang kein Ein- bzw. Auslaufweg programmiert, wird jeweils der aktuelle Wert des Settingdatums verwendet.

SD42010 \$SC\_THREAD\_RAMP\_DISP[ 0 ] = <Wert>

SD42010 \$SC\_THREAD\_RAMP\_DISP[ 1 ] = <Wert>

<Wert>	Bedeutung
0 > <Wert> ≥ -1	Das Beschleunigen der Vorschubachse erfolgt mit der Beschleunigung entsprechend der aktuellen Programmierung <code>BRISK</code> / <code>SOFT</code>
<Wert> = 0	Das Beschleunigen der Vorschubachse erfolgt sprungförmig ( <code>BRISK</code> ).
<Wert> > 0	Es wird der maximale Gewindehochlauf- bzw. Bremsweg vorgegeben. <b>Hinweis</b> Ein zu kurzer Weg kann zu einer Beschleunigungsüberlastung der Achse führen.

Bei Reset / Programmende wird das Settingdatum auf die Werte (-1, -1) zurückgesetzt.

Über MD10710 \$MN\_PROG\_SD\_RESET\_SAVE\_TAB kann eingestellt werden, dass die mit `DITS` und `DITE` geschriebene Wert des Settingdatums bei Reset / Programmende remanent gespeichert werden und somit über Power On erhalten bleiben.

### 16.2.3.2 Linear progressive/degressive Gewindesteigungsänderung bei G34 und G35

#### Funktion

Die Gewindesteigungszunahme (G34) beschreibt die zahlenmäßige Vergrößerung des Steigungswerts. Eine größere Steigung bewirkt einen größeren Abstand zwischen den Gewindegängen auf dem Werkstück. Die Geschwindigkeit der Gewindeachse nimmt also bei angenommener konstanter Spindeldrehzahl zu.

Für die Gewindesteigungsabnahme (G35) gilt sinngemäß das Gegenteil.

Folgende Begriffsbestimmungen werden für die Gewindesteigungsänderung getroffen:

- G34: Gewindesteigungszunahme entspricht progressiver Änderung
- G35: Gewindesteigungsabnahme entspricht degressiver Änderung

Beide Funktionen G34 und G35 implizieren die Funktionalität von G33 und bieten zusätzlich die Möglichkeit unter F eine Steigungsänderung des Gewindes betragsmäßig zu programmieren. Ist die Anfangs- und Endsteigung eines Gewindes bekannt, dann kann die zu programmierende Gewindesteigungsänderung nach folgender Gleichung ermittelt werden:

$$F = \frac{|k_e^2 - k_a^2|}{2 * l_G}$$

Dabei bedeuten:

F: die zu programmierende Gewindesteigungsänderung [mm/U<sup>2</sup>]

k<sub>e</sub>: Gewindesteigung der Achszielpunktcoordinate Gewindeachse [mm/U]

k<sub>a</sub>: Gewindeanfangssteigung (unter I, J oder K programmiert) [mm/U]

l<sub>G</sub>: Gewindelänge [mm]

Der Betrag von F ist entsprechend der gewünschten Steigungszunahme oder Steigungsabnahme auf G34 bzw. G35 anzuwenden.

Bei bekannter Gewindelänge  $l_G$ , Steigungsänderung  $F$  und Anfangssteigung  $k_a$  kann die Gewindesteigung am Satzende  $k_e$  durch Umstellen der Gleichung wie folgt ermittelt werden:

- Bei G34 (zunehmende Steigerung):

$$k_e = \sqrt{k_a^2 + F * 2 * l_G}$$

- Bei G35 (abnehmende Steigung):

$$k_e = \sqrt{k_a^2 - F * 2 * l_G}$$

### Hinweis

Ergibt sich bei der Formel ein negativer Wurzelausdruck, dann kann das Gewinde nicht gefertigt werden!

Die NC meldet in diesem Fall den Alarm 10605 bzw. den Alarm 22275.

### Anwendung

Die Funktionen G34 und G35 können zur Realisierung von selbstscherenden Gewinden eingesetzt werden.

### Beispiel

Gewindeschneiden G33 mit abnehmender Gewindesteigung G35

Programmcode	Kommentar
N1608 M3 S10	; Spindeldrehzahl
N1609 G0 G64 Z40 X216	; Startpunkt anfahren
N1610 G33 Z0 K100 SF=R14	; Gewinde mit konstanter Steigung 100 mm/U
N1611 G35 Z-220 K100 F17.045455	; Gewindesteigungsabnahme 17.045455 mm/U2
	; Gewindesteigung am Satzende 50 mm/U
N1612 G33 Z-240 K50	; Gewindegang ohne Ruck fahren
N1613 G0 X218	
N1614 G0 Z40	
N1616 M17	

### Überwachung während der Satzaufbereitung

Steigungsänderungen, welche bei G34 die Gewindeachse überlasten oder bei G35 einen Achsstillstand bewirken würden, werden frühzeitig bei der Satzaufbereitung erkannt. Es wird der Alarm 10604 "Gewindesteigungszunahme zu hoch" bzw. 10605 "Gewindesteigungsabnahme zu hoch" gemeldet.

Bestimmte praxisrelevante Anwendungen erfordern beim Gewindeschneiden eine Korrektur der Spindeldrehzahl. Der Bediener orientiert sich in diesem Fall an der zulässigen Geschwindigkeit der Gewindeachse.

Dazu ist es möglich, die Ausgabe der Alarme 10604 und 10605 wie folgt zu unterdrücken:

MD11410 \$MN\_SUPPRESS\_ALARM\_MASK Bit 10 = 1

Die Satzaufbereitung wird dann normal fortgesetzt.

### Überwachung während der Abarbeitung

Während der Abarbeitung (Interpolation) des Gewindes werden folgende Situationen zyklisch überwacht:

- Überschreiten der Maximalgeschwindigkeit der Gewindeachse.
- Erreichen des Achsstillstandes bei G35.

Bei Ansprechen der Überwachung wird folgender Alarm gemeldet:

- Alarm 22270 "Maximaler Geschwindigkeit der Gewindeachse erreicht" bzw.
- Alarm 22275 "Geschwindigkeit Null der Gewindeachse erreicht"

### 16.2.3.3 Schnelrückzug während Gewindeschneiden

#### Funktion

Die Funktion "Schnelrückzug für Gewindeschneiden (G33)" ermöglicht eine zerstörungsfreie Unterbrechung des Gewindeschneidens bei:

- NC-Stop (NC/PLC-Nahtstellensignal)
- Alarme, die implizit NC-Stop auslösen
- Schalten eines schnellen Eingangs

#### Literatur

Programmierhandbuch Arbeitsvorbereitung, Kapitel "Schnellabheben von der Kontur"

Die Rückzugsbewegung ist programmierbar über:

- Rückzugweg und Rückzugsrichtung (relativ)
- Rückzugsposition (absolut)

---

#### Hinweis

#### Gewindebohren

Die Funktion "Schnelrückzug" ist **nicht** anwendbar beim Gewindebohren (G331 / G332).

---

## Programmierung

### Syntax

Schnellrückzug freigeben, Rückzugsbewegung über Rückzugweg und Rückzugsrichtung:

```
G33 ... LFON DILF=<Wert> LFTXT/LFWP ALF=<Wert>
```

Schnellrückzug freigeben, Rückzugsbewegung über Rückzugsposition:

```
POLF [<Achsebezeichner>]=<Wert> LFPOS
POLFMASK/POLFMLIN (<Achse1>, <Achse2>, ...)
G33 ... LFON
```

Schnellrückzug für Gewindeschneiden sperren:

```
LFOF
```

### Bedeutung

LFON:	Schnellrückzug für Gewindeschneiden (G33) freigeben
LFOF:	Schnellrückzug für Gewindeschneiden (G33) sperren
DILF= :	Länge des Rückzugswegs festlegen Der durch MD-Projektierung (MD21200 \$MC_LIFFAST_DIST) voreingestellte Wert kann im Teileprogramm durch Programmierung von DILF verändert werden.
	<b>Hinweis:</b> Nach NC-RESET ist immer der projektierte MD-Wert aktiv.
LFTXT	Die Rückzugsrichtung wird in Verbindung mit ALF mit den G-Funktionen LFTXT und LFWP gesteuert.
LFWP:	
	LFTXT: Die Ebene, in welcher die Rückzugsbewegung ausgeführt wird, wird aus der Bahntangente und der Werkzeugrichtung errechnet (Standardeinstellung).
	LFWP: Die Ebene, in welcher die Rückzugsbewegung ausgeführt wird, ist die aktive Arbeitsebene.
ALF= :	In der Ebene der Rückzugsbewegung wird mit ALF die Richtung in diskreten Schritten von Grad programmiert. Bei LFTXT ist für ALF=1 der Rückzug in Werkzeugrichtung festgelegt. Bei LFWP ergibt sich die Richtung in der Arbeitsebene nach folgender Zuordnung:
	<ul style="list-style-type: none"> <li>• G17 (X/Y-Ebene) <ul style="list-style-type: none"> <li>ALF=1 ; Rückzug in X-Richtung</li> <li>ALF=3 ; Rückzug in Y-Richtung</li> </ul> </li> <li>• G18 (Z/X-Ebene) <ul style="list-style-type: none"> <li>ALF=1 ; Rückzug in Z-Richtung</li> <li>ALF=3 ; Rückzug in X-Richtung</li> </ul> </li> <li>• G19 (Y/Z-Ebene) <ul style="list-style-type: none"> <li>ALF=1 ; Rückzug in Y-Richtung</li> <li>ALF=3 ; Rückzug in Z-Richtung</li> </ul> </li> </ul>

**Literatur:**

Zu den Programmiermöglichkeiten mit ALF siehe auch Kapitel "Verfahrriichtung beim Schnellabheben von der Kontur" im Programmierhandbuch Arbeitsvorbereitung.

LFPOS: Rückzug der mit POLFMASK oder POLFMLIN bekannt gemachten Achse auf die mit POLF programmierte absolute Achsposition

POLFMASK: Freigabe der Achsen (<Achsname1>, <Achsname1>, ...) für den unabhängigen Rückzug auf Absolutposition

POLFMLIN: Freigabe der Achsen für den Rückzug auf Absolutposition im linearen Zusammenhang

**Hinweis:**

Der lineare Zusammenhang kann je nach dynamischen Verhalten aller beteiligten Achsen bis zum Erreichen der Abhebeposition nicht immer hergestellt werden.

POLF []: Absolute Rückzugsposition für die im Index angegebene Geometrieachse bzw. Maschinenachse festlegen

Wirksamkeit: modal

=<Wert>: Bei Geometrieachsen wird der zugewiesene Wert als Position im Werkstückkoordinatensystem (WKS) interpretiert, bei Maschinenachsen als Position im Maschinenkoordinatensystem (MKS).

Die Wertzuweisung ist auch als Kettenmaßangabe programmierbar:

=IC<Wert>

<Achsbezeichner>: Bezeichner einer Geometrie- oder Maschinenachse

**Hinweis**

LFON bzw. LFOF können immer programmiert werden, die Auswertung erfolgt jedoch ausschließlich beim Gewindeschneiden (G33).

**Hinweis**

POLF mit POLFMASK/POLFMLIN sind nicht auf den Einsatz bei Gewindeschneiden beschränkt.

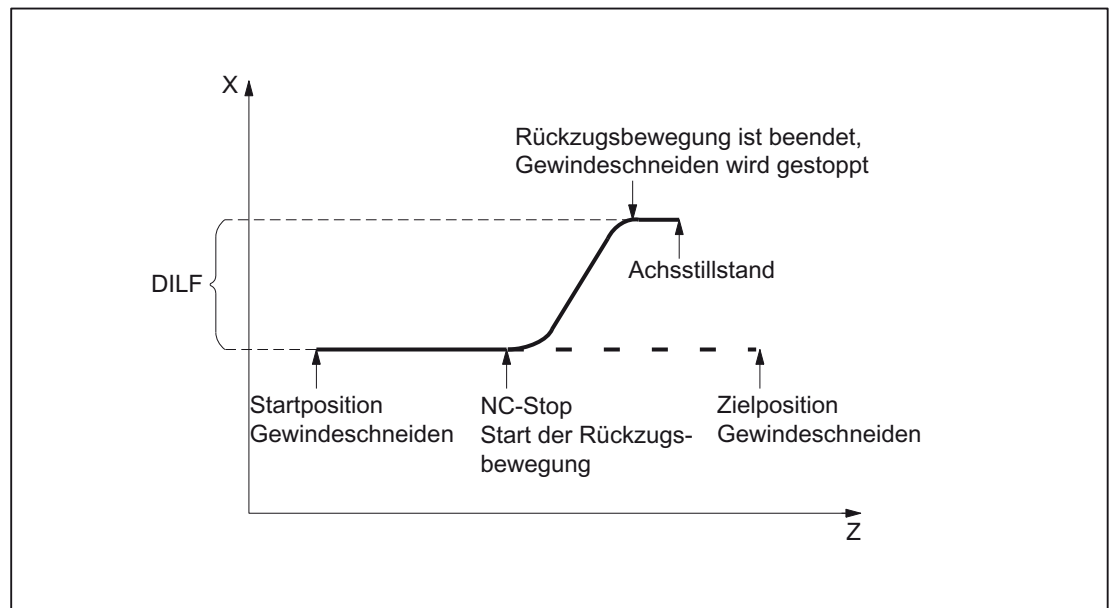


Bild 16-1 Unterbrechung von G33 durch Rückzugsbewegung

### Dynamik der Rückzugsbewegung

Die Rückzugsbewegung wird mit maximaler Achsdynamik ausgeführt:

- MD32000 \$MA\_MAX\_AX\_VELO[<Achse>] (Geschwindigkeit)
- MD32300 \$MA\_MAX\_AX\_ACCEL[<Achse>] (Beschleunigung)
- MD32431 \$MA\_MAX\_AX\_JERK[<Achse>] (Ruck)

## Beispiel

Programmcode	Kommentar
N55 M3 S500 G90 G18	; Aktive Bearbeitungsebene setzen.
...	
N65 MSG ("Gewindeschneiden")	
MM_THREAD:	
N67 \$AC_LIFTFAST=0	; Vor Beginn des Gewindes zurücksetzen.
N68 G0 Z5	
N69 X10	
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7	; Schnellrückzug für Gewindeschneiden freigeben. ; Rückzugsweg=10 mm ; Rückzugsebene Z/X (wegen G18) ; Rückzugsrichtung -X (mit ALF=3 Rückzugsrichtung +X)
N71 G33 Z55 X15	
N72 G1	; Gewindeschneiden abwählen.
N69 IF \$AC_LIFTFAST GOTOB MM_THREAD	; Wenn Gewindeschneiden unterbrochen wurde.
N90 MSG ("")	
...	
N70 M30	
N55 M3 S500 G90 G0 X0 Z0	
...	
N87 MSG ("Gewindebohren")	
N88 LFOF	; Schnellrückzug vor Gewindebohren ausschalten.
N89 CYCLE...	; Gewindebohrzyklus mit G33.
N90 MSG ("")	
...	
N99 M30	

## Verhalten bei Power On und Reset

Nach Power On und Reset werden folgende Einstellungen aktiv:

- Löschstellungen für die Rückzugsbewegung ( $LFON / LFOF$ ) und Rückzugsrichtung ( $LFTXT / LFWP$ ): MD20150 \$MC\_GCODE\_RESET\_VALUES
- Rückzugsweg: MD21200 \$MC\_LIFTFAST\_DIST



## 16.2.4 Vorschub bei Gewindebohren ohne Ausgleichsfutter (G331, G332)

### Funktion

Mit den Funktionen G331 (Gewindebohren) und G332 (Gewindebohren Rückzug) kann ein Gewinde ohne Ausgleichsfutter gebohrt werden.

### Voraussetzung

Technische Voraussetzung für das Gewindebohren ohne Ausgleichsfutter ist eine lagegeregelte Spindel mit Wegmesssystem.

### Drehzahl S, Vorschub F, Gewindesteigung

Bei G331 und G332 wirkt ein Umdrehungsvorschub [mm/Umdr.], der durch die Programmierung der Gewindesteigung [mm/Umdr.] vorgegeben wird.

Die Geschwindigkeit der Achsen für die Gewindelänge errechnet sich aus der programmierten Spindeldrehzahl S und der Gewindesteigung:

Vorschub F [mm/min] = Drehzahl S [U/min] \* Gewindesteigung [mm/U]

### Override

Der Umdrehungsvorschub bei G331 und G332 kann durch einen Override beeinflusst werden.

Je nach Projektierung wirkt der Override dabei entweder auf die Spindeldrehzahl oder auf den Bahnvorschub:

MD12090 \$MN\_OVR\_FUNCTION\_MASK

Bit	Wert	Bedeutung
0	0	Der Override wirkt auf die Spindeldrehzahl (Grundstellung). Abhängig von der Einstellung im Maschinendatum: MD12080 \$MN_OVR_REFERENCE_IS_PROG_FEED bezieht sich der Override entweder auf die programmierte Spindeldrehzahl oder auf die projektierte Spindeldrehzahlbegrenzung.
	1	Der Override wirkt auf den Bahnvorschub Abhängig von der Einstellung im Maschinendatum: MD12082 \$MN_OVR_REFERENCE_IS_MIN_FEED bezieht sich der Override entweder auf den programmierten Bahnvorschub auf die projektierte Bahnvorschubbegrenzung.

---

**Hinweis**

Folgende Korrekturen sind bei G331 und G332 nicht wirksam:

- Programmierbare Bahnvorschubkorrektur OVR
  - Eilgangkorrektur
- 

**Stopp-Ereignisse bei G331/G332 verhindern**

Innerhalb des Gewindebohrers kann, wenn der Satz eine Bahnbewegung oder ein G4 enthält, ein Stopp wie folgt verhindert werden:

MD11550 \$MN\_STOP\_MODE\_MASK, Bit 0 = 0

Nachdem G331/G332 abgearbeitet wurde, ist der vorher aktivierte Stopp wieder möglich.

**Literatur**

Weitere Informationen zu G331/G332 siehe Programmierhandbuch Grundlagen.

## 16.2.5 Vorschub bei Gewindebohren mit Ausgleichsfutter (G63)

### Funktion

G63 ist eine Teilfunktion zum Bohren von Gewinden mit einem Gewindebohrer mit Ausgleichsfutter. Ein Geber (Wegmesssystem) ist hierbei nicht erforderlich.

### Drehzahl S, Vorschub F, Gewindesteigung

Bei G63 muss für die Spindel eine Drehzahl S und für die Zustellachse (Achse für Gewindelänge) ein Vorschub F programmiert werden.

Der Vorschub F ist vom Programmierer aus der Drehzahl S und der Gewindesteigung zu errechnen:

Vorschub F [mm/min] = Drehzahl S [U/min] \* Gewindesteigung [mm/U]

### Literatur

Weitere Informationen zu G63 siehe Programmierhandbuch Grundlagen.

## 16.3 Vorschub für Positionierachsen (FA)

### Funktion

Über den achsspezifischen Vorschub `FA` wird die Geschwindigkeit einer Positionierachse programmiert.

FA ist modal wirksam.

Es gilt immer die Vorschubart G94.

---

### Hinweis

Die maximale Achsgeschwindigkeit (MD32000 `$MA_MAX_AX_VELO`) wird nicht überschritten.

---

### Programmierung

Pro Teileprogrammsatz können max. 5 achsspezifische Vorschübe programmiert werden.

#### Syntax:

`FA[<Positionierachse>]=<Vorschubwert>`

<code>&lt;Positionierachse&gt;</code> :	Bezeichner der Kanalachse (MD20080 <code>\$MC_AXCONF_CHANAX_NAME_TAB</code> )
<code>&lt;Vorschubwert&gt;</code> :	Vorschubgeschwindigkeit Wertebereich: 0,001...999 999,999 mm/min, Grad/min bzw. 0,001...39 999,9999 inch/min

### Voreinstellung

Wird kein axialer Vorschub `FA` programmiert, wirkt die axiale Voreinstellung:

MD32060 `$MA_POS_AX_VELO` (Löschstellung für Positionierachsgeschwindigkeit)

## Ausgabe an die PLC

Der Vorschubwert kann an die PLC ausgegeben werden:

- an die Kanal-spezifische NC/PLC-Nahtstelle über:  
DB21, ... DBB158 - DBB193
- an die Achs-spezifisch NC/PLC-Nahtstelle über:  
DB31, ... DBB78 - DBB81

Der Ausgabezeitpunkt wird festgelegt mit dem Maschinendatum:

MD22240 \$MC\_AUXFU\_F\_SYNC\_TYPE (Ausgabezeitpunkt der F-Funktionen)

In der Grundeinstellung ist die Ausgabe unterdrückt (MD22240 = 3), da es durch die Ausgabe der F-Funktionen an die NC/PLC-Nahtstelle im Bahnsteuerbetrieb zu Geschwindigkeitseinbrüchen kommen kann.

Weitere Informationen siehe Funktionsbeschreibung "H2: Hilfsfunktionsausgaben an PLC (Seite 395)".

## Reset-Verhalten

Das Verhalten nach Programmende oder NC-Reset wird festgelegt durch das Maschinendatum:

MD22410 \$MC\_F\_VALUES\_ACTIVE\_AFTER\_RESET (F-Funktion über Reset hinaus wirksam)

Wert	Bedeutung
0	Nach NC-Reset sind die Standardwerte wirksam.
1	Nach NC-Reset sind die zuletzt programmierten FA-Werte wirksam.

## 16.4 Vorschubbeeinflussung

### 16.4.1 Vorschubsperrung und Vorschub/Spindel Halt

#### Funktion

Bei "Vorschubsperrung" oder "Vorschub/Spindel Halt" werden die Achsen unter Einhaltung der Bremskennlinie zum Stillstand gebracht. Die Bahnkontur wird eingehalten (Ausnahme: G33-Satz).

#### Kanalspezifische Vorschubsperrung

Über das NC/PLC-Nahtstellensignal:  
DB21, ... DBX6.0 (Vorschubsperrung)  
werden alle Achsen (Geometrie- und Zusatzachsen) eines Kanals in allen Betriebsarten stillgesetzt.

Wirksamkeit der kanalspezifischen Vorschubsperrung:	
• Bei aktivem G33, G34, G35:	nicht wirksam
• Bei aktivem G63:	wirksam
• Bei aktivem G331, G332:	wirksam

#### "Vorschub Halt" für Geometrieachsen im JOG-Betrieb

Über die NC/PLC-Nahtstellensignale:  
DB21, ... DBX12.3 (Vorschub Halt Geometrieachse 1)  
DB21, ... DBX16.3 (Vorschub Halt Geometrieachse 2)  
DB21, ... DBX20.3 (Vorschub Halt Geometrieachse 3)  
werden die entsprechenden Geometrieachsen eines Kanals im JOG-Betrieb stillgesetzt.

### Achsspezifischer "Vorschub Halt"

Über das achsspezifische NC/PLC-Nahtstellensignal:  
DB31, ... DBX4.3 (Vorschub Halt)  
wird die jeweilige Maschinenachse stillgesetzt.

Bei Automatikbetrieb gilt:

- Erfolgt "Vorschub Halt" für eine Bahnachse, so werden alle im aktuellen Satz bewegten und am Bahnverbund beteiligten Achsen stillgesetzt.
- Erfolgt "Vorschub Halt" für eine Positionierachse, so wird nur diese Achse stillgesetzt.

Bei JOG-Betrieb wird nur die jeweilige Achse stillgesetzt.

Wirksamkeit des achsspezifischen "Vorschub Halt":	
• Bei aktivem G33, G34, G35:	wirksam (hierbei entstehen Konturabweichungen)
• Bei aktivem G63:	wirksam
• Bei aktivem G331, G332:	wirksam

### Achsen-/Spindelsperre

Bei aktiver "Achsen-/Spindelsperre":

DB31, ... DBX1.3 = 1

wirken die axialen PLC-Verriegelungen "keine Reglerfreigabe" oder "Vorschub Halt" **nicht**.

Der axiale und kanalspezifische Override sind dagegen wirksam.

### "Spindel Halt"

Über das NC/PLC-Nahtstellensignal:

DB31, ... DBX4.3 (Spindel Halt)

wird die jeweilige Spindel stillgesetzt.

Wirksamkeit der Funktion "Spindel Halt":	
• Bei aktivem G33, G34, G35:	wirksam (hierbei können, abhängig von den dynamischen Kennwerten, Konturabweichungen entstehen)
• Bei aktivem G63:	wirksam
• Bei aktivem G331, G332:	<b>nicht</b> wirksam

## 16.4.2 Vorschubkorrektur über Maschinensteuertafel

### Funktion

Mit der "Vorschubkorrektur über Maschinensteuertafel" kann der Bediener vor Ort und mit sofortiger Wirkung an der Maschine den Bahnvorschub prozentual verringern bzw. erhöhen. Dazu werden die programmierten Vorschübe mit den entsprechenden an der NC/PLC-Nahtstelle vorliegenden Korrekturwerten multipliziert.

Für Positionierachsen ist der Vorschub achsspezifisch veränderbar.

Mit der "Spindelkorrektur" kann die Spindeldrehzahl und die Schnittgeschwindigkeit (G96, G961) verändert werden.

Bei einer Vorschubänderung werden die axialen Beschleunigungs- und Geschwindigkeitsgrenzen eingehalten. Auf der Bahn entstehen keine Konturfehler.

Die Vorschubkorrektur kann für Bahn- und Positionierachsen getrennt verändert werden.

Die Korrekturen wirken auf die programmierten Werte bzw. auf die Begrenzungen (z. B. G26, LIMS bei Spindeldrehzahl).

### Kanalspezifische Vorschub- und Eilgangkorrektur

Für Vorschub- und Eilgangkorrektur stehen in der NC/PLC-Nahtstelle eigene Freigabesignale und Korrekturfaktoren zur Verfügung:

DB21, ... DBX6.7 (Vorschubkorrektur wirksam)

DB21, ... DBB4 (Vorschubkorrektur)

DB21, ... DBX6.6 (Eilgangkorrektur wirksam)

DB21, ... DBB5 (Eilgangkorrektur)

Die Korrekturfaktoren können von der PLC binär- oder graycodiert vorgegeben werden. Das vorliegende Format wird der NC über folgende Maschinendaten bekannt gemacht:

MD12020 \$MN\_OVR\_FEED\_IS\_GRAY\_CODE (Bahnvorschub-Korrekturschalter graycodiert)

MD12040 \$MN\_OVR\_RAPID\_IS\_GRAY\_CODE (Eilgang-Korrekturschalter graycodiert)

Bei Binärcode gilt folgende feste Zuordnung:

Binärcode	Dezimal	Korrekturfaktor
00000000	0	0,00 $\pm$ 0%
00000001	1	0,01 $\pm$ 1%
00000010	2	0,02 $\pm$ 2%
00000011	3	0,03 $\pm$ 3%
00000100	4	0,04 $\pm$ 4%
...	...	...
01100100	100	1,00 $\pm$ 100%
...	...	...
11001000	200	2,00 $\pm$ 200%



Bei Graycodierung sind in den folgenden Maschinendaten die der Schalterstellung entsprechenden Korrekturfaktoren einzutragen:

MD12030 \$MN\_OVR\_FACTOR\_FEEDRATE [<n>] (Bewertung des Bahnvorschub-Korrekturschalters)

MD12050 \$MN\_OVR\_FACTOR\_RAPID\_TRA [<n>] (Bewertung des Eilgangkorrekturschalters)

Eine wirksame Vorschubkorrektur wirkt auf alle Bahnachsen, die dem aktuellen Kanal zugeordnet sind. Eine wirksame Eilgangkorrektur wirkt auf alle Achsen, die mit Eilgangverfahren und dem aktuellen Kanal zugeordnet sind.

### Kein Eilgangkorrekturschalter vorhanden

Ist kein eigener Eilgangkorrekturschalter vorhanden, kann zwischen Eilgang- und Vorschubkorrektur umgeschaltet werden. Welche Korrektur wirksam sein soll, kann über PLC oder Bedientafelfront angewählt werden. Bei aktiver Eilgangkorrektur werden die Vorschubkorrekturwerte auf 100% begrenzt.

- Bei Aktivierung der Eilgangkorrektur über die Bedientafelfront wird durch das PLC-Grundprogramm:
  - die Anwahl der Vorschubkorrektur für Eilgang auf das Aktivierungssignal für die Eilgangkorrektur übertragen:
   
DB21, ... DBX6.6 = DB21, ... DBX25.3
  - der Vorschubkorrekturwert in den Eilgangkorrekturwert übertragen:
   
DB21, ... DBB5 = DB21, ... DBB4
- Bei Anwahl der Eilgangkorrektur über die PLC ist durch das PLC-Anwenderprogramm:
  - das Aktivierungssignal für die Eilgangkorrektur zu setzen:
   
DB21, ... DBX6.6 = 1
  - der Vorschubkorrekturwert in den Eilgangkorrekturwert zu übertragen:
   
DB21, ... DBB5 = DB21, ... DBB4

Wirksamkeit der kanalspezifischen Vorschub- und Eilgangkorrektur:	
• Bei aktivem G33, G34, G35:	nicht wirksam
• Bei aktivem G63:	nicht wirksam
• Bei aktivem G331, G332:	nicht wirksam

### Bezugsgeschwindigkeit für Bahnvorschubkorrektur

Die Bezugsgeschwindigkeit für die "Bahnvorschubkorrektur über Maschinensteuertafel" kann abweichend vom Standard (= programmierter Vorschub) gesetzt werden:

MD12082 \$MN\_OVR\_REFERENCE\_IS\_MIN\_FEED

### Achsspezifische Vorschubkorrektur

Für jede Positionierachse sind in der NC/PLC-Nahtstelle ein Freigabesignal und ein Byte für den Vorschubkorrekturfaktor vorhanden:

DB31, ... DBX1.7 (Korrektur wirksam)

DB31, ... DBB0 (Vorschubkorrektur)

Der Korrekturfaktor kann von der PLC binär- oder graycodiert vorgegeben werden. Das vorliegende Format wird der NC über folgendes Maschinendatum bekannt gemacht:

MD12000 \$MN\_OVR\_AX\_IS\_GRAY\_CODE (Achsvorschubkorrekturschalter graycodiert)

Bei Binärcode gilt folgende feste Zuordnung:

Binärcode	Dezimal	Korrekturfaktor
00000000	0	0,00 ± 0%
00000001	1	0,01 ± 1%
00000010	2	0,02 ± 2%
00000011	3	0,03 ± 3%
00000100	4	0,04 ± 4%
...	...	...
01100100	100	1,00 ± 100%
...	...	...
11001000	200	2,00 ± 200%

Bei Graycodierung sind in das folgende Maschinendatum die der Schalterstellung entsprechenden Korrekturfaktoren einzutragen:

MD12010 \$MN\_OVR\_FACTOR\_AX\_SPEED [<n>] (Bewertung des Achsvorschubkorrekturschalters)

Wirksamkeit der achsspezifischen Vorschubkorrektur:	
• Bei aktivem G33, G34, G35:	nicht wirksam
• Bei aktivem G63:	nicht wirksam (die Korrektur wird in der NC fest auf 100% gesetzt)
• Bei aktivem G331, G332:	nicht wirksam (die Korrektur wird in der NC fest auf 100% gesetzt)

### Spindelkorrektur

Für jede Spindel stehen in der NC/PLC-Nahtstelle ein Freigabesignal und ein Byte für den Spindelkorrekturfaktor zur Verfügung:

DB31, ... DBX1.7 (Korrektur wirksam)

DB31, ... DBB19 (Spindelkorrektur)

Der Korrekturfaktor kann von der PLC binär- oder graycodiert vorgegeben werden. Das vorliegende Format wird der NC über folgendes Maschinendatum bekannt gemacht:

MD12060 \$MN\_OVR\_SPIND\_IS\_GRAY\_CODE (Spindel-Korrekturschalter graycodiert)

Bei Binärcode gilt folgende feste Zuordnung:

Binärcode	Dezimal	Korrekturfaktor
00000000	0	0,00 ± 0%
00000001	1	0,01 ± 1%
00000010	2	0,02 ± 2%
00000011	3	0,03 ± 3%
00000100	4	0,04 ± 4%
...	...	...
01100100	100	1,00 ± 100%
...	...	...
11001000	200	2,00 ± 200%

Bei Graycodierung sind in das folgende Maschinendatum die der Schalterstellung entsprechenden Korrekturfaktoren einzutragen:

MD12070 \$MN\_OVR\_FACTOR\_SPIND\_SPEED [<n>] (Bewertung des Spindel-Korrekturschalters)

Wirksamkeit der "Spindelkorrektur":	
• Bei aktivem G33, G34, G35:	wirksam
• Bei aktivem G63:	nicht wirksam
• Bei aktivem G331, G332:	wirksam

### Bezug der Spindelkorrektur

Die Spindelkorrektur kann sich auf die durch Maschinen- oder Settingdatum begrenzte Drehzahl oder die programmierte Drehzahl beziehen. Die Einstellung erfolgt über:

MD12080 \$MN\_OVR\_REFERENCE\_IS\_PROG\_FEED (Override-Bezugsgeschwindigkeit)

### Begrenzung des Korrekturfaktors

Bei binärcodierten Korrekturfaktoren können die maximal möglichen Korrekturen für Bahnvorschub, Achsvorschub und Spindeldrehzahl begrenzt werden:

MD12100 \$MN\_OVR\_FACTOR\_LIMIT\_BIN (Begrenzung bei binärcodiertem Korrekturschalter)

### Korrektur wirksam

Bei freigegebenen Korrekturen wirken die über Maschinensteuertafel vorgegebenen Korrekturwerte ohne Verzögerung in allen Betriebsarten und Maschinenfunktionen.

**Korrektur unwirksam**

Bei nicht wirksam gesetzter Korrektur wirkt intern ein Korrekturfaktor von 100%. Der an der NC/PLC-Nahtstelle anstehende Korrekturfaktor wird nicht ausgewertet.

Eine Ausnahme bildet die Nullstellung bei binärer Schnittstelle und die 1. Schalterstellung für eine graycodierte Schnittstelle. Hier werden die Korrekturfaktoren, die in die NC/PLC-Nahtstelle eingetragen sind, ausgewertet. Bei binärer Schnittstelle ist der Korrekturfaktor immer 0%. Bei graycodierter Schnittstelle wird der in den Maschinendaten für die 1. Schalterstellung eingetragene Wert als Korrekturwert ausgegeben. Dieser sollte mit "0" belegt sein.

**16.4.3 Programmierbare Vorschubkorrektur**

**Funktion**

Mit der Funktion "Programmierbare Vorschubkorrektur" kann das Geschwindigkeitsniveau von Bahn- und Positionierachsen über das Teileprogramm verändert werden.

**Programmierung**

Syntax	Bedeutung
OVR=<Wert>	Vorschubänderung für den Bahnvorschub F
OVRTA [<Achse>=<Wert>	Vorschubänderung für Positioniervorschub FA

Der programmierbare Bereich beträgt 0 - 200%.

Standardeinstellung: 100%

**Wirksamkeit**

Die NC/PLC-Nahtstellensignale DB21, ... DBB6 (Eilgang- bzw. Vorschubkorrektur wirksam) und DB31, ... DBX1.7 (achsspezifische Korrektur wirksam) beziehen sich **nicht** auf die programmierbare Vorschubkorrektur. Bei Deaktivierung dieser Signale bleibt die programmierbare Vorschubkorrektur weiterhin wirksam.

Die wirksame Korrektur errechnet sich aus dem Produkt der "Programmierbaren Vorschubkorrektur" und der "Vorschubkorrektur über Maschinensteuertafel (Seite 1536)".

Die Standardeinstellung für die "Programmierbare Vorschubkorrektur" ist 100%.

Die Standardeinstellung wirkt:

- wenn keine Vorschubkorrektur programmiert ist bzw.
- nach Reset, wenn das Maschinendatum:  
MD22410 \$MC\_F\_VALUES\_ACTIVE\_AFTER\_RESET (F-Funktion über Reset hinaus wirksam)  
nicht gesetzt ist.

---

**Hinweis**

OVR wirkt nicht bei G33, G34, G35.

---

## 16.4.4 Probelaufvorschub

### Funktion

Der Probelaufvorschub wird beim Einfahren von Teileprogrammen verwendet, um das Programm oder Programmteile mit erhöhtem Bahnvorschub ablaufen zu lassen, z. B. bei einem Testlauf ohne Werkstückbearbeitung.

### Aktivierung

Der Probelaufvorschub kann in den Automatikbetriebsarten angewählt und über PLC oder Bedientafelfront aktiviert werden.

Bei Aktivierung über die Bedientafelfront wird das Nahtstellensignal: DB21, ... DBX24.6 (Probelaufvorschub angewählt) gesetzt und vom PLC-Grundprogramm auf das Nahtstellensignal: DB21, ... DBX0.6 (Probelaufvorschub aktivieren) übertragen.

Bei Anwahl über PLC ist das Nahtstellensignal DB21, ... DBX0.6 (Probelaufvorschub aktivieren) vom PLC-Anwenderprogramm zu setzen.

### Wirksamkeit

Solange das Nahtstellensignal "Probelaufvorschub aktivieren" ansteht, wirkt anstelle des programmierten Vorschubs der über SD42100 \$SC\_DRY\_RUN\_FEED eingestellte Vorschubwert in der über SD42101 \$SC\_DRY\_RUN\_FEED\_MODE festgelegten Wirkungsweise (siehe Parametrierung).

Der Probelaufvorschub wird immer als Linearvorschub (G94) interpretiert.

## Parametrierung

### Aktivierung des Probelaufvorschubs

Der Zeitpunkt der Aktivierung ist abhängig von der Einstellung im Maschinendatum:

MD10704 \$MN\_DRYRUN\_MASK (Aktivierung des Probelaufvorschubs)

Wert	Bedeutung
0	Der Probelaufvorschub darf nur am Satzende ein- und ausgeschaltet werden (Standardeinstellung)
1	Der Probelaufvorschub kann auch während der Programmbearbeitung (im Teileprogrammsatz) aktiviert werden. <b>Achtung:</b> Das Einschalten während der Bearbeitung führt steuerungsintern zu einem Reorganisationsvorgang, bei dem die Achsen kurzfristig gestoppt werden. Dies kann Einflüsse auf die Oberflächengüte des bearbeiteten Werkstücks bewirken.
2	Der Probelaufvorschub kann in jeder Phase ein- und ausgeschaltet werden, ohne das die Achsen gestoppt werden. Allerdings wird die Funktion erst mit einem im Programmablauf "späteren" Satz wirksam.

### Änderung des Probelaufvorschubs

Der Vorschub für den Probelauf wird eingetragen in das Settingdatum:

SD42100 \$SC\_DRY\_RUN\_FEED (Probelaufvorschub)

Das Settingdatum kann über die Bedientafelfront im Bedienbereich "Parameter" verändert werden.

Wenn die Anwahl vom NCK akzeptiert wurde, wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB21, ... DBX318.6 (Probelauf Vorschub aktiv)

Ein aktiver Probelaufvorschub wird in der Statuszeile der Bedientafelfront mit "DRY" angezeigt, wenn:

- die Anwahl bei Programmstopp am Satzende erfolgte oder
- das Maschinendatum MD10704 \$MN\_DRYRUN\_MASK während der Programmbearbeitung auf "1" gesetzt wurde.

**Wirkungsweise des Probelaufvorschubs**

Die Wirkungsweise des in SD42100 eingetragenen Probelaufvorschubs kann eingestellt werden über das Settingdatum:

SD42101 \$SC\_DRY\_RUN\_FEED\_MODE

Wert	Bedeutung
0	Es wird der programmierte Vorschub mit dem Probelaufvorschub in SD42100 verglichen und danach mit dem größeren der beiden Vorschübe verfahren (Standardeinstellung).
1	Es wird der programmierte Vorschub mit dem Probelaufvorschub in SD42100 verglichen und danach mit dem kleineren der beiden Vorschübe verfahren.
2	Es wird direkt der in SD42100 eingetragene Probelaufvorschub wirksam, unabhängig von der programmierten Geschwindigkeit.
3 -9	reserviert
10	Wie Projektierung 0, außer Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Diese Funktionen werden wie programmiert ausgeführt.
11	Wie Projektierung 1, außer Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Diese Funktionen werden wie programmiert ausgeführt.
12	Wie Projektierung 2, außer Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Diese Funktionen werden wie programmiert ausgeführt.

**16.4.5 Mehrere Vorschubwerte in einem Satz****Funktion**

Mit der Funktion "Mehrere Vorschubwerte in einem Satz" können abhängig von externen digitalen und/oder analogen Eingängen 6 verschiedene Vorschubwerte eines NC-Satzes, eine Verweilzeit sowie ein Rückzug bewegungssynchron aktiviert werden.

Wenn der Eingang für Ausfeuerzeit bzw. Rückzugsweg aktiviert wird, wird der Restweg für Bahnachsen oder die betreffenden Einzelachsen gelöscht und die Verweilzeit bzw. der Rückzug gestartet.

Die Einleitung des Rückzugs erfolgt innerhalb eines IPO-Takts.

## Signale

Die Eingangssignale werden für die Funktion in einem Eingangsbyte zusammengefasst. Innerhalb des Bytes existiert eine feste funktionelle Zuordnung.

Tabelle 16- 1 Eingangsbyte für die Funktion "Mehrere Vorschubwerte in einem Satz"

	Bit							
	7	6	5	4	3	2	1	0
<b>Eingangs-Nr.</b>	E7	E6	E5	E4	E3	E2	E1	E0
<b>Vorschubadresse</b>	F7	F6	F5	F4	F3	F2	ST	SR

E7 bis E2: Aktivierung der Vorschübe F7 bis F2

E1: Aktivierung der Verweilzeit ST/STA (in Sekunden)

E0: Aktivierung der Rückzugsbewegung SR/SRA

## Priorität der Signale

Die Abfragereihenfolge der Signale erfolgt von E0 aufsteigend. Somit besitzt die Rückzugsbewegung (SR) die höchste und der Vorschub F7 die niedrigste Priorität.

SR und ST beenden die Vorschubbewegungen, die mit F2 bis F7 aktiviert wurden. SR beendet auch ST, d. h. die komplette Funktion.

Das Signal mit der höchsten Priorität bestimmt den aktuellen Vorschub.

Das Verhalten bei Abfall des jeweils höchstpriorien Eingangs (F2 - F7) kann festgelegt werden mit dem Maschinendatum:

MD21230 \$MC\_MULTFEED\_STORE\_MASK (Speicherverhalten für die Funktion "Mehrere Vorschubwerte in einem Satz")

Bit	Wert	Bedeutung
2 ... 7	0	Bei Abfall des jeweils höchstpriorien Eingangs wird der zugehörige Vorschub nicht beibehalten (Standardeinstellung)
	1	Gesetztes Bit 2 ... 7 bewirkt, dass der zugehörige Vorschub (F2 bis F7), der vom jeweils höchstpriorien Eingangssignal angewählt wurde, auch dann beibehalten wird, wenn das Eingangssignal abfällt und ein niederpriorier Eingang ansteht.

Das Satzendeckriterium ist erfüllt bei:

- Erreichen der programmierten Endposition
- Ende der Rückzugsbewegung (SR)
- Nach Ablauf der Verweilzeit (ST)



## Hardware-Zuordnung

Dem Eingangsbyte für die Funktion "Mehrere Vorschubwerte in einem Satz" können maximal zwei digitale Eingangsbytes bzw. Komparator-Eingangsbytes der NCK-Peripherie zugeordnet werden:

MD21220 \$MC\_MULTFEED\_ASSIGN\_FASTIN (Zuordnung der Eingangsbytes der NCK-Peripherie für "Mehrere Vorschubwerte in einem Satz"), Bit 0 ... 15

Weiterhin können die Eingangsbits invertiert werden:

MD21220 \$MC\_MULTFEED\_ASSIGN\_FASTIN, Bit 16 ... 31

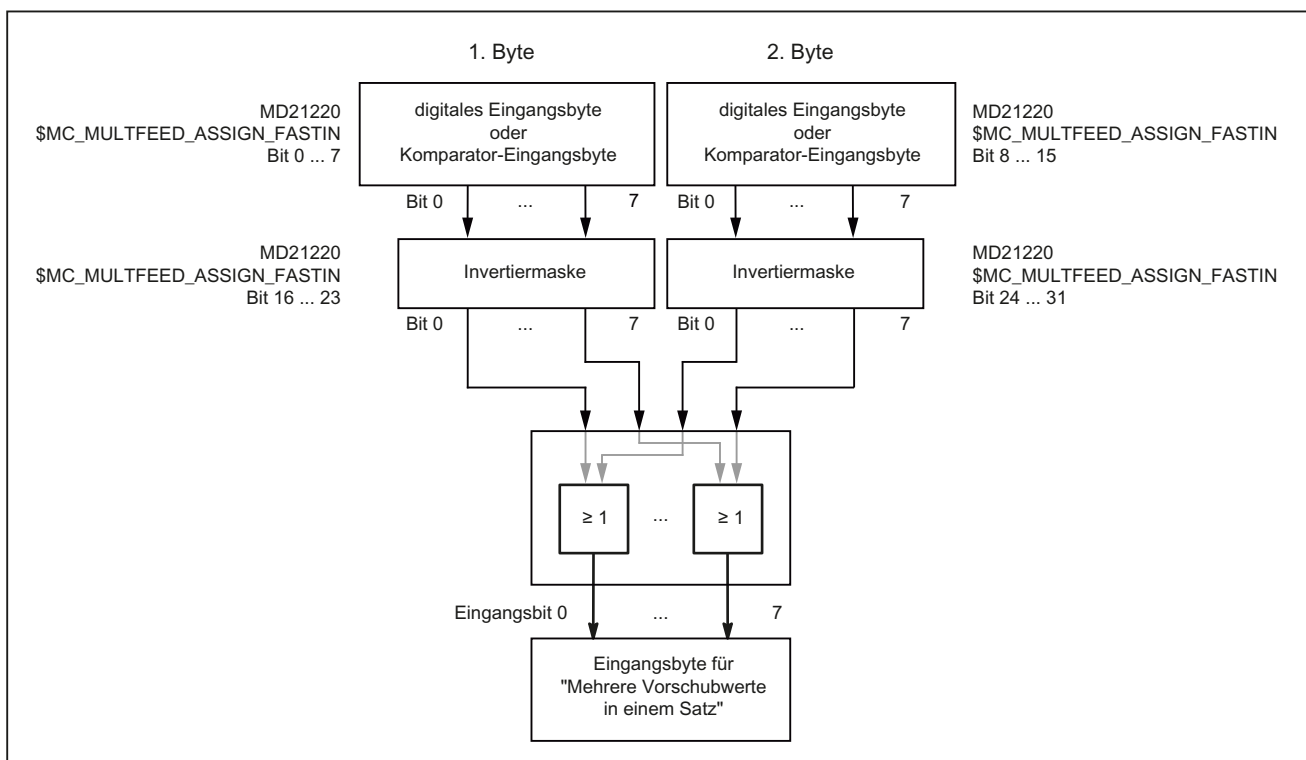


Bild 16-2 Rangierung der Signale für die Funktion "Mehrere Vorschubwerte in einem Satz"

Die Rangierung der digitalen Eingangsbytes und Parametrierung der Komparatoren sind beschrieben in:

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Digitale und analoge NCK-Peripherie (A4)

## Programmierung

### Bahnbewegung

Unter der Adresse  $F$  wird der Bahnvorschub programmiert, der gültig ist, solange kein Eingangssignal ansteht. Dieser ist modal wirksam.

Mit  $F2=...$  bis  $F7=...$  können zusätzlich zum Bahnvorschub bis zu 6 weitere Vorschübe im Satz programmiert werden. Die numerische Erweiterung gibt die Bitnummer des Eingangs an, mit dessen Änderung der Vorschub wirksam wird.

Beispiel:

$F7=1000$  ; 7 entspricht Eingangsbit 7

Die programmierten Werte wirken satzweise. Im Folgesatz gilt der unter  $F$  programmierte Bahnvorschub.

Verweilzeit (Ausfeuerzeit) und Rückzugsweg werden unter zusätzlichen Adressen im Satz programmiert:

$ST=...$  Verweilzeit (bei Schleifen Ausfeuerzeit)

$SR=...$  Rückzugsweg

Diese Adressen wirken satzweise.

### Axiale Bewegung

Unter der Adresse  $F_A$  werden axiale Vorschübe programmiert, die gültig sind, solange kein Eingangssignal ansteht. Sie sind modal wirksam.

Mit  $FMA[2, <Achse>]=...$  bis  $FMA[7, <Achse>]=...$  können zusätzlich bis zu 6 weitere Vorschübe pro Achse im Satz programmiert werden.

Der erste Ausdruck in eckigen Klammern gibt die Bitnummer des Eingangs an, mit dessen Änderung der Vorschub wirksam wird. Der zweite Ausdruck gibt an, für welche Achse der Vorschub gilt.

Beispiel:

$FMA[3, Y]=1000$  ; axialer Vorschub für Y-Achse, entspricht Eingangsbit 3

Die unter  $FMA$  programmierten Werte wirken satzweise. Im Folgesatz gilt der unter  $F_A$  programmierte Vorschub.

Verweilzeit (Ausfeuerzeit) und Rückzugsweg können zusätzlich für einzelne Achsen vorgegeben werden:

STA [<Achse>] = . . .                      axiale Verweilzeit (Ausfeuerzeit)  
SRA [<Achse>] = . . .                      axialer Rückzugsweg

Der Ausdruck in eckiger Klammer gibt an, für welche Achse Ausfeuerzeit bzw. Rückzugsweg gelten.

Beispiele:

STA [X] = 2 . 5                                ; Die Ausfeuerzeit für Achse X beträgt 2,5 Sekunden.  
SRA [X] = 3 . 5                                ; Der Rückzugsweg für Achse X beträgt 3,5 (Einheit z. B. mm).

Diese Adressen wirken satzweise.

---

### Hinweis

#### Rückzugsweg

Die Einheit für den Rückzugsweg bezieht sich auf die aktuell gültige Maßeinheit (mm oder inch).

Der Rückhub erfolgt immer in Gegenrichtung zur aktuellen Bewegung. Mit *SR/SRA* wird immer der Betrag des Rückhubs programmiert. Es wird kein Vorzeichen programmiert.

---

### Hinweis

#### POS statt POSA

Wenn für eine Achse Vorschübe, Ausfeuerzeit (Verweilzeit) oder Rückzugsweg aufgrund eines externen Eingangs programmiert sind, darf diese Achse in diesem Satz nicht als POSA-Achse (Positionierachse über Satzgrenze hinweg) programmiert werden.

---

### Hinweis

#### Statusabfrage

Der Status eines Eingangs kann auch für Synchronbefehle unterschiedlicher Achsen abgefragt werden.

---

### Hinweis

#### LookAhead

LookAhead ist auch bei mehreren Vorschüben in einem Satz wirksam. Damit kann der aktuelle Vorschub durch LookAhead begrenzt werden.

---

## Anwendung

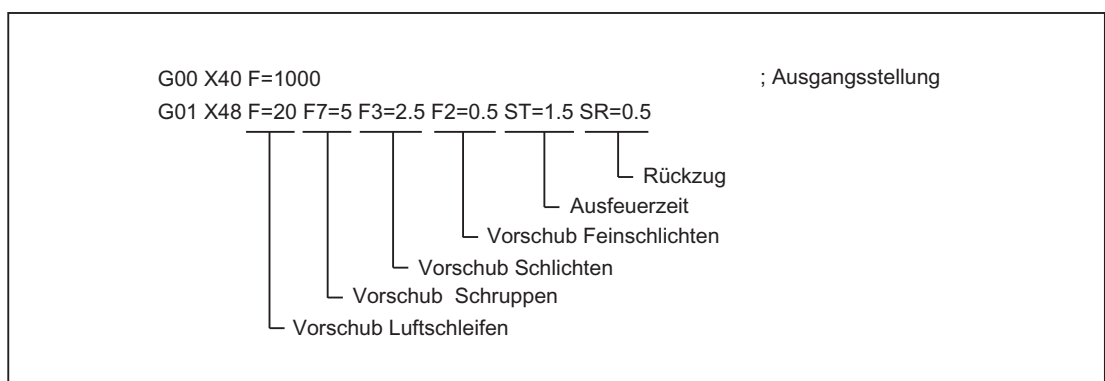
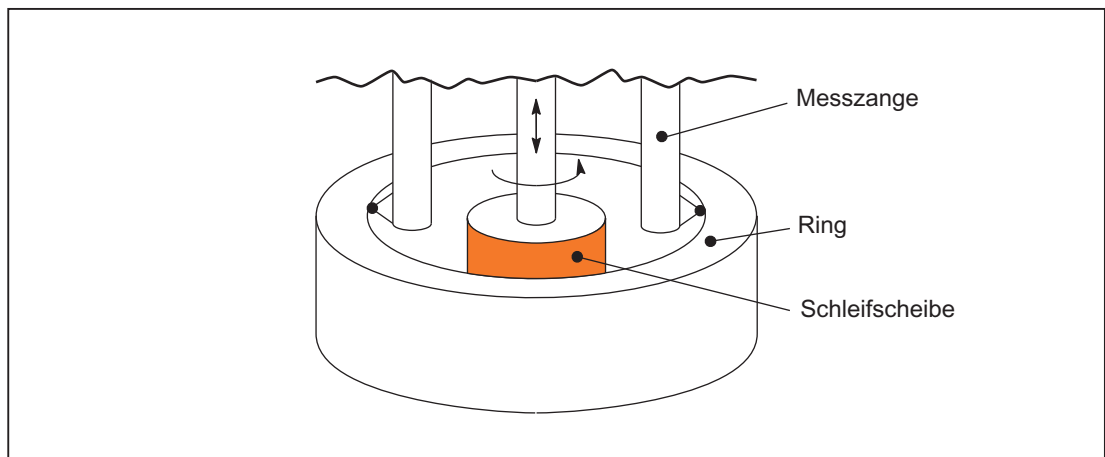
Die Funktion "Mehrere Vorschubwerte in einem Satz" wird schwerpunktmäßig bei der Technologie Schleifen eingesetzt. Sie ist jedoch nicht darauf beschränkt.

Typische Anwendungsfälle sind z. B.:

- Analoge oder digitale Messzange
  - Abhängig von externen analogen oder digitalen Eingängen können verschiedene Vorschubwerte, eine Verweilzeit sowie ein Rückzugsweg aktiviert werden, wobei die Vorgabe der Schwellwerte über Settingdaten erfolgt.
- Umschalten von Zustell- auf Arbeitsvorschub über Näherungsschalter

## Beispiel

Innenschleifen eines Kugellagerrings, wobei der Ist-Durchmesser über eine Messzange erfasst wird und, abhängig von Schwellwerten, der jeweils notwendige Vorschubwert für Schruppen, Schlichten oder Feinschlichten aktiviert wird. Die Messzangenposition liefert auch die Endposition. Somit wird das Satzendekriterium nicht nur von der programmierten Achsposition der Zustellachse, sondern auch von der Messzange bestimmt.



## 16.4.6 Feste Vorschubwerte

### Funktion

Mit der Funktion "Feste Vorschubwerte" können anstelle des programmierten Vorschubs bzw. der projizierten JOG-Geschwindigkeiten über Maschinendatum definierte Festvorschübe (max. 4) aktiviert werden.

Die Funktion steht in den Betriebsarten AUTOMATIK und JOG zur Verfügung.

#### Verhalten in der Betriebsart AUTOMATIK

Anstelle des programmierten Vorschubs wird die Kontur mit dem aktivierten Festvorschub verfahren.

#### Verhalten in der Betriebsart JOG

Anstelle der projizierten JOG-Geschwindigkeit/JOG-Eilgang-Geschwindigkeit wird die Achse mit dem aktivierten Festvorschub verfahren. Die Verfahrrichtung wird über Nahtstellensignale vorgegeben.

### Parametrierung

Die Einstellung der Festvorschübe erfolgt:

- für Linearachsen mit dem Maschinendatum:  
MD12202 \$MN\_PERMANENT\_FEED[<n>]
- für Rundachsen mit dem Maschinendatum:  
MD12204 \$MN\_PERMANENT\_ROT\_AX\_FEED[<n>]

mit <n> = 0, 1, 2, 3 (für Festvorschub 1, 2, 3, 4)

---

#### Hinweis

Die Festvorschübe sind immer Linearvorschubwerte. Auch bei Umdrehungsvorschub wird intern auf Linearvorschub umgestellt.

---

## Aktivierung

Die Aktivierung der Festvorschübe erfolgt über NC/PLC-Nahtstellensignale:

- in der Betriebsart AUTOMATIK für Bahn-/Geometrieachsen über die kanalspezifischen Nahtstellensignale:
  - DB21, ... DBX29.0 (Festvorschub 1 aktivieren)
  - DB21, ... DBX29.1 (Festvorschub 2 aktivieren)
  - DB21, ... DBX29.2 (Festvorschub 3 aktivieren)
  - DB21, ... DBX29.3 (Festvorschub 4 aktivieren)
- in der Betriebsart JOG für Maschinenachsen über die achsspezifischen Nahtstellensignale:
  - DB31, ... DBX3.2 (Festvorschub 1 aktivieren)
  - DB31, ... DBX3.3 (Festvorschub 2 aktivieren)
  - DB31, ... DBX3.4 (Festvorschub 3 aktivieren)
  - DB31, ... DBX3.5 (Festvorschub 4 aktivieren)

## Randbedingungen

### Wirksamkeit

Die Funktion "Feste Vorschubwerte" ist **nicht** wirksam:

- Bei Spindeln
- Bei Positionierachsen
- Beim Gewindebohren

### Override = 0

Das Fahrverhalten bei Override = 0 ist abhängig von der Einstellung im Maschinendatum:

MD12200 \$MN\_RUN\_OVERRIDE\_0

### DRF-Verschiebung

Bei angewähltem Festvorschub kann die DRF-Verschiebung nicht aktiviert werden.

## 16.4.7 Programmierbare Vorschubverläufe

### Funktion

Zur flexiblen Vorgabe des Vorschubverlaufs wird die Vorschubprogrammierung nach DIN 66025 um lineare und kubische Verläufe erweitert.

Die kubischen Verläufe können direkt oder als interpolierender Spline programmiert werden.

### Programmierung

Folgende Vorschubprofile können programmiert werden:

- FNORM

Verhalten gemäß DIN 66025 (Standardeinstellung).

Ein im Satz programmierter F-Wert wird über den gesamten Bahnweg des Satzes konstant vorgegeben und gilt danach als fester modaler Wert.

- FLIN

Ein im Satz programmierter F-Wert wird vom aktuellen Wert am Satzanfang bis zum Satzende linear über den Bahnweg eingefahren und gilt danach als modaler Wert.

- FCUB

Die satzweise programmierten F-Werte werden - bezogen auf den Satzende - durch einen Spline verbunden. Der Spline beginnt und endet tangential zur vorhergehenden bzw. nachfolgenden Vorschubvorgabe. Fehlt die F-Adresse in einem Satz, wird der zuletzt programmierte F-Wert dafür verwendet.

- FPO

Die F-Adresse [Syntax:  $F=FPO(\dots, \dots, \dots)$ ] bezeichnet den Vorschubverlauf über ein Polynom vom aktuellen Wert bis zum Ende des Satzes, in dem sie programmiert wurde. Der Endwert gilt von da an modal.

### Parametrierung

Wenn FLIN und FCUB in Verbindung mit Kompression COMPON verwendet wird, kann für den Bahnvorschub eine Toleranz festgelegt werden:

MD20172 \$MC\_COMPRESS\_VELO\_TOL (maximal erlaubte Abweichung des Bahnvorschubs bei Kompression)

## Randbedingungen

### FLIN/FCUB

Das mit FLIN oder FCUB programmierte Bahngeschwindigkeitsprofil wirkt nicht zusammen mit Umdrehungsvorschub bei G95 sowie mit konstanter Schnittgeschwindigkeit bei G96/G961 und G97/G971.

## Literatur

Weitere Informationen zu den programmierbaren Vorschubverläufen siehe Programmierhandbuch Arbeitsvorbereitung.

### 16.4.8 Vorschub für Fase/Rundung FRC, FRCM

Beim Übergang von Flächen zur Fase/Rundung können sich die Zerspanungsverhältnisse wesentlich ändern. Deshalb benötigen die Konturelemente Fase/Rundung zum Erreichen der gewünschten Oberflächengüte eigene, optimierte Vorschubwerte.

## Funktion

Der Vorschub für Fase/Rundung ist über NC-Adressen programmierbar.

## Programmierung

### Syntax:

... FRC/FRCM=<Wert>

### Bedeutung:

FRC: **satzweise** wirksamer Vorschub für Fase/Rundung

FRCM: **modal** wirksamer Vorschub für Fase/Rundung

<Wert>: Der Vorschubwert wird entsprechend dem aktiven Vorschubtyp interpretiert:

- G94, G961, G971: Vorschub in mm/min, inch/min oder %/min
- G95, G96, G97: Umdrehungsvorschub in mm/Umdr oder inch/Umdr

---

### Hinweis

FRC ist nur wirksam, wenn im Satz eine Fase/Rundung programmiert ist bzw. RNDM aktiviert wurde.

FRC überschreibt im aktuellen Satz den F- bzw. FRCM-Wert.

Der unter FRC programmierte Vorschub muss größer Null sein.

FRCM=0 aktiviert für das Anfasen/Verrunden den unter F programmierten Vorschub.

---



## Parametrierung

### Zuordnung der Fase/Rundung zum Vorgänger- oder Nachfolgesatz

Der Vorschubtyp (G94, G95, G96, G961 ...) und damit die Konvertierung in das interne Format müssen im Satz einheitlich für F und FRC/FRCM sein. In diesem Zusammenhang muss das folgende Maschinendatum berücksichtigt werden:

MD20201 \$MC\_CHFRND\_MODE\_MASK (Verhalten Fase/Rundung)

Bit	Wert	Bedeutung
0	0	Die Technologie der Fase/Rundung (Vorschub, Vorschubtyp, M-Befehle ...) wird vom Nachfolgesatz bestimmt (Standardeinstellung).
	1	Die Technologie der Fase/Rundung wird vom Vorgängersatz bestimmt (empfohlene Einstellung).

### Maximale Anzahl Leersätze

Die Anzahl an Sätzen ohne Verfahrinformationen in der Korrektorebene (Leersätze), die bei aktiver Fase/Rundung zwischen zwei Sätzen mit Verfahrinformation stehen dürfen, ist begrenzt. Die maximale Anzahl wird festgelegt im Maschinendatum:

MD20200 \$MC\_CHFRND\_MAXNUM\_DUMMY\_BLOCKS (Leersätze bei Fase/Radien)

## Randbedingungen

### FLIN/FCUB

Vorschubinterpolation FLIN und FCUB ist für Fase/Rundung **nicht** möglich.

### G0

FRC/FRCM wirkt nicht, wenn eine Fase mit G0 verfahren wird. Die Programmierung ist entsprechend dem F-Wert ohne Fehlermeldung möglich.

### Wechsel G94 ↔ G95

Ist FRCM programmiert, so muss äquivalent zu F der FRCM-Wert bei Wechsel G94 ↔ G95 usw. neu programmiert werden. Wird nur F neu programmiert und ist vor dem Wechsel des Vorschubtyps FRCM > 0, dann erfolgt eine Fehlermeldung.

## Beispiel

**Beispiel 1: MD20201 Bit 0 = 0; Vorschub vom Nachfolgesatz übernehmen (Standardeinstellung!)**

Programmcode	Kommentar
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; Fase N20-N30 mit F=100 mm/min
N30 Y10 CHF=4	; Fase N30-N40 mit FRC=200 mm/min
N40 X20 CHF=3 FRC=200	; Fase N40-N60 mit FRCM=50 mm/min
N50 RNDM=2 FRCM=50	
N60 Y20	; Modale Rundung N60-N70 mit FRCM=50 mm/min
N70 X30	; Modale Rundung N70-N80 mit FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; Fase N80-N90 mit FRC=100 mm/min
N90 X40	; Modale Rundung N90-N100 mit F=100 mm/min (Abwahl FRCM)
N100 Y40 FRCM=0	; Modale Rundung N100-N120 mit G95 FRC=1 mm/Umdr
N110 S1000 M3	
N120 X50 G95 F3 FRC=1	
...	
M02	

**Beispiel 2: MD20201 Bit 0 = 1; Vorschub vom Vorgängersatz übernehmen (empfohlene Einstellung!)**

Programmcode	Kommentar
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; Fase N20-N30 mit F=100 mm/min
N30 Y10 CHF=4 FRC=120	; Fase N30-N40 mit FRC=120 mm/min
N40 X20 CHF=3 FRC=200	; Fase N40-N60 mit FRC=200 mm/min
N50 RNDM=2 FRCM=50	
N60 Y20	; Modale Rundung N60-N70 mit FRCM=50 mm/min
N70 X30	; Modale Rundung N70-N80 mit FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; Fase N80-N90 mit FRC=100 mm/min
N90 X40	; Modale Rundung N90-N100 mit FRCM=50 mm/min
N100 Y40 FRCM=0	; Modale Rundung N100-N120 mit F=100 mm/min
N110 S1000 M3	
N120 X50 CHF=4 G95 F3 FRC=1	; Fase N120-N130 mit G95 FRC=1 mm/Umdr
N130 Y50	; Modale Rundung N130-N140 mit F=3 mm/Umdr
N140 X60	
...	
M02	

## 16.4.9 Satzweiser Vorschub FB

### Funktion

Mit der Funktion "Satzweiser Vorschub" kann für einen einzelnen Teileprogrammsatz ein separater Vorschub vorgegeben werden. Nach diesem Satz ist der zuvor wirksame modale Bahnvorschub wieder aktiv.

### Programmierung

**Syntax:**

```
... FB=<Wert>
```

**Bedeutung:**

FB:	separater Vorschub für den aktuellen Satz
<Wert>:	Der Vorschubwert wird entsprechend dem aktiven Vorschubtyp interpretiert:
	<ul style="list-style-type: none"><li>• G94, G961, G971: Vorschub in mm/min, inch/min oder °/min</li><li>• G95, G96, G97: Umdrehungsvorschub in mm/Umdr oder inch/Umdr</li></ul>

---

**Hinweis**

Der unter FB programmierte Vorschub muss größer Null sein.

Ist im Satz keine Verfahrbewegung programmiert (z. B. Rechensatz), bleibt FB ohne Wirkung.

Ist kein expliziter Vorschub für Fase/Rundung programmiert, gilt der Wert von FB auch für ein in diesem Satz vorhandenes Konturelement Fase/Rundung.

Die gleichzeitige Programmierung von FB und FD (Handradfahren mit Vorschubüberlagerung) oder F (modaler Bahnvorschub) ist nicht möglich.

---

## 16.4.10 Beeinflussung der Einzelachsdynamik

### Einzelachsen

Einzelachsen können im Teileprogramm, in Synchronaktionen und über die PLC programmiert werden:

- Teileprogramm:
  - POS [<Achse>] = . . .
  - POSA [<Achse>] = . . .
  - SPOS [<Achse>] = . . .
  - SPOSA [<Achse>] = . . .
  - OS [<Achse>] = . . .
  - OSCILL [<Achse>] = . . .
- Synchronaktionen:
  - EVERY . . . DO
  - POS [<Achse>] = . . .
  - SPOS [<Spindel>] = . . .
  - MOV [<Achse>] = . . .
- PLC: FC18

### Dynamikverhalten

Das Dynamikverhalten einer Achse wird beeinflusst durch:

- MD32060 \$MA\_POS\_AX\_VELO (Positionierachsgeschwindigkeit)
  - Die wirksame Positionierachsgeschwindigkeit kann verändert werden:
    - Teileprogramm/Synchronaktion: Axialer Vorschub FA oder prozentuale Vorschubkorrektur OVRA
    - PLC: Vorgabe von FRate bzw. Überschreiben des axialen Override
- MD32300 \$MA\_MAX\_AX\_ACCEL (maximale Achsbeschleunigung)
  - Die wirksame maximale Achsbeschleunigung kann verändert werden:
    - Teileprogramm indirekt: Schreiben des Maschinendatums mit anschließendem NewConfig
    - Teileprogramm direkt: Prozentuale Beschleunigungskorrektur ACC
    - Synchronaktionen indirekt: Schreiben des Maschinendatums und Auslösen eines ASUP für die Aktivierung von NewConfig
    - Synchronaktionen direkt: Prozentuale Beschleunigungskorrektur ACC (nicht von der PLC aus vorgebar).

Über PLC bestehen die gleichen Möglichkeiten wie in Synchronaktionen.

- Teileprogrammbefehle: BRISKA, SOFTA, DRIVEA, JERKA  
Nicht in Synchronaktionen programmierbar (nur indirekt über ASUP).  
Nicht von PLC vorgebar (nur indirekt über ASUP).
- Aktiven Servo-Parametersatz  
Der aktive Parametersatz kann verändert werden:
  - Teileprogramm/Synchronaktion: SCPARA
  - PLC: DB31, ... DBX9.0-2 (Regler-Parametersatz)
 Ausführliche Informationen zu den Servo-Parametersätzen siehe "Parametersätze des Lagereglers (Seite 381)".

---

**Hinweis****Dynamikänderungen**

Im Teileprogramm vorgenommene Dynamikänderungen wirken nicht auf Kommando- bzw. PLC-Achsbewegungen. In Synchronaktionen vorgenommenen Dynamikänderungen wirken nicht auf die im Teileprogramm programmierten Verfahrbewegungen.

**Vorsteuerung**

Die Art der Vorsteuerung und welche Bahnachsen mit Vorsteuerung verfahren sollen, kann im Teileprogramm direkt mit FFWON/FFWOF programmiert werden. In Synchronaktionen und von PLC aus kann eine Programmierung nur indirekt über ein ASUP vorgenommen werden.

---

**Prozentuale Beschleunigungskorrektur (ACC)**

Im Teileprogramm oder Synchronaktion kann mit dem Befehl ACC die im Maschinendatum: MD32300 \$MA\_MAX\_AX\_ACCEL (maximale Achsbeschleunigung) vorgegebene Beschleunigung in einem Bereich von 0% – 200% verändert werden.

**Syntax:**

ACC [<Achse>] = <Wert>

**Bedeutung:**

ACC:	Schlüsselwort zur Programmierung der prozentualen Beschleunigungskorrektur
<Achse>:	Name der Kanalachse bzw. Spindel
<Wert>:	Beschleunigungsänderung in Prozent bezogen auf MD32300
Wertebereich:	0 ... 200

Der aktuelle axiale Beschleunigungswert kann über die Systemvariable \$AA\_ACC gelesen werden. Er bestimmt sich aus:

$$\$AA\_ACC[<Achse>] = (MD32300 \$MA\_MAX\_AX\_ACCEL[<Achse>]) * ACC[<Achse>] / 100$$

Über MD32320 \$MA\_DYN\_LIMIT\_RESET\_MASK kann die Löschestellung des mit ACC programmierten Werts bei Kanal-Reset bzw. Teileprogrammende M30 vorgegeben werden.

---

#### Hinweis

Die mit ACC programmierte Beschleunigungskorrektur kann über die Systemvariable \$AA\_ACC gelesen werden. Das Lesen von \$AA\_ACC im Teileprogramm findet jedoch zu einem anderen Zeitpunkt statt als das Lesen in einer Synchronaktion.

Die Systemvariablen \$AA\_ACC enthält nur dann den im Teileprogramm mit ACC programmierten Wert, wenn die Beschleunigungskorrektur nicht zwischenzeitlich durch Programmieren von ACC in einer Synchronaktion verändert wurde. Gleiches gilt für den umgekehrten Fall.

---

### Prozentuale Beschleunigungskorrektur und Hauptlauf-Achsen

Je nachdem, ob die Systemvariable \$AA\_ACC im Teileprogramm oder Synchronaktion gelesen wird, wird der Wert für die mit ACC programmierte Beschleunigungskorrektur für NC-Achsen oder Hauptlaufachsen (Kommandoachsen, PLC-Achsen, asynchrone Pendelachsen etc.) ausgegeben.

Für korrekte Ergebnisse muss die Systemvariable \$AA\_ACC daher immer an der gleichen Stelle (Teileprogramm oder Synchronaktion) gelesen werden, von der aus die Beschleunigungskorrektur mit ACC programmiert wurde.

#### Beispiele:

---

##### Schreiben von ACC im Teileprogramm:

```
N80 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 IPOENDA[X]
```

---

##### Schreiben von ACC in Synchronaktion:

```
N100 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140 IPOENDA[X]
```

---

##### Schreiben von ACC und Lesen von \$AA\_ACC im Teileprogramm:

```
ACC[X]=50 ; Schreiben
RO=$AA_ACC[X] ; Lesen

IF (RO <> $MA_MAX_AX_ACCEL[X] * 0,5) ; Überprüfen
  SETAL(61000)
ENDIF
```

**Schreiben von ACC und Lesen von \$AA ACC in Synchronaktion:**

```

WHEN TRUE DO ACC [X]=25 R0=$AA_ACC [X] ; Schreiben und Lesen
G4 F1

IF (R0 <> $MA_MAX_AX_ACCEL [X] * 0,25) ; Überprüfen
  SETAL (61001)
ENDIF

```

**Bewegungsendekriterium bei Einzelachsen**

Ähnlich dem Satzwechselkriterium bei Bahninterpolation (G601, G602, G603) kann auch das Bewegungsendekriterium bei Verfahrenbewegungen von Einzelachsachsen in Teileprogrammen/Synchronaktionen programmiert werden:

Programmbefehl	Bewegungsendekriterium
FINEA [<Achse>]	"Genauhalt Fein"
COARSEA [<Achse>]	"Genauhalt Grob"
IPOENDA [<Achse>]	"Interpolator-Stopp" (IPO-Stopp)

Nach Programmende oder NC-Reset bleibt der zuletzt programmierte Wert erhalten.

Das wirksame Bewegungsendekriterium kann über die achsspezifische Systemvariable \$AA\_MOTEND gelesen werden.

**Hinweis**

Abhängig davon, ob die Systemvariable \$AA\_MOTEND im Teileprogramm oder Synchronaktion gelesen wird, enthält sie den Wert für NC-Achsen oder Hauptlauf-Achsen.

**Beispiel:****Teileprogramm:**

```
N80 G01 POS [X]=100 FA [X]=1000 ACC [X]=90 COARSEA [X]
```

**Synchronaktion:**

```
N100 EVERY $A_IN [1] DO POS [X]=50 FA [X]=2000 ACC [X]=140 IPOENDA [X]
```

**Literatur:**

Weitere Informationen zum Satzwechsel und zu den Bewegungsendekriterien bei FINEA, COARSEA und IPOENDA siehe:

Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2), Kapitel: Satzwechsel

## Programmierbarer Servo-Parametersatz (SCPARA)

Im Teileprogramm/Synchronaktion kann mit SCPARA der Servo-Parametersatz vorgegeben werden.

### Syntax

SCPARA [<Achse>] =<Parametersatz-Nummer>

### Bedeutung

SCPARA:	Schlüsselwort zur Aktivierung des angegebenen Servo-Parametersatzes
<Achse>:	Name der Kanalachse
<Parametersatz-Nummer>:	Nummer des zu aktivierenden Servo-Parametersatzes

---

### Hinweis

Die Aktivierung des mit SCPARA vorgegebenen Parametersatzes kann vom PLC-Anwenderprogramm aus unterbunden werden:

DB31, ... DBX9.3 = 1 (Parametersatzvorgabe durch SCPARA gesperrt)

Es wird dabei **keine** Meldung angezeigt.

---

Die Nummer des aktiven Parametersatzes kann über die Systemvariable \$AA\_SCPAR gelesen werden.

## Randbedingungen

### Unterschiedliche Bewegungsendekriterien

Bei unterschiedlichen Bewegungsendekriterien werden Teileprogrammsätze unterschiedlich schnell beendet. Dies kann bei Technologiezyklen und PLC-Anwenderteilen zu Seiteneffekten führen.

### Parametersatzwechsel

Falls der Servo-Parametersatz sowohl in einem Teileprogramm bzw. in einer Synchronaktion und der PLC gewechselt werden soll, muss das PLC-Anwenderprogramm erweitert werden.

### Power On

Nach Power On sind folgende Grundstellungswerte eingestellt:

- Prozentuale Beschleunigungskorrektur für alle Einachsinterpolationen: 100%
- Bewegungsendekriterium für alle Einzelachsinterpolationen: FINEA
- Servo-Parametersatz: 1



### **Betriebsartenwechsel**

Beim Betriebsartenwechsel von AUTOMATIK nach JOG behalten die programmierten Dynamikänderungen ihre Gültigkeit.

### **Reset**

Mit Reset bleibt für die Teileprogrammvorgaben der letzte programmierte Wert bestehen. Die Vorgaben für die Hauptlauf-Interpolationen werden nicht geändert.

### **Satzsuchlauf**

Jeweils das zuletzt programmierte Bewegungsendekriterium einer Achse wird aufgesammelt und in einem Aktionsatz ausgegeben. Der letzte im Suchlauf behandelte Satz mit einem programmierten Bewegungsendekriterium dient als Behälter für alle programmierten Bewegungsendekriterien aller Achsen.

## 16.5 Randbedingungen

### Maßeinheit

Die gültige Maßeinheit der Vorschübe ist abhängig vom eingestellten Maßsystem und der eingetragenen Achsart:

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC (Grundsystem der Steuerung metrisch/inch)

MD30300 \$MA\_IS\_ROT\_AX (Rund- oder Linearachse)

### Grundstellung für den Vorschubtyp

Die Grundstellung für den Vorschubtyp ist festgelegt im Maschinendatum:

MD20150 \$MC\_GCODE\_RESET\_VALUES (Löschstellung der G-Gruppen)

Standardeinstellung ist G94.

Die Grundstellung des Vorschubtyps wird erst mit Start eines Teileprogramms angezeigt.

### Wirksamkeit nach Reset

Ob die zuletzt programmierten F-, FA-, OVR-, OVRA- Werte auch nach Reset wirksam sind, ist abhängig von der Einstellung im Maschinendatum:

MD22410 \$MC\_F\_VALUES\_ACTIVE\_AFTER\_RESET (F-Funktion über Reset hinaus wirksam)

### Spindelpositionieren

Bei aktivem G95, G96, G961, G97, G971, G33, G34, G35 sollte kein Spindelpositionieren stattfinden, da nach erfolgter Spindelpositionierung der abgeleitete Bahnvorschub = 0 ist. Wenn dann die programmierte Achsposition noch nicht erreicht ist, kann der Satz nicht beendet werden.

## 16.6 Datenlisten

### 16.6.1 Maschinendaten

#### 16.6.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10704	DRYRUN_MASK	Aktivierung des Probelaufvorschubs
10710	PROG_SD_RESET_SAVE_TAB	Zu aktualisierende Settingdaten
11410	SUPPRESS_ALARM_MASK	Maske zur Unterdrückung spezieller Alarmer
11550	STOP_MODE_MASK	Legt das Stopp-Verhalten fest
12000	OVR_AX_IS_GRAY_CODE	Achs-Vorschubkorrekturschalter Gray codiert
12010	OVR_FACTOR_AX_SPEED	Bewertung des Achs-Vorschubkorrekturschalters
12020	OVR_FEED_IS_GRAY_CODE	Bahnvorschub-Korrekturschalter Gray-codiert
12030	OVR_FACTOR_FEEDRATE	Bewertung des Bahnvorschub-Korrekturschalters
12040	OVR_RAPID_IS_GRAY_CODE	Eilgang-Korrekturschalter Gray-codiert
12050	OVR_FACTOR_RAPID_TRA	Bewertung des Eilgang-Korrekturschalters
12060	OVR_SPIND_IS_GRAY_CODE	Spindel-Korrekturschalter Gray-codiert
12070	OVR_FACTOR_SPIND_SPEED	Bewertung des Spindel-Korrekturschalters
12080	OVR_REFERENCE_IS_PROG_FEED	Override-Bezugsgeschwindigkeit
12082	OVR_REFERENCE_IS_MIN_FEED	Festlegung des Bezugs des Bahn-Override
12090	OVR_FUNCTION_MASK	Auswahl von Override-Spezifikationen
12100	OVR_FACTOR_LIMIT_BIN	Begrenzung bei binärcodiertem Korrekturschalter
12200	RUN_OVERRIDE_0	Fahren bei Override 0
12202	PERMANENT_FEED	Festvorschübe für Linearachsen
12204	PERMANENT_ROT_AX_FEED	Festvorschübe für Rundachsen

### 16.6.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20100	DIAMETER_AX_DEF	Geometrieachsen mit Planachsfunktion
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen
20172	COMPRESS_VELO_TOL	Max. erlaubte Abweichung der Bahnvorschubs bei Kompression
20200	CHFRND_MAXNUM_DUMMY_BLOCKS	Leersätze bei Phase/Radien
20201	CHFRND_MODE_MASK	Verhalten für Fase/Rundung
20750	ALLOW_GO_IN_G96	G0-Logik bei G96, G961
21200	LIFTFAST_DIST	Verfahrstrecke bei Schnellabheben von der Kontur
21220	MULTFEED_ASSIGN_FASTIN	Zuordnung der Eingangsbytes der NCK-Peripherie für "Mehrere Vorschubwerte in einem Satz"
21230	MULTFEED_STORE_MASK	Speicherverhalten für die Funktion "Mehrere Vorschubwerte in einem Satz"
22240	AUXFU_F_SYNC_TYPE	Ausgabezeitpunkt der F-Funktionen
22410	F_VALUES_ACTIVE_AFTER_RESET	F-Funktion über Reset hinaus wirksam

### 16.6.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30300	IS_ROT_AX	Rundachse / Spindel
32000	MAX_AX_VELO	Maximale Achsgeschwindigkeit
32060	POS_AX_VELO	Löschstellung für Positionierachsgeschwindigkeit
32300	MAX_AX_ACCEL	Achsbeschleunigung
32320	DYN_LIMIT_RESET_MASK	Resetverhalten von Dynamikbegrenzungen
34990	ENC_ACTIVATION_SMOOTH_TIME	Glättungszeitkonstante für Istwerte
35100	SPIND_VELO_LIMIT	Maximale Spindeldrehzahl
35130	GEAR_STEP_MAX_VELO_LIMIT	Maximaldrehzahl der Getriebestufe
35140	GEAR_STEP_MIN_VELO_LIMIT	Minimaldrehzahl der Getriebestufe
35160	SPIND_EXTERN_VELO_LIMIT	Spindeldrehzahlbegrenzung von PLC

## 16.6.2 Settingdaten

### 16.6.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42000	THREAD_START_ANGLE	Startwinkel bei Gewinde
42010	THREAD_RAMP_DISP	Beschleunigungsverhalten der Achse beim Gewindeschneiden
42100	DRY_RUN_FEED	Probelaufvorschub
42101	DRY_RUN_FEED_MODE	Mode für Probelaufvorschub
42110	DEFAULT_FEED	Defaultwert für Bahnvorschub
42600	JOG_FEED_PER_REV_SOURCE	Steuerung Umdrehungsvorschub im JOG
43300	ASSIGN_FEED_PER_RES_SOURCE	Umdrehungsvorschub für Positionachsen/Spindel

### 16.6.2.2 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: \$SA_	Beschreibung
43210	SPIND_MIN_VELO_G25	programmierte Spindeldrehzahlbegrenzung G25
43220	SPIND_MAX_VELO_G26	programmierte Spindeldrehzahlbegrenzung G26
43230	SPIND_MAX_VELO_LIMS	Spindeldrehzahlbegrenzung bei G96

## 16.6.3 Signale

### 16.6.3.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Probelaufvorschub aktivieren	DB21, ... .DBX0.6	DB3200.DBX0.6
Vorschubkorrektur	DB21, ... .DBB4	DB3200.DBB4
Eilgangkorrektur	DB21, ... .DBB5	DB3200.DBB5
Vorschubsperr	DB21, ... .DBX6.0	DB3200.DBX6.0
Eilgangkorrektur wirksam	DB21, ... .DBX6.6	DB3200.DBX6.6
Vorschubkorrektur wirksam	DB21, ... .DBX6.7	DB3200.DBX6.7
Vorschub Halt, Geometrieachse 1	DB21, ... .DBX12.3	DB3200.DBX1000.3
Vorschub Halt, Geometrieachse 2	DB21, ... .DBX16.3	DB3200.DBX1004.3
Vorschub Halt, Geometrieachse 3	DB21, ... .DBX20.3	DB3200.DBX1008.3

**16.6.3.2 Signale von Kanal**

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Probelaufvorschub angewählt	DB21, ... .DBX24.6	DB1700.DBX0.6
Vorschubkorrektur für Eilgang angewählt	DB21, ... .DBX25.3	DB1700.DBX1.3
Festvorschub 1 aktivieren für Bahn-/Geometrieachsen	DB21, ... .DBX29.0	-
Festvorschub 2 aktivieren für Bahn-/Geometrieachsen	DB21, ... .DBX29.1	-
Festvorschub 3 aktivieren für Bahn-/Geometrieachsen	DB21, ... .DBX29.2	-
Festvorschub 4 aktivieren für Bahn-/Geometrieachsen	DB21, ... .DBX29.3	-
Probelauf Vorschub aktiv	DB21, ... .DBX318.6	DB3300.DBX4002.6

**16.6.3.3 Signale an Achse/Spindel**

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschub-/Spindelkorrektur	DB31, ... .DBB0	DB380x.DBB0
Korrektur wirksam	DB31, ... .DBX1.7	DB380x.DBX1.7
Festvorschub 1 aktivieren für Maschinenachse	DB31, ... .DBX3.2	-
Festvorschub 2 aktivieren für Maschinenachse	DB31, ... .DBX3.3	-
Festvorschub 3 aktivieren für Maschinenachse	DB31, ... .DBX3.4	-
Festvorschub 4 aktivieren für Maschinenachse	DB31, ... .DBX3.5	-
Vorschub Halt / Spindel Halt	DB31, ... .DBX4.3	DB380x.DBX4.3

**16.6.3.4 Signale von Achse/Spindel**

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Umdrehungsvorschub aktiv	DB31, ... .DBX62.2	DB390x.DBX2.2
F-Funktion für Positionierachse	DB31, ... .DBB81	-
Programmierte Drehzahl zu hoch	DB31, ... .DBX83.1	DB390x.DBX2001.1

## W1: Werkzeugkorrektur

### 17.1 Kurzbeschreibung

#### Verrechnung der Werkzeugkorrekturdaten

Die Steuerung SINUMERIK 840D sl ermöglicht eine Verrechnung folgender Werkzeugkorrekturdaten:

- Längenkorrektur
- Radiuskorrektur
- Ablage der Werkzeugdaten in flexiblem Werkzeugkorrekturspeicher:
  - Werkzeug-Kennzeichnung durch T-Nummern von 0 bis 32000
  - Definition eines Werkzeugs durch maximal 9 Schneiden
  - Schneide wird durch max. 25 Werkzeugparameter beschrieben

- Werkzeug-Anwahl wählbar: Sofort oder über wählbare M-Funktion
- Werkzeug-Radiuskorrektur:
  - An- und Abwahlstrategie wählbar: Normal oder konturbezogen
  - Korrektur wirkt für alle Interpolationsarten:
    - Linear
    - Kreis
    - Helikal
    - Spline
    - Polynom
    - Evolvente
  - Korrektur an Außenecken wählbar:
    - Übergangskreis / Ellipse ( $G_{450}$ ) oder Schnittpunkt der Äquidistanten ( $G_{451}$ )
  - Parametergesteuerte Anpassung der Funktionen  $G_{450}/G_{451}$  an die Kontur
  - Freifahren an Außenecken bei  $G_{450}$  mit Parameter  $DISC$
  - Anzahl der Zwischensätze ohne Achsbewegung in der Korrektorebene wählbar
  - Kollisionsüberwachung wählbar:
    - Mögliche Konturverletzungen werden vorausschauend erkannt, wenn:
      - Bahnweg kürzer ist als Werkzeugradius
      - Breite einer Innenecke kleiner ist als Werkzeug-Durchmesser
  - Werkzeugradiuskorrektur konstant halten
  - Schnittpunktverfahren für Polynome

### Orientierbare Werkzeugträger

Die Funktion ermöglicht die Bearbeitung schräger Flächen mit Berücksichtigung von Werkzeuglängenkorrektur, wenn die Kinematik des Werkzeugträgers statisch (ohne NC-Achsen) die Orientierung des Werkzeuges ermöglicht. Die aufwendigere 5-Achstransformation ist für diesen Fall nicht erforderlich.

#### Literatur:

Funktionshandbuch Sonderfunktionen; Mehrachstransformationen (F2)



Durch geeignete Wahl von Werkzeugdaten und Werkzeugträgerdaten wird für die Steuerung die Kinematik so beschrieben, dass diese die Werkzeuglängenkorrektur berücksichtigen kann. Einige der Beschreibungsdaten können vom aktuellen Frame direkt durch die Steuerung übernommen werden.

---

#### Hinweis

Weitere Erläuterungen zu Werkzeugen und Werkzeugkorrekturen sowie die vollständigen programmtechnischen Möglichkeiten der Werkzeugkorrektur (WLK und WRK) mit ihren Sonderfällen sind zu entnehmen:

#### Literatur:

Programmierhandbuch Grundlagen

---

### Flache/eindeutige D-Nummern-Struktur

Korrekturanwahl über eindeutige D-Nummern mit Verwaltungsfunktion verfügbar.

### Werkzeugkorrekt.-Sonderbehandlung

Die Bewertung der Vorzeichen lässt sich für Werkzeuglänge und Verschleiß steuern mit den Settingdaten:

SD42900 \$SC\_MIRROR\_TOOL\_LENGTH (Vorzeichenwechsel Werkzeuglänge beim Spiegeln)

SD42960 \$SC\_TOOL\_TEMP\_COMP (Temperaturkompens. bezogen auf Werkzeug)

Das gilt ebenfalls für das Verhalten der Verschleißkomponenten beim Spiegeln von Geometrieachsen oder beim Wechsel der Bearbeitungsebene über Settingdaten.

#### Literatur:

Programmierhandbuch Grundlagen; Werkzeugkorrekturen

### G461/G462

Um in bestimmten Fällen beim Aktivieren und Deaktivieren der WRK auch das Ausräumen von Innenecken zu ermöglichen, werden die Befehle G461 und G462 eingeführt und damit die An-/Abfahrstrategie bei der Werkzeugradiuskorrektur erweitert.

- G461

Wenn kein Schnittpunkt des letzten WRK-Satzes mit einem Vorgängersatz möglich ist, berechnet die Steuerung einen Schnittpunkt, indem die Offsetkurve dieses Satzes mit einem Kreis verlängert wird, dessen Mittelpunkt im Endpunkt des nicht korrigierten Satzes liegt, und dessen Radius gleich dem Werkzeugradius ist.

- G462

Wenn kein Schnittpunkt des letzten WRK-Satzes mit einem Vorgängersatz möglich ist, berechnet die Steuerung einen Schnittpunkt, indem im Endpunkt des letzten Satzes mit Werkzeugradiuskorrektur eine Gerade eingefügt wird (der Satz wird durch seine Endtangente verlängert).

## Wechsel von G40 nach G41/42

Der Wechsel von G40 nach G41/G42 und umgekehrt wird auch bei Werkzeugen mit relevanter Schneidenlage (Dreh- und Schleifwerkzeuge) nicht mehr als Werkzeugwechsel behandelt.

## Werkzeugkorrektur-Umgebungen

Mit SW-Stand 7.1 werden Funktionen bereitgestellt, mit denen bezüglich den aktuellen Zuständen von Werkzeugdaten folgende Aktionen möglich sind:

- Abspeichern
- Löschen
- Lesen
- Modifizieren

Einige der Funktionen waren bisher innerhalb von Messzyklen realisiert. Sie stehen jetzt allgemein zur Verfügung.

Mit einer weiteren Funktion kann Information über die Zuordnung der Werkzeuglängen des aktiven Werkzeuges zu Abszisse, Ordinate und Applikate ermittelt werden.

## 17.2 Werkzeug

### 17.2.1 Allgemeines

#### WZ anwählen

Ein Werkzeug wird im Programm mit der T-Funktion angewählt.

Ob mit der T-Funktion sofort das neue Werkzeug eingewechselt wird, hängt ab von der Einstellung im Maschinendatum:

MD22550 \$MC\_TOOL\_CHANGE\_MODE (neue Werkzeugkorrektur bei M-Funktion)

#### WZ-Wechsel sofort

MD22550 \$MC\_TOOL\_CHANGE\_MODE = 0 (neue Werkzeugkorrektur bei M-Funktion)

Das neue Werkzeug wird mit der T-Funktion sofort eingewechselt.

Bei Drehmaschinen mit Werkzeugrevolver wird hauptsächlich diese Einstellung verwendet.

#### WZ-Wechsel mit M06

MD22550 \$MC\_TOOL\_CHANGE\_MODE = 1 (neue Werkzeugkorrektur bei M-Funktion)

Das neue Werkzeug wird mit der T-Funktion zum Wechsel vorbereitet.

Bei Fräsmaschinen mit Werkzeugmagazin wird hauptsächlich diese Einstellung verwendet, um das neue Werkzeug hauptzeitparallel (die Bearbeitung wird nicht unterbrochen) auf die Werkzeugwechselposition zu bringen.

Das alte WZ wird aus der Spindel entfernt und das neue WZ in die Spindel eingewechselt mit der eingegebenen M-Funktion im Maschinendatum:

MD22560 \$MC\_TOOL\_CHANGE\_M\_CODE (M-Funktion für Werkzeugwechsel)

Nach DIN 66025 soll dieser Werkzeugwechsel mit der M-Funktion  $M_{06}$  programmiert werden.

Das nächste Werkzeug wird vorgewählt mit dem Maschinendatum:

MD20121 \$MC\_TOOL\_PRESEL\_RESET\_VALUE (Vorgewähltes Werkzeug bei RESET)

Dessen Werkzeuglängenkorrekturwerte sollen berücksichtigt werden bei `RESET` und Hochlauf entsprechend Maschinendatum:

MD20110 \$MC\_RESET\_MODE\_MASK (Feststellung der Steuerungs-Grundstellung nach RESET/TP-Ende)

#### Wertebereich von T

Die T-Funktion kann folgende ganzzahlige Werte annehmen:

- Von  $T_0$  (kein Werkzeug)
- Bis  $T_{32000}$  (Werkzeug mit der Nummer 32000)

### Werkzeugschneide

Jedes Werkzeug kann bis zu 9 WZ-Schneiden besitzen. Die 9 WZ-Schneiden sind den D-Funktionen D1 bis D9 zugeordnet.

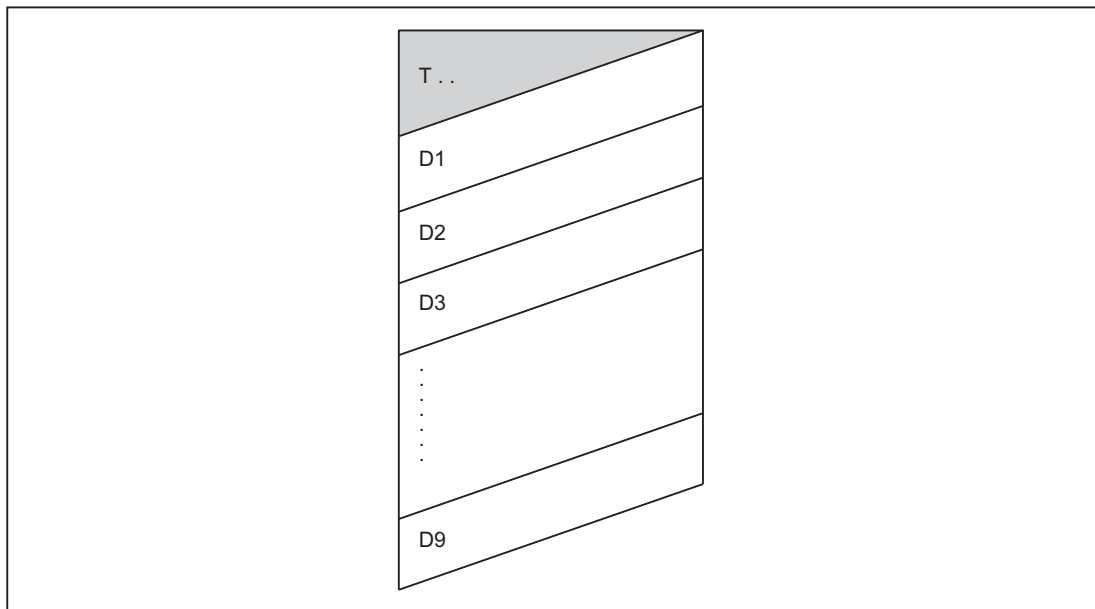


Bild 17-1 Beispiel für ein Werkzeug T... mit 9 Schneiden (D1 bis D9)

### D-Funktion

Die Werkzeugschneide wird mit D1 (Schneide 1) bis D9 (Schneide 9) programmiert. Die Werkzeugschneide bezieht sich immer auf das gerade aktive Werkzeug. Eine aktive Werkzeugschneide (D1 bis D9) ohne aktives Werkzeug (T0) ist unwirksam. Eine Werkzeugschneide D0 wählt alle Werkzeugkorrekturen des aktiven Werkzeugs ab.

### Anwahl der Schneide bei WZ-Wechsel

Nach der Programmierung eines neuen Werkzeugs (neue T-Nummer) und dem Einwechseln dieses WZ gibt es folgende Möglichkeiten zur Auswahl der Schneide:

- Die Schneidenummer wird programmiert.
- Die Schneidenummer wird vorgegeben durch das Maschinendatum:

MD20270 \$MC\_CUTTING\_EDGE\_DEFAULT (Grundstellung der Werkzeugschneide ohne Programmierung)

Wert	Bedeutung
= 0	Nach M06 erfolgt keine automatische Schneidenauswahl.
< > 0	Nummer der Schneide, die nach M06 ausgewählt ist.
= -1	Die Schneiden-Nr. des alten Werkzeugs bleibt erhalten und wird nach M06 auch für das neue Werkzeug ausgewählt.

## Aktivieren der WZK

Mit  $D_1$  bis  $D_9$  wird die Werkzeugkorrektur einer WZ-Schneide für das aktive Werkzeug aktiviert. Die Werkzeuglängenkorrektur und die Werkzeugradiuskorrektur werden jedoch zu unterschiedlichen Zeitpunkten wirksam:

- Die **Werkzeuglängenkorrektur (WLK)** wird mit der ersten Verfahrbewegung der Achse, in der die WLK wirken soll, herausgefahren.

Diese Verfahrbewegung muss eine Linearinterpolation ( $G_0, G_1, POS, POSA$ ) oder Polynominterpolation ( $POLY$ ) sein. Ist die  $POS/POSA$ -Achse eine der aktiven Geometrieachsen, wird die Werkzeuglängenkorrektur mit der ersten Verfahrbewegung der Achse, in der die WLK wirken soll, herausgefahren.

- Die **Werkzeugradiuskorrektur (WRK)** wird durch Programmierung von  $G_{41}/G_{42}$  in der aktiven Ebene ( $G_{17}, G_{18}$  oder  $G_{19}$ ) wirksam.

Die Anwahl der WRK mit  $G_{41}/G_{42}$  darf nur in einem Programmsatz mit  $G_0$  (Eilgang) oder  $G_1$  (Linearinterpolation) erfolgen.

## 17.2.2 Korrekturspeicherstruktur

### WZK-Speichergroße

Jeder Kanal kann einen eigenen WZK-Speicher (TO-Einheit) haben.

Welcher WZK-Speicher für den entsprechenden Kanal existiert, wird eingestellt mit dem Maschinendatum:

MD28085 \$MC\_MM\_LINK\_TOA\_UNIT (Zuordnung der TO-Einheit zu einem Kanal)

Die maximale Anzahl von WZ-Schneiden für alle von der NCK verwalteten Werkzeuge wird eingestellt mit dem Maschinendatum:

MD18100 \$MN\_MM\_NUM\_CUTTING\_EDGES\_IN\_TOA (Anzahl der Werkzeugschneiden in der NCK)

### Werkzeuge

Der WZK-Speicher besteht aus Werkzeugen mit den Nummern  $T_1$  bis  $T_{32000}$ .

Jedes Werkzeug kann über TOA-Dateien oder einzeln mit dem Softkey "Neues Werkzeug" eingerichtet werden. Nicht benötigte Korrekturen sind mit dem Wert Null zu belegen. Dies ist auch die Voreinstellung beim Anlegen der Korrekturspeicher. Die einzelnen Werte der Korrekturspeicher (WZ-Parameter) sind über Systemvariablen vom Programm les- und schreibbar.

---

#### Hinweis

Die Werkzeuge ( $T_1$  bis  $T_{32000}$ ) müssen nicht in aufsteigender Reihenfolge, nicht lückenlos und nicht mit  $T_1$  beginnend in den WZK-Speicher eingebracht werden.

---

### WZ-Schneiden

Jedes Werkzeug kann bis zu 9 WZ-Schneiden ( $D_1$  bis  $D_9$ ) besitzen. Wird ein neues Werkzeug in den WZK-Speicher eingebracht, wird automatisch die erste WZ-Schneide ( $D_1$ ) eingerichtet. Weitere WZ-Schneiden (max. 8) werden mit dem Softkey "Neue Schneide" in aufsteigender Reihenfolge lückenlos eingerichtet. Auf diese Weise kann jedem Werkzeug eine unterschiedliche Anzahl von WZ-Schneiden zugeordnet werden.

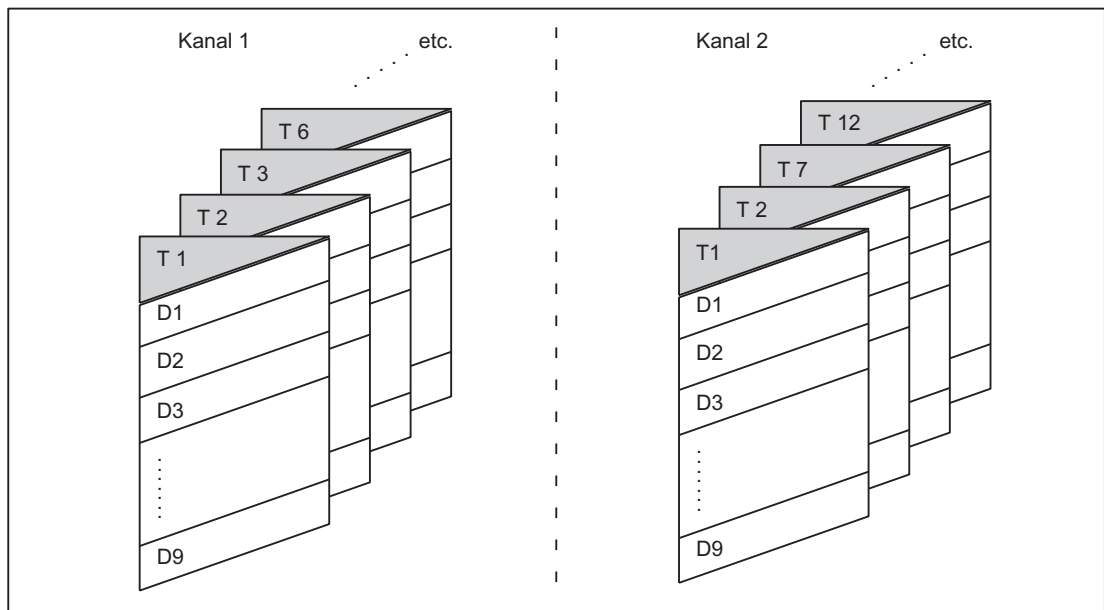


Bild 17-2 Beispiel einer WZK-Speicherstruktur für 2 Kanäle

### 17.2.3 Verrechnung der WZ-Korrektur

#### D-Nr.

Die **D-Nr.** reicht zur Ermittlung der WZ-Korrekturen aus (über MD einstellbar).

D-Nr.	DP1	DP2	DP3	...	...	DP25
D1						
D2						
:						
Dn						

Korrekturwert

#### Programmierung

Obiger Korrekturblock soll in der NC verrechnet werden.

Aufruf im Teileprogramm:

```

...
Dn

```

### 17.2.4 Adresserweiterung für die NC-Adressen T und M

#### MD20096

Ob auch bei **nicht** aktivierter Werkzeugverwaltung die Adresserweiterung von T und M als Spindelnummer interpretiert werden soll, kann festgelegt werden über das Maschinendatum:

MD20096 \$MC\_T\_M\_ADDRESS\_EXT\_IS\_SPINO (Spindelnummer als Adresserweiterung)

Für den Bezug der D-Nummer zur T-Nummer gelten dann dieselben Regeln wie bei aktiver Funktion "Werkzeugverwaltung".

### Auswirkung auf die D-Nummer

Über die D-Nummer wird ein Korrekturdatensatz bestimmt.

Die D-Adresse kann nicht mit Adresserweiterung programmiert werden.

Die Auswertung der D-Adresse bezieht sich immer auf das momentan aktive Werkzeug.

Die programmierte D-Adresse bezieht sich auf das bzgl. der Masterspindel aktive Werkzeug (analog zur Funktion Werkzeugverwaltung) bei gesetztem Maschinendatum:

MD20096 \$MC\_T\_M\_ADDRESS\_EXT\_IS\_SPINO = TRUE (Spindelnummer als Adresserweiterung)

### Auswirkung auf die T-Nummer

Bei aktiver Funktion "Werkzeugverwaltung" werden die bzgl. der Masterspindel (bzw. Master-WZ-Halter) programmierten Werte als programmierte/aktive T-Nummern angezeigt.

Ohne aktive Werkzeugverwaltung werden **alle** programmierten T-Werte als programmiert/aktiv angezeigt (unabhängig davon, welche Adresserweiterung dazu programmiert wurde).

Nur noch der bzgl. der Masterspindel programmierte T-Wert wird als programmiert/aktiv angezeigt, wenn:

MD20096 \$MC\_T\_M\_ADDRESS\_EXT\_IS\_SPINO = TRUE (Spindelnummer als Adresserweiterung)

### Beispiel

Im folgenden Beispiel wird die Auswirkung von MD20096 gezeigt.

Es werden zwei Spindeln betrachtet. Die Spindel 1 ist Masterspindel. M6 wurde als WZ-Wechselsignal festgelegt.

```
T1 = 5  
M1 = 6  
T2 = 50  
M2 = 6  
D4
```



- Bei aktiver Werkzeugverwaltung bezieht sich  $D_4$  auf das Werkzeug "5".  
T2=50 bestimmt das Werkzeug für die Nebenspindel, deren Werkzeug nicht die Korrektur der Bahn beeinflusst. Die Bahn wird ausschließlich durch das für die Masterspindel programmierte Werkzeug bestimmt.
- $D_4$  bezieht sich auf das Werkzeug "50" ohne aktive Werkzeugverwaltung und mit dem Maschinendatum:  
MD20096 \$MC\_T\_M\_ADDRESS\_EXT\_IS\_SPINO = FALSE (Bedeutung der Adresserweiterung bei T, M Werkzeugwechsel)  
Weder die Adresserweiterung von T noch von M werden im NCK ausgewertet.  
Jeder WZ-Wechselbefehl bestimmt eine neue Korrektur der Bahn.
- $D_4$  bezieht sich auf das Werkzeug "5" (wie bei aktiver WZV) ohne aktive Werkzeugverwaltung und mit dem Maschinendatum:  
MD20096 \$MC\_T\_M\_ADDRESS\_EXT\_IS\_SPINO = TRUE.  
Die Adresserweiterung 1 ( $T_1=...$ ,  $M_1=...$ ) bezeichnet die Masterspindel.

---

**Hinweis**

Bisher führt bei nicht aktivierter Werkzeugverwaltung jede Programmierung eines Werkzeugwechsels (mit T bzw. M) zu einer Neueinrechnung der WZ-Korrektur in die Bahn. Die Adresserweiterung wird dabei nicht näher definiert. Die Bedeutung der Erweiterung wird vom Anwender festgelegt (über PLC-Anwenderprogramm).

---

## 17.2.5 Freie D-Nummernvergabe

### "Relative" D-Nummern

Im NCK haben Sie die Möglichkeit, die D-Nummern für die WZ-Korrekturdatensätze als "relative" D-Nummern zu führen. Dabei werden jeder T-Nummer die entsprechenden D-Nummern zugeordnet. Bisher war die Anzahl der D-Nummern auf maximal 9 begrenzt.

## Funktionen

Erweiterungen der Funktionen bei der Vergabe der D-Nummern:

- Die maximal erlaubte D-Nummer wird festgelegt über das Maschinendatum:  
MD18105 \$MN\_MM\_MAX\_CUTTING\_EDGE\_NO (Max. Wert der D-Nummern (DRAM))  
Es ist der Wert 9 voreingestellt, so dass bestehende Anwendungen weiterhin lauffähig bleiben.
- Die Anzahl der Schneiden (bzw. Korrekturdatensätze) **je Werkzeug** kann festgelegt werden über das Maschinendatum:  
MD18106 \$MN\_MM\_MAX\_CUTTING\_EDGE\_PERTOOL (Max. Anzahl der D-Nummern pro Werkzeug (DRAM))  
Damit kann die Anzahl der zu parametrierenden Schneiden pro Werkzeug an die Anzahl der real möglichen Schneiden angepasst und so überwacht werden.
- Es gibt außerdem die Möglichkeit, D-Nummern im NCK umzubenennen und damit beliebige D-Nummern für die Schneiden zu vergeben.

---

### Hinweis

Neben der relativen D-Nummernvergabe können die D-Nummern auch als "flache" bzw. "absolute" D-Nummern (1-32000) ohne Bezug zu einer T-Nummer vergeben werden (innerhalb der Funktion "Flache D-Nummernstruktur").

---

## Schneidenummer CE

Beim Umbenennen der D-Nummern geht die Information über die im Werkzeugkatalog für diese Schneiden festgelegte Nummer verloren. Es kann somit nach dem Umbenennen nicht mehr erkannt werden, um welche Schneide des Kataloges es sich handelt.

Da diese Information bei Umrüstvorgängen benötigt wird, wurde für jede Schneide eine **Schneidenummer CE** eingeführt, die beim Umbenennen der D-Nummer erhalten bleibt.

Die D-Nummer bezeichnet im Teileprogramm die Schneidenkorrektur. Diese **Korrekturnummer D** wird getrennt von der **Schneidenummer CE** (die Nummer im WZ-Katalog) verwaltet und ist eine beliebige Zahl, mit der eine Korrektur im Teileprogramm und auf der Anzeige benannt wird.

Die CE-Nummer bezeichnet beim Umrüsten die konkrete physikalische Schneide. Die Schneidenummer CE wird vom NCK bei der Korrekturanwahl beim Werkzeugwechsel nicht ausgewertet (nur über BTSS verfügbar).

Die Schneidenummer CE wird über die Systemvariable **\$TC\_DPCE[t,d]** beschrieben:

- **t** steht für die interne T-Nummer
- **d** steht für die D-Nummer.

Schreibvorgänge werden auf Kollision überwacht; d.h. die Schneidennummern eines Werkzeugs müssen alle verschieden sein. Die Variable \$TC\_DPCE ist Bestandteil des Schneidenparameterdatensatzes \$TC\_DP1,..., \$TC\_DP25.

Die Parametrierung von \$TC\_DPCE ist nur sinnvoll, wenn die maximale Schneidnummer (MD18105) größer ist als die maximale Anzahl der Schneiden je Werkzeug (MD18106).

Dann ist die Vorbelegung der Schneidnummer gleich der Ordnungsnummer der Schneide. Die angelegten Korrekturen eines Werkzeugs werden mit 1 beginnend bis zur maximalen Anzahl der Schneiden je Werkzeug hochgezählt (MD18106).

Die Schneidnummer CE ist gleich der D-Nummer (in Kompatibilität zum bisherigen Verhalten) falls:

$MD18105 \leq MD18106$ .

Eine Leseoperation liefert  $CE=D$ . Eine Schreiboperation wird ohne Alarmmeldung ignoriert.

---

**Hinweis**

Die Korrekturwerte \$TC\_DP1,..., \$TC\_DP25 der aktiven WZ-Korrektur können mit der Systemvariablen \$P\_AD[n]; mit  $n=1,...,25$  gelesen werden. Die Schneidnummer CE der aktiven Korrektur erhält man für  $n=26$ .

---

**Befehle**

Wenn die maximale Schneidenummer:  
 MD18105 \$MN\_MM\_MAX\_CUTTING\_EDGE\_NO (Max. Wert der D-Nummern (DRAM))  
 größer ist als die maximale Anzahl der Schneiden je Werkzeug:  
 MD18106 \$MN\_MM\_MAX\_CUTTING\_EDGE\_PERTOOL (Max. Anzahl der D-Nummern pro  
 Werkzeug (DRAM))  
 sind folgende Befehle verfügbar:

Befehl	Bedeutung
CHKDNO	Prüft die vorhandenen D-Nummern auf Eindeutigkeit; Die D-Nummern aller innerhalb einer TO-Einheit definierten Werkzeuge dürfen nur einmal auftreten. Ersatzwerkzeuge werden dabei nicht berücksichtigt.
GETDNO	Ermittelt zu einer Schneide eines Werkzeuges die D-Nummer. Existiert keine D-Nummer zu den eingegebenen Parametern, wird d=0 gesetzt. Ist die D-Nummer ungültig wird ein Wert größer 32000 zurückgegeben.
SETDNO	Setzt oder ändert die D-Nummer der Schneide CE eines Werkzeuges T. Existiert kein Datensatz zu den eingegebenen Parametern wird FALSE zurückgegeben. Syntaxfehler erzeugen einen Alarm. Die D-Nummer kann nicht explizit auf 0 gesetzt werden.
GETACTTD	Ermittelt zu einer absoluten D-Nummer die dazugehörige T-Nummer. Es erfolgt keine Prüfung auf Eindeutigkeit. Gibt es mehrere gleiche D-Nummern innerhalb einer TO-Einheit, wird die T-Nummer des ersten gefundenen Werkzeuges zurückgegeben. Bei Verwendung "flacher" D-Nummern ist die Verwendung des Befehls nicht sinnvoll, da hier immer der Wert 1 zurückgegeben wird (keine T-Nummer in der Datenhaltung).
DZERO	Kennzeichnet alle D-Nummern der TO-Einheit als ungültig; Der Befehl dient zur Unterstützung während dem Umrüsten. So gekennzeichnete Korrekturdatensätze werden nicht mehr vom Sprachbefehl CHKDNO geprüft. Um sie wieder zugänglich zu machen, muss die D-Nummer wieder mit SETDNO gesetzt werden.

**Hinweis**

Ist die maximale Schneidenummer kleiner als die maximale Anzahl der Schneiden pro Werkzeug (**MD18105 < MD18106**) sind die beschriebenen Sprachbefehle ohne Wirkung auf das System.

Um vorhandene Anwendungen lauffähig zu halten, ist deshalb diese Relation im NCK standardmäßig voreingestellt.

Die einzelnen Befehle sind ausführlich beschrieben in:

**Literatur:**  
 Programmierhandbuch Grundlagen

## Aktivierung

Um mit eindeutigen D-Nummern und damit mit den beschriebenen Sprachbefehlen arbeiten zu können, müssen die D-Nummern für die Werkzeuge frei benannt werden können.

Dazu müssen folgende Voraussetzungen erfüllt sein:

- MD18105 > MD18106
- Die Funktion "Flache D-Nummer" ist nicht aktiviert.  
MD18102 \$MN\_MM\_TYPE\_OF\_CUTTING\_EDGE (Art der D-Nummer Programmierung (SRAM))

## Beispiele

**MD18105 \$MN\_MM\_MAX\_CUTTING\_EDGE\_NO = 1** (Max. Wert der D-Nummern (DRAM))

Damit kann pro Werkzeug maximal eine Korrektur definiert werden (mit D-Nummer = 1).

---

### Hinweis

Bei aktivierter Funktion "Flache D-Nummern" kann damit in der TO-Einheit nur genau eine D-Korrektur definiert werden.

---

**MD18105 \$MN\_MM\_MAX\_CUTTING\_EDGE\_NO = 9999**

Damit können Werkzeuge mit eindeutigen D-Nummern versehen werden.

Es werden z.B.:

- der T-Nr. 1 die D-Nummern 1, 2, 3 zugeordnet
- der T-Nr. 2 die D-Nummern 10, 20, 30, 40, 50 zugeordnet
- der T-Nr. 3 die D-Nummern 100, 200 zugeordnet
- etc.

CHKDNO; **MD18105 \$MN\_MM\_MAX\_CUTTING\_EDGE\_NO = 9999**

Es sollen folgende Daten auf Eindeutigkeit der D-Nummern geprüft werden:

- T-Nr. 1 mit D-Nummern 1, 2, 3
- T-Nr. 2 mit D-Nummern 10, 20, 30, 40, 50
- T-Nr. 3 mit D-Nummern 100, 200, 30

(Tippfehler bei der Definition: Statt 300 wurde 30 eingegeben)

CHKDNO	Wird beim Prüfen dieser Konstellation den Zustand <code>FALSE</code> zurückgeben, denn D=30 ist doppelt vorhanden.
CHKDNO (2, 3, 30)	Wird beim Prüfen der angegebenen D-Nummer 30, den Zustand <code>FALSE</code> zurückgeben, da D=30 doppelt vorkommt
CHKDNO (2, 3, 100)	Wird den Zustand <code>TRUE</code> zurückgeben, denn D=100 gibt es genau einmal.
CHKDNO (1, 3)	Wird den Zustand <code>TRUE</code> zurückgeben, obwohl D=30 vom dritten Werkzeug mit D=30 vom zweiten Werkzeug kollidiert.

MD18106 \$MN\_MM\_MAX\_CUTTING\_EDGE\_PERTOOL = 1 (Max. Anzahl der D-Nummern pro Werkzeug (DRAM))

Es werden nur Werkzeuge benutzt, die genau eine Schneide haben. Der Wert 1 des Maschinendatums verhindert die Definition einer zweiten Schneide für ein Werkzeug.

MD18106 \$MN\_MM\_MAX\_CUTTING\_EDGE\_PERTOOL = 12

Es können maximal 12 Schneiden für ein Werkzeug definiert werden.

## Programmierbeispiele

### Umbenennen einer D-Nummer

Es soll die D-Nummer der Schneide mit CE = 3 von 2 auf den Wert 17 umbenannt werden. Dazu sollen folgende Vorgaben gelten:

- Interne T-Nummer T = 1
- D-Nummer = 2
- Werkzeug mit einer Schneide mit:

Programmcode	Kommentar
\$TC_DP2[ 1, 2 ] = 120	
\$TC_DP3[ 1, 2 ] = 5.5	
\$TC_DPCE[ 1, 2 ] = 3	; Schneidenummer CE

- MD18105 \$MN\_MM\_MAX\_CUTTING\_EDGE\_NO = 20 (Max. Wert der D-Nummern (DRAM))

Innerhalb des Teileprogramms wird diese Korrektur standardmäßig mit `T1, . . . .D2` programmiert.

Sie weisen die momentane D-Nummer der Schneide 3 einer Variablen (DNrAlt) zu und definieren für die neue D-Nummer die Variable DNrNeu:

Programmcode
def int DNrAlt, DNrNeu = 17
DNrAlt = GETDNO( 1, 3 )
SETDNO( 1, 3, DNrNeu )

Damit wird der Schneide CE=3 der neue D-Wert 17 zugewiesen.

Jetzt werden die Daten dieser Schneide über die D-Nummer 17 angesprochen; sowohl über die Systemvariable, als auch in der Programmierung mit der NC-Adresse D.

Innerhalb des Teileprogramms wird diese Korrektur nun standardmäßig mit T1, . . . .D17 programmiert und die Daten werden folgendermaßen angesprochen:

Programmcode	Kommentar
\$TC_DP2[ 1, 17 ] = 120	
\$TC_DP3[ 1, 17 ] = 5.5	
\$TC_DPCE[ 1, 17 ] = 3	; Schneidenummer CE

### Hinweis

Sofern das Werkzeug noch eine weitere Schneide definiert hat, z.B.: \$TC\_DPCE[ 1, 2 ] = 1 ; = CE, kann die D-Nummer 2 der Schneide 1 nicht gleich benannt werden wie die D-Nummer der Schneide 3; d.h.: SETDNO( 1, 1, 17 ) liefert den Status = FALSE als Rückgabewert.

### DZERO - D-Nummern ungültig setzen

Mit Aktivieren dieses Befehls werden alle D-Nummern der Werkzeuge der TO-Einheit ungültig. Es kann keine Korrektur mehr aktiviert werden, bis wieder gültige D-Nummern im NCK verfügbar sind. Die D-Nummern werden über den Befehl SETDNO neu vergeben.

Es sind folgende Werkzeuge vorgegeben (alle mit der Schneidenummer1):

T1, D1	D-Nr. der Schneide CE=1
T2, D10	D-Nr. der Schneide CE=1
T3, D100	D-Nr. der Schneide CE=1

Nun wird programmiert:

Programmcode
DZERO

Wird jetzt eine der Korrekturen aktiviert (z.B. mit T3 D100), wird ein Alarm erzeugt, da D100 momentan nicht definiert ist.

Die D-Nummern werden neu definiert über:

Programmcode	Kommentar
SETDNO( 1, 1, 100 )	; T=1, Schneide 1 erhält die (neue) D-Nummer 100
SETDNO( 2, 1, 10 )	; T=2, Schneide 1 erhält die (alte) D-Nummer 10
SETDNO( 3, 1, 1 )	; T=3, Schneide 1 erhält die (neue) D-Nummer 1

### Hinweis

Der Befehl DZERO kann bei Spannungsausfall den NCK bzgl. der D-Nummern in einem undefinierten Zustand hinterlassen. Um dies zu beheben, wiederholen Sie nach dem Wiedereinschalten den Befehl DZERO.

### Prinzipieller Ablauf eines Umrüstprogramms

Es soll sichergestellt werden, dass die benötigten Werkzeuge und Schneiden vorhanden sind. Das Magazin, der Werkzeugspeicher der NCK, befindet sich in einem beliebigen Zustand. Die D-Nummern in den Teileprogrammen der neuen Bearbeitung stimmen meist nicht mit den D-Nummern der konkreten Schneiden überein. Das Umrüstprogramm kann wie folgt aussehen:

Programmcode	Kommentar
DZERO	; alle D-Nummern der TO-Einheit werden als ungültig gekennzeichnet
....	; Schleife oder mehrere Schleifen über die Plätze des Magazins bzw. der Magazine mit Prüfung der Werkzeuge und ihrer Schneidenummern.  Wird ein WZ gefunden, das noch nicht gesperrt ist (\$TC_TP8) und das die gewünschte Schneidenummer CE hat (GETDNO), dann wird die Schneide mit der neuen D-Nummer versehen (SETDNO).
....	; Entlade- und Beladevorgänge werden durchgeführt.  Es kann mit dem Status 'zu entladen' bzw. 'zu beladen' des Werkzeuges gearbeitet werden.
CHKDNO	; Das Ent-/ Beladen und der Vorgang zum Umbenennen der D-Nummern sind abgeschlossen.  Es können einzelne Werkzeuge und / oder D-Nummern geprüft und anhand des Rückgabewertes auftretende Kollisionen automatisch behandelt werden.



## 17.2.6 Korrektursatz bei Fehler im Werkzeugwechsel

### MD22550

Wenn im Teileprogramm eine Werkzeugvorbereitung programmiert ist und der NCK stellt dabei einen Fehler fest (z.B. der Datensatz zur programmierten T-Nummer ist im NCK nicht vorhanden), hat der Bediener die Möglichkeit, die Fehlersituation zu beurteilen und entsprechende Handlungen vorzunehmen, um anschließend die Bearbeitung fortzusetzen.

Der Werkzeugwechsel kann unabhängig programmiert werden, in Abhängigkeit des Maschinendatums:

MD22550 \$MC\_TOOL\_CHANGE\_MODE (Neue Werkzeugkorrektur bei M-Funktion)

**MD22550 \$MC\_TOOL\_CHANGE\_MODE = 0**

```
T= 'T-Nr.'      ; WZ-Vorbereitung + Werkzeugwechsel in einem NC-Satz,
                ; d.h. mit Programmierung von T erfolgt im NCK das
                ; Aktivwerden einer neuen D-Korrektur. Siehe dazu das
                ; Maschinendatum:
                ; MD20270 $MC_CUTTING_EDGE_DEFAULT (Grundstellung der
                ; Werkzeugschneide ohne Programmierung)
```

**MD22550 \$MC\_TOOL\_CHANGE\_MODE = 1**

```
T= 'T-Nr.'      ; Werkzeugvorbereitung
M06             ; Werkzeugwechsel
                ; (die Nummer des WZ-Wechsel-M-Codes ist einstellbar)
                ; d.h. mit Programmierung von M06 erfolgt in NCK das
                ; Aktivwerden einer neuen D-Korrektur; (siehe dazu das
                ; Maschinendatum MD20270 $MC_CUTTING_EDGE_DEFAULT)
```

Bei nicht aktiver Werkzeugverwaltung können folgende Probleme erkannt werden:

- D-Korrekturdatensatz fehlt
- Fehler im Teileprogramm

---

#### Hinweis

Das Problem "Werkzeug ist nicht im Magazin" kann nicht erkannt werden, da bei der Werkzeugkorrektur dem NCK keine Magazininformationen vorliegen.

---

### D-Korrekturdatensatz fehlt

Die Programmabarbeitung bleibt auf dem Satz mit dem fehlerhaften D-Wert stehen (unabhängig vom Wert des Maschinendatums MD22550). Der Bediener muss entweder das Programm korrigieren, oder er muss den fehlenden Datensatz nachladen.

Dazu benötigt er bei flacher D-Nummernfunktion die D-Nummer und ansonsten zusätzlich die T-Nummer. Diese Parameter werden durch den ausgelösten Alarm mit übergeben.

### Fehler im Teileprogramm

Die Bearbeitungsmöglichkeiten im Fehlerfall hängen davon ab, wie der Werkzeugwechsel programmiert wurde, festgelegt über das Maschinendatum:

MD22550 \$MC\_TOOL\_CHANGE\_MODE (Neue Werkzeugkorrektur bei M-Funktion)

#### Werkzeugwechsel mit T-Programmierung (MD22550 = 0)

Für diesen Fall erfolgt die Bearbeitung über die im NCK verfügbare Funktion "Korrektursatz". Das NC-Programm stoppt auf dem NC-Satz, in dem ein Fehler beim programmierten T-Wert erkannt wurde. Beim Fortsetzen des Programms wird der "Korrektursatz" erneut abgearbeitet.

Durch den Bediener können folgende Bearbeitungen vorgenommen werden:

- Das fehlerhafte Teileprogramm kann korrigiert werden.
- Fehlende Schneidenkorrektur-Daten können von HMI nachgeliefert werden.
- Fehlende Schneidenkorrektur-Daten können in "Überspeichern" im NCK eingebracht werden.

Nach dem Bedienereingriff wird die *START*-Taste gedrückt und der beanstandete Satz wird wieder abgearbeitet. Wurde der Fehler richtig korrigiert, wird das Programm fortgesetzt. Ansonsten wird erneut ein Alarm erzeugt.

#### Werkzeugwechsel mit T und M06-Programmierung (MD22550 = 1)

In diesem Fall wird im NC-Satz der WZ-Vorbereitung (Programmierung von T) ein Fehler bemerkt, der aber zunächst ignoriert werden soll. Die Verarbeitung wird fortgesetzt, bis die WZ-Wechselanforderung (meist *M06*) im NC-Programm abgearbeitet wird. An dieser Stelle soll das Programm "halten".

Die programmierte T-Adresse kann beliebig viele Programmzeilen vor dem M06-Befehl stehen, oder die beiden Anweisungen können in unterschiedlichen (Unter-)Programmen stehen. Deshalb ist es im Allgemeinen nicht möglich, einen abgearbeiteten Satz über Korrektursatz zu ändern.

Der Bediener hat dieselben Bearbeitungsmöglichkeiten wie bei = 0.

Das Nachladen fehlender Daten ist möglich. Dann muss aber beim "Überspeichern" T programmiert werden.

Sofern ein Programmfehler vorliegt, kann nicht die Zeile mit dem Fehler korrigiert werden (Txx), sondern nur die Zeile, auf der die Programmabarbeitung gestoppt und den Alarm erzeugt hat. Nur wenn das Maschinendatum:

MD22562 \$MC\_TOOL\_CHANGE\_ERROR\_MODE Bit0 = 1 (Verhalten bei Fehlern im Werkzeugwechsel)

Es ergibt sich folgender Ablauf:

```
Txx          ; Fehler! Datensatz mit xx gibt es nicht
              ; Zustand merken; xx merken;
              ; weitermachen im Programm
....
M06          ; Merker 'xx fehlt' erkennen → Alarm ausgegeben,
              ; Programm stoppen
              ; Satz korrigieren mit z.B. Tyy M06, starten,
              ; der Satz Tyy M06 wird interpretiert und ist o.k.
              ; Die Bearbeitung geht weiter.
```

Bei erneuter Abarbeitung der Programmstelle ergibt sich folgendes:

```
Txx          ; Fehler! Datensatz mit xx gibt es nicht,
              ; Zustand merken; xx merken;
              ; weitermachen im Programm
....
Tyy M06      ; Merker "xx fehlt" erkennen → ohne weitere Reaktion verwerfen,
              ; denn Tyy M06 ist korrekt → Programm stoppt nicht (richtig).
```

Falls erforderlich kann nach Programmende die ursprüngliche Stelle des T-Aufrufs korrigiert werden. Falls die WZ-Wechsellogik an der Maschine dies nicht verarbeiten kann, muss das Programm abgebrochen und die fehlerhafte Stelle korrigiert werden.

Sofern nur der Datensatz fehlt, wird dieser zum NCK übertragen, in "Überspeichern" Txx programmiert und anschließend das Programm fortgesetzt.

Wie bei "fehlender D-Nummer" wird auch bei "fehlender T-Nummer" über den entsprechenden Alarm (17191) der erforderliche Parameter (T-Nummer) dem Bediener zur Verfügung gestellt.

---

### Hinweis

Um die Programmkorrektur zu ermöglichen, wird sofort bei dem fehlerhaften Txx-Satz angehalten.

Bei Programmtestbetrieb wird auch angehalten beim Maschinendatum:

MD22562 \$MC\_TOOL\_CHANGE\_ERROR\_MODE Bit0=1 (Verhalten bei Fehlern im Werkzeugwechsel)

---

## 17.2.7 Definition der Wirkung der Werkzeugparameter

### MD20360

Die Wirkung der Werkzeugparameter auf die Planachse kann im Zusammenhang mit Durchmesserprogrammierung gezielt gesteuert werden mit dem Maschinendatum:

MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK (Definition der WZ-Parameter)

Details sind beim genannten MD beschrieben.

### DRF-Handradverfahren mit halbem Weg

Beim DRF-Handradverfahren kann eine Planachse nur mit dem halben Weg des vorgegebenen Inkrements wie folgt verfahren werden:

Festlegung der Wegvorgabe mit Handrad über das Maschinendatum:

MD11346 \$MN\_HANDWHEEL\_TRUE\_DISTANCE = 1 (Handrad Weg- oder Geschwindigkeitsvorgabe)

Definition der DRF-Verschiebung in der Planachse als Durchmesserkorrekturwert einrechnen mit dem Maschinendatum:

MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK Bit 9 = 1 (Definition der WZ-Parameter)

Bei der axialen Abwahl  $DRFOF$  einer DRF-Verschiebung wird eine existierende Werkzeugkorrektur (Handradüberlagerung in Werkzeugrichtung) auch gelöscht.

---

#### Hinweis

Weitere Informationen zu überlagerten Bewegungen mit Handrad entnehmen Sie bitte:

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1)

Programmierhandbuch Grundlagen

(Die Programmieranleitung beschreibt die vollständigen programmtechnischen Möglichkeiten, um die DRF-Verschiebung achsspezifisch abzuwählen.)

---

## 17.3 Flache D-Nummern-Struktur

### 17.3.1 Allgemeines

#### Einfache WZV

Für Drehmaschinen ist eine einfache WZV (keine Schwester-Werkzeuge, keine Magazine) über D-Nummern möglich.

Die Funktion ist in der Grundstufe (ohne aktivierte WZV-Funktion) verfügbar. Schleifwerkzeuge können damit nicht definiert werden.

#### Aktivierung

Welche Art der D-Nummern-Verwaltung wirksam ist, kann festgelegt werden durch das Maschinendatum:

MD18102 \$MN\_MM\_TYPE\_OF\_CUTTING\_EDGE (Art der D-Nummer Programmierung)

Wert	Bedeutung
0	wie bisher = Standard-Einstellung
1	flache D-Nummern-Struktur mit absoluter <b>direkter D-Programmierung</b>

Schneiden können einzeln über PI-Befehl oder NC-Programmierbefehl gelöscht werden. Über HMI können Schneiden auch gezielt mit einer bestimmten Nummer angelegt werden.

### 17.3.2 Neuanlegen einer D-Nummer (Korrektursatz)

#### Programmierung

Mit den Systemvariablen \$TC\_DP1... \$TC\_DP25 können Werkzeugkorrekturen programmiert werden. Inhaltlich ist die Bedeutung dieselbe wie bisher.

Die Schreibweise ändert sich: es wird keine T-Nummer mehr angegeben.

- Funktion "Flache D-Nummer" **aktiv**:

**\$TC\_DPx[d]** = wert ;mit x=Parameternr., d=D-Nummer

D.h., Daten mit dieser Syntax können nur nach NCK geladen werden, wenn die Funktion "Flache D-Nummer" aktiviert ist.

- Funktion 'Flache D-Nummer' **inaktiv**:

**\$TC\_DPx[t][d]** = wert ;mit t=T-Nummer, d=D-Nummer

Eine D-Nummer kann je Werkzeug nur einmal vergeben werden; d.h. jede D-Nummer steht genau für einen Korrekturdatensatz.

Ein neuer Datensatz wird im Speicher von NCK dann angelegt, wenn zum ersten Mal eine nicht existierende D-Nummer programmiert wird.

Die max. Anzahl der D- bzw. Korrektur-Datensätze (max. 600) wird eingestellt über das Maschinendatum:

MD18100 \$MN\_MM\_NUM\_CUTTING\_EDGES\_IN\_TOA (Werkzeugkorrekturen im TO-Bereich)

### Datensicherung

Die Datensicherung erfolgt im selben Format; d.h. eine Sicherungsdatei, die mit der Funktion "Flache D-Nummer" erstellt wird, kann nicht in den NCK einer Steuerung gespielt werden, die die Funktion nicht aktiviert hat.

Das gilt auch für die umgekehrte Richtung einer Übertragung.

### D-Nummernbereich

1 - 99 999 999

## 17.3.3 D-Nummer programmieren

### MD18102 = 1

Wenn MD18102 \$MN\_MM\_TYPE\_OF\_CUTTING\_EDGE = 1, dann wird die D-Korrektur ohne Bezug zu einem bestimmten Werkzeug aktiviert.

D<sub>0</sub> behält weiterhin die bisherige Bedeutung "Abwahl der aktiven Korrektur in NCK".

### Adresserweiterung von D

Eine Adresserweiterung von D ist nicht möglich. Für die WZ-Bahn gibt es zu einem Zeitpunkt nur einen möglichen aktiven Korrekturdatensatz.

## Programmierung

Im Teileprogramm wird wie bisher programmiert. Nur der Wertebereich der programmierten D-Nummer erhöht sich.

### Beispiel 1:

MD-Parametrierung		Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE	= 0	Werkzeugwechsel mit der Programmierung von T.
MD18102 \$MN_MM_TYPE_OF_CUTTING_EDGE	= 1	D-Nummer-Programmierung ohne Bezug zu einem bestimmten Werkzeug.
MD20270 \$MC_CUTTING_EDGE_DEFAULT	= -1	D-Korrektur bleibt bei Werkzeug-Wechsel unverändert.

Programmcode	Kommentar
...	
D92	
X0	; Verfahre mit den Korrekturen von D92.
T17	; Gibt T=17 an PLC aus.
X1	; Verfahre mit den Korrekturen von D92.
D16	
X2	; Verfahre mit den Korrekturen von D16.
D32000	
X3	; Verfahre mit den Korrekturen von D32000.
T29000	; Gibt T=29000 an PLC aus.
X4	; Verfahre mit den Korrekturen von D32000.
D1	
X5	; Verfahre mit den Korrekturen von D1.
...	

**Beispiel 2:**

MD22550 = 0

Programmcode	Kommentar
T1	
T2	
T3	
D777	; Kein Warten, D777 wird aktiviert, T3= programmiertes und aktives WZ in der Anzeige, D777= programmierte und aktive Korrektur.

**Hinweis**

Der WZ-Wechsel und die Zuordnung einer D-Korrektur zu einem konkreten WZ liegt in der Verantwortung des NC-Programms und gegebenenfalls des PLC-Programms.

**D-Nr. über Teileprogramm löschen**

- **Mit** flacher D-Nummer:  
\$TC\_DP1[d] = 0  
Korrekturdatensatz mit der Nummer D in der TO-Einheit wird gelöscht.  
Der Speicher ist anschließend frei für die Definition einer anderen D-Nummer.
- **Ohne** flache D-Nummer:  
\$TC\_DP1[t][d] = 0  
Schneide d des Werkzeugs t wird gelöscht.
- \$TC\_DP1[0] = 0  
Alle D-Korrekturen der TO-Einheit werden gelöscht.

Aktive Korrektur-Datensätze (D-Nummern) können nicht gelöscht werden. D. h. es kann eventuell erforderlich sein, vor dem Löschen ein D0 zu programmieren.



## Werkzeug-MDs

Folgende Maschinendaten beeinflussen den Umgang mit Werkzeugen und Schneiden (D-Nummern) im NCK:

Maschinendatum	Bedeutung
MD20270 \$MC_CUTTING_EDGE_DEFAULT	Standard-Werkzeugschneide nach Werkzeugwechsel
MD20130 \$MC_CUTTING_EDGE_RESET_VALUE	Werkzeugschneide - Längenkorrektur im Hochlauf (RESET / TP-Ende)
MD20120 \$MC_TOOL_RESET_VALUE	Werkzeug - Längenkorrektur im Hochlauf (RESET / TP-Ende)
MD20121 \$MC_TOOL_PRESEL_RESET_VALUE	Vorgewähltes Werkzeug bei RESET
MD22550 \$MC_TOOL_CHANGE_MODE	Werkzeugwechsel mit M-Funktion statt mit T-Funktion
MD22560 \$MC_TOOL_CHANGE_M_CODE	M-Funktion für Werkzeugwechsel
MD20110 \$MC_RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach RESET / TP-Ende
MD20112 \$MC_START_MODE_MASK	Festlegung der Steuerungs-Grundstellung bei NC-START

### 17.3.4 T-Nummer programmieren

Bei aktiver Funktion "Flache D-Nummern-Struktur" wird die NC-Adresse T weiterhin ausgewertet, d.h. die programmierte wie die aktive T-Nr. werden angezeigt. Die D-Nr. ermittelt die NC aber unabhängig vom programmierten T-Wert.

Die NC kennt je Kanal 1 Masterspindel (über MD einstellbare Spindelnummer). Korrekturen und der Befehl  $M_6$  (WZ-Wechsel) werden nur in Bezug auf die Masterspindel berechnet.

Eine Adresserweiterung von T wird als Spindel-Nr. erkannt (z.B.  $T_2 = 1$ ; WZ 1 soll an Spindel 2 angewählt werden), ein WZ-Wechsel wird nur erkannt, wenn Spindel 2 die Masterspindel ist.

### 17.3.5 M6 programmieren

#### MD22550 und MD22560

Die NC kennt je Kanal 1 Masterspindel (über MD einstellbare Spindelnummer). Korrekturen und der Befehl M6 (WZ-Wechsel) werden nur in Bezug auf die Masterspindel berechnet.

Ob mit einer M-Funktion der WZ-Wechselbefehl erfolgt wird festgelegt über das Maschinendatum:

MD22550 \$MC\_TOOL\_CHANGE\_MODE (Neue Werkzeugkorrektur bei M-Funktion)

T ist dabei der WZ-Vorbereitungsbefehl.

Der Name der M-Funktion für den WZ-Wechsel wird festgelegt über das Maschinendatum:

MD22560 \$MC\_TOOL\_CHANGE\_M\_CODE (M-Funktion für Werkzeugwechsel)

Standard ist M6. Eine Adresserweiterung von M6 wird als Spindel-Nr. erkannt.

#### Beispiel

Es sind zwei Spindeln definiert, Spindel 1 und Spindel 2, und es gilt:

```
MD20090 = 2      ; Spindel-Nr. 2 ist Masterspindel.
M6           ; WZ-Wechsel gewünscht, Befehl bezieht sich implizit auf
              Masterspindel
M1 = 6        ; kein WZ-Wechsel, da Spindel-Nr. 2 die Masterspindel ist
M2 = 6        ; WZ-Wechsel erfolgt, da Spindel-Nr. 2 die Masterspindel ist
```

### 17.3.6 Programmtest

#### MD20110

Dass das aktive WZ und die WZ-Korrektur wie folgt übernommen werden, kann eingestellt werden mit dem Maschinendatum:

MD20110 \$MC\_RESET\_MODE\_MASK, Bit 3 (Festlegung der Steuerungs-Grundstellung nach RESET/TP-Ende)

Wert		Bedeutung
Bit 3	= 1	aus dem zuletzt beendeten Testprogramm im Testbetrieb
	= 0	aus dem zuletzt beendeten Programm vor Einschalten des Programmtests

## Voraussetzung

Die Bits 0 und 6 müssen gesetzt sein vom Maschinendatum:

MD20110 \$MC\_RESET\_MODE\_MASK, Bit 3 (Festlegung der Steuerungs-Grundstellung nach RESET/TP-Ende)

## 17.3.7 Werkzeugverwaltung oder "Flache D-Nummern"

### Werkzeugverwaltung

Die aktive Werkzeugverwaltung der NCK arbeitet mit folgenden Annahmen:

1. Werkzeuge werden in Magazinen verwaltet.
2. Schneiden werden überwacht, erreichte Grenzen bewirken Sperren des Werkzeugs.
3. Idee der Schwester-Werkzeuge: Werkzeuge werden für die Anwahl nur mit ihrem Bezeichner programmiert. NCK wählt dann - je nach Strategie - das konkrete Werkzeug aus.

---

#### Hinweis

Diese Funktionalität steht bei SINUMERIK 828D nur als Option zur Verfügung!

---

Das bedeutet, dass der Einsatz der Werkzeugverwaltung nur Sinn ergibt, wenn konkrete Werkzeuge definiert sind, und wenn diese auch in der Verantwortung von NCK behandelt werden.

### Flache D-Nummer

Flache D-Nummer bedeutet, dass die Werkzeugverwaltung außerhalb NCK gemacht wird und kein Bezug zu T-Nummern hergestellt wird.

### Keine Mischung von WZV und Flacher D-Nr.

Eine Mischung bzw. Verteilung der Funktionen der Werkzeugverwaltung auf NCK und PLC ergibt wenig Sinn, da der Hauptgrund für die Werkzeugverwaltung auf NCK das Argument **Zeit** ist.

Das kann nur funktionieren, wenn die Aufgaben, die zeitkritisch sind, im NCK getan werden. Das ist bei "Flacher D-Nummer" aber nicht der Fall.

---

#### Hinweis

Die Aktivierung der beiden Funktionen "Flache D-Nummernstruktur" und "Werkzeugverwaltung" wird überwacht. Wenn beide aktiviert sind, besitzt die "Werkzeugverwaltung" die höhere Priorität.

---

## 17.4 Werkzeugschneide

### 17.4.1 Allgemeines

#### Werkzeugschneide

Eine WZ-Schneide ist eindeutig durch folgende Angaben beschrieben:

- Werkzeugtyp (Schafffräser, Bohrer, etc.)
- Geometrische Beschreibung
- Technologische Beschreibung

#### WZ-Parameter

Die geometrische Beschreibung, die technologische Beschreibung und der WZ-Typ werden für jede WZ-Schneide auf WZ-Parameter abgebildet.

Zu den jeweiligen Werkzeugtypen sind folgende WZ-Parameter vorhanden:

WZ-Parameter	Bedeutung	Bemerkung
1	Werkzeugtyp	
2	WZ-Schneidenlage	für Drehwerkzeuge oder für Fräs- /Schleifwerkzeuge bei 2D-WRK Kontur-WZ
Geometrie - Werkzeuglängen		
3	Länge 1	
4	Länge 2	
5	Länge 3	
Geometrie - Werkzeugform		
6	Radius 1/Länge 1	bei 3D Stirnfräsen
7	Länge 2	bei 3D Stirnfräsen
8	Radius 1	bei 3D Stirnfräsen
9	Radius 2	bei 3D Stirnfräsen
10	Winkel 1 / Minimaler Grenzwinkel	bei 3D Stirnfräsen bei 2D-WRK Kontur-WZ
11	Winkel 2 / Maximaler Grenzwinkel	bei 3D Stirnfräsen bei 2D-WRK Kontur-WZ
Verschleiß - Werkzeuglängen		
12	Länge 1	
13	Länge 2	
14	Länge 3	

WZ-Parameter	Bedeutung	Bemerkung
Verschleiß - Werkzeugform		
15	Radius 1/Länge 1	bei 3D Stirnfräsen
16	Länge 2	bei 3D Stirnfräsen
17	Radius 1	bei 3D Stirnfräsen
18	Radius 2	bei 3D Stirnfräsen
19	Winkel 1 / minimaler Grenzwinkel	bei 3D Stirnfräsen bei 2D-WRK Kontur-WZ.
20	Winkel 2 / maximaler Grenzwinkel	bei 3D Stirnfräsen bei 2D-WRK Kontur-WZ.
Basismaß / Adaptermaß		
21	Basis-Länge 1	
22	Basis-Länge 2	
23	Basis-Länge 3	
Technologie		
24	Freischneidwinkel	nur für Drehwerkzeuge
25	Reserviert*	

\* "Reserviert" bedeutet, dass dieser WZ-Parameter nicht benutzt wird und für Erweiterungen reserviert ist.

### 3D-Stirnfräsen

Beim 3D-Stirnfräsen werden die Fräserarten 111, 120, 121, 130, 155, 156 und 157 speziell behandelt, indem gegebenenfalls die Werkzeugparameter (1-23) ausgewertet werden.

### Literatur

Weitere Informationen zu verschiedenen Werkzeugtypen siehe:

- Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1), Kapitel: "Werkzeugtyp (WZ-Parameter)"
- Programmierhandbuch Grundlagen; Kapitel: "Werkzeugkorrekturen" > "Liste der Werkzeugtypen"
- Funktionshandbuch Sonderfunktionen; 3D-Werkzeugradiuskorrektur (W5)

## 17.4.2 WZ-Parameter 1: Werkzeugtyp

### Bedeutung

Der Werkzeugtyp (3-stellige Nummer) legt fest, um welches Werkzeug es sich handelt. Mit der Auswahl des Werkzeugtyps sind die anderen Komponenten wie Geometrie, Verschleiß oder Basismaß vorbestimmt.

### Bedingungen

Für den Parameter "Werkzeugtyp" gilt:

- Der WZ-Typ muss für jede WZ-Schneide angegeben werden.
- Der WZ-Typ darf nur die festgelegten Werte annehmen.
- Der WZ-Typ "0" bedeutet, dass kein gültiges Werkzeug definiert ist.

## Werkzeugtypen und Werkzeugparameter

In der folgenden Tabelle sind verschiedene Werkzeugtypen und die wichtigsten Werkzeugparameter aufgelistet. Die für einen bestimmten Werkzeugtyp verfügbaren Werkzeugparameter sind mit "x" gekennzeichnet:

Tabelle der wichtigsten verfügbaren Werkzeugparameter (notwendig ist nur der Werkzeugtyp)																
WZ-Parameter \$TC_DP	2	3	4	5	6	7	8/9	12/13/14	15/16	21/22/23	24	TPG 3	TPG 4/5	TPG 6	TPG 7	TPG 8
Werkzeugtyp	Schneidenlage	Geometrie – Länge 1	– Länge 2	– Länge 3	Geometrie – Radius 1	Nutbreite / – Radius 2	Überstand / – Länge 4/5	Verschleiß – Länge 1/2/3	Verschleiß – Radius 1/2	Adap/Basismaß – Länge 1/2/3	Freischnidwinkel	– Scheibenradius min	– Scheibenbreite min/akt.	– Drehzahl max	– Umfangsgeschwindigkeit max	– Winkel ischräge Scheibe
Fräswerkzeuge und Bohrwerkzeuge (alle übrigen)																
100	Fräswerkzeug nach CLDATA*1		x	x	x	x		x	x	x						
110	Kugelkopffräser		x	x	x	x		x	x	x						
120	Schaftfräser		x	x	x	x		x	x	x						
121	Schaftfräser mit Eckenverrundung		x	x	x	x	x	x	x	x						
130	Winkelkopffräser		x	x	x	x		x	x	x						
131	Winkelkopffräser mit Eckenverrundung		x	x	x	x	x	x	x	x						
140	Planfräser		x	x	x	x		x	x	x						
145	Gewindefräser		x	x	x	x		x	x	x						
150	Scheibenfräser		x	x	x	x		x	x	x						
155	Kegelstumpffräser		x	x	x	x		x	x	x						
200	Spiralbohrer		x	x	x	x		x		x						
205	Vollbohrer		x	x	x	x		x		x						
210	Bohrstange		x	x	x	x		x		x						
220	Zentrierbohrer		x	x	x	x		x		x						
230	Spitzsenker		x	x	x	x		x		x						
231	Flachsenker		x	x	x	x		x		x						
240	Gewindebohrer Regelgewinde		x	x	x	x		x		x						
241	Gewindebohrer Feingewinde		x	x	x	x		x		x						
242	Gewindebohrer Whitworthgewinde		x	x	x	x		x		x						
250	Reibahle		x	x	x	x		x		x						
Schleifwerkzeuge und Drehwerkzeuge (400 – 599)																
400	Umfangsschleifscheibe		x	x	x	x	x	x	x	x					x	x
401	Umfangsschleifscheibe mit Überwachung		x	x	x	x	x	x	x	x		x	x	x	x	x
403	wie 401 aber ohne Basismaß für SUG*2		x	x	x	x	x	x	x	x		x	x	x		x
410	Planschleibe		x	x	x	x	x	x	x	x						x
411	Planschleibe mit Überwachung		x	x	x	x	x	x	x	x		x	x	x	x	
413	wie 411 aber ohne Basismaß für SUG*2		x	x	x	x	x	x	x	x		x	x	x		
490	Abrichter		x	x	x	x	x	x	x	x						
500	Schruppstahl		x	x	x	x	x	x	x	x	x					
510	Schlichtstahl		x	x	x	x	x	x	x	x	x					
520	Einstechstahl		x	x	x	x	x	x	x	x	x					
530	Abstechstahl		x	x	x	x	x	x	x	x	x					
540	Gewindestahl		x	x	x	x	x	x	x	x	x					
Sonderwerkzeuge (700)																
700	Nutsäge		x	x	x	x	x	x	x	x	x					
CLDATA*1 "cutter Location Data" — Werkzeugpositionsdaten nach DIN 66215, BI 1SUG*2 — Schleifenscheibenumfangsgeschwindigkeit																

**Hinweis**

Innerhalb der Gruppe Drehwerkzeuge hat der Werkzeugtyp keine Bedeutung.

Insbesondere bei den Schleifwerkzeugen (400-499) sind auch nicht aufgeführte Nummern zulässig.

**Werkzeugkorrekturdaten**

Die Werkzeugkorrekturdaten (TOA-Daten) werden in Systemvariablen abgelegt.

Beispiel: Werkzeugtyp Nutsäge (Typ 700)

	Geometrie	Verschleiß	Basis	Einheit
<b>Längenkorrektur</b>				
Länge 1	\$TC_DP3	\$TC_DP12	\$TC_DP21	mm
Länge 2	\$TC_DP4	\$TC_DP13	\$TC_DP22	mm
Länge 3	\$TC_DP5	\$TC_DP14	\$TC_DP23	mm
<b>Sägeblattkorrektur</b>				
Durchmesser d	\$TC_DP6	\$TC_DP15		mm
Nutbreite b	\$TC_DP7	\$TC_DP16		mm
Überstand k	\$TC_DP8	\$TC_DP17		mm



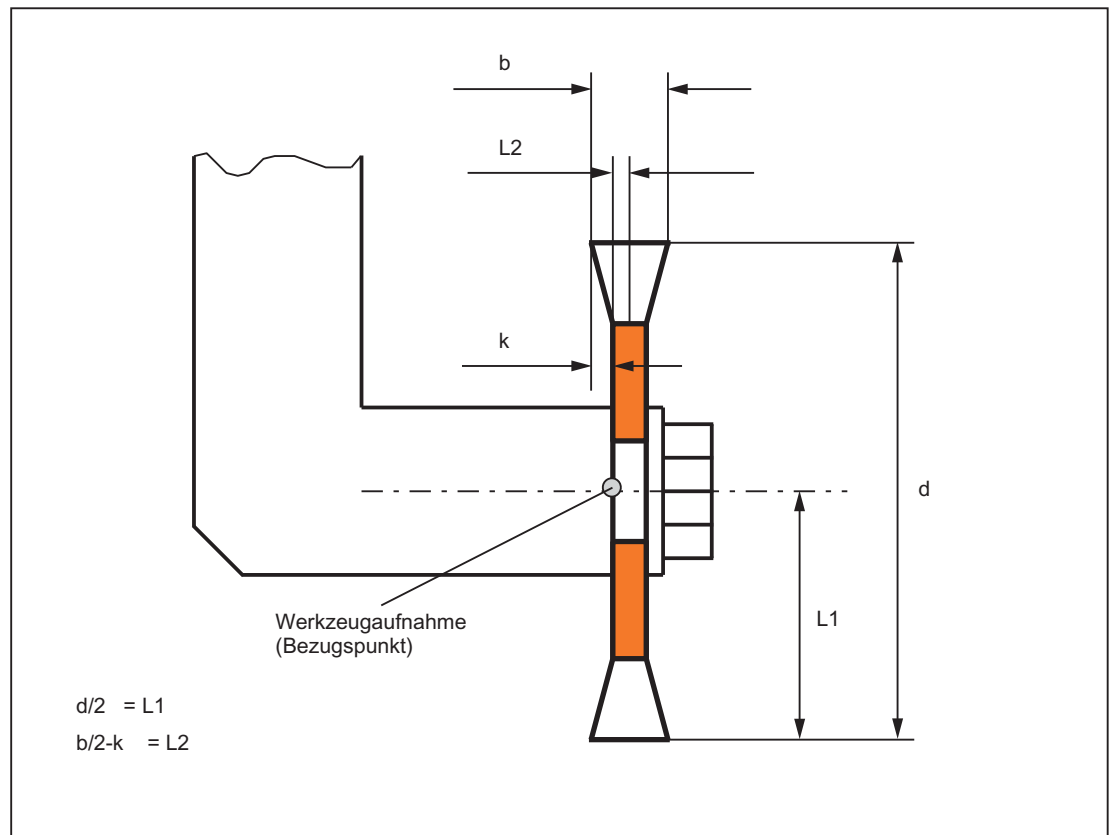


Bild 17-3 Geometrie der Nutsäge (analog zum Winkelkopfräser)

Die Breite des Sägeblattes wird bei der Werkzeugradiuskorrektur ( $G40$  bis  $G42$ ) wie folgt berücksichtigt:

Befehl	Bedeutung
G40	keine Sägeblattkorrektur
G41	Sägeblattkorrektur links
G42	Sägeblattkorrektur rechts

### 17.4.3 WZ-Parameter 2: Schneidenlage

#### Bedeutung

Die Schneidenlage beschreibt die Lage der Werkzeugspitze P in Bezug auf den Schneidenmittelpunkt S. Sie wird in den WZ-Parameter 2 eingetragen.

Die Schneidenlage wird zusammen mit dem Schneidenradius (WZ-Parameter 8) zur Berechnung der Werkzeugradiuskorrektur bei Drehwerkzeugen (Werkzeugtyp 5xx) benötigt.

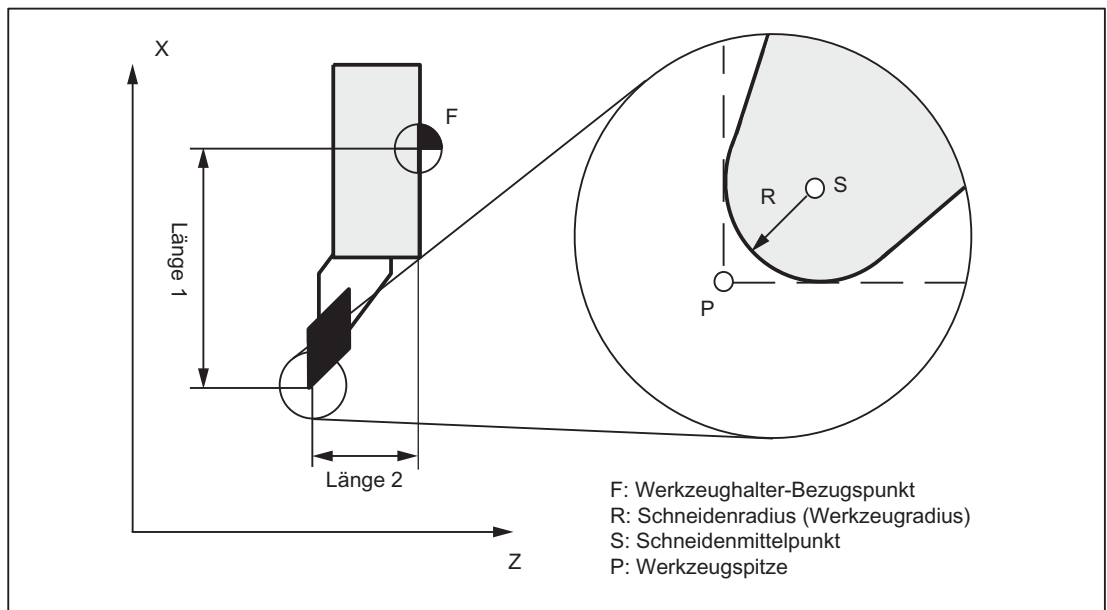


Bild 17-4 Abmessungen bei Drehwerkzeugen: Drehmeisel

#### Schneidenlagen-Parameterwerte

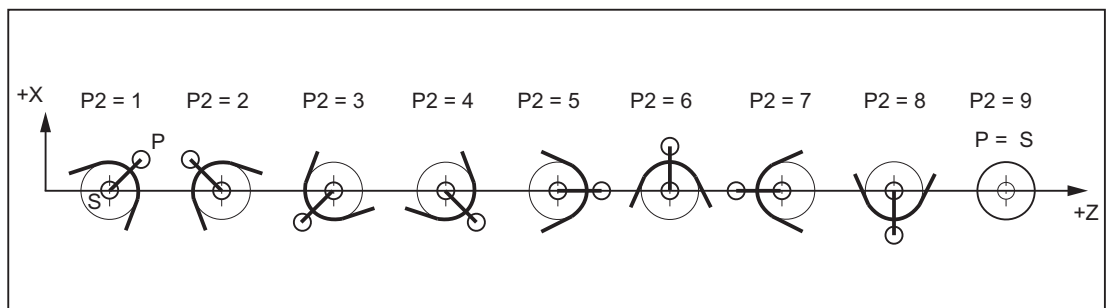


Bild 17-5 Werkzeugparameter 2 (P2): Bei Bearbeitung hinter der Drehmitte

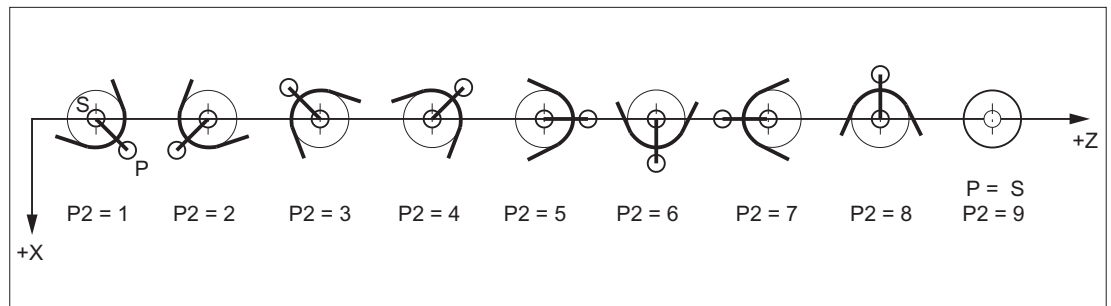


Bild 17-6 Werkzeugparameter 2 (P2): Bei Bearbeitung vor der Drehmitte

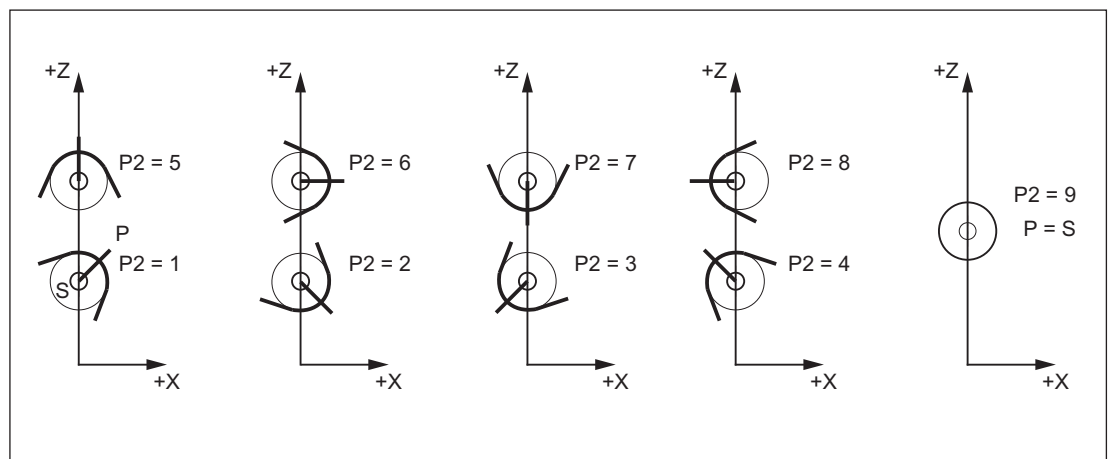


Bild 17-7 Werkzeugparameter 2 (P2): Schneidenlage bei Karusselldrehmaschinen

### Besonderheiten

- Wird zur Ermittlung der WZ-Längenkorrektur als Bezugspunkt nicht der Punkt P, sondern der Schneidenmittelpunkt S benützt, ist als WZ-Schneidenlage die Kennung 9 einzugeben.
- Die Kennung 0 (null) als WZ-Schneidenlage ist nicht zulässig.

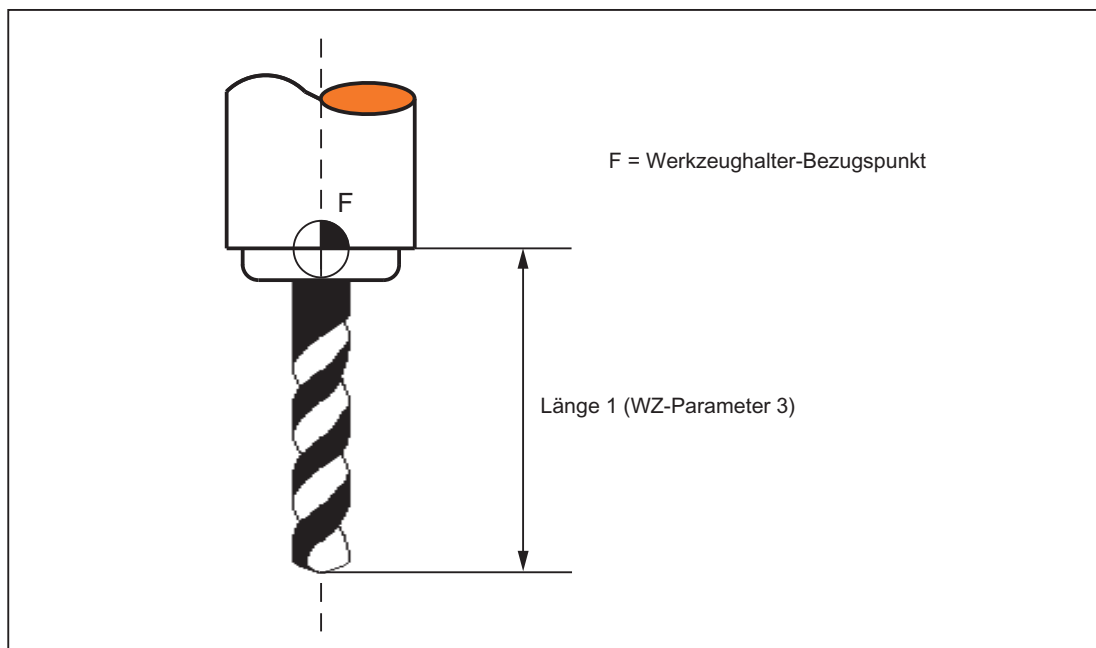
### 17.4.4 WZ-Parameter 3 - 5: Geometrie - Werkzeuglängen

#### Bedeutung

Die Längen der Werkzeuge werden für die Geometrie-Werkzeuglängenkorrektur benötigt. Sie werden als WZ-Länge 1 bis 3 in die WZ-Parameter 3 bis 5 eingegeben. Je nach WZ-Typ sind folgende Längenangaben mindestens erforderlich:

Werkzeugtyp	Erforderliche WZ-Längen
WZ-Typ 12x, 140, 145, 150:	WZ-Länge 1
WZ-Typ 13x:	WZ-Länge 1-3 (je nach Ebene G17-G19)
WZ-Typ 2xx:	WZ-Länge 1
WZ-Typ 5xx:	WZ-Länge 1-3

Beispiel: Spiralbohrer (WZ-Typ 200) mit WZ-Länge 1 (WZ-Parameter 3)



#### Hinweis

Unabhängig vom WZ-Typ werden immer alle drei WZ-Parameter 3 bis 5 (WZ-Länge 1 bis 3) in den drei Geometrieachsen verrechnet.

Wenn für einen WZ-Typ mehr WZ-Längen in die WZ-Parameter 3 bis 5 eingegeben werden als mindestens erforderlich sind, dann werden diese zusätzlichen WZ-Längen in den Geometrieachsen ohne Alarm verrechnet.

## Besonderheit

Erst die Addition der Geometrie-Werkzeuglängenkorrektur (WZ-Parameter 3 bis 5) und der Verschleiß-Werkzeuglängenkorrektur (WZ-Parameter 12 bis 14) legt die aktive Größe des Werkzeugs fest. Zur Verrechnung der gesamten Werkzeuglängenkorrektur in den Geometrieachsen wird auch die Basismaß/Adaptermaß-Werkzeuglängenkorrektur addiert.

## Literatur

Zur Eingabe der Werkzeuggröße (Längen) in die WZ-Parameter 3 bis 5 (WZ-Länge 1 bis 3) und die Verrechnung in den drei Geometrieachsen siehe → Bedienhandbuch.

## 17.4.5 WZ-Parameter 6 - 11: Geometrie - Werkzeugform

### Bedeutung

Mit den WZ-Parametern 6 bis 11 wird die Form des Werkzeugs festgelegt. Die Angaben werden für die Geometrie-Werkzeugradiuskorrektur benötigt.

In den meisten Fällen wird nur der WZ-Parameter 6 (WZ-Radius 1) benutzt:

WZ-Parameter	Bedeutung	Verwendung	
6	WZ-Länge 1	nicht verwendet	
7	WZ-Länge 2	nicht verwendet	
8	WZ-Radius 1	In den WZ-Parameter 6 (WZ-Radius 1) ist der Werkzeugradius für folgende WZ-Typen einzugeben:	
		WZ-Typ 1xx	Fräswerkzeuge
		WZ-Typ 5xx	Drehwerkzeuge
		Für Bohrerwerkzeuge (WZ-Typ 2xx) muss kein WZ-Radius eingegeben werden. Bei Drehwerkzeugen (WZ-Typ 5xx) muss zusätzlich noch die WZ-Schneidenlage (WZ-Parameter 2) eingegeben werden.	
9	WZ-Radius 2	nicht verwendet	
10	WZ-Winkel 1	nicht verwendet	
11	WZ-Winkel 2	nicht verwendet	

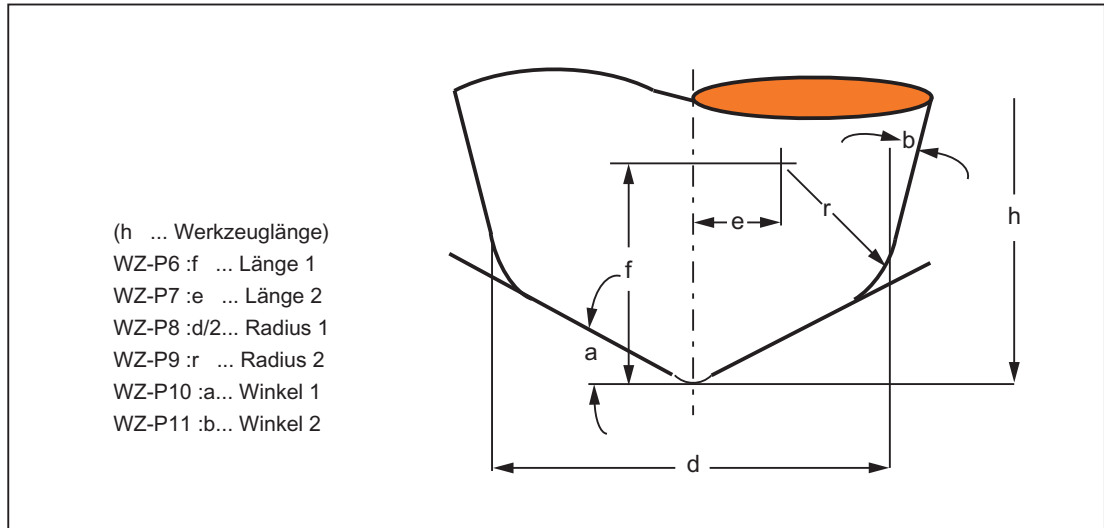
### 2D-WRK mit Konturwerkzeugen

Zur Definition von Konturwerkzeugen mit mehreren Werkzeugschneiden können der minimale und maximale Grenzwinkel angegeben werden. Die beiden Grenzwinkel beziehen sich jeweils auf den Vektor vom Schneidenmittelpunkt zum Schneidenbezugspunkt und werden im Gegenuhrzeigersinn gezählt.

WZ-Winkel 1	Minimaler Grenzwinkel je Werkzeugschneide
WZ-Winkel 2	Maximaler Grenzwinkel je Werkzeugschneide

### 3D-Stirnfräsen

Für die Werkzeugbeschreibung beim 3D-Stirnfräsen sind alle WZ-Parameter 6 bis 11 erforderlich:



### Literatur

Zur Eingabe der Werkzeugform (Radius für die WRK) in die WZ-Parameter 6 bis 11 und die Verrechnung in den drei Geometrieachsen durch die Geometrie-Werkzeugradiuskorrektur siehe:

- Programmierhandbuch Grundlagen; Kapitel: "Werkzeugkorrekturen" > "2½ D-Werkzeugkorrektur"
- Funktionshandbuch Sonderfunktionen; 3D-Werkzeugradiuskorrektur (W5)

Bezüglich 3D-Stirnfräsen siehe:

- Programmierhandbuch Arbeitsvorbereitung; Kapitel: "Transformationen" > "Drei-, Vier- und Fünf-Achs-Transformation (TRAORI)"

## 17.4.6 WZ-Parameter 12 - 14: Verschleiß - Werkzeuglängen

### Bedeutung

Während mit der Geometrie-Werkzeuglängenkorrektur (WZ-Parameter 3 bis 5) die Größe des Werkzeugs eingestellt wird, kann mit der Verschleiß-Werkzeuglängenkorrektur die Veränderung der aktiven Werkzeuggröße korrigiert werden.

Die aktive Werkzeuggröße kann sich ändern durch:

- Unterschiede zwischen der WZ-Aufnahme in der WZ-Messmaschine und der WZ-Aufnahme in der Werkzeugmaschine
- Verschleiß des Werkzeugs durch die Einsatzdauer bei der Bearbeitung
- Definition eines Schlichtaufmaßes

### Aktive WZ-Größe

Die Geometrie-Werkzeuglängenkorrektur (WZ-Parameter 3 bis 5) und die Verschleiß-Werkzeuglängenkorrektur (WZ-Parameter 12 bis 14) werden addiert (Geometrie-WZ-Länge 1 mit Verschleiß-WZ-Länge 1, etc.) und stellen so die Größe des aktiven Werkzeugs dar.

## 17.4.7 WZ-Parameter 15 - 20: Verschleiß - Werkzeugform

### Bedeutung

Während mit der Geometrie-Werkzeugradiuskorrektur (WZ-Parameter 6 bis 11) die Form des Werkzeugs eingestellt wird, kann mit der Verschleiß-Werkzeugradiuskorrektur die Veränderung der aktiven Werkzeugform korrigiert werden.

Die aktive Werkzeuggröße kann sich ändern durch:

- Verschleiß des Werkzeugs durch die Einsatzdauer bei der Bearbeitung
- Definition eines Schlichtaufmaßes

### Aktive WZ-Form

Die Geometrie-Werkzeugradiuskorrektur (WZ-Parameter 6 bis 11) und die Verschleiß-Werkzeugradiuskorrektur (WZ-Parameter 15 bis 20) werden addiert (Geometrie-WZ-Radius 1 mit Verschleiß-WZ-Radius 1) und stellen so die Form des aktiven Werkzeugs dar.

### 17.4.8 WZ-Parameter 21 - 23: Basismaß / Adaptermaß

#### Bedeutung

Das Basismaß/Adaptermaß kann verwendet werden, wenn der Bezugspunkt des Werkzeughalters (WZ-Größe) nicht mit dem Bezugspunkt des Werkzeugträgers übereinstimmt.

Das ist der Fall, wenn:

- WZ und WZ-Adapter getrennt vermessen, aber schon zusammengebaut in die Maschine eingesetzt werden (WZ-Größe und Adaptergröße werden getrennt in eine WZ-Schneide eingegeben).
- das WZ in eine zweite Werkzeugaufnahme eingesetzt wird, die sich an einer anderen Position befindet (z. B: Vertikal - und Horizontalspindel).
- die Werkzeugaufnahmen eines Werkzeugrevolvers unterschiedliche Positionen besitzen.

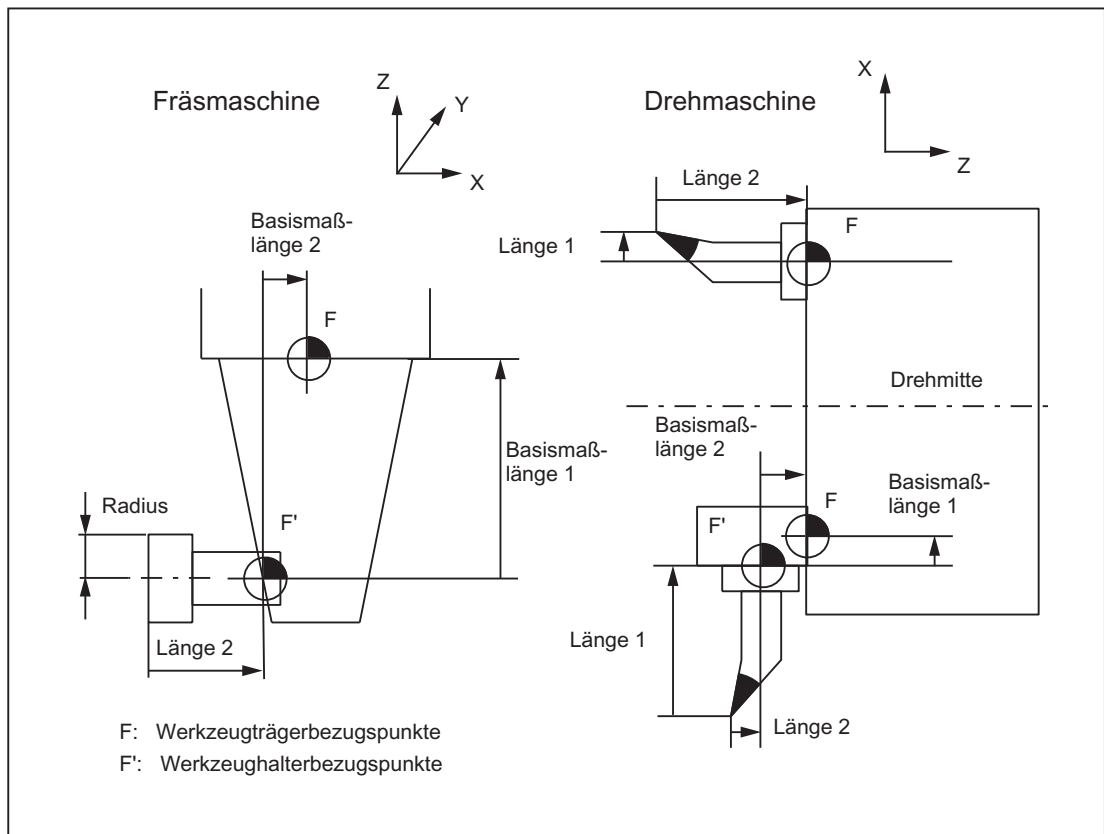


Bild 17-8 Anwendungsbeispiele für Basismaß / Adaptermaß-WLK



### Basis-Länge 1 bis 3 (WZ-Parameter 21 bis 23)

Damit der Werkzeughalterbezugspunkt F zum Werkzeugträgerbezugspunkt F' in den drei Geometrieachsen (3-dimensional) korrigiert werden kann, wirken unabhängig von WZ-Typ immer alle 3 Basis-Längen. D. h. auch ein Spiralbohrer (WZ-Typ 200) mit einer Werkzeuglängenkorrektur (Länge 1) kann ein Basismaß/Adaptermaß in 3 Achsen haben.

### Literatur

Zur genaueren Beschreibung der Basismaß/Adaptermaß-Werkzeuglängenkorrektur siehe:

- Programmierhandbuch Grundlagen

### 17.4.9 WZ-Parameter 24: Freischneidwinkel

#### Bedeutung

Bestimmte Drehzyklen, in denen Verfahrbewegungen mit Hinterschneiden erzeugt werden, überwachen den Freischneidwinkel des aktiven Werkzeugs auf eine mögliche Konturverletzung.

#### Wertebereich

Der Winkel (0 bis 90° ohne Vorzeichen) wird als Freischneidwinkel in den WZ-Parameter 24 eingegeben.

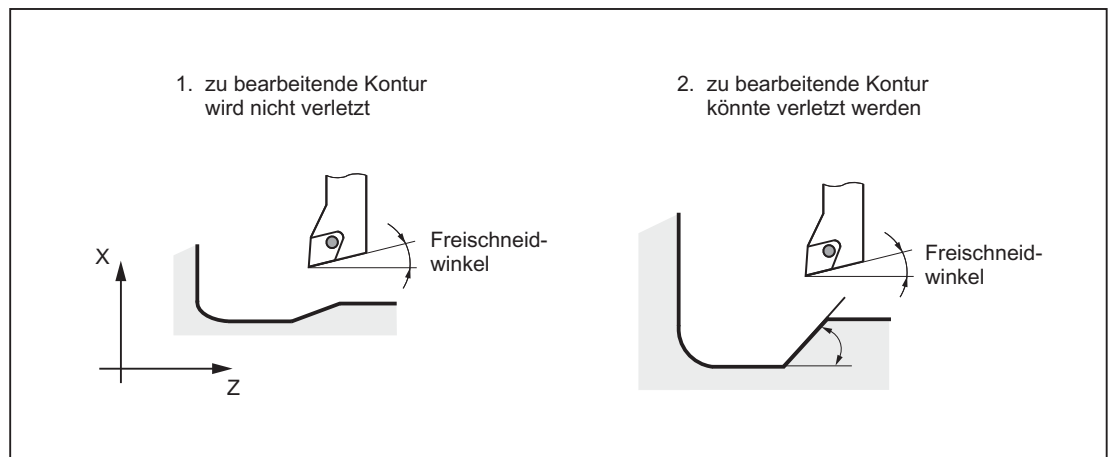


Bild 17-9 Freischneidwinkel des Drehwerkzeugs beim Hinterschneiden

### Bearbeitungsart Längs oder Plan

Der Freischneidwinkel ist abhängig von der Bearbeitungsart Längs oder Plan unterschiedlich einzugeben. Soll ein Werkzeug für Längs- und Planbearbeitung eingesetzt werden, müssen bei unterschiedlichen Freischneidwinkeln zwei WZ-Schneiden definiert werden.

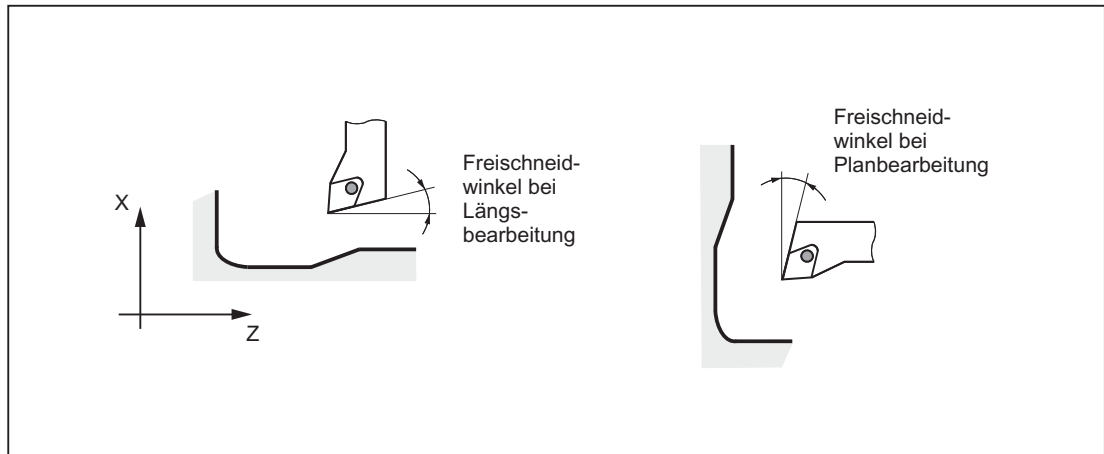


Bild 17-10 Freischneidwinkel für Längs- und Planbearbeitung

### Hinweis

Wird der Freischneidwinkel (WZ-Parameter 24) mit Null eingegeben, erfolgt in den Drehzyklen keine Überwachung beim Hinterschneiden.

### Literatur

Zur genaueren Beschreibung des Freischneidwinkels siehe:

- Programmierhandbuch Zyklen

## 17.4.10 Werkzeuge mit relevanter Schneidenlage

Für Werkzeuge mit relevanter Schneidenlage ist Folgendes zu beachten:

- Für die Berechnung von Schnittpunkten mit dem An- bzw. Abfahrtsatz wird die Gerade zwischen den Schneidenmittelpunkten am Satzanfang und am Satzende verwendet. Die Differenz zwischen Schneidenbezugspunkt und Schneidenmittelpunkt wird dieser Bewegung überlagert.  
Beim An- bzw. Abfahren mit `KONT` erfolgt die Überlagerung im linearen Teilsatz der An- bzw. Abfahrbewegung. Die geometrischen Verhältnisse sind deshalb bei Werkzeugen mit und ohne relevante Schneidenlage identisch.
- Der Wechsel eines Werkzeugs bei aktiver Werkzeugradiuskorrektur, bei dem sich der Abstand zwischen Schneidenmittelpunkt und Schneidenbezugspunkt ändert, ist in Kreissätzen und in Verfahrsätzen mit rationalen Polynomen mit einem Nennergrad  $> 4$  verboten. Bei anderen Interpolationsarten ist ein Wechsel im Gegensatz zum bisherigen Zustand auch bei aktiver Transformation (z. B. `TRANSMIT`) zulässig.
- Bei der Werkzeugradiuskorrektur mit variabler Werkzeugorientierung ist die Transformation vom Schneidenbezugspunkt auf den Schneidenmittelpunkt nicht durch eine einfache Nullpunktverschiebung realisierbar. Werkzeuge mit relevanter Schneidenlage werden deshalb beim 3D-Umfangfräsen verboten (Alarm).

---

### Hinweis

Für das Stirnfräsen ist das Thema nicht relevant, da hier ohnehin nur definierte Werkzeugtypen ohne relevante Schneidenlage zugelassen sind. (Werkzeuge mit einem nicht ausdrücklich zugelassenen Werkzeugtyp werden als Kugelkopffräser mit dem angegebenen Radius behandelt. Die Angabe einer Schneidenlage wird ignoriert.)

---

## 17.5 Werkzeugradiuskorrektur-2D (WRK)

### 17.5.1 Allgemeines

---

#### Hinweis

Zur Werkzeugradiuskorrektur (WRK) siehe:

#### Literatur:

Programmierhandbuch Grundlagen

Nur die Programmieranleitung beschreibt die vollständigen programmtechnischen Möglichkeiten der Werkzeugradiuskorrektur (WRK) mit ihren Sonderfällen.

---

#### Warum WRK?

Die im Teileprogramm programmierte Kontur (Geometrie) des Werkstücks soll unabhängig von den in der Fertigung eingesetzten Werkzeugen sein. Dazu ist es nötig, die Werte für Werkzeuglänge und Werkzeugradius einem aktuellen Korrekturspeicher zu entnehmen. Durch die WRK kann mit dem aktuellen Werkzeugradius die Äquidistante zur programmierten Kontur in der Ebene bestimmt werden.

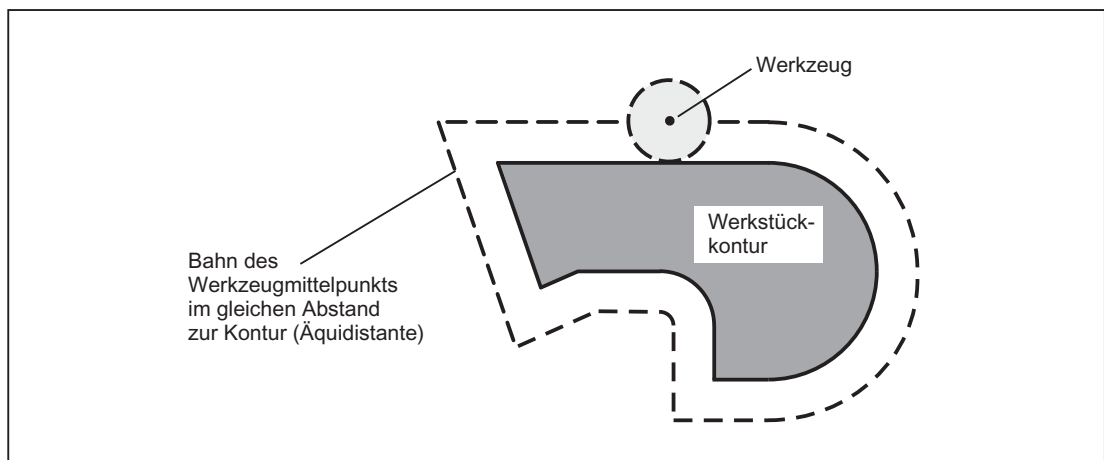


Bild 17-11 Werkstückkontur (Geometrie) mit Äquidistante

## WRK in der Ebene

Die WRK wirkt in der aktuellen Ebene (G17 bis G19) bei folgenden Interpolationsarten:

• Linearinterpolation	...	G0, G1
• Kreisinterpolation	...	G2, G3, CIP
• Helikalinterpolation	...	G2, G3
• Splineinterpolation	...	ASPLINE, BSPLINE, CSPLINE
• Polynominterpolation	...	POLY

## 17.5.2 Anwahl der WRK (G41/G42)

### Korrekturrichtung

Die WRK ermittelt eine äquidistante Bahn zur programmierten Kontur. Die Korrektur kann in Bewegungsrichtung links oder rechts von der programmierten Kontur erfolgen.

Befehl	Bedeutung
G41	WRK links von der Kontur in Bewegungsrichtung
G42	WRK rechts von der Kontur in Bewegungsrichtung
G40	Abwahl der WRK

### Zwischensätze

Während der aktiven WRK werden in der Regel nur Programmsätze mit Positionen von Geometrieachsen in der aktuellen Ebene programmiert. Trotzdem können bei aktiver WRK auch einzelne Zwischensätze programmiert werden. Zwischensätze sind Programmsätze, die nicht mindestens die Position einer Geometrieachse in der aktuellen Ebene enthalten:

- Positionen in der Zustellachse
- Hilfsfunktionen
- etc.

Die max. Anzahl der Zwischensätze kann vorgegeben werden im Maschinendatum:

MD20250 \$MC\_CUTCOM\_MAXNUM\_DUMMY\_BLOCKS (Satzanzahl ohne Verfahrbewegung bei WRK)

**Besonderheiten**

- Die Anwahl der WRK kann nur in einem Programmsatz mit G0 (Eilgang) oder G1 (Linearinterpolation) erfolgen.
- Spätestens im Programmsatz mit der WRK-Anwahl muss ein Werkzeug eingewechselt (T-Funktion) und die WZ-Schneide/WZK (D1 bis D9) aktiviert werden.
- Bei einer WZ-Schneide/WZK von D0 erfolgt keine WRK-Anwahl.
- Wird bei der WRK-Anwahl nur eine Geometrieachse der Ebene programmiert, wird die zweite Achse der Ebene automatisch ergänzt (letzte progr. Position).
- Wird im Satz mit der WRK-Anwahl keine Geometrieachse der aktuellen Ebene programmiert, findet keine Anwahl statt.
- Wird im Satz nach der WRK-Anwahl die WRK abgewählt (G40), findet keine Anwahl statt.
- Bei Anwahl der WRK wird das Anfahrverhalten durch die Anweisungen NORM/KONT bestimmt.

**17.5.3 An- und Abfahrverhalten (NORM/KONT/KONTC/KONTT)**

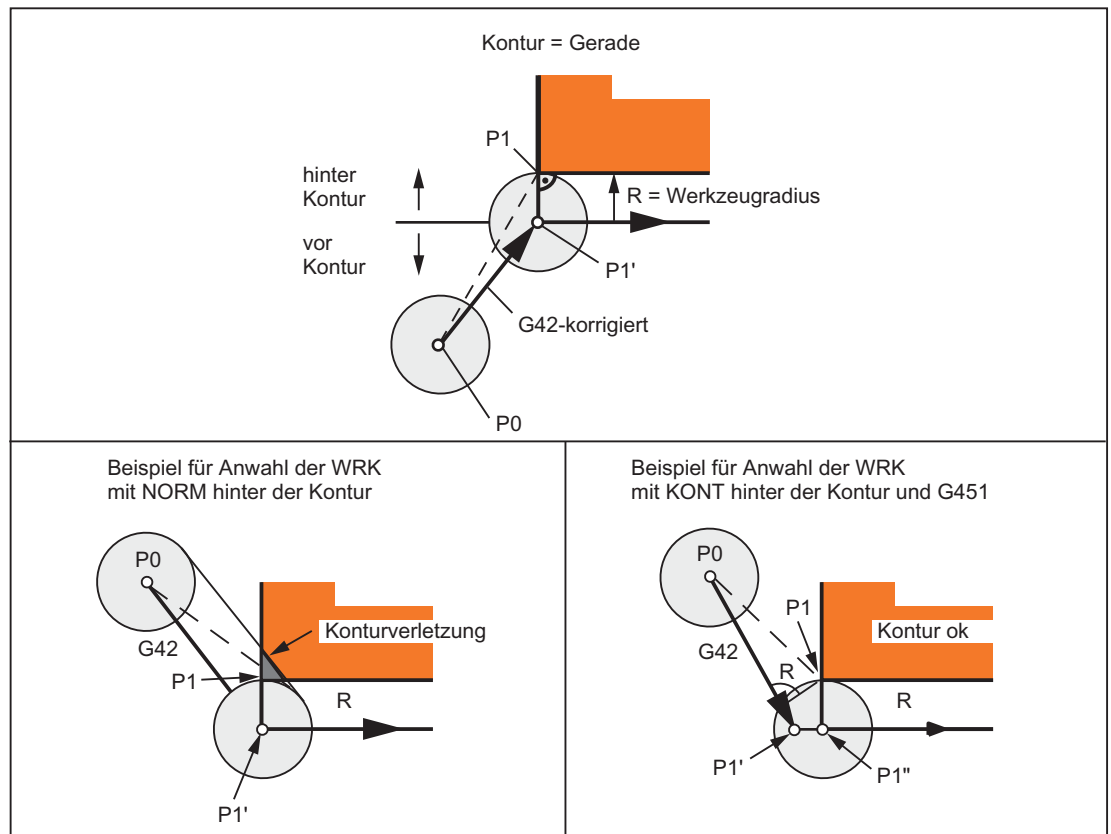
**NORM und KONT**

Mit den Anweisungen NORM und KONT kann das Anfahrverhalten (Anwahl der WRK mit G41/G42) und das Abfahrverhalten (Abwahl der WRK mit G40) gesteuert werden:

Befehl	Bedeutung
NORM	Normaleinstellung im Anfangspunkt/Endpunkt (Grundstellung)
KONT	Kontur umfahren im Anfangspunkt/Endpunkt
KONTC	krümmungsstetiges An-/Abfahren
KONTT	tangentenstetiges An-/Abfahren

## Besonderheiten

- `KONT` unterscheidet sich von `NORM` nur bei einer WZ-Startposition, die hinter der Kontur liegt.

Bild 17-12 Beispiel für Anwahl der WRK mit `KONT` oder `NORM` vor und hinter der Kontur

- `KONT` und `G450/G451` (Eckenverhalten an Außenecken) wirkt gemeinsam und bestimmt so das An- Abfahrverhalten bei WRK.
- Bei Abwahl der WRK wird das Abfahrverhalten durch die Anweisungen `NORM/KONT` bestimmt.

## Randbedingungen

Bei den beiden folgenden Varianten sind die An- bzw. Abfahrsätze Polynome. Sie stehen deshalb nur bei Steuerungsvarianten zur Verfügung, bei denen die Polynominterpolation freigegeben ist:

- KONTT

Mit KONTT wird tangential an die Kontur angefahren bzw. tangential von der Kontur abgefahren. Die Krümmungen am Satzübergang sind im Allgemeinen nicht stetig.

- KONTC

Mit KONTC ist beim Übergang vom Anfahrsatz bzw. zum Abfahrsatz nicht nur die Tangente stetig, sondern auch die Krümmung, so dass in diesem Fall bei der Aktivierung / Deaktivierung **kein Beschleunigungssprung** mehr auftreten kann.

Obwohl KONTC die Eigenschaft KONTT mit umfasst, ist die tangentialstetige Variante KONTT allein verfügbar, weil durch die geforderte Krümmungstetigkeit bei KONTC mitunter unerwünschte Konturen entstehen können.

## Achsen

Die Stetigkeitsbedingung wird in allen **drei** Achsen eingehalten. Es ist somit zulässig, beim An/Abfahren gleichzeitig eine Wegkomponente senkrecht zur Korrektorebene zu programmieren.

Für die originalen An- bzw. Abfahrsätze mit KONTT/KONTC sind nur **Linearsätze** zugelassen. Diese programmierten Linearsätze werden in der Steuerung durch die entsprechenden Polynomkurven ersetzt.

## Ausschluss

KONTT und KONTC stehen bei den 3D-Varianten der Werkzeugradiuskorrektur (CUT3DC, CUT3DCC, CUT3DF) nicht zur Verfügung.

Werden sie dennoch programmiert, wird steuerungsintern ohne Fehlermeldung auf NORM umgeschaltet.



## Beispiel KONTC

In den beiden folgenden Bildern ist eine typische Anwendung für das krümmungsstetige An- und Abfahren dargestellt:

In der Kreismitte beginnend wird an den Vollkreis angefahren. Dabei sind im Satzendpunkt des Anfahrsatzes dessen Richtung und dessen Krümmungsradius gleich den Werten des folgenden Kreises. In den beiden An-/Abfahrätzen wird gleichzeitig in Z-Richtung zugestellt.

Das zugehörige NC-Programmsegment sieht folgendermaßen aus:

```
$TC_DP1 [1,1]=121 ; Fräser
$TC_DP6 [1,1]=10 ; Radius 10 mm
N10 G1 X0 Y0 Z60 G64 T1 D1 F10000
N20 G41 KONTC X70 Y0 Z0
N30 G2 I-70 ; Vollkreis
N40 G40 G1 X0 Y0 Z60
N50 M30
```

### Erklärung:

In diesem Beispiel wird ein Vollkreis mit dem Radius 70 mm in der X-Y-Ebene bearbeitet. Da das Werkzeug einen Radius von 10 mm hat, ergibt sich als resultierende Werkzeugmittelpunktsbahn ein Kreis mit dem Radius 60 mm. Start-/ Endpunkt liegen bei X0 Y0 Z60, so dass gleichzeitig zur An-/Abfahrbewegung in der Korrektorebene eine Bewegung in Z-Richtung ausgeführt wird.

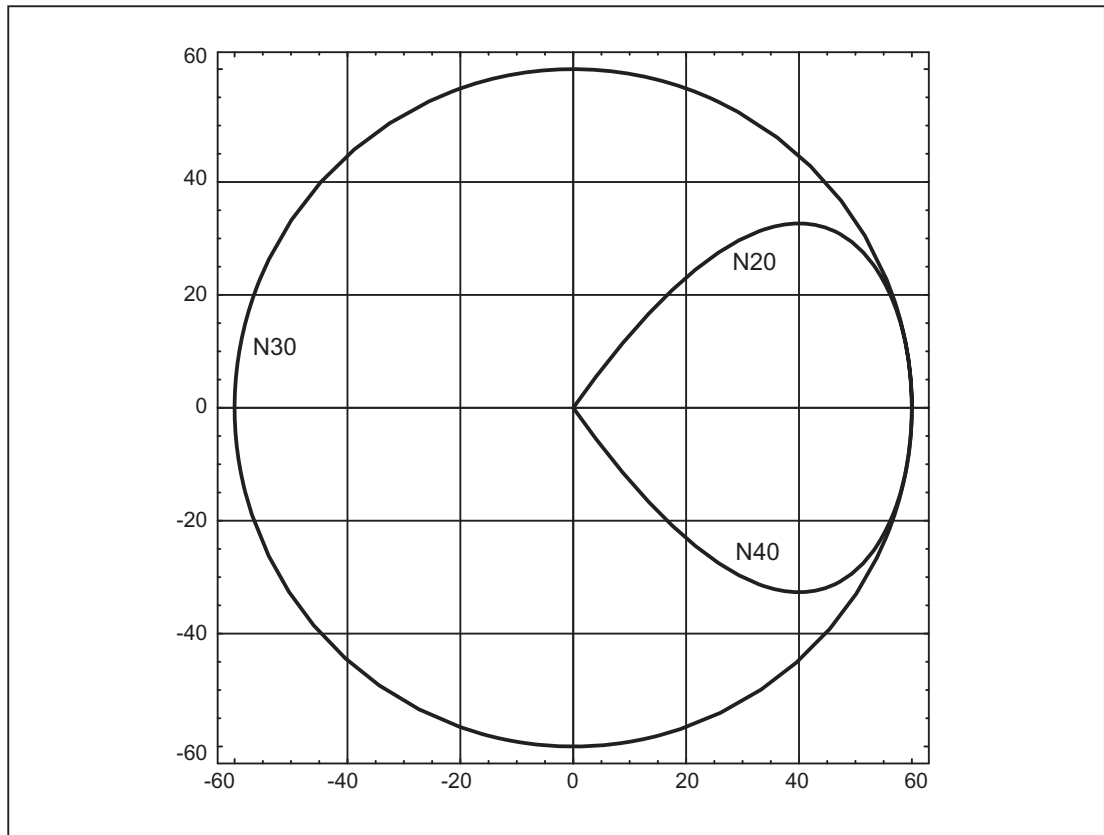


Bild 17-13 Krümmungsstetiges An- und Abfahren bei der Innenbearbeitung eines Vollkreises:  
Projektion in die X-Y-Ebene

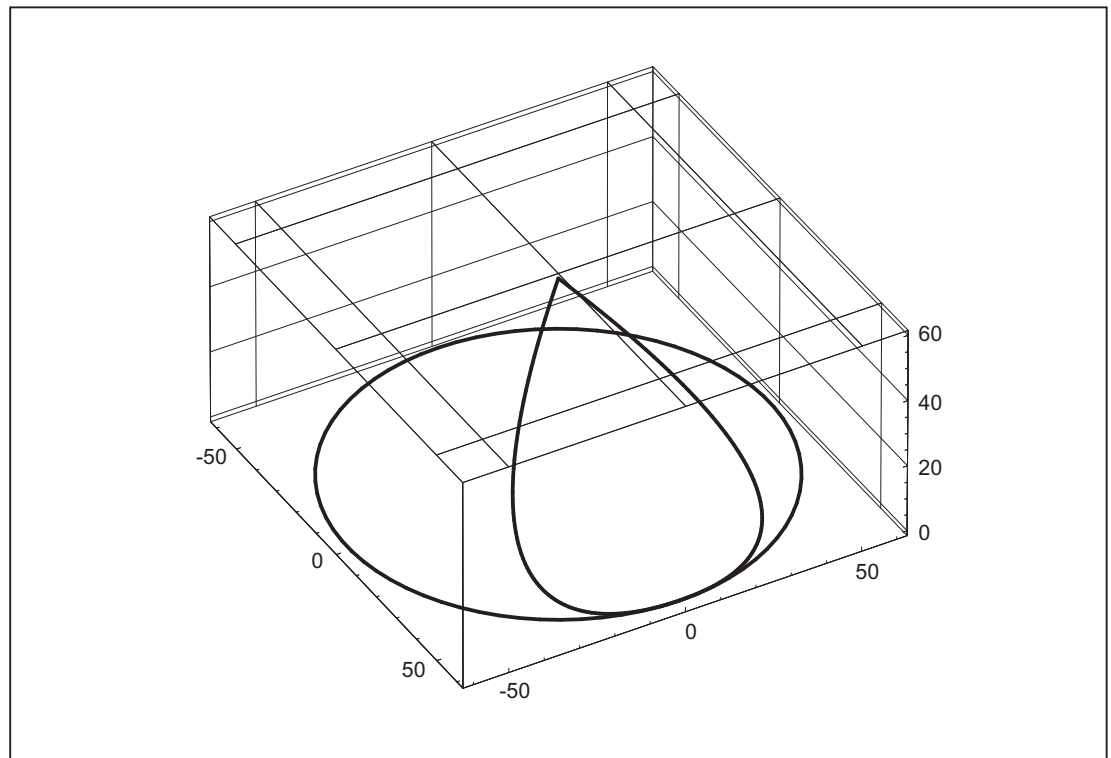


Bild 17-14 Krümmungsstetiges An- und Abfahren bei der Innenbearbeitung eines Vollkreises:  
Räumliche Darstellung

### KONTT und KONTC im Vergleich

Im folgenden Bild ist das unterschiedliche An-/Abfahrverhalten bei  $KONTT$  und  $KONTC$  dargestellt. Ein Kreis mit dem Radius 20 mm um den Mittelpunkt bei X0 Y-40 wird mit einem Werkzeug mit 20 mm Radius an der Außenseite korrigiert. Es ergibt sich deshalb eine kreisförmige Bewegung des Werkzeugmittelpunkts mit dem Radius 40 mm. Der Endpunkt des Abfahrtsatzes liegt bei X40 Y30. Der Übergang zwischen dem Kreissatz und dem Abfahrtsatz liegt im Nullpunkt. Wegen der verlangten Krümmungsstetigkeit bei  $KONTC$  führt der Abfahrtsatz zunächst eine Bewegung mit negativer Y-Komponente aus. Dies wird häufig unerwünscht sein. Der Abfahrtsatz mit  $KONTT$  zeigt dieses Verhalten nicht. Allerdings tritt in diesem Fall am Satzübergang ein Beschleunigungssprung auf.

Ist der  $KONTT$ - bzw.  $KONTC$ -Satz nicht der Ab- sondern der Anfahrtsatz, ergibt sich exakt die gleiche Kontur, die lediglich in umgekehrter Richtung durchlaufen wird, d.h. **An- und Abfahrverhalten sind symmetrisch.**

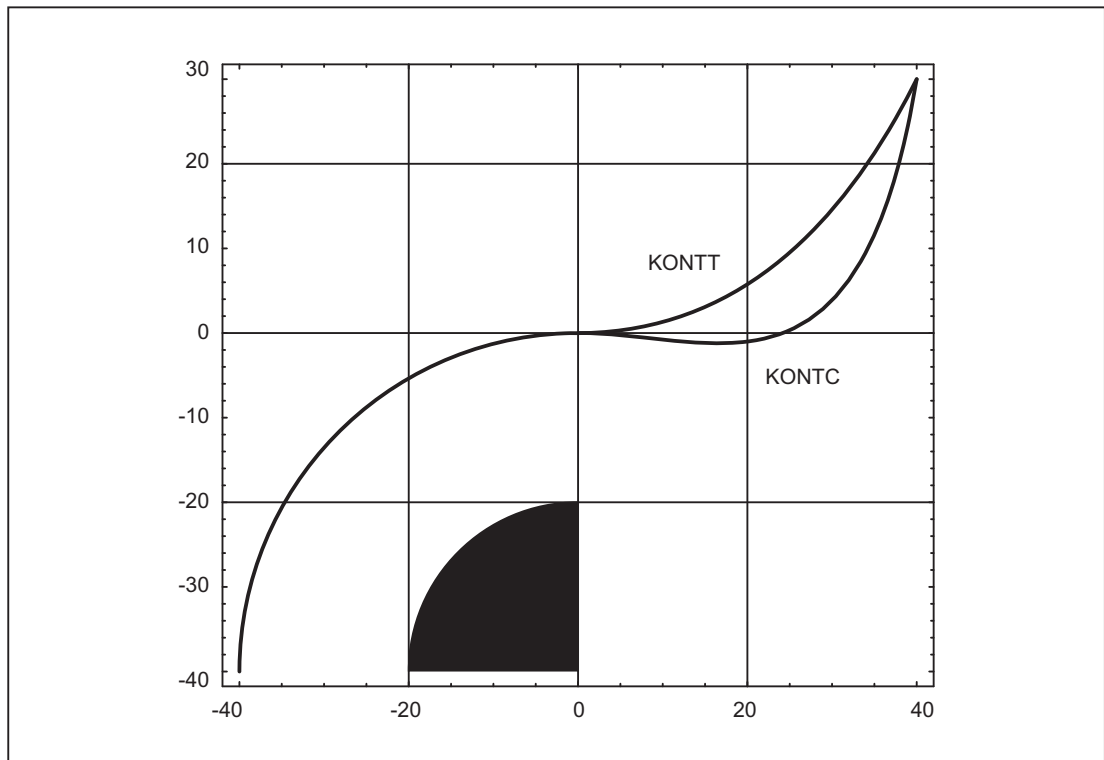


Bild 17-15 Unterschiede von KONTT und KONTC

**Hinweis**

Aus dem Bild ist ersichtlich, dass eine an den Viertelkreis der Kontur anschließende Gerade z. B. bis X20 Y-20 bei Ab-/Anfahren an X0, Y0 mit KONTC eine Konturverletzung zur Folge hätte.

**17.5.4 Weiches An- und Abfahren**

**17.5.4.1 Funktion**

**Bedeutung**

Die Funktion "Weiches An- und Abfahren (WAB)" dient dazu, im Startpunkt einer Kontur unabhängig von der Lage des Ausgangspunktes tangential anzufahren.

Das Anfahrverhalten kann dabei durch eine Reihe von ergänzenden Parametern variiert und speziellen Erfordernissen angepasst werden.

Die beiden Funktionen "Weiches Anfahren" und "Weiches Abfahren" sind weitgehend symmetrisch zueinander. Im Folgenden wird deshalb nur das Anfahren ausführlich beschrieben und auf Abweichungen beim Abfahren hingewiesen.

## Teilbewegungen

Es gibt beim weichen An- und Abfahren maximal 4 Teilbewegungen mit folgenden Positionen:

- Startpunkt der Bewegung  $P_0$
- Zwischenpunkte  $P_1$ ,  $P_2$ , und  $P_3$
- Endpunkt  $P_4$

Die Punkte  $P_0$ ,  $P_3$  und  $P_4$  sind immer definiert. Die Zwischenpunkte  $P_1$  und  $P_2$  können je nach Parametrierung und den geometrischen Verhältnissen entfallen.

Beim Abfahren werden die Punkte in der umgekehrten Reihenfolge, d. h. beginnend mit  $P_4$  und endend mit  $P_0$  durchlaufen.

### 17.5.4.2 Parameter

Das Verhalten der Funktion weiches An- und Abfahren wird durch maximal 9 Parameter bestimmt.

### Satzweise wirksamer G-Code zur Bestimmung der An- bzw. Abfahrkontur

Dieser G-Code kann nicht entfallen.

- G147: Anfahren mit einer Geraden
- G148: Abfahren mit einer Geraden
- G247: Anfahren mit einem Viertelkreis
- G248: Abfahren mit einem Viertelkreis
- G347: Anfahren mit einem Halbkreis
- G348: Abfahren mit einem Halbkreis

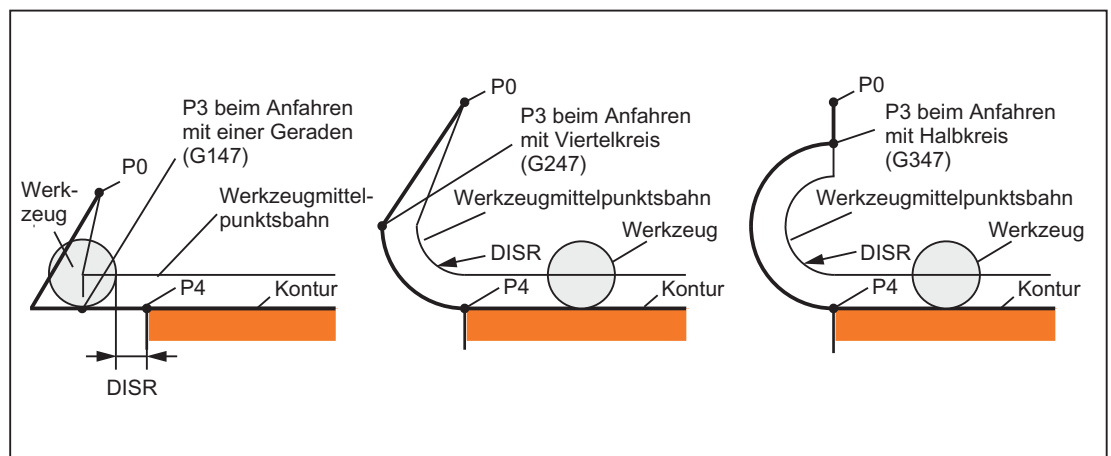


Bild 17-16 Anfahrverhalten in Abhängigkeit von G147 bis G347 und DISR (bei gleichzeitiger Aktivierung der Werkzeugradiuskorrektur)

### Modaler G-Code zur Bestimmung der An- bzw. Abfahrriichtung

Dieser G-Code ist nur dann von Bedeutung, wenn die Anfahrkontur ein Viertel- oder ein Halbkreis ist. Die An- bzw. Abfahrriichtung kann wie folgt bestimmt werden:

- G140:

Bestimmung der An- bzw. Abfahrriichtung mit Hilfe der aktiven Werkzeugradiuskorrektur. (G140 ist Grundstellungswert.)

Bei positivem Werkzeugradius:

- G41 aktiv → anfahren von links
- G42 aktiv → anfahren von rechts

Ist keine Werkzeugradiuskorrektur aktiv (G40), ist das Verhalten wie bei G143. Ein Alarm wird in diesem Fall nicht ausgegeben. Ist der Radius des aktiven Werkzeugs 0, so wird die Anfahr- bzw. Abfahrseite so bestimmt, als wäre der Werkzeugradius positiv.

- G141:

Anfahren von links an die Kontur bzw. nach links von der Kontur wegfahren.

- G142:

Anfahren von rechts an die Kontur bzw. nach rechts von der Kontur wegfahren.

- G143:

Automatische Bestimmung der Anfahrriichtung, d. h. es wird von der Seite aus an die Kontur angefahren, auf der der Startpunkt relativ zur Tangente im Startpunkt des Folgesatzes (P<sub>4</sub>) liegt.

---

#### Hinweis

Beim **Abfahren** wird entsprechend die Tangente im Endpunkt des Vorgängersatzes verwendet. Ist beim Abfahren der Endpunkt nicht programmiert, d. h. er soll implizit bestimmt werden, so ist G143 beim Abfahren nicht zulässig, da zwischen Anfahrseite und Lage des Endpunktes eine wechselseitige Abhängigkeit besteht. Wird G143 in diesem Fall dennoch programmiert, wird ein Alarm ausgegeben. Gleiches gilt, wenn bei aktivem G140 wegen nicht aktiver Werkzeugradiuskorrektur automatisch auf das G143-Verhalten umgeschaltet wird.

---

Modaler G-Code (G340, G341), der die Aufteilung der Bewegung vom Start- zum Endpunkt auf die einzelnen Sätze definiert

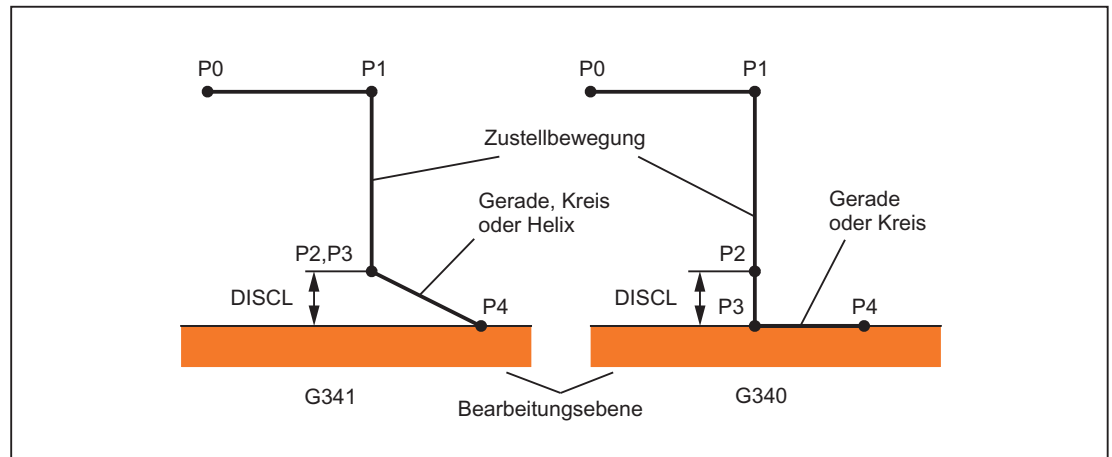


Bild 17-17 Ablauf der Anfahrbewegung abhängig von G340/G341

- G340:** Das charakteristische Anfahren von  $P_0$  bis  $P_4$  ist im Bild dargestellt. Ist  $G_{247}$  oder  $G_{347}$  aktiv (Viertel- oder Halbkreis) und der Startpunkt  $P_3$  liegt nicht in der durch den Endpunkt  $P_4$  definierten Bearbeitungsebene, wird statt eines Kreises eine Helix eingefügt. Der Punkt  $P_2$  ist nicht definiert bzw. er fällt mit  $P_3$  zusammen. Die Kreisebene bzw. die Helixachse wird dabei durch die im WAB-Satz aktive Ebene ( $G_{17} - G_{19}$ ) bestimmt, d. h. vom Folgesatz wird nicht die Starttangente selbst zur Bestimmung des Kreises verwendet, sondern deren Projektion in die aktive Ebene. Die Bewegung vom Punkt  $P_0$  zum Punkt  $P_3$  erfolgt in zwei Geraden mit der Geschwindigkeit, die vor dem WAB-Satz wirksam war.
- G341:** Das charakteristische Anfahren von  $P_0$  bis  $P_4$  ist im Bild dargestellt.  $P_3$  und  $P_4$  liegen in der Bearbeitungsebene, so dass bei  $G_{247}$  bzw.  $G_{347}$  nie eine Helix, sondern immer ein Kreis eingefügt wird.

### Hinweis

In allen Fällen, in denen die Lage der aktiven Ebene  $G_{17} - G_{19}$  eingeht (Kreisebene, Helixachse, Zustellbewegungen senkrecht zur aktiven Ebene), wird ein eventuell aktiver drehender Frame berücksichtigt.

## DISR

DISR gibt die Länge der Anfahrgeraden bzw. den Radius bei Anfahrkreisen an.

### An-/Abfahren mit Geraden

Bei An-/Abfahren mit Geraden gibt DISR den Abstand der Fräserkante vom Startpunkt der Kontur an, d. h. die Länge der Geraden ergibt sich bei aktiver WRK als Summe von Werkzeugradius und programmierten Wert von DISR.

Ein Alarm wird ausgegeben:

- wenn DISR negativ ist und der Betrag größer ist als der Werkzeugradius (die Länge der resultierenden Anfahrgeraden ist kleiner oder gleich null).

### An-/Abfahren mit Kreisen

Bei An-/Abfahren mit Kreisen gibt DISR immer den Radius der Werkzeugmittelpunktsbahn an. Ist Werkzeugradiuskorrektur aktiviert, wird intern ein Kreis mit einem solchen Radius erzeugt, dass auch in diesem Fall die Werkzeugmittelpunktsbahn aus dem programmierten Radius resultiert.

Ein Alarm wird ausgegeben beim An- bzw. Abfahren mit Kreisen:

- wenn der Radius des intern erzeugten Kreises Null oder negativ ist.
- wenn DISR nicht programmiert ist.
- wenn der Radius einen Wert  $\leq 0$  hat.

## DISCL

DISCL gibt den Abstand des Punktes P2 von der Bearbeitungsebene an.

Soll die Position des Punktes P2 auf der Achse senkrecht zur Kreisebene absolut angegeben werden, ist der Wert in der Form  $DISCL = AC(\dots)$  zu programmieren.

Ist DISCL nicht programmiert, so sind die Punkte P1, P2 und P3 bei G340 identisch und die Anfahrkontur wird von P1 zu P4 gebildet.

Es wird überwacht, dass der durch DISCL definierte Punkt zwischen P1 und P3 liegt, d. h. bei allen Bewegungen, die eine Komponente senkrecht zur Bearbeitungsebene haben (Zustellbewegungen, Anfahrbewegung von P3 nach P4), muss diese Komponente das gleiche Vorzeichen haben. Eine Richtungsumkehr ist nicht zugelassen. Gegebenenfalls wird ein Alarm ausgegeben.

Bei der Erkennung der Richtungsumkehr wird eine Toleranz zugelassen, definiert durch das Maschinendatum:

MD20204 \$MC\_WAB\_CLEARANCE\_TOLERANCE (Richtungsumkehr bei WAB)

Wenn P2 außerhalb des durch P1 und P3 definierten Bereichs liegt, die Abweichung aber kleiner oder gleich der parametrisierten Toleranz ist, dann wird angenommen, dass P2 in der durch P1 bzw. P3 definierten Ebene liegt.



**Beispiel:**

Es wird bei G17 ausgehend von der Position Z=20 des Punktes P<sub>1</sub> angefahren. Die durch P<sub>3</sub> definierte WAB-Ebene ist bei Z=0. Der durch DISCL definierte Punkt muss deshalb zwischen diesen beiden Punkten liegen. MD20204=0.010. Liegt P<sub>2</sub> zwischen 20.000 und 20.010 bzw. zwischen 0 und -0.010, so wird angenommen, es sei der Wert 20.0 bzw. 0.0 programmiert. Der Alarm wird ausgegeben, wenn die Z-Position von P<sub>2</sub> größer 20.010 oder kleiner -0.010 ist.

Abhängig von der relativen Lage von Startpunkt P<sub>0</sub> und Endpunkt P<sub>4</sub> in Bezug auf die Bearbeitungsebene, erfolgen die Zustellbewegungen in negative (Normalfall beim Anfahren) oder in positive (Normalfall beim Abfahren) Richtung, d. h. dass es z. B. bei G17 zulässig ist, dass die Z-Komponente des Endpunkt P<sub>4</sub> größer ist als die des Startpunkts P<sub>0</sub>.

**Programmierung des Endpunktes P<sub>4</sub> (bzw. P<sub>0</sub> beim Abfahren) in der Regel mit X... Y... Z...****Mögliche Programmierung des Endpunktes P<sub>4</sub> beim Anfahren**

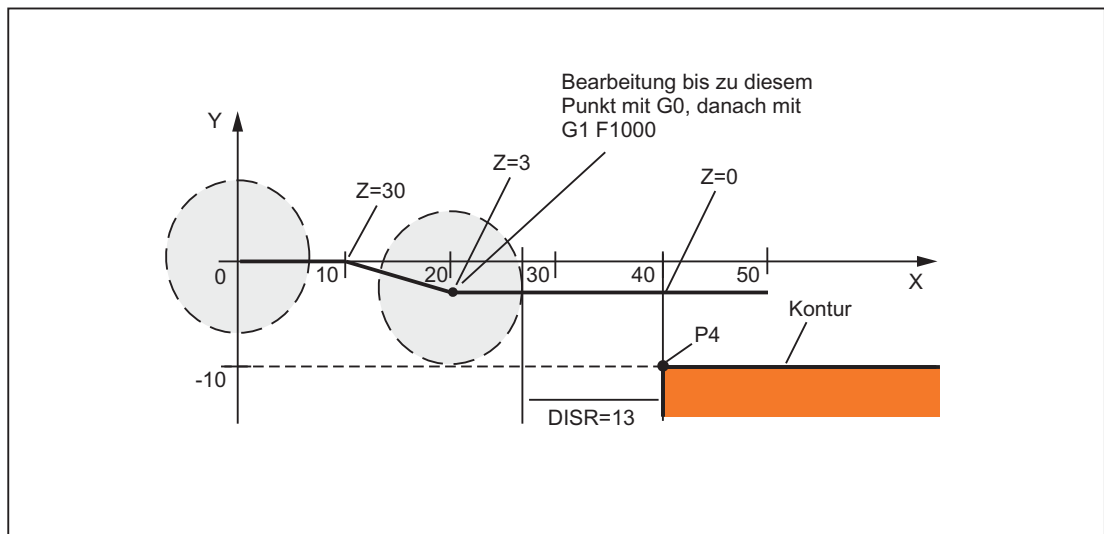
Endpunkt P<sub>4</sub> kann im WAB-Satz selbst programmiert sein.

P<sub>4</sub> kann durch den Endpunkt des nächsten Verfahrssatzes bestimmt werden.

Zwischen dem WAB-Satz und dem nächsten Verfahrssatz können weitere Sätze (Dummysätze) ohne Bewegung der Geometrieachsen eingefügt werden.

Der Endpunkt gilt beim Anfahren dann als im WAB-Satz selbst programmiert, wenn mindestens eine Geometrieachse der Bearbeitungsebene (X oder Y bei G17) programmiert wurde. Ist im WAB-Satz nur die Position der Achse senkrecht zur Bearbeitungsebene (Z bei G17) programmiert, so wird diese Komponente aus dem WAB-Satz entnommen, die Position in der Ebene dagegen aus dem Folgesatz. Ist in diesem Fall die Achse senkrecht zur Bearbeitungsebene zusätzlich auch im Folgesatz programmiert, wird ein Alarm ausgegeben.

**Beispiel:**



```

$TC_DP1 [1,1]=120 ; Fräserwerkzeug T1/D1
$TC_DP6 [1,1]=7 ; Werkzeug mit 7mm Radius

N10 G90 G0 X0 Y0 Z30 D1 T1
N20 X10
N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000
N40 G1 X40 Y-10
N50 G1 X50
...
...
    
```

N30/N40 kann ersetzt werden durch:

```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000
```

oder:

```
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0
```

### Mögliche Programmierung des Endpunktes $P_0$ beim Abfahren

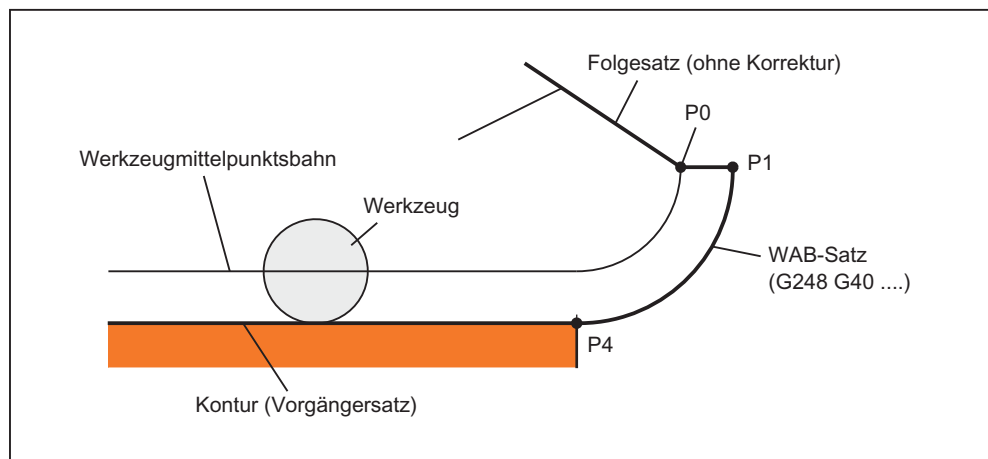
Die Endposition wird immer aus dem WAB-Satz selbst entnommen, unabhängig wie viele Achsen programmiert wurden. Es sind folgende Fälle zu unterscheiden:

1. Im WAB-Satz ist keine Geometrieachse programmiert.

Die Kontur endet in diesem Fall im Punkt  $P_2$  (bzw. im Punkt  $P_1$  wobei  $P_1$  und  $P_2$  zusammenfallen). Die Position in den Achsen, die die Bearbeitungsebene bilden, ergibt sich aus der Wegfahrkontur (Endpunkt der Geraden bzw. des Kreises). Die Achskomponente senkrecht dazu wird durch  $DISCL$  definiert. Ist in diesem Fall  $DISCL = 0$ , verläuft die Bewegung deshalb vollständig in der Ebene.

2. Im WAB-Satz ist nur die Achse senkrecht zur Bearbeitungsebene programmiert.

Die Kontur endet in diesem Fall im Punkt  $P_1$ . Die Position der übrigen beiden Achsen ergibt sich wie unter 1.



#### Abfahren mit WAB bei gleichzeitiger Deaktivierung der WRK

Ist der WAB-Abfahrsatz gleichzeitig der Deaktivierungssatz der Werkzeugradiuskorrektur, so wird in 1. und 2. ein zusätzlicher Weg von  $P_1$  nach  $P_0$  derart eingefügt, dass sich bei der Deaktivierung der Werkzeugradiuskorrektur am Ende der Wegfahrkontur keine Bewegung ergibt, d. h. dieser Punkt definiert dann nicht eine Position auf einer zu korrigierenden Kontur, sondern den Werkzeugmittelpunkt.

3. Es ist mindestens eine Achse der Bearbeitungsebene programmiert.

Die eventuell fehlende zweite Achse der Bearbeitungsebene wird aus ihrer letzten Position im Vorgängersatz modal ergänzt. Die Position der Achse senkrecht zur Bearbeitungsebene wird - abhängig davon, ob diese Achse programmiert ist oder nicht - wie unter 1. oder 2. gebildet. Die so gebildete Position definiert den Endpunkt  $P_0$ .

Die Abwahl der Werkzeugradiuskorrektur muss nicht gesondert behandelt werden, da der programmierte Punkt  $P_0$  bereits unmittelbar die Position des Werkzeugmittelpunktes am Ende der Gesamtkontur definiert.

Start- und Endpunkt der WAB-Kontur ( $P_0$  bzw.  $P_4$ ) dürfen sowohl beim An- als auch beim Abfahren zusammenfallen.

### Geschwindigkeit des Vorgängersatzes (typisch G0)

Mit dieser Geschwindigkeit werden alle Bewegungen vom Punkt P<sub>0</sub> bis zum Punkt P<sub>2</sub> ausgeführt, d. h. die Bewegung parallel zur Bearbeitungsebene und der Teil der Zustellbewegung bis zum Sicherheitsabstand.

### Programmierung der Vorschubgeschwindigkeit mit FAD

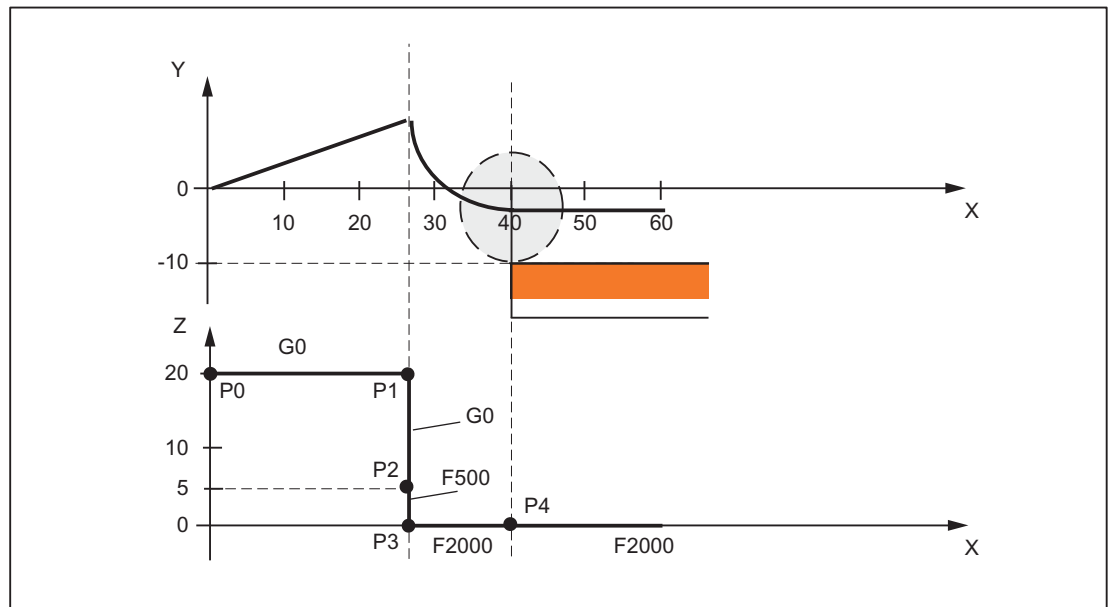
FAD programmiert bei ...	
G340	Vorschubgeschwindigkeit von P <sub>2</sub> bzw. P <sub>3</sub> zu P <sub>4</sub> .
G341	Vorschubgeschwindigkeit der Zustellbewegung senkrecht zur Bearbeitungsebene von P <sub>2</sub> nach P <sub>3</sub> .

Wird FAD nicht programmiert, wird dieser Teil der Kontur mit der modal wirksamen Geschwindigkeit des Vorgängersatzes verfahren, falls im WAB-Satz kein F-Wort programmiert ist, das die Geschwindigkeit festlegt.

#### Programmierverhalten:

- FAD=0 oder negativ → Alarmausgabe
- FAD=... → programmierter Wert wirkt entsprechend dem aktiven G-Code der Gruppe 15 (Vorschubtyp; G<sub>93</sub>, G<sub>94</sub> usw.)
- FAD=PM(... ) → programmierter Wert wird unabhängig vom aktiven G-Code der Gruppe 15 als Linearvorschub (wie G<sub>94</sub>) interpretiert
- FAD=PR(... ) → programmierter Wert wird unabhängig vom aktiven G-Code der Gruppe 15 als Umdrehungsvorschub (wie G<sub>95</sub>) interpretiert

Beispiel:



```

$TC_DP1 [1,1]=120 ; Fräserwerkzeug T1/D1
$TC_DP6 [1,1]=7 ; Werkzeug mit 7mm Radius

N10 G90 G0 X0 Y0 Z20 D1 T1
N20 G41 G341 G247 DISCL=AC(5) DISR=13FAD 500 X40 Y-10 Z=0 F2000
N30 X50
N40 X60
...

```

### Programmierung Vorschub F

Dieser Vorschubwert ist ab dem Punkt P<sub>3</sub> (bzw. ab dem Punkt P<sub>2</sub>, falls F<sub>AD</sub> nicht programmiert ist) wirksam. Wird im WAB-Satz kein F-Wort programmiert, so wirkt die Geschwindigkeit des Vorgängersatzes. Die durch F<sub>AD</sub> definierte Geschwindigkeit wird für Folgesätze nicht übernommen.

### 17.5.4.3 Geschwindigkeiten

#### Geschwindigkeiten beim Anfahren

In den beiden folgenden Darstellungen für das Anfahren wird davon ausgegangen, dass in dem auf den WAB-Satz folgenden Satz keine neue Geschwindigkeit programmiert wurde. Ist dies nicht der Fall, wird ab dem Punkt P<sub>4</sub> die neue Geschwindigkeit wirksam.

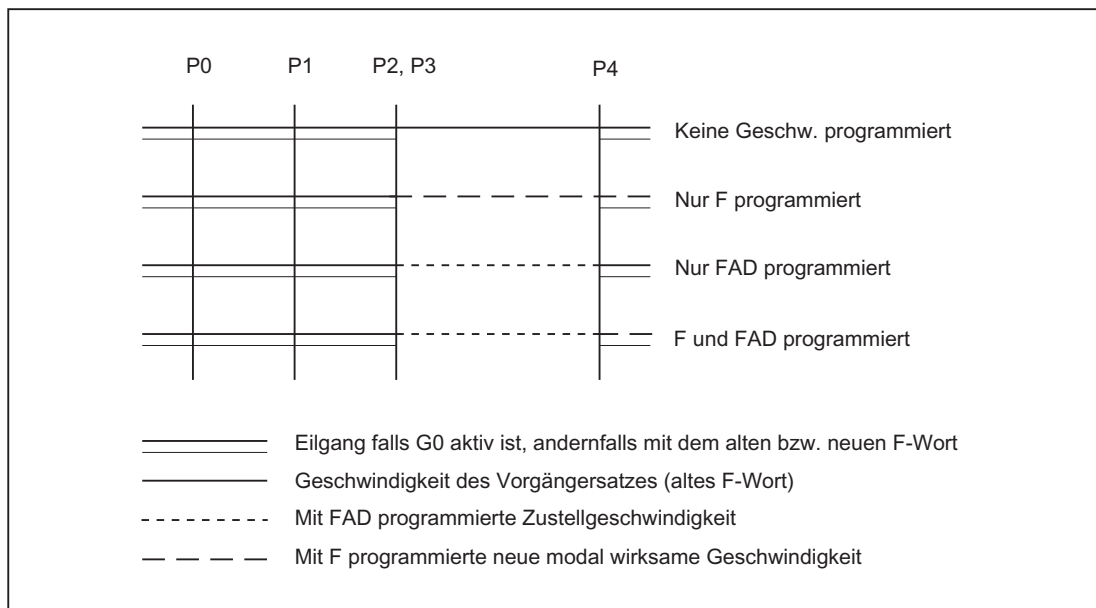


Bild 17-18 Geschwindigkeiten in den WAB-Teilsätzen beim Anfahren mit G340

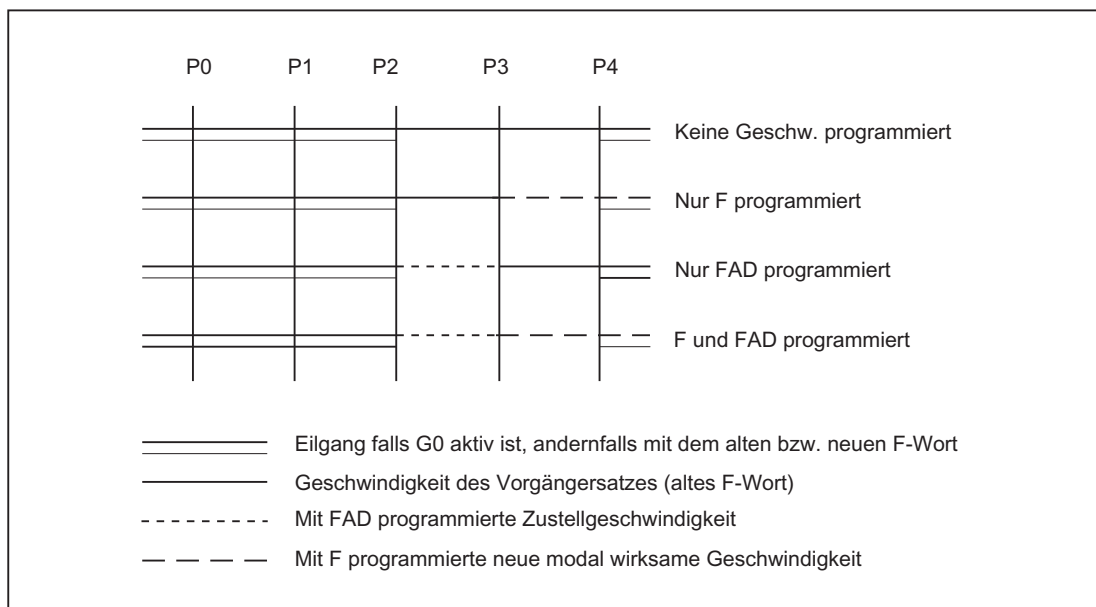


Bild 17-19 Geschwindigkeiten in den WAB-Teilsätzen beim Anfahren mit G341

## Geschwindigkeiten beim Abfahren

Beim Abfahren sind die Rollen von modal wirksamem Vorschub aus dem Vorgängersatz und dem im WAB-Satz programmierten Vorschubwert vertauscht, d. h. die eigentliche Abfahrkontur (Gerade, Kreis, Helix) wird mit dem alten Vorschub verfahren, eine neu mit F-Wort programmierte Geschwindigkeit gilt entsprechend ab dem Punkt  $P_2$  bis zum Punkt  $P_0$ .

Ist ebenes Wegfahren aktiv und  $F_{AD}$  ist programmiert, wird die Strecke  $P_3$  nach  $P_2$  mit  $F_{AD}$  verfahren, andernfalls mit der alten Geschwindigkeit. Für die Strecke  $P_4$  nach  $P_2$  gilt immer das letzte in einem Vorgängersatz programmierte F-Wort.  $G_0$  wird in diesen Sätzen nicht wirksam.

Von  $P_2$  nach  $P_0$  wird mit dem im WAB-Satz programmierten F-Wort verfahren bzw., falls kein F-Wort programmiert wurde, mit dem modal wirksamen F-Wort aus einem Vorgängersatz. Dies gilt unter der Bedingung, dass  $G_0$  nicht aktiv ist.

Sollen beim Abfahren die Sätze von  $P_2$  nach  $P_0$  mit Eilgang verfahren werden, so muss  $G_0$  vor dem WAB-Satz oder im WAB-Satz selbst aktiviert werden. Ein eventuell zusätzlich programmiertes F-Wort hat dann in den WAB-Sätzen selbst keine Bedeutung mehr. Es bleibt jedoch modal für Folgesätze wirksam.

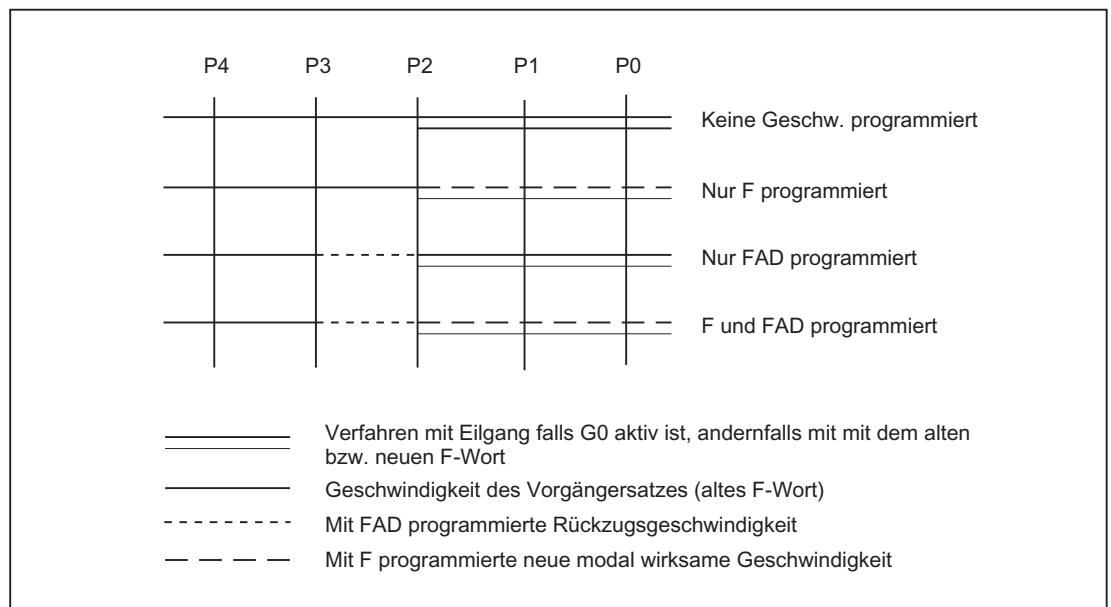


Bild 17-20 Geschwindigkeiten in den WAB-Teilsätzen beim Abfahren

### 17.5.4.4 Systemvariablen

Die Punkte P<sub>3</sub> und P<sub>4</sub> können beim Anfahren als Systemvariablen im WKS gelesen werden:

\$P_APR:	Lesen von P <sub>3</sub> (Aufstartpunkt) im WKS	
\$P_AEP:	Lesen von P <sub>4</sub> (Konturanfangspunkt) im WKS	
\$P_APDV	=1	Wenn der Inhalt von \$P_APR und \$P_AEP gültig ist, d. h. wenn diese die zum letzten programmierten WAB-Anfahr Satz gehörenden Positionswerte enthalten.
	=0	Die Positionen eines älteren WAB-Anfahr Satzes werden gelesen.

Eine Änderung des WKS zwischen WAB-Satz und Leseoperation hat keinen Einfluss auf die Positionswerte.

### 17.5.4.5 Randbedingungen

#### Randbedingungen

- In einem WAB-Satz können beliebige weitere NC-Befehle programmiert werden (z. B. Hilfsfunktionsausgaben, Synchronachsbewegungen, Positionierachsbewegungen usw.).  
Diese werden beim Anfahren im ersten und beim Abfahren im letzten Teilsatz ausgeführt.
- Wird der Endpunkt P<sub>4</sub> nicht aus dem WAB-Satz, sondern aus einem nachfolgenden Verfahrsatz entnommen, so wird die eigentliche WAB-Kontur (Gerade, Viertel- oder Halbkreis) in diesem Satz abgefahren.  
Der letzte Teilsatz des ursprünglichen WAB-Satzes enthält dann keine Verfahrinformation für die Geometrieachsen. Er wird aber dennoch stets ausgegeben, da in diesem Satz eventuell weitere Aktionen (z. B. Einzelachsen) ausgeführt werden müssen.
- Es müssen immer mindestens zwei Sätze betrachtet werden:
  - der WAB-Satz selbst
  - der Satz, der die Anfahr- bzw. Abfahr-richtung definiert
 Zwischen diesen beiden Sätzen können weitere Sätze programmiert werden.  
Die Anzahl möglicher Zwischensätze wird begrenzt mit dem Maschinendatum:  
MD20202 \$MC\_WAB\_MAXNUM\_DUMMY\_BLOCKS (Max. Satzanzahl ohne Verfahrbewegung bei WAB)



- Wird in einem Anfahr Satz gleichzeitig die Werkzeugradiuskorrektur aktiviert, so ist der erste Geraden Satz der WAB-Kontur der Aktivierungssatz.  
Die gesamte Kontur, die von der WAB-Funktion erzeugt wurde, wird von der Werkzeugradiuskorrektur so behandelt, als ob sie explizit programmiert worden wäre (Kollisionsüberwachungen, Schnittpunktberechnungen, Anfahrverhalten  $NORM / KONT$ ).
- Die Richtung der Zustellbewegung und die Lage der Kreisebene bzw. der Helixachse wird ausschließlich durch die aktive Ebene ( $G17 - G19$ ) - gegebenenfalls gedreht mit einem aktiven Frame - bestimmt.
- Beim Anfahren darf zwischen dem WAB-Satz und dem Folgesatz, der die Tangentenrichtung definiert, kein Vorlaufstopp eingefügt werden.  
Ein Vorlaufstopp in diesem Fall - gleichgültig ob explizit programmiert oder von der Steuerung automatisch eingefügt - führt zu einem Alarm.

### Verhalten bei REPOS

Wird ein WAB-Zyklus unterbrochen und repositioniert, so wird bei  $RMIBL$  an der Unterbrechungsstelle wieder aufgesetzt. Der Aufsetzpunkt bei  $RMEBL$  ist der Endpunkt des letzten WAB-Satzes, bei  $RMBBL$  entsprechend der Startpunkt des ersten WAB-Satzes.

Wird  $RMIBL$  zusammen mit  $DISPR$  (Wiederanfahren im Abstand  $DISPR$  vor dem Unterbrechungspunkt) programmiert, kann der Wiederanfahrpunkt in einem Teilsatz des WAB-Zyklus vor dem Unterbrechungsteilsatz liegen.

### 17.5.4.6 Beispiele

#### Beispiel 1

Folgende Bedingungen sind gegeben:

- Weiches Anfahren wird in Satz N20 aktiviert
- X=40 (Endpunkt); Y=0; Z=0
- Anfahrbewegung erfolgt mit Viertelkreis (G247)
- Anfahrriichtung nicht programmiert, es wirkt G140, d.h. da WRK aktiv (G42) und Korrekturwert positiv ist (10), wird Kontur von rechts angefahren
- Intern erzeugte Anfahrkreis (WAB-Kontur) hat Radius 20, so dass der Radius der Werkzeugmittelpunktsbahn gleich dem programmierten Wert DISR=10 ist
- Die Anfahrbewegung erfolgt wegen G341 mit einem Kreis in der Ebene, so dass sich der Startpunkt (20, -20, 0) ergibt
- Der Punkt P2 hat wegen DISCL=5 die Position (20, -20, 5) und der Punkt P1 liegt wegen Z30 in N10 bei (20, -20, 30)

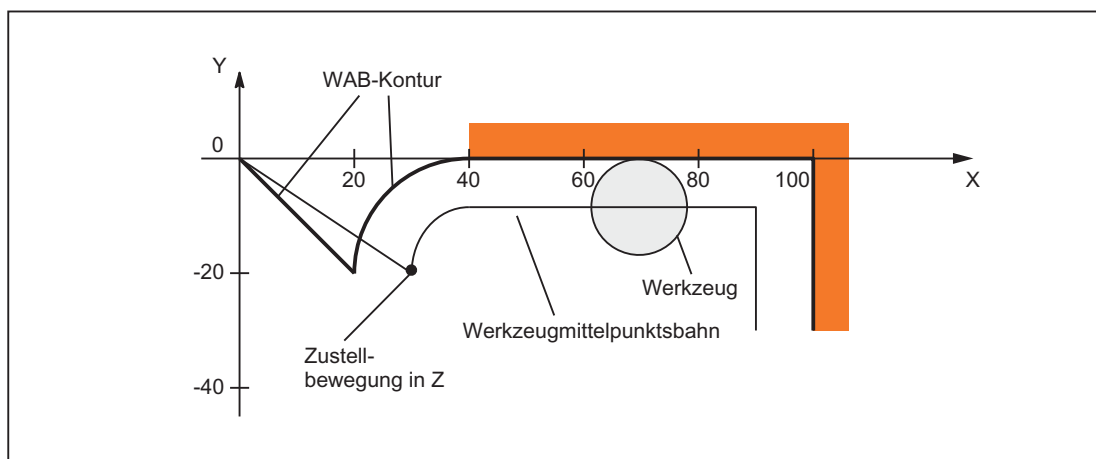


Bild 17-21 Kontur Beispiel 1

#### Teileprogramm:

Programmcode	Kommentar
\$TC_DP1 [1,1]=120	; Werkzeugdefinition T1/D1
\$TC_DP6 [1,1]=10	; Radius
N10 G0 X0 Y0 Z30	
N20 G247 G341 G42 NORM D1 T1 Z0 FAD=1000 F=2000 DISCL=5 DISR=10	
N30 X40	
N40 X100	
N50 Y-30	
...	

## Beispiel 2

Folgende Bedingungen sind beim Anfahren gegeben:

- Weiches Anfahren wird in Satz N20 aktiviert
- Anfahrbewegung erfolgt mit Viertelkreis (G247)
- Anfahrriechung nicht programmiert, es wirkt G140, d. h. da WRK aktiv (G41) ist, wird Kontur von links angefahren
- Konturoffset OFFN=5 (N10)
- aktueller Werkzeugradius=10, damit ist der effektive Korrekturradius für WRK=15; der Radius der WAB-Kontur ist damit gleich 25, so dass der Radius der Werkzeugmittelpunktsbahn gleich DISR=10 wird
- Endpunkt des Kreises ergibt sich aus N30, da in N20 nur Z-Position programmiert ist
- Zustellbewegung
  - von Z20 nach Z7 (DISCL=AC(7)) im Eilgang
  - anschließend nach Z0 mit FAD=200
  - Anfahrkreis in X-Y-Ebene und Folgesätze mit F1500

(Damit diese Geschwindigkeit in den Folgesätzen wirksam wird, muss der aktive G-Code G0 in N30 mit G1 überschrieben werden. Andernfalls würde die Kontur mit G0 weiter bearbeitet werden.)

Folgende Bedingungen sind beim Abfahren gegeben:

- Weiches Abfahren wird in Satz N60 aktiviert
- Abfahrbewegung erfolgt mit Viertelkreis (G248) und Helix (G340)
- FAD nicht programmiert, da bei G340 ohne Bedeutung
- Z=2 im Startpunkt; Z=8 im Endpunkt, da DISCL=6
- Bei DISR=5 ist Radius der WAB-Kontur=20, der der Werkzeugmittelpunktsbahn=5
- Nach dem Kreissatz folgen die Wegfahrbewegung von Z8 nach Z20 und die Bewegung parallel zur X-Y-Ebene zum Endpunkt bei X70 Y0

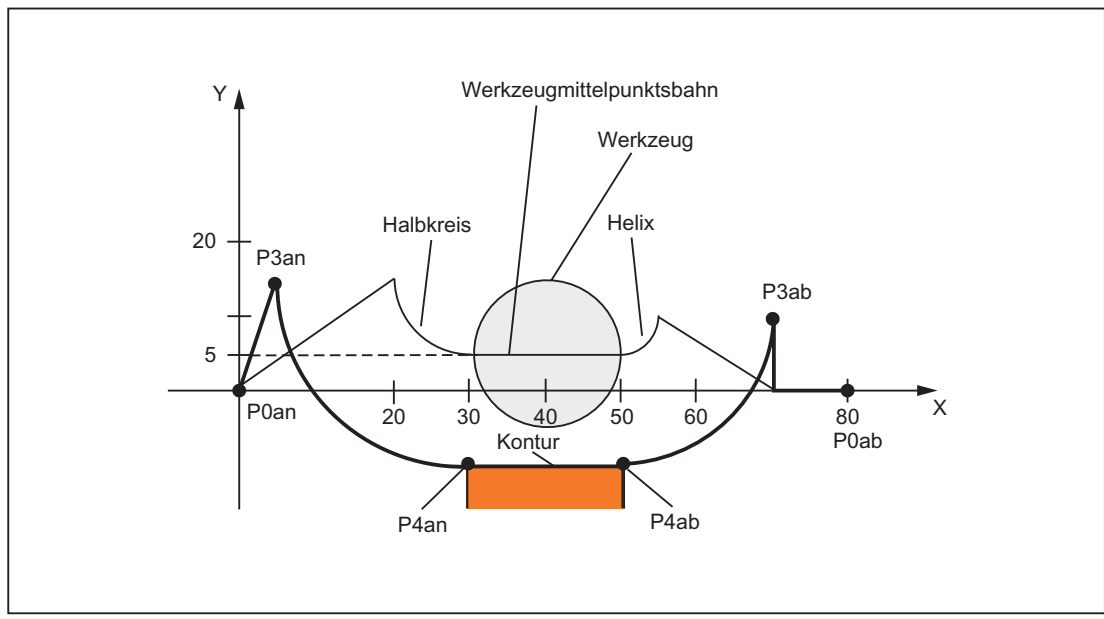


Bild 17-22 Kontur Beispiel 2

**Teileprogramm:**

Programmcode	Kommentar
\$TC_DP1 [1,1]=120	; Werkzeugdefinition
\$TC_DP6 [1,1]=10	T1/D1
	; Radius
N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN = 5	; (P0an)
N20 G41 G247 G341 Z0 DISCL = AC(7) DISR = 10 F1500 FAD=200	; (P3an)
N30 G1 X30 Y-10	; (P4an)
N40 X40 Z2	
N50 X50	; (P4ab)
N60 G248 G340 X70 Y0 Z20 DISCL = 6 DISR = 5 G40 F10000	; (P3ab)
N70 X80 Y0	; (P0ab)
N80 M 30	

**Hinweis**

Die derart erzeugte Kontur wird durch die Werkzeugradiuskorrektur modifiziert, die im WAB-Anfahrtsatz aktiviert und im WAB-Abfahrtsatz deaktiviert wird.

Die Werkzeugradiuskorrektur berücksichtigt einen effektiven Radius von 15, der sich aus der Summe von Werkzeugradius (10) und Konturoffset (5) zusammensetzt. Der resultierende Radius der Werkzeugmittelpunktsbahn im Anfahrtsatz ist deshalb 10, im Abfahrtsatz 5.

## 17.5.5 Abwahl der WRK (G40)

### Anweisung G40

Die Abwahl der WRK erfolgt mit der Anweisung G40.

### Besonderheiten

- Die Abwahl der WRK kann nur in einem Programmsatz mit G0 (Eilgang) oder G1 (Linearinterpolation) erfolgen.
- Wird bei aktiver WRK D0 programmiert, dann erfolgt keine Abwahl, sondern die Fehlermeldung 10750.
- Wird im Satz mit der WRK-Abwahl eine Geometrieachse programmiert, findet eine Abwahl der WRK statt, auch wenn diese nicht in der aktuellen Ebene liegt.

## 17.5.6 Korrektur an Außenecken

### G450/G451

Mit den G-Funktionen G450/G451 kann das Verhalten bei unstetigen Satzübergängen an Außenecken gesteuert werden:

Befehl	Bedeutung
G450	Unstetige Satzübergänge mit Übergangskreis
G451	Unstetige Satzübergänge mit Schnittpunkt der Äquidistanten

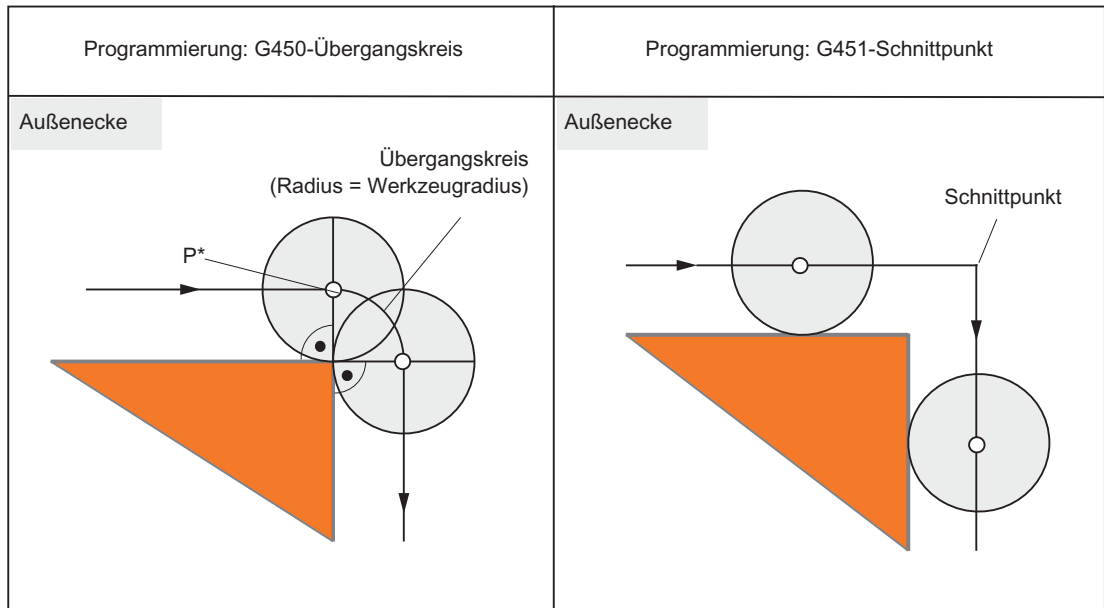


Bild 17-23 Beispiel für 90°-Außenecke mit G450 und G451

### G450 (Übergangskreis)

Bei aktiver G-Funktion  $G_{450}$  vollzieht der Mittelpunkt des Werkzeugs bei Außenecken eine Kreisbahn mit dem Werkzeugradius. Die Kreisbahn beginnt mit der Normalenstellung (senkrecht zur Bahntangente) im Endpunkt des vorherigen Bahnstückes (Programmsatzes) und endet in Normalenstellung im Anfangspunkt des neuen Bahnstückes (Programmsatzes).

Bei sehr flachen Außenecken nähert sich das Verfahren mit  $G_{450}$  (Übergangskreis) und  $G_{451}$  (Schnittpunkt) immer mehr an (siehe Abschnitt "Sehr flache Außenecken").

Sollen spitze Außenkonturrecken entstehen, muss das Werkzeug von der Kontur abgehoben werden (siehe Abschnitt "DISC").

### DISC

Mit  $G_{450}$  - Übergangskreis können keine scharfen Außenkonturrecken entstehen, weil durch den Übergangskreis die Werkzeugmittelpunktsbahn so geführt wird, dass die WZ-Schneide auf der Außenecke (programmierte Position) stillsteht. Sollen mit  $G_{450}$  trotzdem scharfe Außenecken bearbeitet werden, kann mit der Anweisung  $DISC$  im Programm eine Überhöhung programmiert werden. Dadurch wird aus dem Übergangskreis ein Kegelschnitt und die WZ-Schneide hebt von der Außenecke ab.

Der Wertebereich der Anweisung  $DISC$  beträgt 0 bis 100 in Schritten von 1.

Wert	Bedeutung
$DISC = 0$	Überhöhung abgeschaltet, Übergangskreis wirksam
$DISC = 100$	Überhöhung so groß, dass sich theoretisch ein Verhalten wie bei Schnittpunkt ( $G_{451}$ ) ergibt.

Der programmierbare Maximalwert für  $DISC$  kann eingestellt werden über das Maschinendatum:

MD20220 \$MC\_CUTCOM\_MAX\_DISC (max. Wert für DISC)

Sinnvolle Werte für  $DISC$  liegen in der Regel nicht über 50.

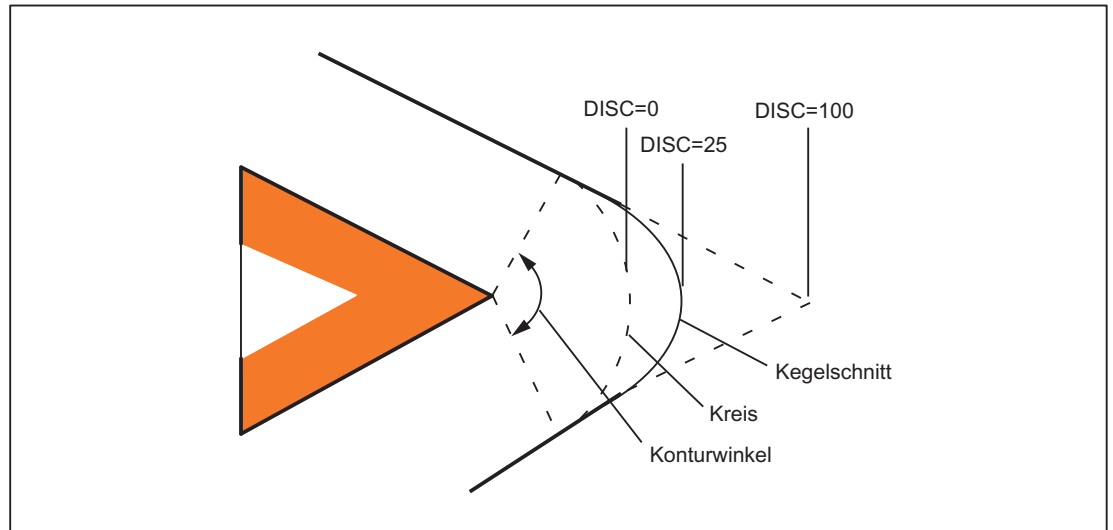


Bild 17-24 Beispiel: Überhöhung mit  $DISC=25$

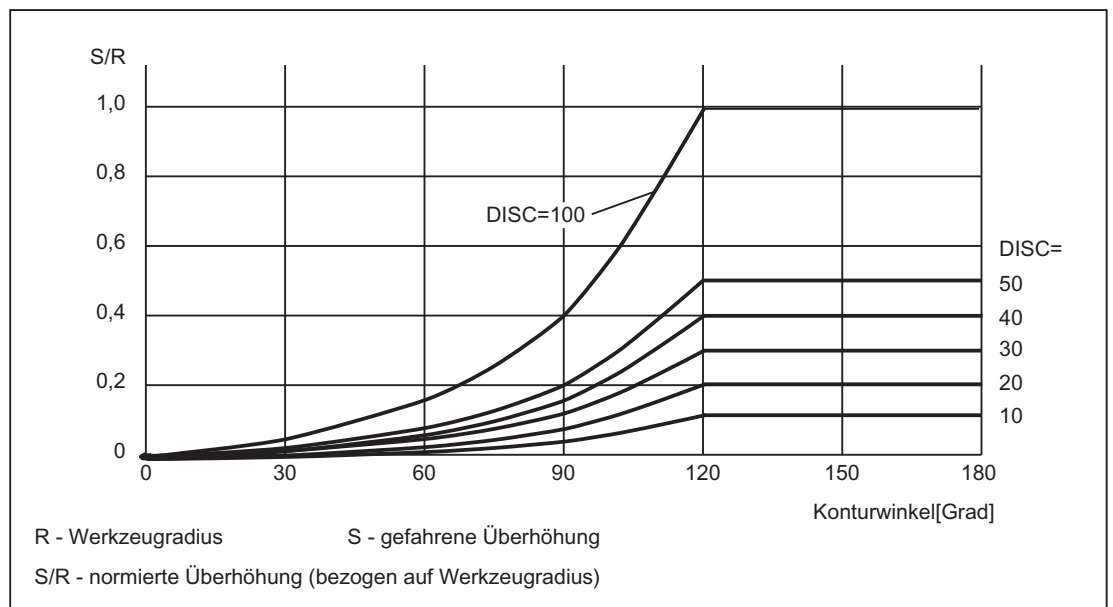


Bild 17-25 Überhöhung mit  $DISC$  abhängig von Konturwinkel

### G451 (Schnittpunkt)

Bei aktiver G-Funktion G451 wird die Position (der Schnittpunkt) angefahren, die sich aus den Bahnlinien (nur bei Gerade, Kreis oder Helix) ergibt, die sich im Abstand des Werkzeugradius zur programmierten Kontur befinden (Mittelpunktsbahn des Werkzeugs). Splines und Polynome werden grundsätzlich nicht verlängert.

### Sehr spitze Außenecken

Bei sehr spitzen Außenecken kann es mit G451 zu langen Leerwegen kommen. Deshalb wird bei sehr spitzen Außenecken automatisch von G451 (Schnittpunkt) auf G450 (Übergangskreis, ggf. mit DISC) umgeschaltet.

Der Grenzwinkel (Konturwinkel) für diese automatische Umschaltung (Schnittpunkt → Übergangskreis) kann vorgegeben werden im Maschinendatum:

MD20210 \$MC\_CUTCOM\_CORNER\_LIMIT (Maximalwinkel für Ausgleichssätze bei WRK)

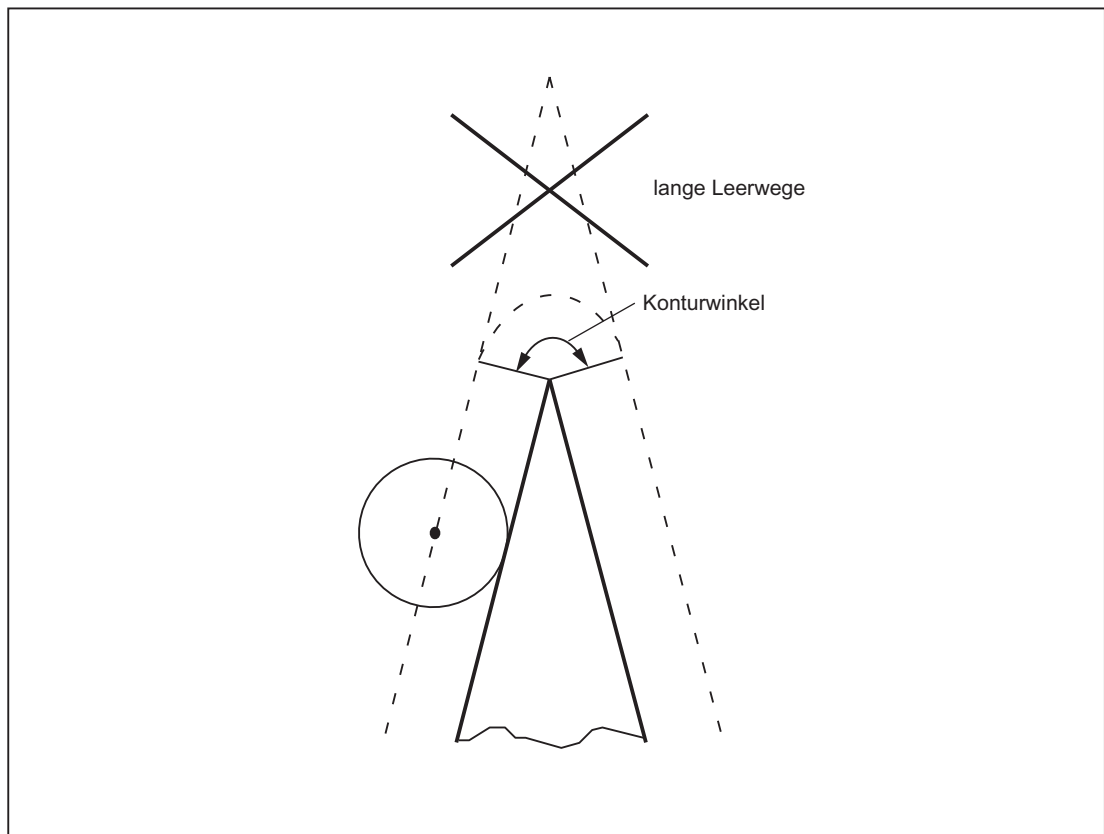


Bild 17-26 Beispiel für automatische Umschaltung auf Übergangskreis



## Sehr flache Außenecken

Bei sehr flachen Außenecken nähern sich die Verfahren mit G450 (Übergangskreis) und G451 (Schnittpunkt) immer mehr an. In diesem Fall macht es keinen Sinn mehr, einen Übergangskreis einzufügen. Insbesondere bei der 5-Achsbearbeitung darf an diesen Außenecken kein Übergangskreis eingefügt werden, weil es sonst im Bahnsteuerbetrieb (G64) zu Geschwindigkeitseinbußen kommt. Deshalb wird bei sehr flachen Außenecken automatisch von G450 (Übergangskreis, ggf. mit DISC) auf G451 (Schnittpunkt) umgeschaltet.

Der Grenzwinkel (Konturwinkel) für diese automatische Umschaltung (Übergangskreis → Schnittpunkt) kann vorgegeben werden im Maschinendatum:

MD20230 \$MC\_CUTCOM\_CURVE\_INSERT\_LIMIT (Maximalwinkel für Schnittpunktberechnung bei WRK)

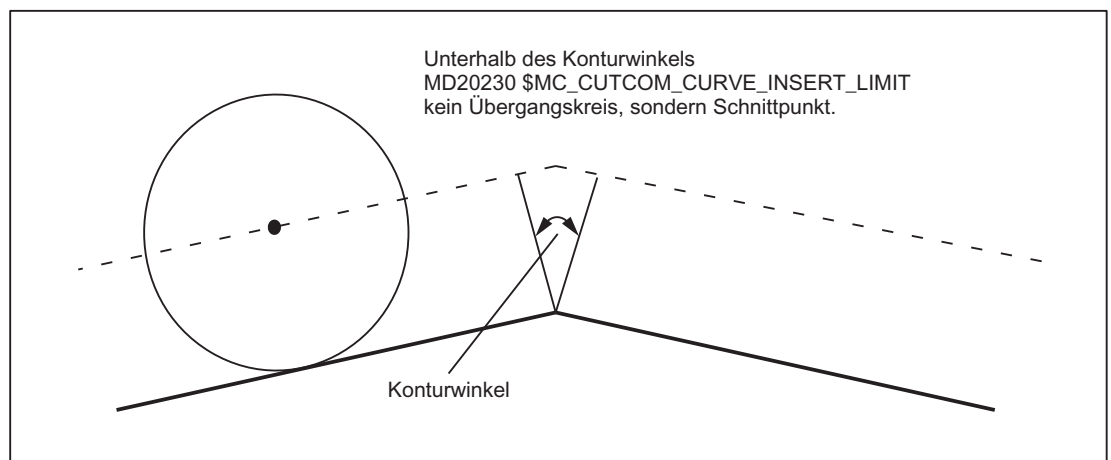


Bild 17-27 Beispiel für automatische Umschaltung auf Schnittpunkt

## 17.5.7 Korrektur an Innenecken

### Schnittpunkt

Bilden zwei aufeinanderfolgende Sätze eine Innenecke, so wird versucht, einen Schnittpunkt der beiden Äquidistanten zu finden. Wird ein Schnittpunkt gefunden, wird die programmierte Kontur bis zum Schnittpunkt verkürzt (erster Satz am Ende verkürzt, zweiter Satz am Anfang verkürzt).

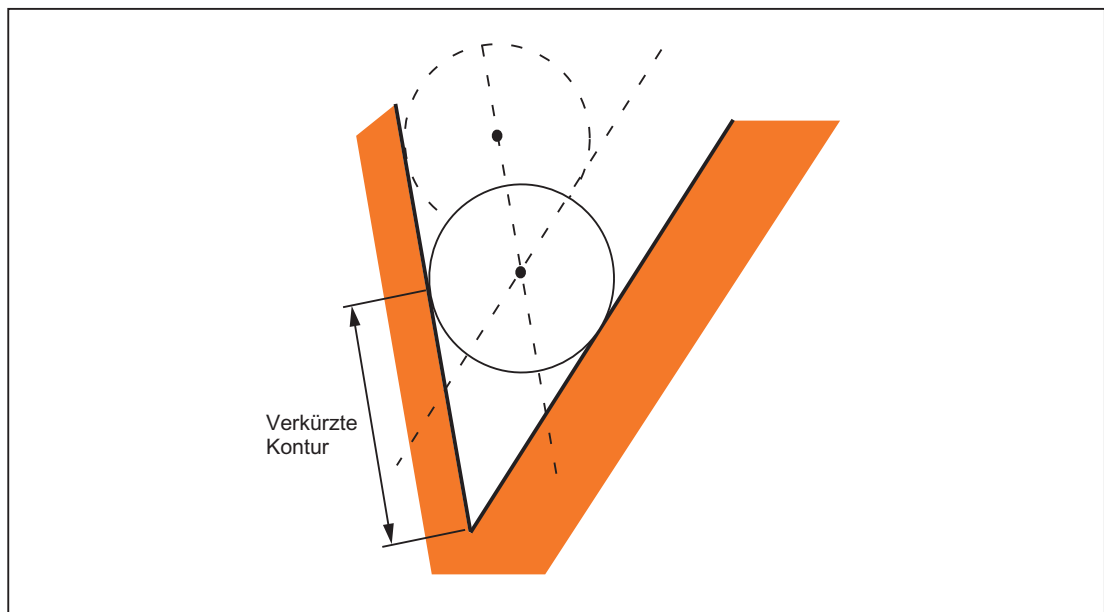


Bild 17-28 Beispiel für eine verkürzte Kontur

### Kein Schnittpunkt

Bei Innenecken kann der Fall eintreten, dass zwischen zwei aufeinander folgenden Sätzen kein Schnittpunkt gefunden wird (siehe folgendes Bild).

### Vorausschauende Konturberechnung

Wird zwischen zwei aufeinander folgenden Sätzen kein Schnittpunkt gefunden, betrachtet die Steuerung automatisch den nächsten Satz und versucht, mit der Äquidistanten dieses Satzes einen Schnittpunkt zu finden (siehe folgendes Bild: Schnittpunkt S). Diese automatische Betrachtung der nächsten Sätze (die vorausschauende Konturberechnung) wird immer so lange durchgeführt, bis eine per Maschinendatum einstellbare Satzanzahl erreicht ist:

MD20240 \$MC\_CUTCOM\_MAXNUM\_CHECK\_BLOCKS (Sätze für vorausschauende Konturberechnung bei WRK)

Wird innerhalb der vorausschauend betrachteten Sätze kein Schnittpunkt gefunden, bleibt die Programmbearbeitung stehen und ein Alarm wird gesetzt.

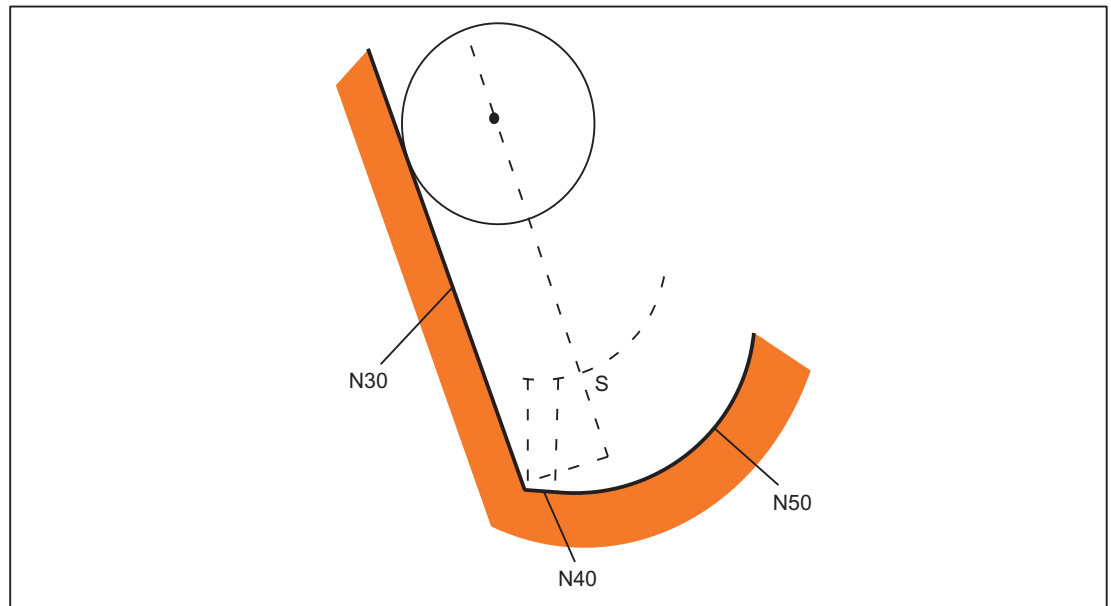


Bild 17-29 Gibt es keinen Schnittpunkt zwischen Satz N30 und Satz N40, dann wird der Schnittpunkt zwischen Satz N30 und Satz N50 berechnet.

### Mehrere Schnittpunkte

Bei Innenecken kann der Fall eintreten, dass die vorausschauende Konturberechnung mehrere Schnittpunkte der Äquidistanten in mehreren aufeinander folgenden Sätzen findet. Dabei wird immer der letzte Schnittpunkt als gültiger Schnittpunkt festgelegt:

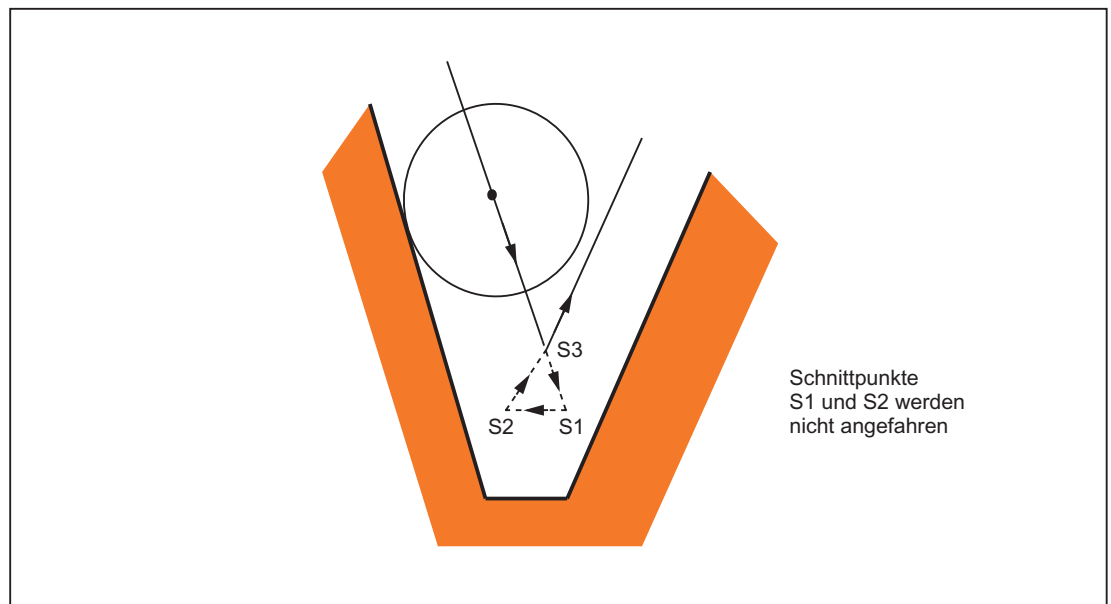


Bild 17-30 Beispiel: Innenecke mit WRK ohne Konturverletzung (3 Sätze vorausschauend)

Für weitere Informationen siehe auch Kapitel "Kollisionsüberwachung und Flaschenhalskennung (Seite 1644)" Abschnitt "Kollisionsüberwachung".

### Besonderheiten

Werden mehrere Schnittpunkte mit dem nächsten Satz gefunden, ist der Schnittpunkt gültig, der am nächsten beim Startpunkt des ersten Satzes liegt.

## 17.5.8 Kollisionsüberwachung und Flaschenhalserkennung

### Kollisionsüberwachung

Die Kollisionsüberwachung (Flaschenhalserkennung) überprüft, ob sich die Äquidistanten von nicht benachbarten Sätzen schneiden. Wird ein Schnittpunkt gefunden, ist das Verhalten wie bei Innenecken mit mehreren Schnittpunkten: Der letzte gefundene Schnittpunkt ist gültig:

Die max. Anzahl von Sätzen die vorausschauend betrachtet wird, kann eingegeben werden im Maschinendatum:

MD20240 \$MC\_CUTCOM\_MAXNUM\_CHECK\_BLOCKS (Sätze für vorausschauende Konturberechnung bei WRK)

### Programmierung

Die Kollisionsüberwachung kann im Programm ein- bzw. ausgeschaltet werden:

Befehl	Bedeutung
CDON	Kollisionsüberwachung ein
CDOF	Kollisionsüberwachung aus
CDOF2	Kollisionsüberwachung aus

Bei CDOF wird zunächst zwischen zwei aufeinander folgenden Sätzen ein Schnittpunkt gesucht. Weitere Sätze werden nicht berücksichtigt. Wird zwischen benachbarten Sätzen ein Schnittpunkt gefunden, werden keine weiteren Sätze betrachtet. Bei Außenecken kann zwischen zwei aufeinander folgenden Sätzen immer ein Schnittpunkt gefunden werden.

Die vorausschauende Betrachtung von mehr als zwei benachbarten Sätzen ist somit bei CDON und auch bei CDOF möglich.

---

### Hinweis

CDOF2 wirkt nur beim 3D-Umfangasfräsen und hat bei allen anderen Bearbeitungsarten (z. B. 3D-Stirnfräsen) die gleiche Bedeutung wie CDOF.

---

## Satz auslassen

Wird ein Schnittpunkt zwischen zwei nicht benachbarten Sätzen erkannt, so werden die dazwischen liegenden Bewegungen in der Korrekturebene nicht ausgeführt. Alle anderen in den ausgelassenen Sätzen enthaltenen Bewegungen und ausführbaren Anweisungen (M-Befehle, Verfahren von Positionierachsen usw.) werden an der Position des Schnittpunktes in der Reihenfolge ausgeführt, in der sie im NC-Programm enthalten sind.

## Warnung 10763

Wenn ein Satz als Folge der Kollisionsüberwachung oder der Flaschenhalserkennung ausgelassen worden ist, erscheint die Warnung 10763. Das Programm wird nicht unterbrochen.

Die Ausgabe dieser Warnung wird unterdrückt, wenn das Bit 1 gesetzt ist im Maschinendatum:

MD11410 \$MN\_SUPPRES\_ALARM\_MASK (Maske zur Unterstützung spezieller Alarmausgaben)

## Besonderheiten

Bei der Schnittpunktüberprüfung nicht benachbarter Sätze werden nicht die programmierten Originalkonturen überprüft, sondern die zugehörigen berechneten Äquidistanten. Dies kann dazu führen, dass bei Außenecken ein "Flaschenhals" erkannt wird, obwohl das tatsächlich nicht der Fall ist. Die Ursache hierfür liegt darin, dass bei  $DISC > 0$  die berechnete Werkzeugbahn nicht äquidistant zur programmierten Originalkontur verläuft.

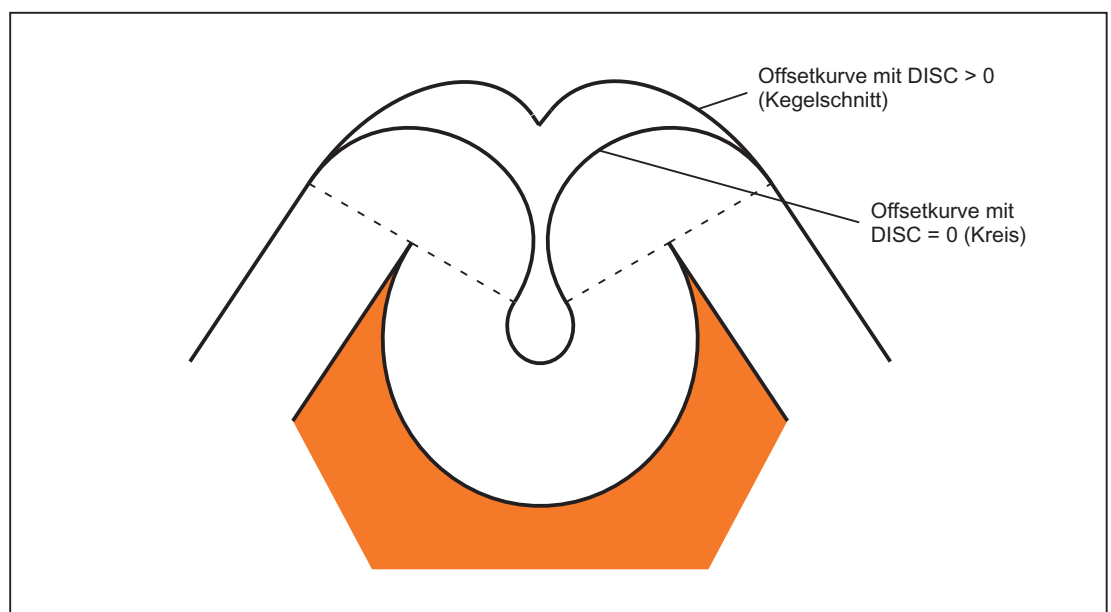


Bild 17-31 Flaschenhalserkennung und Außenecken

## 17.5.9 Sätze mit veränderlichem Korrekturwert

### Randbedingungen

Ein veränderlicher Korrekturwert ist bei allen Interpolationsarten zulässig (auch Kreis- und Spline-Interpolation).

Ebenso ist ein Wechsel des Vorzeichens (und damit ein Wechsel der Korrekturseite) zulässig.

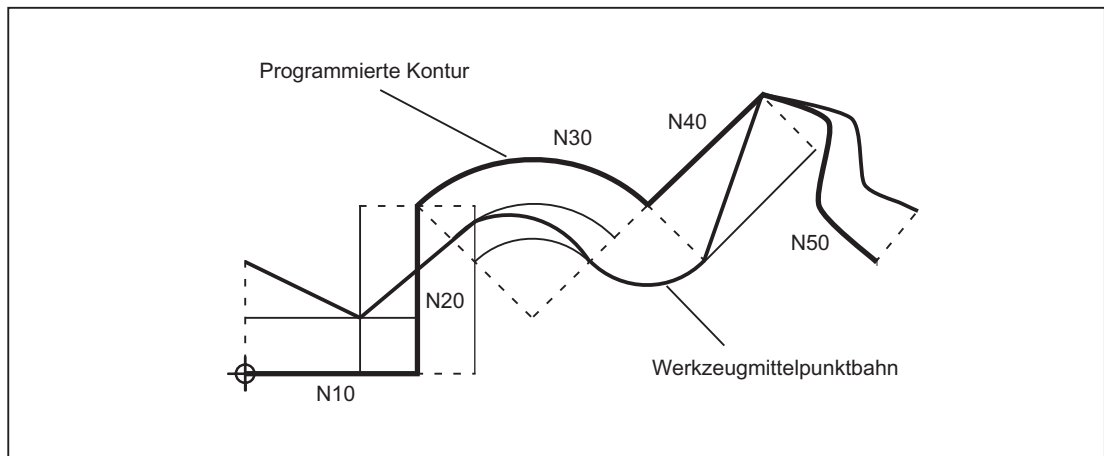


Bild 17-32 Werkzeugradiuskorrektur bei veränderlichem Korrekturwert

### Schnittpunktberechnung

Bei der Schnittpunktberechnung in Sätzen mit veränderlichem Korrekturwert wird der Schnittpunkt der Offsetkurven (Werkzeugbahnen) immer mit der Annahme eines konstanten Korrekturwertes berechnet.

Ist der Satz mit dem veränderlichen Korrekturwert der erste der beiden zu betrachtenden Sätze in Fahrrichtung, so wird der Korrekturwert am Satzende zur Berechnung herangezogen, andernfalls der Korrekturwert am Satzanfang.

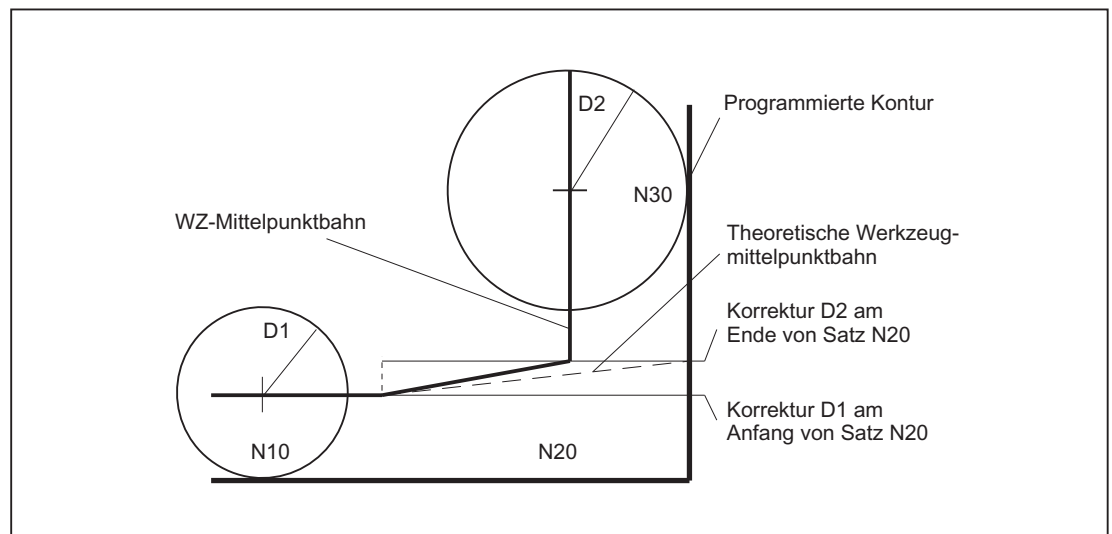


Bild 17-33 Schnittpunktberechnung bei veränderlichem Korrekturwert

### Einschränkung

Wird bei der Bearbeitung an der Kreisinnenseite der Korrekturradius größer als der programmierte Kreisradius, so wird die Bearbeitung mit dem folgenden Alarm abgelehnt:

Alarm 10758 "Krümmungsradius mit veränderlichem Korrekturwert zu klein"

### Geschlossene Kontur stabil erhalten

Wird ein Radius zweier Kreise geringfügig erhöht, kann dies einen dritten Satz erfordern, damit die geschlossene Kontur stabil erhalten bleibt. Dies ist dann der Fall, wenn zwei benachbarte Sätze, die zwei mögliche Schnittpunkte für eine geschlossene Kontur bilden, aufgrund der Korrektur übersprungen werden.

Eine stabil geschlossene Kontur kann dadurch erreicht werden, dass nicht der zweite mögliche Schnittpunkt gewählt wird, sondern der erste Schnittpunkt:

SD42496 \$SC\_CUTCOM\_CLSD\_CONT  $\neq$  0 (Verhalten der WRK bei geschlossener Kontur)

Damit wird im Falle des Überspringens dieses Satzes immer der zweite mögliche Schnittpunkt erreicht. Ein dritter Satz wird damit überflüssig.

## 17.5.10 Werkzeugradiuskorrektur konstant halten

### Bedeutung

Die Funktion "Werkzeugradiuskorrektur konstant halten" dient dazu, die Werkzeugradiuskorrektur für eine Anzahl von Sätzen zu unterdrücken, wobei jedoch eine durch die Werkzeugradiuskorrektur in vorhergehenden Sätzen aufgebaute Differenz zwischen der programmierten und der tatsächlich abgefahrenen Bahn des Werkzeugmittelpunktes als Verschiebung beibehalten wird.

Sie kann z. B. dann vorteilhaft eingesetzt werden, wenn beim Zeilenfräsen in den Umkehrpunkten mehrere Verfahrssätze notwendig sind, die von der Werkzeugradiuskorrektur erzeugten Konturen (Umfahrungsstrategien) jedoch nicht erwünscht sind.

### Aktivierung

Die Funktion "Werkzeugradiuskorrektur konstant halten" wird mit dem G-Code `CUTCNON` (CUTter compensation CONstant ON) aktiviert und mit dem G-Code `CUTCNOF` (CUTter compensation CONstant OFF) deaktiviert.

`CUTCNON` und `CUTCNOF` bilden eine modale G-Code-Gruppe.

Grundstellungswert ist `CUTCNOF`.

Die Funktion ist unabhängig von der Art der Werkzeugradiuskorrektur (2<sup>1</sup>/<sub>2</sub>D, 3D-Stirnfräsen, 3D-Umfangfräsen) einsetzbar.

### Normalfall

Im Normalfall ist vor der Aktivierung der Korrekturunterdrückung die Werkzeugradiuskorrektur bereits aktiv, und sie ist noch aktiv, wenn die Korrekturunterdrückung wieder deaktiviert wird.

Im letzten Verfahrssatz vor `CUTCNON` wird auf den Offsetpunkt im Satzendpunkt gefahren. Alle folgenden Sätze, in denen die Korrekturunterdrückung aktiv ist, werden ohne Korrektur verfahren. Sie werden dabei jedoch um den Vektor vom Endpunkt des letzten Korrektursatzes zu dessen Offsetpunkt verschoben. Der Interpolationstyp dieser Sätze (linear, zirkular, polynomial) ist beliebig.

Der Deaktivierungssatz der Korrekturunterdrückung, d. h. der Satz, der `CUTCNOF` enthält, wird normal korrigiert. Er beginnt im Offsetpunkt des Startpunktes. Zwischen dem Endpunkt des Vorgängersatzes, d. h. des letzten programmierten Verfahrssatzes mit aktivem `CUTCNON`, und diesem Punkt wird ein linearer Satz eingefügt.

Kreissätze, bei denen die Kreisebene senkrecht auf der Korrekturebene steht (vertikale Kreise), werden so behandelt, als ob in ihnen `CUTCNON` programmiert wäre. Diese implizite Aktivierung der Korrekturunterdrückung wird im ersten Verfahrssatz, der eine Verfahrbewegung in der Korrekturebene enthält und der kein derartiger Kreis ist, automatisch rückgängig gemacht. Vertikale Kreise in diesem Sinne können nur beim Umfangfräsen auftreten.



**Beispiel:**

```
N10 ; Definition des Werkzeugs d1
N20 $TC_DP1[1,1]= 110 ; Typ
N30 $TC_DP6[1,1]= ; Radius
N40
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000
N60
N70 X20 G42 NORM
N80 X30
N90 Y20
N100 X10 CUTCONON ; Einschalten der Korrekturunterdrückung
N110 Y30 KONT ; Beim Ausschalten der Konturunterdrückung
; gegebenenfalls Umfahungskreis einfügen
N120 X-10 CUTCONOF
N130 Y20 NORM ; Kein Umfahungskreis beim Ausschalten der
; WRK
N140 X0 Y0 G40
N150 M30
```

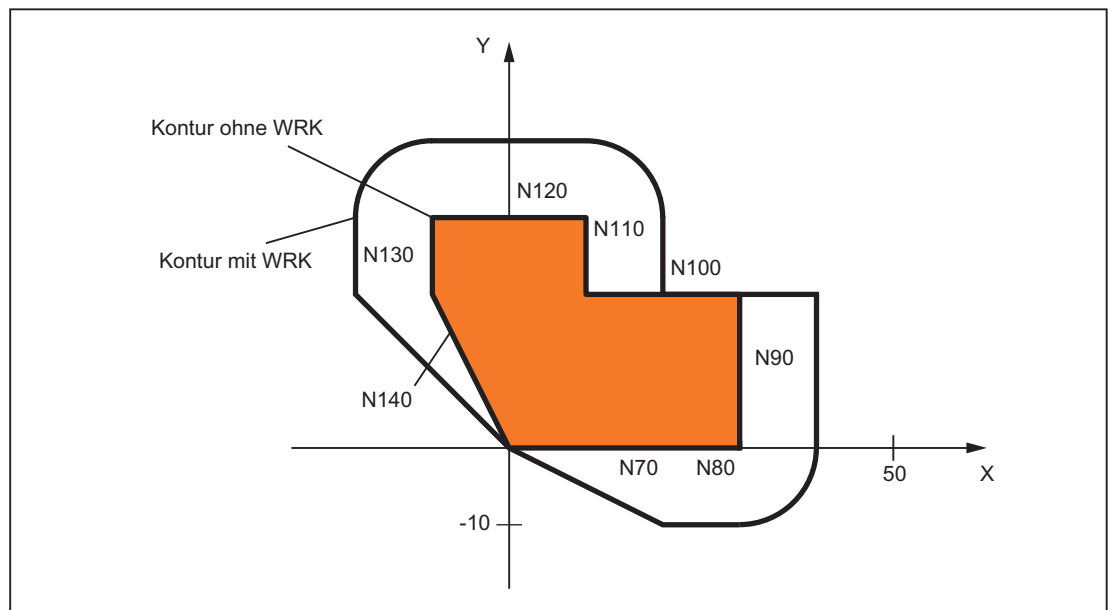


Bild 17-34 Beispielprogramm zur Konturunterdrückung

## Sonderfälle

- Ist keine Werkzeugradiuskorrektur aktiv ( $G_{40}$ ), hat `CUTCNON` keine Wirkung. Es wird kein Alarm erzeugt. Der G-Code bleibt jedoch aktiv.  
Dies ist dann von Bedeutung, wenn in einem späteren Satz die Werkzeugradiuskorrektur mit  $G_{41}$  bzw. mit  $G_{42}$  eingeschaltet werden soll.
- Ein Wechsel des G-Codes in der 7. G-Code-Gruppe (Werkzeugradiuskorrektur;  $G_{40}$  /  $G_{41}$  /  $G_{42}$ ) bei aktivem `CUTCNON` ist zulässig. Ein Wechsel nach  $G_{40}$  wird sofort wirksam.  
Die Verschiebung, mit der die Vorgängersätze verfahren wurden, wird dabei herausgefahren.
- Wird `CUTCNON` oder `CUTCNOF` in einem Satz ohne Verfahrbewegung in der aktiven Korrektorebene programmiert, so wird die Wirksamkeit bis zum nächsten Satz mit einer solchen Verfahrbewegung verzögert.
- Wird `CUTCNON` bei aktiver Werkzeugradiuskorrektur programmiert und vor Programmende nicht mehr aufgehoben, so werden die betroffenen Verfahrätze mit dem letzten gültigen Offset verfahren.  
Gleiches gilt bei erneuter Programmierung von  $G_{41}$  bzw.  $G_{42}$  im letzten Verfahratz eines Programms.
- Wird die Werkzeugradiuskorrektur mit  $G_{41}$  bzw.  $G_{42}$  eingeschaltet und ist gleichzeitig bereits `CUTCNON` aktiv, wird die Aktivierung der Korrektur bis zum ersten Verfahratz mit `CUTCNOF` verzögert.
- Beim Wiederanfahren an die Kontur mit `CUTCNOF` wird die 17. G-Code-Gruppe (An- und Abfahrverhalten bei Werkzeugkorrektur; `NORM` / `KONT`) ausgewertet, d. h. bei `KONT` wird gegebenenfalls ein Umfahrungskreis eingefügt. Ein Umfahrungskreis wird unter den gleichen Bedingungen wie bei der Aktivierung der Werkzeugradiuskorrektur mit  $G_{41}$  bzw.  $G_{42}$  eingefügt.
- Die Zahl der Sätze mit unterdrückter Werkzeugradiuskorrektur ist begrenzt:  
`MD20252 $MC_CUTCOM_MAXNUM_SUPPR_BLOCKS` (Maximale Satzzahl mit Korrekturunterdrückung)  
Wird sie überschritten, wird die Bearbeitung mit einer Fehlermeldung abgebrochen.  
Die Begrenzung ist notwendig, weil beim Repositionieren die interne Satzbearbeitung im letzten Satz vor `CUTCNON` wieder aufgenommen werden muss.
- Das Verhalten bei erneuter Programmierung von  $G_{41}$  bzw.  $G_{42}$  bei bereits aktiver Werkzeugradiuskorrektur ist ähnlich wie bei der Korrekturunterdrückung.  
Es gelten folgende Abweichungen:
  - Es sind nur Linearsätze zugelassen
  - Ein einzelner Verfahratz, der  $G_{41}$  bzw.  $G_{42}$  enthält, wird so modifiziert, dass er im Versatzpunkt des Startpunktes des Folgesatzes endet. Dadurch entfällt das Einfügen eines Zwischensatzes. Gleiches gilt für den letzten einer Folge von Verfahrätzen, von denen jeder  $G_{41}$  bzw.  $G_{42}$  enthält.
  - Das erneute Anfahren der Kontur erfolgt unabhängig vom G-Code der 17. Gruppe (An- und Abfahrverhalten bei Werkzeugkorrektur; `NORM` / `KONT`) immer mit `NORM`.

- Wird  $G_{41}$  /  $G_{42}$  mehrfach in aufeinander folgenden Verfahrssätzen programmiert, so werden alle Sätze bis auf den letzten wie bei `CUTCNON` bearbeitet.
- Die Art der Konturunterdrückung wird nur im ersten Verfahrssatz einer Folge von Sätzen ausgewertet.

Ist im ersten Satz sowohl `CUTCNON` als auch  $G_{41}$  bzw.  $G_{42}$  programmiert, wird das Verhalten beim Deaktivieren der Konturunterdrückung von `CUTCNON` bestimmt.

Ein Wechsel von  $G_{41}$  nach  $G_{42}$  bzw. umgekehrt kann in diesem Fall sinnvoll dazu verwendet werden, die Korrekturseite (links bzw. rechts von der Kontur) beim Wiederanfahren zu wechseln.

Ein Wechsel der Korrekturseite ( $G_{41}$  /  $G_{42}$ ) kann bei aktiver Konturunterdrückung auch in einem späteren Satz programmiert werden.

- Für alle Sätze mit aktiver Konturunterdrückung ist die Kollisionsüberwachung bzw. Flaschenhalserkennung ausgeschaltet.

## 17.5.11 Alarmverhalten

### Alarm im Vorlauf

Tritt im Vorlauf ein Alarm der Werkzeugradiuskorrektur auf, wird die Bearbeitung im Hauptlauf am nächsten erreichbaren Satzende angehalten, d. h. in der Regel am Ende des aktuell interpolierten Satzes (bei aktivem Look-Ahead dann, wenn die Achsen zum Stillstand gekommen sind).

### Alarmer bei Vorlaufstopp und aktiver Werkzeugradiuskorrektur

Die Werkzeugradiuskorrektur benötigt zur Bestimmung des Endpunktes eines Satzes in der Regel mindestens einen nachfolgenden Verfahrssatz (bei Flaschenhälsen auch mehr). Da bei aktivem Vorlaufstopp ein solcher Satz nicht zur Verfügung steht, wird deshalb auf den Offsetpunkt im letzten Satz verfahren. Entsprechend wird im ersten Satz nach einem Vorlaufstopp der Offsetpunkt im Startpunkt angefahren.

Die Kontur, die sich dabei ergibt, kann von der Kontur, die sich ohne Vorlaufstopp ergeben würde, erheblich abweichen. Insbesondere sind Konturverletzungen möglich. Es wurde deshalb folgendes Settingdatum eingeführt:

`SD42480 $MC_STOP_CUTCOM_STOPRE` (Alarmreaktion bei WRK und Vorlaufstopp)

Abhängig von dessen Wert bleibt das Verhalten der Werkzeugradiuskorrektur gegenüber dem bisherigen Stand unverändert, bzw. wird bei einem Vorlaufstopp während aktiver Werkzeugradiuskorrektur ein Alarm ausgegeben und die Programmbearbeitung angehalten.

Der Bediener kann diesen Alarm quittieren und das NC-Programm mit NC-Start unverändert fortsetzen oder mit `RESET` abbrechen.

## 17.5.12 Schnittpunktverfahren für Polynome

### Funktion

Bilden zwei Kurven bei aktiver Werkzeugradiuskorrektur eine Außenecke, wird abhängig vom G-Code der 18. Gruppe (Eckenverhalten bei Werkzeugkorrektur; G450 / G451) und unabhängig vom Typ der beteiligten Kurven (Geraden, Kreise, Polynome):

- ein Kegelschnitt zum Umfahren der Ecke eingefügt  
oder
- die beteiligten Kurven werden so extrapoliert, dass sie einen Schnittpunkt bilden

Wird bei aktivem G451 kein Schnittpunkt gefunden oder ist der Winkel, den die beiden beteiligten Kurven bilden, zu spitz, wird automatisch auf das Einfügeverfahren umgeschaltet.

Das Schnittpunktverfahren für Polynome wird freigegeben mit dem Maschinendatum:

MD20256 \$MC\_CUTCOM\_INTERS\_POLY\_ENABLE (Schnittpunktverfahren für Polynome möglich)

---

### Hinweis

Wird dieses Maschinendatum auf inaktiv gesetzt, dann wird stets ein (gegebenenfalls sehr kurzer) Satz eingefügt (auch wenn die Übergänge nahezu tangential sind). Diese kurzen Sätze führen im G64-Betrieb zwangsläufig zu unerwünschten Geschwindigkeitseinbrüchen.

---

## 17.5.13 G461/G462: Erweiterung An- / Abfahrstrategie

### Funktion

In bestimmten geometrischen Sonderfällen werden gegenüber der bisherigen Realisierung erweiterte An- und Abfahrstrategien beim Aktivieren bzw. Deaktivieren der Werkzeugradiuskorrektur benötigt (siehe folgendes Bild).

---

### Hinweis

Im Folgenden wird immer nur die Situation bei Deaktivieren der Werkzeugradiuskorrektur dargestellt. Das Verhalten beim Anfahren ist dazu völlig symmetrisch.

---

**Beispiel**

```

G42 D1 T1 ; Werkzeugradius 20mm
...
G1 X110 Y0
N10 X0
N20 Y10
N30 G40 X50 Y50

```

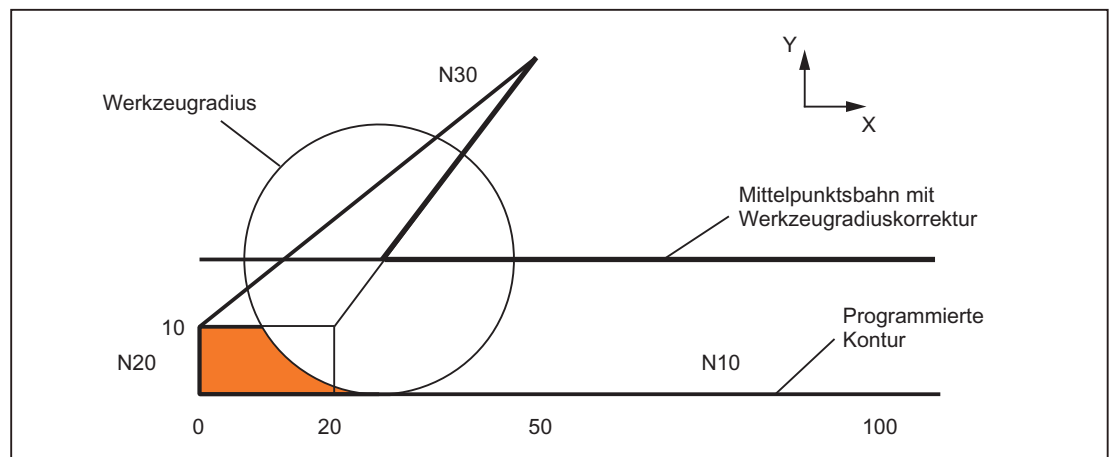


Bild 17-35 Abfahrverhalten bei G460

Der letzte Satz mit aktiver Werkzeugradiuskorrektur (N20) ist so kurz, dass beim aktuellen Werkzeugradius kein Schnittpunkt der Offsetkurve mit dem Vorgängersatz (oder einem weiter davor liegenden Satz) mehr existiert. Es wird deshalb ein Schnittpunkt zwischen den Offsetkurven von Folgesatz und Vorgängersatz gesucht, d. h. im Beispiel zwischen N10 und N30. Die für den Abfahrsatz verwendete Kurve ist dabei keine echte Offsetkurve, sondern eine Gerade vom Offsetpunkt im Endpunkt des Satzes N20 zum programmierten Endpunkt von N30. Wird ein Schnittpunkt gefunden, wird er angefahren. Der im Bild farbig dargestellte Bereich wird dann nicht bearbeitet, wie es mit dem verwendeten Werkzeug eigentlich möglich wäre.

**G460**

Bei G460 ist das An- bzw. Abfahrverhalten wie bisher.

## G461

Wenn kein Schnittpunkt des letzten WRK-Satzes mit einem Vorgängersatz möglich ist, wird die Offsetkurve dieses Satzes mit einem Kreis verlängert, dessen Mittelpunkt im Endpunkt des nicht korrigierten Satzes liegt, und dessen Radius gleich dem Werkzeugradius ist.

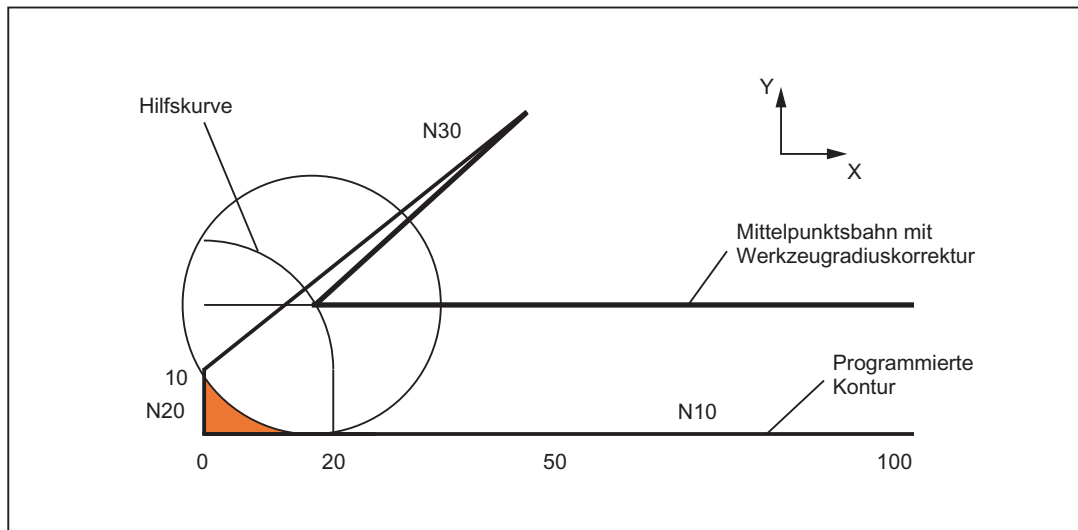


Bild 17-36 Abfahrverhalten bei G461

Die Steuerung versucht, diesen Kreis mit einem der Vorgängersätze zu schneiden. Dabei wird bei aktivem `CDOF` die Suche abgebrochen, wenn ein Schnittpunkt gefunden wurde, d. h. es wird nicht überprüft, ob auch noch Schnittpunkte mit weiter in der Vergangenheit liegenden Sätzen existieren.

Bei aktivem `CDON` wird auch dann, wenn bereits ein Schnittpunkt gefunden wurde, nach weiteren Schnittpunkten gesucht.

Ein so gefundener Schnittpunkt ist der neue Endpunkt eines Vorgängersatzes und der Startpunkt des Deaktivierungssatzes. Der eingefügte Kreis dient nur zur Schnittpunktberechnung und hat selbst keine Verfahrbewegung zur Folge.

**Hinweis**

Wird kein Schnittpunkt gefunden, wird folgender Alarm ausgegeben:

Alarm "10751 Kollisionsgefahr"

## G462

Wenn kein Schnittpunkt des letzten WRK-Satzes mit einem Vorgängersatz möglich ist, wird beim Abfahren mit G462 (Grundstellung) im Endpunkt des letzten Satzes mit Werkzeugradiuskorrektur eine Gerade eingefügt (der Satz wird durch seine Endtangente verlängert):

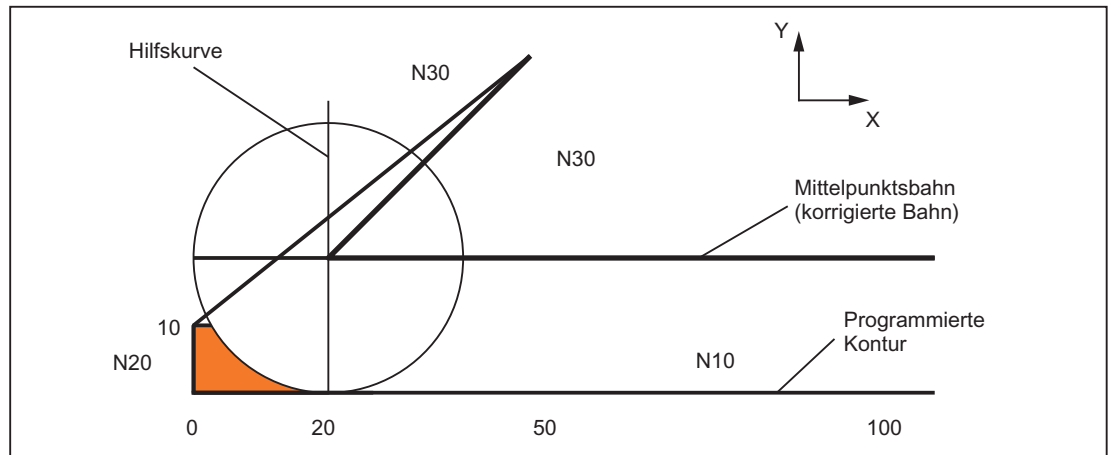


Bild 17-37 Abfahrverhalten bei G462

Die Schnittpunktsuche verläuft dann identisch zu der bei G461.

Bei G462 wird die im Beispielprogramm von N10 und N20 gebildete Ecke nicht soweit ausgeräumt, wie es mit dem verwendeten Werkzeug möglich wäre. Dieses Verhalten kann aber dennoch notwendig sein, wenn die Teilekontur (abweichend von der programmierten Kontur) im Beispiel links von N20 auch bei größeren Werten von y als 10 mm nicht verletzt werden darf.

Ist KONT aktiv (Kontur im Start- oder Endpunkt umfahren), wird unterschieden, ob der Endpunkt vor oder hinter der Kontur liegt.

### Endpunkt vor der Kontur

Liegt der Endpunkt vor der Kontur, ist das Abfahrverhalten gleich wie bei NORM. Diese Eigenschaft ändert sich auch nicht, wenn der letzte Kontursatz bei G451 mit einer Geraden oder einem Kreis verlängert wird. Zusätzliche Umfahrungsstrategien, um eine Konturverletzung in der Nähe des Konturendpunktes zu vermeiden, sind deshalb nicht notwendig.

### Endpunkt hinter der Kontur

Liegt der Endpunkt hinter der Kontur, wird immer abhängig von G450 / G451 ein Kreis bzw. eine Gerade eingefügt. G460 - G462 hat dann keine Bedeutung.

Hat der letzte Verfahrersatz in dieser Situation keinen Schnittpunkt mit einem Vorgängersatz, kann sich nun ein Schnittpunkt mit dem eingefügten Konturelement oder mit dem Geradenstück vom Endpunkt des Umfahungskreises zum programmierten Endpunkt ergeben.

Ist das eingefügte Konturelement ein Kreis (G450) und bildet dieser mit dem Vorgängersatz einen Schnittpunkt, ist dieser gleich dem Schnittpunkt, der sich auch bei NORM und G461 ergeben würde. Im Allgemeinen bleibt jedoch ein zusätzliches Stück des Kreises zu verfahren. Für den linearen Teil des Abfahrersatzes ist keine Schnittpunktberechnung mehr notwendig.

Im zweiten Fall (wenn kein Schnittpunkt des eingefügten Konturelements mit den Vorgängersätzen gefunden wird) wird auf den Schnittpunkt zwischen der Abfahrgeraden und einem Vorgängersatz verfahren.

Es kann sich somit bei aktivem G461 bzw. G462 nur dann ein gegenüber G460 verändertes Verhalten ergeben, wenn entweder NORM aktiv ist, oder das Verhalten bei KONT geometrisch bedingt identisch zu dem bei NORM ist.

---

#### Hinweis

Das Anfahrverhalten ist symmetrisch zum Abfahrverhalten.

Das An- bzw. Abfahrverhalten wird vom Zustand des G-Befehles im An- bzw. Abfahrersatz bestimmt. Das Anfahrverhalten kann deshalb unabhängig vom Abfahrverhalten eingestellt werden.

---

#### Beispiel:

Programm für die Verwendung von G461 beim Anfahren:

```
N10 $TC_DP1[1,1]=120 ; Werkzeugtyp Fraeser
N20 $TC_DP6[1,1]=10 ; Radius
N30 X0 Y0 F10000 T1 D1
N40 Y20
N50 G42 X50 Y5 G461
N60 Y0 F600
N70 X30
N80 X20 Y-5
N90 X0 Y0 G40
N100 M30
```



## 17.6 Orientierbare Werkzeugträger

### 17.6.1 Allgemeines

#### Einführung

Bei einer Klasse von Werkzeugmaschinen ist die Orientierung des Werkzeugs (z. B. **durch Umrüsten**) veränderbar. Im Betrieb ist die einmal eingestellte Orientierung jedoch **fest** und kann insbesondere während des Verfahrens nicht verändert werden. Aus diesem Grund ist für derartige Maschinen eine kinematische Orientierungstransformation (3-, 4- oder 5-Achstransformationen, TRAORI) weder notwendig noch sinnvoll. Es besteht jedoch die Notwendigkeit, die durch eine Orientierungsänderung bedingten Änderungen der Werkzeuglängenkomponenten zu berücksichtigen, ohne den Anwender mit den dazu erforderlichen Berechnungen zu belasten. Die Steuerung übernimmt diese Berechnungen.

#### Verfügbarkeit

Die Funktion "Orientierbarer Werkzeugträger" steht bei SINUMERIK 828D nur für die Fräsvarianten zur Verfügung.

#### Datenvorgaben

Wenn die Steuerung Werkzeugkorrekturen bei orientierbaren Werkzeugträgern berücksichtigen soll, müssen vorliegen:

- Werkzeugdaten (Geometrie, Verschleiß ...)
- Werkzeugträgerdaten (Angaben zur Geometrie des orientierbaren Werkzeugträgers)

#### Werkzeugträgeranwahl

Für die Funktion "Orientierbarer Werkzeugträger" muss ein definierter Werkzeugträger der Steuerung angegeben werden. Dafür existiert der NC-Programmbefehl:

TCARR = m

m: Nummer des Werkzeugträgers

Mit dem Werkzeugträger ist ein Werkzeugträgerdatensatz verbunden, der dessen Geometrie beschreibt.

Die Aktivierung des Werkzeugträgers und seines Datensatzes wirkt sofort, d. h. ab dem nächsten Bewegungssatz.

### Zuordnung Werkzeug / Werkzeugträger

Das bisher aktive Werkzeug wird dem neu angewählten Werkzeugträger zugeordnet.

Aus Sicht der Steuerung sind Werkzeugträgernummern  $m$  und Werkzeugnummern  $T$  frei kombinierbar. In der realen Anwendung können sich Kombinationen aus Bearbeitungsgründen und mechanischen Gründen ausschließen. Die Steuerung prüft nicht auf sinnvolle Kombinationen.

### Beschreibung der Kinematik des WZ-Trägers

Die Kinematik des orientierbaren Werkzeugträgers wird mit insgesamt 33 Parametersätzen beschrieben.

Die Daten des Datensatzes sind durch den Anwender bearbeitbar.

### Orientierbare Werkzeugträger

Beispiel: Kardanischer Werkzeugträger mit zwei Achsen für die Werkzeugorientierung

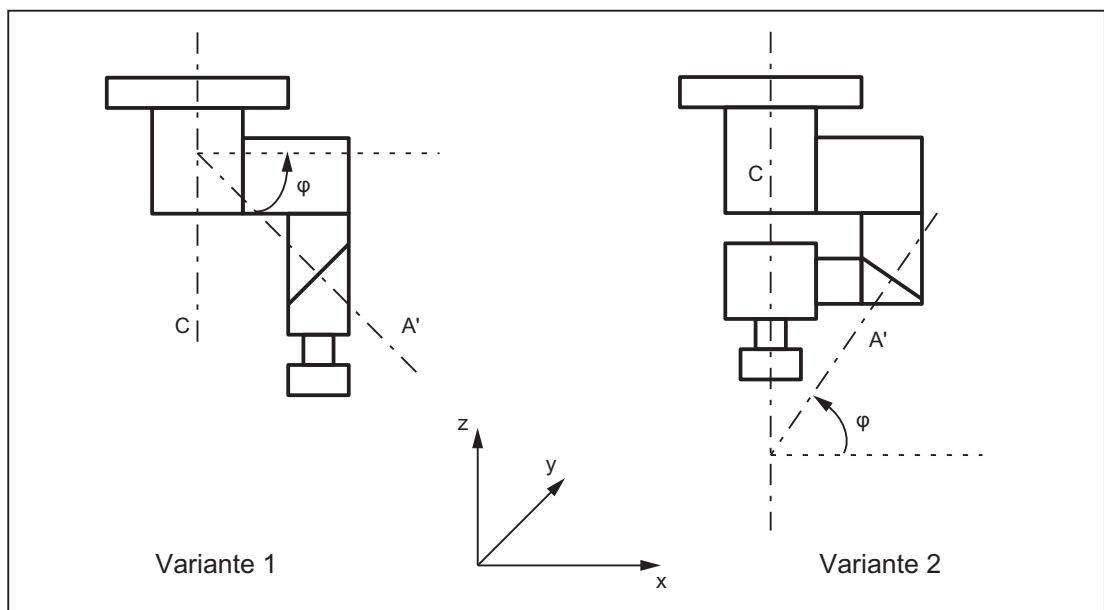


Bild 17-38 Kardanischer Werkzeugträger mit zwei Achsen

## Bearbeitung der Werkzeugträger-Datensätze

Es werden zwei Möglichkeiten angeboten:

- Explizite Eintragung im Werkzeugträger-Datensatz aus dem Teileprogramm
- Automatische Übernahme einiger Werte (Winkel) aus einem Frame

Voraussetzung hierfür ist, dass bei der Werkzeugträgeranwahl zusätzlich `TCOFR` (Tool Carrier Orientation FRame) angegeben wird.

Die für die Werkzeuglängenberechnung verwendete Werkzeugorientierung wird bei einem Werkzeugträgerwechsel neu aus dem zu diesem Zeitpunkt aktiven Frame ermittelt.

## Orientierung in der Z-Richtung

Die G-Funktion `TOFRAME` definiert einen Frame so, dass die Z-Richtung in diesem Frame gleich der aktuellen Werkzeugorientierung ist.

Wenn kein Werkzeugträger oder ein Werkzeugträger ohne Orientierungsänderung aktiv ist, dann ist die Z-Richtung im neuen Frame:

- bei `G17` gleich der alten Z-Richtung.
- bei `G18` gleich der alten Y-Richtung.
- bei `G19` gleich der alten X-Richtung.

## TCOABS bei aktivem Frame

Die absolute Werkzeugträgerorientierung wird eingestellt durch:

`TCOABS` (Tool Carrier Orientation ABSolut)

Die für die Werkzeuglängenkorrektur berücksichtigte Orientierung ist **unabhängig** von der Orientierung des aktiven Frames.

Es kann jeweils nur eine der Anweisungen `TCOABS` oder `TCOFR` gültig sein.

## Wechsel des Frames

Der Anwender kann nach der Anwahl des Werkzeuges den Frame wechseln. Das hat keinen Einfluss auf die Komponenten der Werkzeuglängenkorrektur.

### Winkel in den Werkzeugträgerdaten:

Die in den Werkzeugträgerdaten abgelegten programmierten Drehwinkel werden durch die mit Frames festgelegten Drehwinkel nicht beeinflusst. Beim Wechsel von `TCOFR` nach `TCOABS` werden die ursprünglichen (programmierten) Drehwinkel in den Werkzeugträgerdaten wieder wirksam.

## Werkzeugkorrektur-Typen

Die Werkzeuggradientkorrektur WRK berücksichtigt die aktuelle Werkzeugorientierung, wenn CUT2D oder CUT3DFS aktiv ist.

### Alle anderen Werkzeugkorrektur-Typen

Das sind alle Korrekturtypen der G-Code-Gruppe 22 mit Ausnahme von CUT3DC und CUT3DF. Das Verhalten ist unverändert bezüglich der für die Korrektur verwendeten Ebene. Diese wird unabhängig von der Werkzeug-Orientierung aus dem aktiven Frame bestimmt.

Bei CUT2DF und CUT3DF wird die für WRK verwendete Korrektorebene **unabhängig** von der aktuellen Werkzeugorientierung aus dem Frame ermittelt. Es wird die aktive Ebene (G17/G18/G19) berücksichtigt.

### CUT3DC und CUT3DF

3D-Werkzeugkorrektur für das Umfangsfräsen

3D-Werkzeugkorrektur für das Stirnfräsen bei aktiver 5-Achs-Transformation sind von der Funktion "Orientierbare Werkzeugträger" nicht betroffen.

Die Orientierungsinformation wird durch die aktive kinematische 5-Achs-Transformation ermittelt.

## Eingeschränkte Orientierung des Werkzeugträgers

Wird durch den Frame eine Orientierung festgelegt, die mit der definierten Werkzeugträgerkinematik nicht erreicht werden kann, wird ein Alarm ausgegeben.

Folgende Kinematiken können nicht jede Orientierung erreichen:

- wenn die beiden Drehachsen, die zur Definition der Kinematik notwendig sind, nicht aufeinander senkrecht stehen und wenn die Werkzeugachse, die die Werkzeugrichtung definiert, nicht senkrecht auf der zweiten Drehachse steht,  
oder
- wenn weniger als zwei Achsen definiert sind.

### Nicht drehende Werkzeugträger

Die intern verwendete Werkzeugorientierung ist nur von der Grundorientierung des Werkzeugs und der aktiven Ebene (G17 - G19) abhängig.

## Mehrdeutigkeiten

Eine bestimmte, durch Frame vorgegebene Werkzeugorientierung, kann mit zwei Achsen im Allgemeinen durch **zwei** verschiedene Drehwinkelpaare eingestellt werden. Die Steuerung wählt von diesen zwei möglichen Stellungen diejenige aus, bei der die Drehwinkel möglichst nahe an den programmierten Drehwinkeln liegen.

### Ablage von Winkeln in den Werkzeugträgerdaten

In ungefähr allen Fällen, in denen eine Mehrdeutigkeit auftreten kann, ist es erforderlich, die aus dem Frame zu erwartenden Winkel in den Werkzeugträgerdaten abzulegen.

## Parametersätze

Ein vollständiger Parametersatz für einen orientierbaren Werkzeugträger besteht aus 33 Werten.

Folgende Systemvariablen sind verfügbar:

- \$TC\_CARR1 bis \$TC\_CARR33
- \$TC\_CARR34 bis \$TC\_CARR65 zusätzlich zur freien Verfügung des Anwenders und für Feinverschiebungen.

Die Bedeutung der einzelnen Parameter wird im Folgenden unterschieden.

### Kinematik der Maschine:

\$TC\_CARR1 bis \$TC\_CARR20 sowie \$TC\_CARR23

Dabei definieren \$TC\_CARR18 bis \$TC\_CARR20 einen weiteren Vektor  $l_4$ , der zur Beschreibung der Maschine bei aufgelösten Kinematiken (sowohl Werkzeug als auch Werkstück sind drehbar) benötigt wird.

\$TC\_CARR21 und \$TC\_CARR22 enthalten die Kanalachsbezeichner der Rundachsen, auf deren Position gegebenenfalls bei der Bestimmung der Orientierung des orientierbaren Werkzeugträgers zugegriffen wird.

### Kinematiktyp:

\$TC\_CARR23 mittels eines Buchstabens T, P oder M

Beim Kinematiktyp gibt es die folgenden drei Möglichkeiten, wobei Groß- und Kleinschreibung zulässig ist:

T:	Nur das Werkzeug ( <b>T</b> ool) ist drehbar (Basiswert).
P:	Nur das Werkstück ( <b>P</b> art) ist drehbar.
M:	Werkstück und Werkzeug sind drehbar ( <b>M</b> ixed mode).

Jedes andere außer den genannten drei Zeichen führt bei dem Versuch, den orientierbaren Werkzeugträgern zu aktivieren, zu dem Alarm:

Alarm "14153 Kanal %1 Satz %2 Unbekannter Werkzeugtraegertyp: %3"

**Parametrierung der Drehachsen:**

\$TC\_CARR24 bis \$TC\_CARR33

Mit den Systemvariablen in \$TC\_CARR24 bis \$TC\_CARR33 können Offsets, Winkeloffsets, Hirth-Verzahnung sowie Achsgrenzen parametrierbar werden.

---

**Hinweis**

Die Systemvariablen sind sowohl mit als auch ohne aktive Werkzeugverwaltung verfügbar.

---

**Komponenten und Vorbesezung der Kette / Datensatz**

Die Werte \$TC\_CARR1 bis \$TC\_CARR20 und \$TC\_CARR24 - \$TC\_CARR33 im Werkzeugträger-Datensatz sind vom NC-Sprachformat-Typ *REAL*.

Die Werte \$TC\_CARR21 und \$TC\_CARR22 für die Achsbezeichner der ersten Drehachse ( $v_1$ ) und der zweiten Drehachse ( $v_2$ ) sind vom NC-Sprachformat-Typ *AXIS*. Sie sind alle mit Null vorbelegt.

Der Wert \$TC\_CARR23 wird mit dem Großbuchstaben "T" (nur das Werkzeug ist drehbar) vorbelegt.

\$TC\_CARRn[m]

\$TC\_CARR[0]= 0 hat Sonderbedeutung

**Systemvariablen für orientierbare Werkzeugträger**

\$TC\_CARRn[m]

n: Parameter 1...33

m: Nummer des orientierbaren WZ-Trägers 1...Wert des Maschinendatums:  
MD18088 \$MN\_MM\_NUM\_TOOL\_CARRIER (Maximale Anzahl definierbarer  
Werkzeugträger)

Beschreibung	NCK-Variable	Sprach-Format	Vorbelegung
x-Komponente des Offsetvektors $l_1$	\$TC_CARR1	REAL	0
y-Komponente des Offsetvektors $l_1$	\$TC_CARR2	REAL	0
z-Komponente des Offsetvektors $l_1$	\$TC_CARR3	REAL	0
x-Komponente des Offsetvektors $l_2$	\$TC_CARR4	REAL	0
y-Komponente des Offsetvektors $l_2$	\$TC_CARR5	REAL	0
z-Komponente des Offsetvektors $l_2$	\$TC_CARR6	REAL	0
x-Komponente der Drehachse $v_1$	\$TC_CARR7	REAL	0
y-Komponente der Drehachse $v_1$	\$TC_CARR8	REAL	0
z-Komponente der Drehachse $v_1$	\$TC_CARR9	REAL	0
x-Komponente der Drehachse $v_2$	\$TC_CARR10	REAL	0
y-Komponente der Drehachse $v_2$	\$TC_CARR11	REAL	0
z-Komponente der Drehachse $v_2$	\$TC_CARR12	REAL	0
Drehwinkel $\alpha_1$ (in Grad)	\$TC_CARR13	REAL	0
Drehwinkel $\alpha_2$ (in Grad)	\$TC_CARR14	REAL	0
x-Komponente des Offsetvektors $l_3$	\$TC_CARR15	REAL	0
y-Komponente des Offsetvektors $l_3$	\$TC_CARR16	REAL	0
z-Komponente des Offsetvektors $l_3$	\$TC_CARR17	REAL	0
x-Komponente des Offsetvektors $l_4$	\$TC_CARR18	REAL	0
y-Komponente des Offsetvektors $l_4$	\$TC_CARR19	REAL	0
z-Komponente des Offsetvektors $l_4$	\$TC_CARR20	REAL	0
Achsbezeichner der Drehachse $v_1$	\$TC_CARR21	AXIS	0
Achsbezeichner der Drehachse $v_2$	\$TC_CARR22	AXIS	0
Kinematiktyp	\$TC_CARR23	CHAR	T
Offset der Drehachse $v_1$	\$TC_CARR24	REAL	0
Offset der Drehachse $v_2$	\$TC_CARR25	REAL	0
Winkeloffset der Drehachse $v_1$ (Hirthverz.)	\$TC_CARR26	REAL	0
Winkeloffset der Drehachse $v_2$ (Hirthverz.)	\$TC_CARR27	REAL	0
Winkelinkrement Drehachse $v_1$ (Hirthverz.)	\$TC_CARR28	REAL	0
Winkelinkrement Drehachse $v_2$ (Hirthverz.)	\$TC_CARR29	REAL	0
Minimalposition Drehachse $v_1$ (SW-Limit)	\$TC_CARR30	REAL	0
Minimalposition Drehachse $v_2$ (SW-Limit)	\$TC_CARR31	REAL	0
Maximalposition Drehachse $v_1$ (SW-Limit)	\$TC_CARR32	REAL	0
Maximalposition Drehachse $v_2$ (SW-Limit)	\$TC_CARR33	REAL	0

**Systemvariablen für den Anwender und für Feinverschiebungen**

- \$TC\_CARR34 bis \$TC\_CARR40  
Enthalten Parameter, die dem Anwender zur freien Verfügung stehen.
- \$TC\_CARR41 bis \$TC\_CARR65  
Enthalten Feinverschiebungsparameter, die zu den Werten in den Basisparametern addiert werden können. Der einem Basisparameter zugeordnete Feinverschiebungswert ergibt sich, wenn zur Parameternummer der Wert 40 addiert wird.
- \$TC\_CARR47 bis \$TC\_CARR54 sowie \$TC\_CARR61 bis \$TC\_CARR63  
Sind nicht definiert und führen bei dem Versuch, lesend oder schreibend darauf zuzugreifen, zu einem Alarm.

Beschreibung	NCK-Variable	Sprach-Format	Vorbelegung
Werkzeugträgername *	\$TC_CARR34	String[32]	""
Achsname 1 **	\$TC_CARR35	String[32]	""
Achsname 2 **	\$TC_CARR36	String[32]	""
Kennung **	\$TC_CARR37	INT	0
Positionskomponente X **	\$TC_CARR38	REAL	0
Positionskomponente Y **	\$TC_CARR39	REAL	0
Positionskomponente Z **	\$TC_CARR40	REAL	0
x-Komp. Feinversch. des Offsetvektors l <sub>1</sub>	\$TC_CARR41	REAL	0
y-Komp. Feinversch. des Offsetvektors l <sub>1</sub>	\$TC_CARR42	REAL	0
z-Komp. Feinversch. des Offsetvektors l <sub>1</sub>	\$TC_CARR43	REAL	0
x-Komp. Feinversch. des Offsetvektors l <sub>2</sub>	\$TC_CARR44	REAL	0
y-Komp. Feinversch. des Offsetvektors l <sub>2</sub>	\$TC_CARR45	REAL	0
z-Komp. Feinversch. des Offsetvektors l <sub>2</sub>	\$TC_CARR46	REAL	0
x-Komp. Feinversch. des Offsetvektors l <sub>3</sub>	\$TC_CARR55	REAL	0
y-Komp. Feinversch. des Offsetvektors l <sub>3</sub>	\$TC_CARR56	REAL	0
z-Komp. Feinversch. des Offsetvektors l <sub>3</sub>	\$TC_CARR57	REAL	0
x-Komp. Feinversch. des Offsetvektors l <sub>4</sub>	\$TC_CARR58	REAL	0
y-Komp. Feinversch. des Offsetvektors l <sub>4</sub>	\$TC_CARR59	REAL	0
z-Komp. Feinversch. des Offsetvektors l <sub>4</sub>	\$TC_CARR60	REAL	0
Offset der Feinversch. der Drehachse v <sub>1</sub>	\$TC_CARR64	REAL	0
Offset der Feinversch. der Drehachse v <sub>2</sub>	\$TC_CARR65	REAL	0
<b>Anmerkungen:</b>			
*	Mit der Systemvariable \$TC_CARR34 kann ein orientierbarer Werkzeugträger später nicht mehr mit einer Zahl, sondern mit einem Namen bezeichnet werden.		
**	Die Systemvariablen \$TC_CARR35 bis \$TC_CARR40 beziehen sich auf die beabsichtigte Verwendung des orientierbaren Werkzeugträgers innerhalb der Messzyklen und können auch für andere Zwecke verwendet werden.		



## 17.6.2 Kinematische Zusammenhänge und Maschinenaufbau

### Darstellung der kinematischen Kette

Zur Beschreibung der kinematischen Zusammenhänge zwischen einem Bezugspunkt und der Werkzeugspitze wird das Konzept der kinematischen Kette verwendet.

Die Kette gibt alle für den Werkzeugträger-Datensatz erforderlichen Angaben schematisch an. Für den konkreten Fall einer bestimmten Kinematik müssen die entsprechenden Komponenten der Kette mit den realen Vektoren, Längen und Winkeln besetzt werden. Die Kette stellt die maximal zulässige Konstellation dar. Einzelkomponenten können in einfacheren Fällen zu Null werden (z. B. Kinematiken, die nur eine oder überhaupt keine Drehachse haben).

An der Maschine müssen keine Achsen vorhanden sein, die Werkzeug und / oder Werkstück drehen. Die Funktion ist auch dann nutzbar, wenn die Orientierungen z. B. manuell durch Handräder oder Umbau eingestellt werden.

Der Maschinenaufbau wird durch folgende Angaben beschrieben:

- Zwei Drehachsen ( $v_1$  und  $v_2$ ) mit Angabe von jeweils einem Drehwinkel ( $\alpha_1$  bzw.  $\alpha_2$ ), der bei Drehung im Uhrzeigersinn und Blick in Richtung des Drehvektors positiv zählt.
- Bis zu vier Offsetvektoren ( $l_1$  bis  $l_4$ ) für relevante Maschinenabmessungen (Achsabstände, Abstände zu Maschinen- oder Werkzeugreferenzpunkten).

### Nullvektoren

Die Vektoren  $v_1$  bzw.  $v_2$  können Null sein. Der jeweils zugehörige Drehwinkel (explizit programmiert oder aus dem aktiven Frame berechnet) muss dann, da die Richtung der Drehachse nicht definiert ist, ebenfalls Null sein. Ist diese Bedingung nicht erfüllt, wird bei der Aktivierung des Werkzeugträgers ein Alarm ausgegeben.

### Weniger als zwei Drehachsen

Die Möglichkeit, eine Drehachse nicht zu definieren, ist dann sinnvoll, wenn der zu beschreibende Werkzeugträger das Werkzeug nur in einer Ebene drehen kann. Ein sinnvoller minimaler Datensatz kann deshalb aus einem einzigen von Null verschiedenen Eintrag in den Werkzeugträgerdaten bestehen, nämlich einem Wert in einer der Komponenten von  $v_1$  bzw.  $v_2$  zur Beschreibung einer achsparallelen Drehachse, wobei der zugehörige Drehwinkel  $\alpha_1$  bzw.  $\alpha_2$  aus einem Frame bestimmt wird.

### Weitere Sonderfälle

Die beiden Vektoren  $v_1$  bzw.  $v_2$  können kollinear sein. Damit geht jedoch ein Freiheitsgrad der Orientierung verloren, d. h. eine derartige Kinematik ist gleichartig zu einer solchen, bei der nur eine Drehachse definiert ist. Alle möglichen Orientierungen liegen dann auf einem Kegelmantel. Der Kegelmantel entartet zu einer Geraden, wenn zusätzlich auch Werkzeugorientierung  $t$  und  $v_1$  bzw.  $v_2$  kollinear werden, d. h. in diesem Spezialfall ist deshalb eine Orientierungsänderung nicht mehr möglich. Der Kegelmantel entartet zu einer Kreisfläche (d. h. es sind alle Orientierungen in einer Ebene möglich), wenn Werkzeugorientierung  $t$  und  $v_1$  bzw.  $v_2$  aufeinander senkrecht stehen.

Es ist zulässig, dass beide Vektoren  $v_1$  und  $v_2$  Null werden. Eine Orientierungsänderung ist dann allerdings nicht mehr möglich. In diesem Spezialfall wirken die eventuell von Null verschiedenen Längen  $l_1$  und  $l_2$  wie zusätzliche Werkzeuglängenkorrekturen, deren Komponenten in den einzelnen Achsen durch einen Wechsel der Ebene ( $G_{17} - G_{19}$ ) nicht beeinflusst werden.

### Erweiterungen der Kinematikdaten

- Möglichkeit, unmittelbar auf vorhandene Maschinenachsen zuzugreifen, um über die Rundachspalten die Einstellung der Werkzeugträger zu definieren.
- Erweiterung auf Kinematiken mit drehbarem Werkstück bzw. auf Kinematiken mit drehbarem Werkzeug und drehbarem Werkstück.
- Möglichkeit, für die Rundachspalten nur diskrete Werte in einem Raster zuzulassen (Hirth-Verzahnungen).

Die Erweiterungen sind kompatibel zu früheren Softwareständen und umfassen die Kinematikdatensätze von \$TC\_CARR18 bis \$TC\_CARR23.

## Maschinen mit drehbarem Werkzeug

Bei Maschinen mit drehbarem Werkzeug ändert sich an der Definition der Kinematik gegenüber älteren Softwareständen nichts. Insbesondere hat der neu eingeführte Vektor  $l_4$  keine Bedeutung. Ein eventuell von Null verschiedener Inhalt von  $l_4$  wird nicht ausgewertet.

Der Begriff "Orientierbare Werkzeugträger" ist für die neuen Kinematiktypen, bei denen auch der Werkzeuggestisch allein oder auch zusätzlich zum Werkzeug gedreht werden kann, eigentlich nicht mehr angemessen. Er wird jedoch aus Kompatibilitätsgründen beibehalten.

Die kinematischen Ketten zur Beschreibung der Maschine mit drehbarem Werkzeug (allgemeiner Fall) sind im folgenden Bild dargestellt:

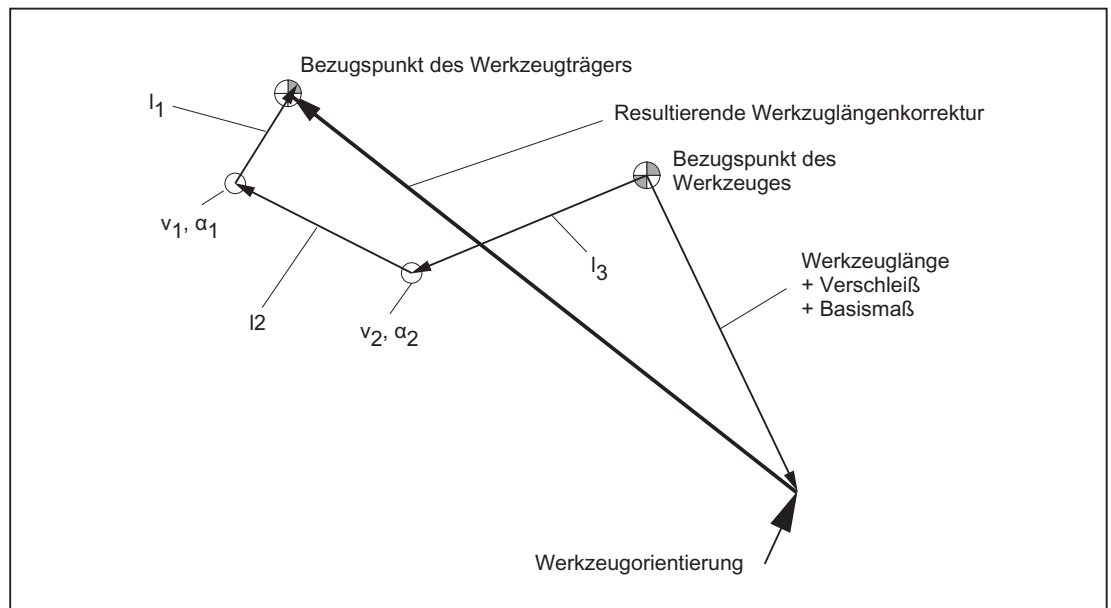


Bild 17-39 Kinematische Kette zur Beschreibung eines Werkzeuges mit Orientierung

Vektoren, die Offsets im drehbaren Kopf beschreiben, sind in Richtung von der Werkzeugspitze zum Bezugspunkt des Werkzeugträgers positiv definiert.

Für Maschinen mit drehbarem Werkzeug wird folgender Kinematiktyp definiert:

\$TC\_CARR23 mittels des Buchstabens T

### Maschinen mit drehbarem Werkstück

Bei Maschinen mit drehbarem Werkstück hat der Vektor  $l_1$  keine Bedeutung. Ein eventuell von Null verschiedener Inhalt wird nicht ausgewertet.

Die zugehörige kinematische Kette der Maschine mit drehbarem Werkstück ist im folgenden Bild dargestellt.

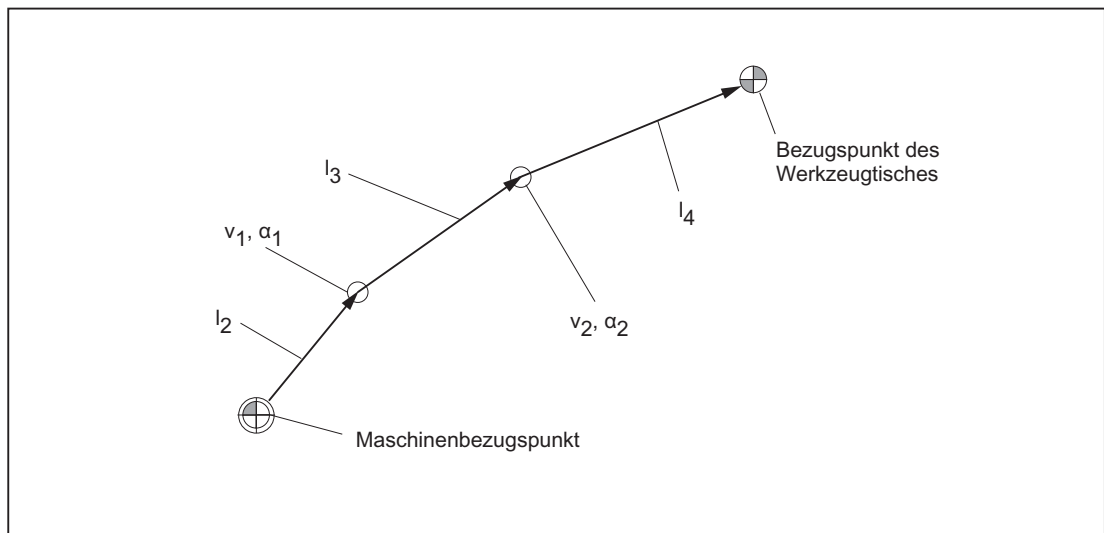


Bild 17-40 Kinematische Kette zur Beschreibung eines drehbaren Werkzeugtisches

Vektoren, die Offsets im drehbaren Tisch beschreiben, sind in Richtung vom Maschinenbezugspunkt zum Tisch positiv definiert.

Für Maschinen mit drehbarem Werkstück wird folgender Kinematiktyp definiert:

\$TC\_CARR23 mittels des Buchstabens P

#### Hinweis

Bei Maschinen mit drehbarem Werkstück wird es in der Regel sinnvoll sein, den Maschinenbezugspunkt und den Bezugspunkt des Tisches identisch zu wählen. Eine derartige Wahl der Bezugspunkte hat den Vorteil, dass sich die Lage des Werkstücknullpunkts in Grundstellung (d.h. bei nicht gedrehten Rundachsen) beim Aktivieren des drehbaren Tisches nicht verändert. Die (offene) kinematische Kette (siehe Bild) wird dann geschlossen.

In diesem Sonderfall gilt deshalb:  $l_2 = - (l_3 + l_4)$

## Maschinen mit aufgelöster Kinematik

Bei Maschinen mit aufgelöster Kinematik (sowohl Werkzeug als auch Werkstück sind drehbar) kann jede der beiden Komponenten nur mit einer Achse gedreht werden.

Die Kinematik des drehbaren Werkzeugs wird mit der ersten Drehachse ( $v_1$ ) und den beiden Vektoren  $l_1$  und  $l_2$  beschrieben, der drehbare Tisch mit der zweiten Drehachse ( $v_2$ ) und den beiden Vektoren  $l_3$  und  $l_4$ . Die beiden kinematischen Teilketten der Maschine mit drehbarem Werkzeug und drehbarem Werkstück sind im folgenden Bild dargestellt.

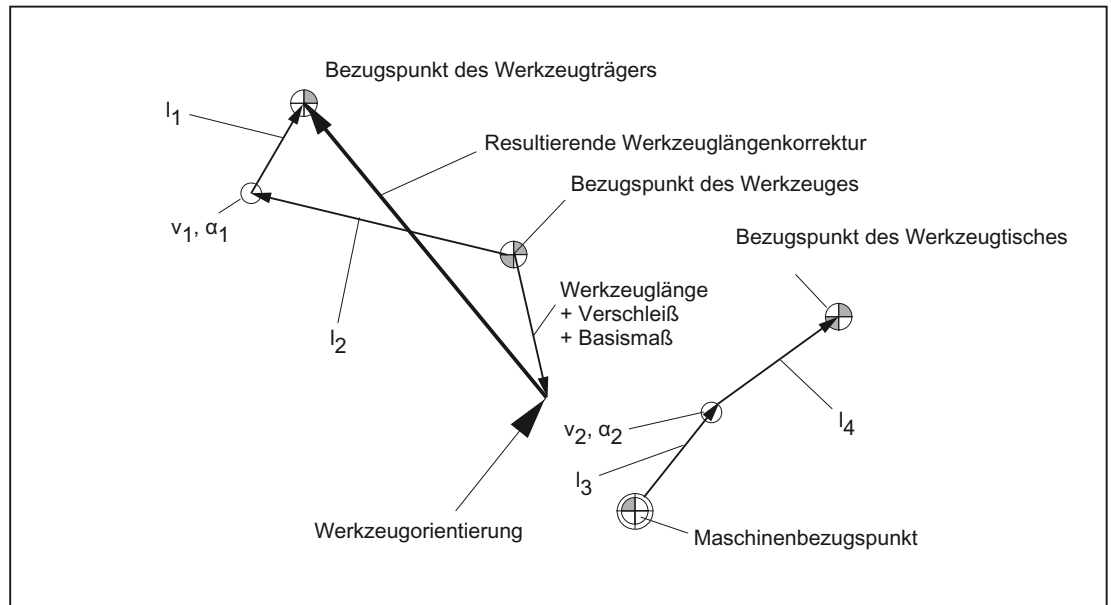


Bild 17-41 Kinematische Kette bei aufgelöster Kinematik

Für Maschinen mit drehbarem Werkzeug und drehbarem Werkstück wird folgender Kinematiktyp definiert:

\$TC\_CARR23 mittels des Buchstabens M (aufgelöste Kinematik)

### Hinweis

Bei Maschinenkinematiken mit aufgelöster Kinematik wird es aus dem gleichen Grund wie bei Maschinen, bei denen nur der Werkzeugschwenktisch drehbar ist, in der Regel sinnvoll sein, den Maschinenbezugspunkt und den Bezugspunkt des Tisches identisch zu wählen. Die (offene) Teilkette zur Beschreibung des Werkzeugschwenktisches (siehe Bild) wird dann geschlossen.

In diesem Sonderfall gilt:  $l_3 = -l_4$

### Werkzeug drehbar Typen T und M

Bei Maschinenkinematiken mit einem drehbaren Werkzeug (Typen T und M) wirkt der orientierbare Werkzeugträgeranteil, der den Werkzeug- oder Kopfanteil (im Gegensatz zum Tischanteil) beschreibt, zusammen mit dem aktiven Werkzeug wie ein neues Gesamtwerkzeug.

### Feinverschiebung

Die Offsetvektoren  $l_1$  bis  $l_4$  sowie die Offsets der Drehachsen  $v_1$  und  $v_2$  können jeweils als Summe eines Basiswertes und einer Feinverschiebung dargestellt werden. Die den Basiswerten zugeordneten Feinverschiebungsparameter ergeben sich, indem **zum Index des Basiswertes der Wert 40 addiert** wird.

#### Beispiel:

Dem Parameter \$TC\_CARR5 ist der Feinverschiebungswert \$TC\_CARR45 zugeordnet.

---

#### Hinweis

Zur Bedeutung der für die Feinverschiebung verfügbaren Systemvariablen \$TC\_CARR41 bis \$TC\_CARR65 siehe:

#### Literatur:

Programmierhandbuch Arbeitsvorbereitung; Werkzeugkorrekturen

---

### Aktivierung

Mit der folgenden Einstellung werden die Feinverschiebungswerte zu den Basiswerten addiert:

SD42974 \$SC\_TOCARR\_FINE\_CORRECTION = 1 (Feinverschiebung TCARR ein/aus)

### Randbedingungen

Der Betrag der zulässigen Feinverschiebung ist begrenzt.

Der maximal zulässige Wert wird festgelegt:

für:	mit dem Maschinendatum:
<ul style="list-style-type: none"><li>die Komponenten der Vektoren <math>l_1</math> bis <math>l_4</math>:</li></ul>	MD20188 \$MC_TOCARR_FINE_LIM_LIN
<ul style="list-style-type: none"><li>die Offsets der beiden Drehachsen <math>v_1</math> und <math>v_2</math>:</li></ul>	MD20190 \$MC_TOCARR_FINE_LIM_ROT

Ein unzulässiger Feinverschiebungswert wird erst erkannt, wenn:

- ein orientierbarer Werkzeugträger aktiviert wird, der solch einen Wert enthält und
- gleichzeitig folgendes Settingdatum gesetzt ist:  
SD42974 \$SC\_TOCARR\_FINE\_CORRECTION

## Beschreibung einer Drehung

Ein Datensatz zur Beschreibung einer Drehung besteht aus jeweils einem Vektor  $v_1$  bzw.  $v_2$  zur Angabe der Richtung der Drehachse im Grundzustand und jeweils einem Winkel  $\alpha_1$  bzw.  $\alpha_2$ . Der Drehwinkel wird bei Drehung im Uhrzeigersinn und in Blickrichtung des Drehvektors positiv gezählt.

Die Bestimmung der beiden Werkzeugträgerwinkel  $\alpha_1$  und  $\alpha_2$  mittels eines Frames ist unabhängig von der aktuell angewählten aktiven Ebene ( $G_{17} - G_{19}$ ).

Die Werkzeugorientierung in Grundstellung (beide Winkel  $\alpha_1$  und  $\alpha_2$  gleich Null) ist (wie im Standardfall auch):

- bei  $G_{17}$  parallel zu Z.
- bei  $G_{18}$  parallel zu Y.
- bei  $G_{19}$  parallel zu X.

## Werkzeugträger Daten zuordnen

### Beispiel an einer Maschine mit drehbaren Werkzeugträger

An einer Maschine mit orientierbarem Werkzeugträger vom Kinematiktyp T ergeben sich am dargestellten Fräskopf die folgenden Einstellungen:

Komponente des Offsetvektors $l_1 =$	(-200, 0, 0)
Komponente des Offsetvektors $l_2 =$	(0, 0, 0)
Komponente des Offsetvektors $l_3 =$	(-100, 0, 0)
Komponente der Drehachse $v_1 =$	(1, 0, 0)
Komponente der Drehachse $v_2 =$	(-1, 0, 1)
Basismaß Bezugspunkt Werkzeug	(0, 0, 250)

### Hinweis

Der Werkzeug-Bezugspunkt für das Basismaß wird durch den Referenzpunkt an der Maschine festgelegt.

Weitere Informationen zu den Bezugspunkten im Arbeitsraum siehe Kapitel "K2: Achsen, Koordinatensysteme, Frames (Seite 721)".

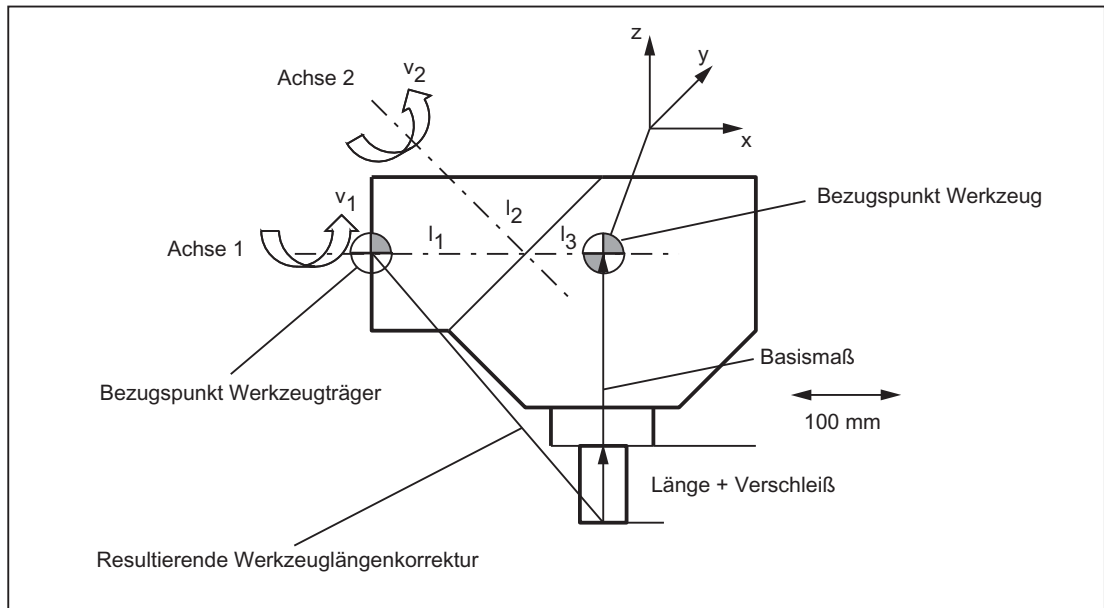


Bild 17-42 Zuordnung der Werkzeugträger-Daten

Für die folgenden Datensatzwerte wurden geeignete Annahmen getroffen:

- Die beiden Drehachsen schneiden sich in einem Punkt.  
Deshalb sind alle Komponenten von  $l_2$  Null.
- Die erste Drehachse liegt in der x-z-Ebene, die zweite Rundachse ist parallel zur x-Achse.  
Durch diese Bedingungen sind die Richtungen von  $v_1$  bzw.  $v_2$  festgelegt (deren Längen sind - solange sie ungleich Null sind - ohne Bedeutung).
- Der Bezugspunkt des Werkzeugträgers liegt vom Schnittpunkt der beiden Rundachsen aus gesehen um 200 mm in negativer x-Richtung.  
Durch diese Bedingung wird  $l_1$  festgelegt.



### Zugehörige Datensatzwerte angeben

Für den abgebildeten Werkzeugträger einer Maschine mit drehbarem Werkzeugträger werden folgende zugehörige Datensatzwerte angegeben:

Beschreibung	NCK-Variable	Wert
x-Komponente des Offsetvektors $l_1$	\$TC_CARR1	- 200
y-Komponente des Offsetvektors $l_1$	\$TC_CARR2	0
z-Komponente des Offsetvektors $l_1$	\$TC_CARR3	0
x-Komponente des Offsetvektors $l_2$	\$TC_CARR4	0
y-Komponente des Offsetvektors $l_2$	\$TC_CARR5	0
z-Komponente des Offsetvektors $l_2$	\$TC_CARR6	0
x-Komponente der Drehachse $v_1$	\$TC_CARR7	1
y-Komponente der Drehachse $v_1$	\$TC_CARR8	0
z-Komponente der Drehachse $v_1$	\$TC_CARR9	0
x-Komponente der Drehachse $v_2$	\$TC_CARR10	-1
y-Komponente der Drehachse $v_2$	\$TC_CARR11	0
z-Komponente der Drehachse $v_2$	\$TC_CARR12	1
Drehwinkel $\alpha_1$ (in Grad)	\$TC_CARR13	0
Drehwinkel $\alpha_2$ (in Grad)	\$TC_CARR14	0
x-Komponente des Offsetvektors $l_3$	\$TC_CARR15	-100
y-Komponente des Offsetvektors $l_3$	\$TC_CARR16	0
z-Komponente des Offsetvektors $l_3$	\$TC_CARR17	0

### Erklärungen

Bei der im Beispiel gewählten Werkzeugträgerkinematik bilden die beiden Drehachsen einen Winkel von  $45^\circ$ , so dass die Orientierung nicht jeden beliebigen Wert annehmen kann. Konkret sind in diesem Beispiel keine Orientierungen mit negativer X-Komponente darstellbar.

x-Komponente des Basismaßes: 0  
y-Komponente des Basismaßes: 0  
z-Komponente des Basismaßes: 250

### Hinweis

Aus der Geometrie eines Werkzeugträgers sind die erforderlichen Daten nicht eindeutig zu bestimmen, d. h. der Anwender hat eine gewisse Wahlfreiheit, wo welche Daten abgelegt werden sollen. So besteht im Beispiel die Möglichkeit, für das Basismaß nur eine z-Komponente bis zur zweiten Achse anzugeben. In diesem Fall wäre dann  $l_2$  nicht mehr Null, sondern enthielte die Komponenten des Abstandes dieses Punktes auf der zweiten Achse von einem weiteren Punkt auf der ersten Achse. Dieser Punkt auf der ersten Achse kann ebenfalls frei gewählt werden. Abhängig von der Wahl dieses Punktes muss dann  $l_1$  so gewählt werden, dass der (ebenfalls frei wählbare) Bezugspunkt erreicht wird.

**Allgemein gilt:** Die Vektorkomponenten, die durch die Drehung einer Achse nicht verändert werden, können beliebig auf die Vektoren "vor" und "nach" der Drehung verteilt werden.

---

## 17.6.3 Schrägbearbeitung mit 3 + 2 Achsen

### Funktionale Beschreibung

Schrägbearbeitung mit 3 + 2 Achsen beschreibt die Erweiterung des Konzepts der orientierbaren Werkzeugträger sowie die Übertragung dieses Konzepts auf Maschinen mit drehbarem Werkzeuggestisch auf solche Maschinen, bei denen die Orientierungen von Werkzeug und Tisch gleichzeitig veränderbar ist.

Die Funktion "Schrägbearbeitung mit 3 + 2 Achsen" dient zur Bearbeitung von Flächen im Raum, die gegenüber den Hauptebenen [X-Y-Ebene ( $G_{17}$ ), Z-X-Ebene ( $G_{18}$ ), Y-Z-Ebene ( $G_{19}$ )] beliebig gedreht sind.

Eine entsprechende Orientierung des Werkzeugs relativ zum Werkstück kann erreicht werden, indem entweder das Werkzeug, das Werkstück oder sowohl Werkzeug als auch Werkstück gedreht wird.

Die Software berechnet dabei automatisch die sich aus den Werkzeuglängen, Hebelarmen und Drehachswinkeln erforderlichen Ausgleichsbewegungen. Dabei wird stets davon ausgegangen, dass zunächst die erforderliche Orientierung eingestellt und während eines Bearbeitungsganges, z. B. Fräsen einer Tasche auf einer schrägen Ebene, nicht mehr verändert wird.

Weiterhin werden folgende 3 Funktionen beschrieben, die bei der Schrägbearbeitung erforderlich sind:

- **Wegprogrammierung** in Richtung der Werkzeugorientierung unabhängig von einem aktiven Frame
- Definition einer **Framedrehung** durch Angabe der Raumwinkel
- Definition des **Rotationsanteils in Werkzeugrichtung** im programmierbaren Frame unter Erhaltung der übrigen Frameanteile

## Abgrenzung zur 5-Achs-Transformation

Wird die Funktionalität benötigt, dass die Werkzeugspitze (TCP-Tool Center Point) bei einer Umorientierung in Bezug auf das Werkstück auch während der Interpolation unverändert bleibt, ist die 5-Achs-Software erforderlich.

Weitere Erläuterungen zur 5-Achs-Transformation siehe:

### Literatur:

Funktionshandbuch Sonderfunktionen; 3- bis 5-Achs-Transformation (F2)

## Festlegung der orientierbaren Werkzeugträger

Der orientierbare Werkzeugträger wird durch eine allgemeine 5-Achs-Kinematik nachgebildet, die durch einen Datensatz im Werkzeugkorrekturspeicher mit insgesamt 33 REAL-Werten beschrieben wird. Bei einem Werkzeugträger (z. B. einem Fräskopf), der über zwei rotatorische Achsen zur Orientierungseinstellung verfügt, sind 31 dieser Werte konstant.

In der aktuellen SW-Version wird ein Datensatz im Werkzeugkorrekturspeicher mit insgesamt 47-Real-Werte beschrieben. Beim Werkzeugträger, der über zwei rotatorische Achsen zur Orientierungseinstellung verfügt, sind 45 dieser Werte konstant.

Die restlichen beiden Werte sind variabel und dienen zur Festlegung der Orientierung. Die konstanten Werte beschreiben Offsets und die Richtungen und Einstellmöglichkeiten der Drehachsen, die variablen Werte beschreiben die Winkel der Drehachsen.

## 17.6.4 Maschine mit drehbarem Werkstück

### Systemvariablen

Bei der Berechnung der aktiven Werkzeuglänge mit `TCOABS` wurde bisher immer auf die in `$TC_CARR13` bzw. `$TC_CARR14` abgelegten Winkel zugegriffen. Das gilt nach wie vor, wenn `$TC_CARR21` bzw. `$TC_CARR22` nicht auf Rundachsen verweisen. Enthält `$TC_CARR21` bzw. `$TC_CARR22` einen Verweis auf eine Rundachse im Kanal, wird als Winkel anstelle des Eintrags in `$TC_CARR13` bzw. `$TC_CARR14` die Achsposition der betreffenden Achse am Beginn des aktuellen Satzes verwendet.

Eine gemischte Betriebsart ist zulässig, d. h. die Winkel können für eine Achse aus dem Eintrag der Systemvariablen `$TC_CARR13` bzw. `$TC_CARR14` und für die andere Achse aus der Position einer Kanalachse ermittelt werden.

Damit ist es möglich, bei Maschinen, bei denen die Achsen innerhalb der NC bekannt sind, die zur Einstellung des orientierbaren Werkzeugträgers dienen, direkt auf deren Positionen zuzugreifen, während dazu bisher beispielsweise die Systemvariable `$AA_IM[Achse]` gelesen und das Ergebnis der Leseoperation in `$TC_CARR13/14` geschrieben werden musste. Damit entfällt insbesondere auch der implizite Vorlaufstopp beim Lesen der Achspositionen.

## MD20180

Die Rundachsposition wird mit ihrem programmierten bzw. berechneten Wert verwendet wenn das Maschinendatum:

MD20180  $\$MC\_TOCARR\_ROT\_ANGLE\_INCR[i] = 0$  (Rundachseninkrement des orientierbaren Werkzeugträgers)

Ist dieses Maschinendatum dagegen ungleich Null, so wird die Position des nächstliegenden Rasterpunktes verwendet, die sich bei geeignetem gewähltem ganzzahligem  $n$  aus der folgenden Gleichung ergibt:

$$\varphi = \$MC\_TOCARR\_ROT\_ANGLE\_OFFSET[i] + n * \$MC\_TOCARR\_ROT\_ANGLE\_INCR[i]$$

Diese Funktionalität wird dann benötigt, wenn die Rundachsen indexiert werden müssen und deshalb nicht beliebige Positionen einnehmen können (z. B. bei Hirth-Verzahnungen). In der Systemvariablen  $\$P\_TCANG[i]$  wird der gerasterte Wert, in der Systemvariablen  $\$P\_TCDIFF[i]$  die Differenz zwischen exaktem und gerastertem Wert geliefert.

## Orientierung Frame TCOFR

Bei  $TCOFR$  (Bestimmung der Winkel aus der durch einen aktiven Frame definierten Orientierung) erfolgt die Rasterung nach der Bestimmung der Winkel aus der aktiven Framedrehung. Ist die verlangte Orientierung aufgrund der Maschinenkinematik nicht einstellbar, wird die Bearbeitung mit einem Alarm abgebrochen. Das gilt auch dann, wenn die Sollorientierung sehr nahe bei einer erreichbaren Orientierung liegt. Insbesondere kann in einer solchen Situation der Alarm durch die Winkelraasterung nicht verhindert werden.

## TCARR Frameverschiebung

Eine Frameverschiebung als Folge eines Werkzeugträgerwechsels wird sofort bei der Anwahl mit  $TCARR=...$  wirksam. Dagegen wird eine Änderung der Werkzeuglänge nur dann sofort wirksam, wenn ein Werkzeug aktiv ist.

## TCOFR / TCOABS Framedrehung

Eine Framedrehung wird mit der Aktivierung nicht bewirkt, bzw. eine bereits wirksame Drehung wird nicht verändert. Die Stellung der Rundachsen, die für die Berechnung verwendet wird, wird wie im Fall T (nur das Werkzeug ist drehbar) abhängig vom G-Code  $TCOFR / TCOABS$  aus dem Drehanteil eines aktiven Frames bzw. aus den Einträgen  $\$TC\_CARRn$  ermittelt.

Durch die Aktivierung eines Frames verändert sich die Position im Werkstückkoordinatensystem entsprechend, ohne dass es zu einer Ausgleichsbewegung der Maschine kommt. Die Verhältnisse sind im folgenden Bild dargestellt:

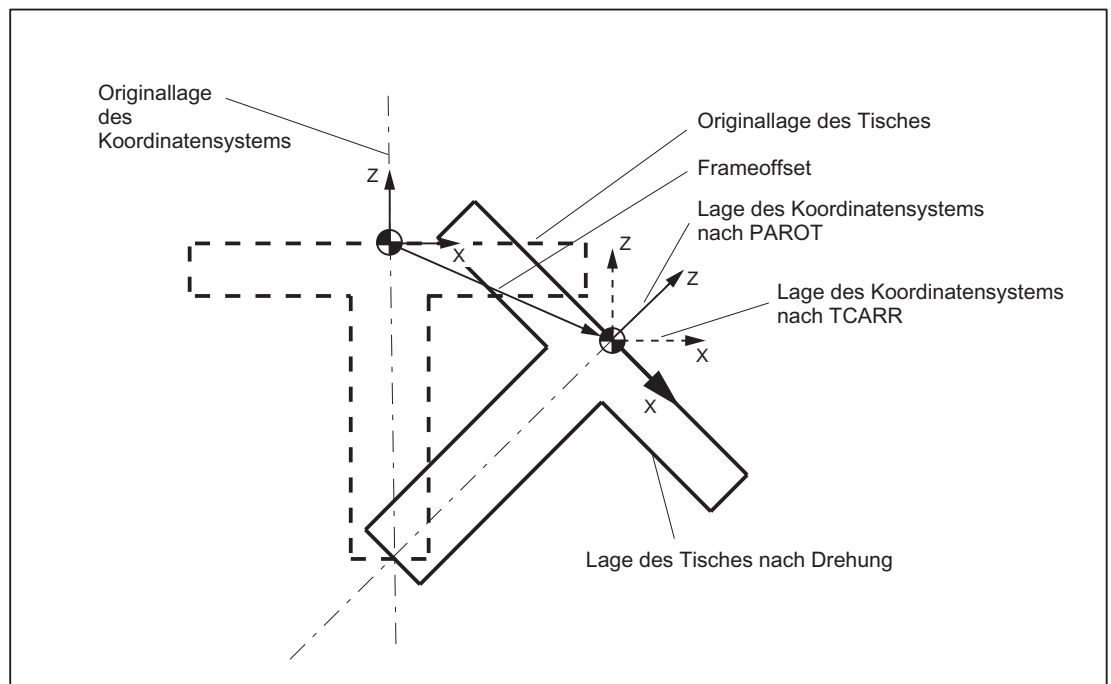


Bild 17-43 Nullpunktverschiebung bei Aktivierung eines drehbaren Werkzeugtisches mit  $TCARR$

## Beispiel

Bei der Maschine im Bild zeigt die Drehachse des Tisches in die positive Y-Richtung. Der Tisch ist um +45 Grad gedreht. Mit  $PAROT$  wird ein Frame definiert, der ebenfalls eine Drehung von 45 Grad um die Y-Achse beschreibt. Das gegenüber der Außenwelt nicht gedrehte Koordinatensystem (im Bild mit "Lage des Koordinatensystems nach  $TCARR$ " gekennzeichnet) ist gegenüber dem mitgeführten Koordinatensystem (Lage nach  $PAROT$ ) aber um -45 Grad gedreht. Wird dieses Koordinatensystem z. B. mit  $ROT Y-45$  definiert und anschließend der Werkzeugträger bei aktivem  $TCOFR$  angewählt, wird für die Drehachse des Werkzeugträgers ein Winkel von +45 Grad ermittelt.

## Drehbarer Tisch

Bei drehbaren Tischen (Kinematiktypen P und M) hat die Aktivierung mit `TCARR` zunächst ebenfalls keine Drehung des Koordinatensystems zur Folge (siehe Bild), d.h. der Nullpunkt des Koordinatensystems verschiebt sich zwar in Bezug auf die Maschine und bleibt fest in Bezug auf den Nullpunkt des Werkstücks, die Orientierung bleibt jedoch unverändert raumfest.

## Aktivierung Kinematiktyp P und M

Bei Kinematiken des Typs P und des Typs M wird bei der Anwahl eines Werkzeugträgers ein additiver Frame aktiviert (Tischoffset des orientierbaren Werkzeugträgers), der die Verschiebung des Nullpunktes als Folge der Drehung des Tisches berücksichtigt.

Die Nullpunktverschiebung kann in ein eigenes Systemframe `$P_PARTFR` geschrieben werden. Dazu muss das Bit 2 gesetzt werden im Maschinendatum:

MD28082 `$MC_MM_SYSTEM_FRAME_MASK` (Systemframes (SRAM))

Das durch das folgende Maschinendatum bezeichnete Basisframe wird dann für die Nullpunktverschiebung nicht mehr benötigt:

MD20184 `$MC_TOCARR_BASE_FRAME_NUMBER` (Nummer des Basisframes für Aufnahme des Tischoffsets)

## Aktivierung Kinematiktyp M

Bei Kinematiken des Typs M (Werkzeug und Tisch sind jeweils um eine Achse drehbar), bewirkt die Aktivierung eines Werkzeugträgers mit `TCARR` gleichzeitig eine entsprechende Änderung der effektiven Werkzeuglänge (falls ein Werkzeug aktiv ist) und der Nullpunktverschiebung.

## Drehungen

Je nach Bearbeitungsaufgabe muss beim Einsatz drehbarer Werkzeugträger bzw. Tische nicht nur eine Nullpunktverschiebung (sei es als Frame oder als Werkzeuglänge), sondern auch eine Drehung berücksichtigt werden. Die Aktivierung eines orientierbaren Werkzeugträgers führt jedoch in keinem Fall unmittelbar zu einer Drehung des Koordinatensystems.

## TOROT

Ist nur das Werkzeug drehbar, kann ein Frame mittels `TOFRAME` bzw. `TOROT` definiert werden, dessen Z-Achse in Werkzeugrichtung zeigt.

## PAROT

Wird ein Koordinatensystem benötigt, das in Bezug auf das Werkstück fest ist, d. h. gegenüber der Originallage nicht nur verschoben sondern auch entsprechend der Tischdrehung gedreht ist, so kann analog zur Situation bei drehbarem Werkzeug mit `PAROT` eine entsprechende Drehung aktiviert werden.

Mit `PAROT` bleiben Translationen, Skalierungen und Spiegelungen im aktiven Frame erhalten, der Rotationsanteil wird jedoch durch den Rotationsanteil eines orientierbaren Werkzeugträgers, der dem Tisch entspricht, gedreht.

`PAROT` und `TOROT` berücksichtigen in den Fällen, in denen der Tisch bzw. das Werkzeug mit zwei Drehachsen orientiert werden, die gesamte Orientierungsänderung. Bei gemischten Kinematiken wird jeweils nur der entsprechende, durch eine Drehachse verursachte Anteil berücksichtigt. Damit ist es z. B. bei der Verwendung von `TOROT` möglich, ein Werkstück so zu drehen, dass eine schräge Ebene parallel zur raumfesten X-Y-Ebene liegt, bei der Bearbeitung aber eine Drehung des Werkzeugs zu berücksichtigen ist, wenn beispielsweise Bohrungen angebracht werden müssen, die nicht senkrecht zu dieser Ebene sind.

Der Sprachbefehl `PAROT` wird nicht abgelehnt, wenn kein orientierbarer Werkzeugträger aktiv ist. Er bewirkt keine Änderung im programmierten Frame.

---

### Hinweis

Für weitere Erläuterungen zu den Funktionen `TCARR` und `TOROT` sowie `PAROT` bezüglich kanalspezifischer Systemframes siehe Kapitel "K2: Achsen, Koordinatensysteme, Frames (Seite 721)".

---

## 17.6.5 Vorgehen bei Benutzung der orientierbaren Werkzeugträger

### Neuanlegen eines Werkzeugträgers

Die Anzahl der verfügbaren Werkzeugträgerdatensätze in der NCK wird festgelegt mit dem Maschinendatum:

MD18088 \$MN\_MM\_NUM\_TOOL\_CARRIER (Maximale Anzahl definierbarer Werkzeugträger)

1. Der Wert wird wie folgt berechnet:

**MD18088 = "Anzahl TO-Einheiten" \* "Anzahl Werkzeugträgerdatensätze einer TO-Einheit"**

Jeder TO-Einheit werden **MD18088 / "Anzahl der TO-Einheiten"** fest zugeteilt.

---

#### Hinweis

Weitere Erläuterungen zur Definition und Zuordnung einer TO-Einheit durch das Maschinendatum:

MD28085 \$MC\_MM\_LINK\_TOA\_UNIT (Zuordnung einer TO-Einheit zu einem Kanal (SRAM))

#### Literatur:

Funktionshandbuch Erweiterungsfunktionen; Speicherkonfigurationen (S7)

---

2. Nullsetzen der Werkzeugträgerdaten:

Zum Nullsetzen aller Werte aller Datensätze kann der Befehl \$TC\_CARR1[0] = 0 verwendet werden.

Einzelne Werkzeugträgerdatensätze können selektiv mit dem Sprachbefehl `DELTC` bzw. über den PI-Dienst `_N_DELTCAR` gelöscht werden.

3. Zugriff auf die Daten eines Werkzeugträgers:

- Teileprogramm

→ \$TC\_CARRn[m] = Wert

Damit wird der bisherige Wert der Systemvariablen n für den Werkzeugträger m mit dem neuen Wert "Wert" beschrieben.

→ Wert = \$TC\_CARRn[m]

Mit "def real wert" - damit können die Parameter eines Werkzeugträger m gelesen werden, sofern sie bereits definiert wurden (z.B. MD18088 setzen). Andernfalls wird ein Alarm gemeldet.

- BTSS-Schnittstelle

Die Parameter eines orientierbaren WZ-Trägers können über die Variablendienste NCK-HMI (BTSS) mit der Systemvariablen \$P\_TCANG[<n>] gelesen und geschrieben werden.

4. Datensicherung:

Die genannten Systemvariablen werden im Rahmen der allgemeinen NCK-Datensicherung gesichert.



## Werkzeugträger anwählen

Ein Werkzeugträger mit der Nummer  $m$  wird mit dem NC-Programmbefehl  $TCARR = m$  angewählt ( $TCARR$  Tool Carrier).

Mit  $TCARR = 0$  wird ein aktiver Werkzeugträger abgewählt.

### Neues Werkzeug oder neuer Werkzeugträger

Ein neu aktiviertes Werkzeug wird so behandelt, als wäre es auf den aktiven Werkzeugträger montiert.

Ein neuer Werkzeugträger wird sofort bei seiner Programmierung wirksam, ein Wechsel oder eine Neuprogrammierung des aktiven Werkzeugs ist nicht notwendig. Werkzeugträger(-nummer) und Werkzeug(-nummer) sind voneinander unabhängig und können beliebig miteinander kombiniert werden.

## Werkzeugträger aus G-Code der Gruppe 42

Werkzeugorientierung absolut  $TCOABS$  (Tool Carrier Orientation **ABS**olute):

Die Werkzeugorientierung wird explizit bestimmt, wenn die entsprechenden Werte in die Systemvariablen  $\$TC\_CARR13$  bzw.  $\$TC\_CARR14$  eingetragen werden und in der G-Code-Gruppe 42 der G-Code  $TCOABS$  aktiviert wird.

Werkzeugorientierung Frame  $TCOFR$  (Tool Carrier Orientation **FR**ame):

Ebenso kann die Werkzeugorientierung automatisch aus der aktuellen Orientierung eines aktiven Frames bei Werkzeuganwahl bestimmt werden, wenn bei der Werkzeugträgeranwahl in der G-Code-Gruppe 42 einer der folgenden G-Codes aktiv ist:

- $TCOFR$  bzw.  $TCOFRZ$   
Der orientierbare Werkzeugträger wird so eingestellt, dass das Werkzeug in Z-Richtung zeigt.
- $TCOFRX$   
Der orientierbare Werkzeugträger wird so eingestellt, dass das Werkzeug in X-Richtung zeigt.
- $TCOFRY$   
Der orientierbare Werkzeugträger wird so eingestellt, dass das Werkzeug in Y-Richtung zeigt.

`TCOFR` bringt zum Ausdruck, dass bei Bearbeitung auf einer schrägen Fläche implizit die Werkzeugkorrekturen so berücksichtigt werden, als ob das Werkzeug auf dieser Fläche senkrecht stünde.

---

#### Hinweis

Die Werkzeugorientierung ist nicht starr an die Frameorientierung gebunden. Bei aktivem Frame und bei aktivem G-Code `TCOABS` kann ein Werkzeug angewählt werden, bei dem die Orientierung des Werkzeugs von der Orientierung des aktiven Frames unabhängig ist.

Nach der Werkzeuganwahl kann der Anwender den Frame wechseln, welcher die Komponenten der Werkzeuglängenkorrektur nicht beeinflusst. Es ist dann auch nicht mehr sichergestellt, dass das Werkzeug senkrecht auf der Bearbeitungsebene steht. Der Anwender sollte deshalb die Einhaltung der gewollten Werkzeugorientierung auf einer schrägen Fläche vorher kontrollieren.

Die für die Werkzeuglängenberechnung verwendete Werkzeugorientierung wird bei aktivem `TCOFR` usw. bei einem Werkzeugträgerwechsel immer neu aus dem zu diesem Zeitpunkt aktiven Frame ermittelt.

---

### Werkzeugträger aus G-Code der Gruppe 53

Die Werkzeugorientierung kann mit Hilfe der G-Codes der Gruppe 53 (`TOFRAME`, `TOROT` usw.) einen Frame so zu definieren, dass eine Achsrichtung (Z, Y oder X) in diesem Frame gleich der aktuellen Werkzeugorientierung ist.

Für die Werkzeugorientierung ist zum Zeitpunkt des Aufrufs von `TOFRAME` der aktive G-Code der Gruppe 6 (`G17` - `G19`) entscheidend.

Ist kein Werkzeugträger aktiv, oder ist ein Werkzeugträger aktiv, der keine Orientierungsänderung des Werkzeugs bewirkt, dann ist die Z-Richtung im neuen Frame:

- bei `G17` gleich der alten Z-Richtung.
- bei `G18` gleich der alten Y-Richtung.
- bei `G19` gleich der alten X-Richtung.

Bei drehenden Werkzeugträgern werden diese Richtungen entsprechend modifiziert. Entsprechendes gilt für die neuen X- bzw. Y-Richtungen.

Statt `TOFRAME` bzw. `TOROT` kann einer der G-Codes `TOFRAMEX`, `TOFRAMEY`, `TOROTX` oder `TOROTY` verwendet werden. Dann sind die Bedeutungen der Achsen entsprechend vertauscht.

### Gruppenwechsel

Ein Wechsel des G-Codes aus der Gruppe 42 (`TCOABS`, `TCOFR` usw.) bewirkt eine Neuberechnung der relevanten Werkzeuglängenkomponenten.

Die in den Werkzeugträgerdaten abgelegten (programmierten) Drehwinkel werden nicht beeinflusst, so dass bei einem Wechsel von `TCOFR` nach `TCOABS` wieder die ursprünglich in den Werkzeugträgerdaten abgelegten Winkel wirksam werden.

#### **Drehwinkel ( $\alpha_1$ bzw. $\alpha_2$ ) lesen:**

Die aktuell zur Berechnung der Orientierung verwendeten Winkel können über die Systemvariable  $\$P\_TCANG[n]$  mit  $n = 1$  oder  $n = 2$  gelesen werden.

Gibt es zu einer bestimmten Orientierung zwei zulässige Lösungen (ein zweites gültiges Winkelpaar), so kann auf die entsprechenden Werte mit  $\$P\_TCANG[3]$  bzw.  $\$P\_TCANG[4]$  zugegriffen werden. Die Anzahl der gültigen Lösungen 0 bis 2 kann mit  $\$P\_TCSOL$  gelesen werden.

#### **Werkzeugradienkorrektur mit CUT2D oder CUT3DFS:**

Bei der Werkzeugradiuskorrektur wird die aktuelle Werkzeugorientierung berücksichtigt, wenn in der G-Code-Gruppe 22 (Werkzeugkorrekturtyp) entweder `CUT2D` oder `CUT3DFS` aktiv ist.

Bei nicht drehenden Werkzeugträgern ist das Verhalten nur von der aktiven Ebene von G-Code der Gruppe 6 ( $G_{17} - G_{19}$ ) abhängig und somit identisch zum bisherigen Verhalten.

#### **Alle anderen Werkzeugkorrektur Typen:**

Das Verhalten bei allen anderen Werkzeugkorrekturtypen ist unverändert.

Insbesondere wird bei `CUT2DF` und bei `CUT3DF` die für die Werkzeugradiuskorrektur verwendete Korrekturebene unabhängig von der aktuellen Werkzeugorientierung aus dem aktiven Frame ermittelt. Dabei wird die aktive Ebene ( $G_{17} - G_{19}$ ) berücksichtigt, so dass das Verhalten identisch zum bisherigen ist.

Die übrigen beiden G-Codes der Gruppe 22 `CUT3DC` und `CUT3DF` sind durch die Werkzeugträgerfunktionalität nicht betroffen, da die Werkzeugorientierungsinformation in diesen Fällen von der aktiven kinematischen Transformation bereitgestellt wird.

## **Zwei Drehachsen**

Für zwei Drehachsen existieren allgemein zwei Lösungen. Die Auswahl dieser beiden Lösungspaare wird von der Steuerung selbst so getroffen, dass die Orientierungswinkel, die sich aus dem Frame ergeben, möglichst nahe an den vorgegebenen Winkeln liegen.

Für die Vorgabe der Winkel gibt es zwei Möglichkeiten:

1. Verweist  $\$TC\_CARR21$  bzw.  $\$TC\_CARR22$  auf eine Rundachse, so wird als Vorgabe die Position dieser Achse am Anfang des Satzes verwendet, in dem der Werkzeugträger aktiviert wird.
2. Verweist  $\$TC\_CARR21$  bzw.  $\$TC\_CARR22$  nicht auf eine Rundachse, werden die  $\$TC\_CARR13$  bzw.  $\$TC\_CARR14$  enthaltenen Werte verwendet.

## **Beispiel**

Die Steuerung berechnet zunächst einen Winkel 10 Grad für eine Achse. Der vorgegebene Winkel ist 750 Grad. Zum Ausgangswinkel werden deshalb 720 Grad ( $= 2 * 360$  Grad) addiert, so dass der endgültige Winkel 730 Grad ist.

## Rundachsenoffset

Mit Hilfe der beiden systemvariablen \$TC\_CARR24 und \$TC\_CARR25 kann ein Offset der Rundachsen vorgegeben werden. Ein Wert ungleich Null in einem dieser Parameter bedeutet, dass die zugehörige Rundachse in Grundstellung nicht die Position Null, sondern die durch den Parameter angegebene Position hat. Alle Winkelangaben beziehen sich dann auf das um diesen Wert verschobene Koordinatensystem.

Beim Wechsel der Bearbeitungsebene ( $G_{17} - G_{19}$ ) werden nur die Werkzeuglängenkomponenten des aktiven Werkzeugs vertauscht. Die des Werkzeugträgers werden nicht getauscht. Der resultierende Werkzeuglängenvektor wird dann entsprechend dem aktuellen Werkzeugträger gedreht und gegebenenfalls mit den zum Werkzeugträger gehörenden Offsets modifiziert.

Die Bestimmung der beiden Werkzeugträgerwinkel  $\alpha_1$  und  $\alpha_2$  mittels eines Frames ist unabhängig von der aktuell angewählten aktiven Ebene ( $G_{17} - G_{19}$ ).

## Grenzwertlimits

Im Systemvariablensatz (\$TC\_CARR30 bis \$TC\_CARR33) zur Beschreibung des orientierbaren Werkzeugträgers können für jede Drehachse Grenzwinkel (Softwarelimits) angegeben werden. Sind sowohl der Minimal- als auch der Maximalwert Null, werden diese Grenzen nicht ausgewertet.

Ist mindestens einer der beiden Grenzwerte ungleich Null, wird überprüft, ob die zuvor ermittelte Lösung innerhalb der zulässigen Grenzen liegt. Ist das nicht der Fall, wird zunächst versucht, durch Addition oder Subtraktion von Vielfachen von 360 Grad zu der unzulässigen Achsposition eine gültige Einstellung zu erreichen. Ist dies nicht möglich und es existieren zwei verschiedene Lösungen, wird die erste Lösung verworfen und die zweite Lösung verwendet. Die zweite Lösung wird bezüglich der Achsgrenzen wie die erste Lösung behandelt.

Wird die erste Lösung verworfen und stattdessen die zweite Lösung verwendet, werden die Inhalte von \$P\_TCANG[1/2] und \$P\_TCANG[3/4] getauscht, so dass auch in diesem Fall die tatsächlich verwendete Lösung in \$P\_TCANG[1/2] steht.

Die Achsgrenzen werden auch dann überwacht, wenn die Achswinkel nicht berechnet, sondern vorgegeben werden. Dies ist der Fall, wenn bei der Aktivierung eines orientierbaren Werkzeugträgers TCOABS aktiv ist.

## 17.6.6 Programmierung

### Werkzeugträger-Anwahl

Ein Werkzeugträger wird mit **m** Nummer des Werkzeug-Trägers angewählt mit:

`TCARR = m`

### Zugriff auf die Werkzeugträger Datensätze

Aus dem Teileprogramm heraus sind folgende Zugriffe möglich:

Der bisherige Wert des Parameters **n** für den Werkzeugträger **m** wird mit dem neuen "Wert" **beschrieben** mit:

`$TC_CARRn[m] = Wert`

Die Parameter eines Werkzeugträgers **m** können, sofern der Werkzeugträger-Datensatz bereits definiert wurde, **gelesen** werden mit:

`wert = $TC_CARRn[m]` (Wert muss eine `REAL`-Variable sein)

Die Werkzeugträger-Datensatz-Nr muss in dem Bereich liegen der definiert wird durch das Maschinendatum:

MD18088 \$MN\_MM\_NUM\_TOOL\_CARRIER (Gesamtzahl der definierbaren Werkzeugträger-Datensätze)

Für einen NC-Kanal können diese Anzahl geteilt durch die Anzahl der aktiven Kanäle Werkzeugträger-Datensätze definiert werden.

#### **Ausnahme:**

Wenn vom Standard abweichende Einstellungen gewählt werden über das Maschinendatum:

MD28085 \$MC\_MM\_LINK\_TOA\_UNIT (Zuordnung einer TO-Einheit zu einem Kanal)

### Nullsetzen aller Werkzeugträger-Datensätze

Aus dem Teileprogramm heraus können alle Werte aller Werkzeugträger-Datensätze mit einem Kommando gelöscht werden.

`$TC_CARR1[0] = 0`

Vom Anwender nicht gesetzte Werte sind mit Wert=0 vorbelegt.

## Aktivierung

Ein Werkzeugträger wird wirksam, wenn sowohl ein **Werkzeugträger** als auch ein **Werkzeug** aktiviert wurden. Die Anwahl eines Werkzeugträgers allein hat keine Wirkung. Die Wirkung der Anwahl eines Werkzeugträgers ist abhängig vom G-Code  $T_{COABS}$  /  $T_{COFR}$  (modale G-Code-Gruppe Werkzeugträger).

Ein Wechsel des G-Codes in der Gruppe  $T_{COABS}$  /  $T_{COFR}$  bewirkt eine Neuberechnung der Werkzeuglängenkomponenten, wenn ein Werkzeugträger aktiv ist. Bei  $T_{COABS}$  werden die in den Werkzeugträgerdaten abgelegten Werte für die beiden Drehwinkel  $\alpha_1$  und  $\alpha_2$  für die Ermittlung der Werkzeugorientierung verwendet.

Bei  $T_{COFR}$  werden die beiden Winkel aus dem aktuellen Frame ermittelt. Die in den Werkzeugträgerdaten abgelegten Werte bleiben jedoch unverändert erhalten. Diese werden darüber hinaus auch dazu verwendet, die Mehrdeutigkeit, die sich bei der Berechnung der Drehwinkel aus einem Frame ergeben kann, aufzulösen. Dabei werden unter mehreren möglichen Winkeln die gewählt, die von den programmierten Winkeln am wenigsten abweichen.

---

### Hinweis

Weitere Erläuterungen zur Programmierung von Werkzeugkorrekturen mit Werkzeugträgerkinematik und zu den Systemvariablen siehe:

### Literatur:

Programmierhandbuch Arbeitsvorbereitung

---

## 17.6.7 Randbedingungen und Steuerungsverhalten für Orientierungen

### Volle Orientierung

Mit einem gegebenen Datensatz, der eine bestimmte Kinematik beschreibt, sind nur dann alle denkbaren räumlichen Orientierungen darstellbar, wenn:

- die beiden Vektoren  $v_1$  und  $v_2$ , die Drehachsen beschreiben, auch definiert sind (d.h. beide Vektoren müssen ungleich Null sein).
- die beiden Vektoren  $v_1$  und  $v_2$  aufeinander senkrecht stehen.
- die Werkzeugorientierung senkrecht auf der zweiten Drehachse steht.

### Nicht definierte Orientierung

Sind diese Bedingungen nicht erfüllt und wird mit  $T_{COFR}$  eine durch einen aktiven Frame bestimmte nicht erreichbare Orientierung angefordert, wird ein Alarm ausgegeben.

### Abhängigkeiten Vektor / Drehwinkel

Ist ein die Richtung einer Drehachse beschreibender Vektor  $v_1$  oder  $v_2$  gleich Null, muss auch der zugehörige Drehwinkel  $\alpha_1$  bzw.  $\alpha_2$  gleich Null sein. Andernfalls wird ein Alarm ausgegeben. Der Alarm wird erst ausgegeben, wenn der Werkzeugträger aktiv wird, d.h. bei einem Werkzeugträgerwechsel.

### Werkzeugfeinkorrektur kombiniert mit Orientierung

Die Kombination von Werkzeugfeinkorrektur und Werkzeugträgern ist **nicht** zulässig. Die Aktivierung der Werkzeugfeinkorrektur bei aktivem Werkzeugträger und umgekehrt die Aktivierung eines Werkzeugträgers bei aktiver Werkzeugfeinkorrektur führt zu einem Alarm.

### Automatische Werkzeugträger Anwahl, RESET

Bei **RESET** oder bei Programmstart kann automatisch ein Werkzeugträger angewählt werden über das Maschinendatum:

MD20126 \$MC\_TOOL\_CARRIER\_RESET\_VALUE (Wirksamer Werkzeugträger bei RESET)

Er wird analog zu der gesteuerten Anwahl eines Werkzeugs behandelt über das Maschinendatum:

MD20120 \$MC\_TOOL\_RESET\_VALUE (Werkzeuglängenkorr. Hochlauf (RESET/TP-Ende))

Das Verhalten bei **RESET** oder Programmstart wird wie bei Werkzeuganwahl gesteuert über das gleiche Bit 6 in den Maschinendaten:

MD20110 \$MC\_RESET\_MODE\_MASK (Festlegung der Steuerungsgrundstellung nach RESET/TP-Ende)

Oder:

MD20112 \$MC\_START\_MODE\_MASK (Festlegung der Steuerungsgrundstellung bei NC-START)

Für weitere Informationen siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)".

### Ab SW 6.3

War bei der letzten Anwahl vor Reset **TCOABS** aktiv, ist das Verhalten gegenüber den Vorgängerständen unverändert. Ein anderer aktiver G-Codes führt dazu, dass der orientierbare Werkzeugträger mit dem Frame aktiviert wird, der zum letzten Anwahlzeitpunkt vor Reset aktiv war. Dabei werden auch geänderte Werkzeugträgerdaten (\$TC\_CARR...) berücksichtigt. Sind diese Daten unverändert, wird der Werkzeugträger wieder im exakt gleichen Zustand wie vor Reset aktiviert. Wurden nach der letzten Werkzeugträgeranwahl vor Reset Werkzeugträgerdaten verändert, ist eine Anwahl in Übereinstimmung mit dem alten Frame nicht immer möglich. In diesem Fall wird der orientierbare Werkzeugträger entsprechend den zu diesem Zeitpunkt gültigen Werten des G-Codes der Gruppe 42 und der aktiven Frames angewählt.

### MD22530 Hilfsfunktionsausgabe an PLC

Dass bei Anwahl des Werkzeugträgers wahlweise ein konstanter oder ein M-Code, dessen Nummer von der Werkzeugträgernummer abgeleitet wird, ausgegeben wird kann eingestellt werden mit dem Maschinendatum:

MD22530 \$MC\_TOCARR\_CHANGE\_M\_CODE (M-Code bei Werkzeugträgerwechsel)

Für weitere Informationen siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)".

## Werkzeugträgerkinematik

Für die Werkzeugträgerkinematik sind folgende Randbedingungen zu beachten:

- Die Werkzeugorientierung in Grundstellung, beide Winkel  $\alpha_1$  und  $\alpha_2$  sind Null, ist wie im Standardfall auch bei:
  - $G_{17}$  parallel zu Z
  - $G_{18}$  parallel zu Y
  - $G_{19}$  parallel zu X
- Bezüglich der Achsgrenzwerte muss eine zulässige Stellung erreichbar sein.
- Nur wenn die beiden Rundachsen aufeinander senkrecht stehen sind auch beliebige Orientierungen einstellbar.

Zusätzlich muss bei Maschinen, bei denen beide Achsen den Tisch drehen, die Werkzeugorientierung senkrecht auf der ersten Drehachse stehen.

Bei Maschinen mit gemischter Kinematik muss die Werkzeugorientierung senkrecht auf der Achse stehen, die das Werkzeug dreht, d.h. ebenfalls auf der ersten Drehachse.

Für eine **in einem Frame** festgelegte Orientierung gilt:

- Die in einem Frame festgelegte Orientierung muss mit der definierten Werkzeugträgerkinematik erreichbar sein, sonst wird ein Alarm ausgegeben.  
Dieser Fall kann immer dann auftreten, wenn die beiden Drehachsen, die zur Definition der Kinematik notwendig sind, nicht aufeinander senkrecht stehen.  
Dies gilt, wenn weniger als zwei Drehachsen definiert sind und ist der Fall:
  - beim **Kinematiktyp T mit drehbarem Werkzeug**, wenn die Werkzeugachse, die die Werkzeugrichtung definiert, nicht senkrecht auf der **zweiten** steht.
  - beim **Kinematiktyp M und P mit drehbarem Werkstück**, wenn die Werkzeugachse, die die Werkzeugrichtung definiert, nicht senkrecht auf der **ersten** steht.
- Drehachsen, die zum Erreichen einer bestimmten Position einen Frame mit vorgegebener Werkzeugorientierung erfordern, sind nur im Falle einer Drehachse eindeutig bestimmt. Für zwei Drehachsen existieren allgemein zwei Lösungen.
- In allen Fällen, in denen eine Mehrdeutigkeit zu berücksichtigen ist, ist es besonders wichtig, dass in den Werkzeugdaten die ungefähr aus den Framedaten zu erwartenden Winkel abgelegt werden, bzw. dass die Drehachsen in der Nähe der zu erwartenden Positionen stehen.

## Verhalten bei ASUP, REPOS

Der Werkzeugträger kann in einem asynchronen Unterprogramm ASUP gewechselt werden. Bei Fortsetzung des unterbrochenen Programms mit `REPOS` wird bei der Anfahrbewegung der neue Werkzeugträger berücksichtigt und das Programm mit diesem fortgesetzt. Auch hier ist die Behandlung analog zu der des Werkzeugwechsels in einem ASUP.

Für weitere Informationen siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)".



## 17.7 Schneidenden-Modifikation bei drehbaren Werkzeugen

### 17.7.1 Funktion

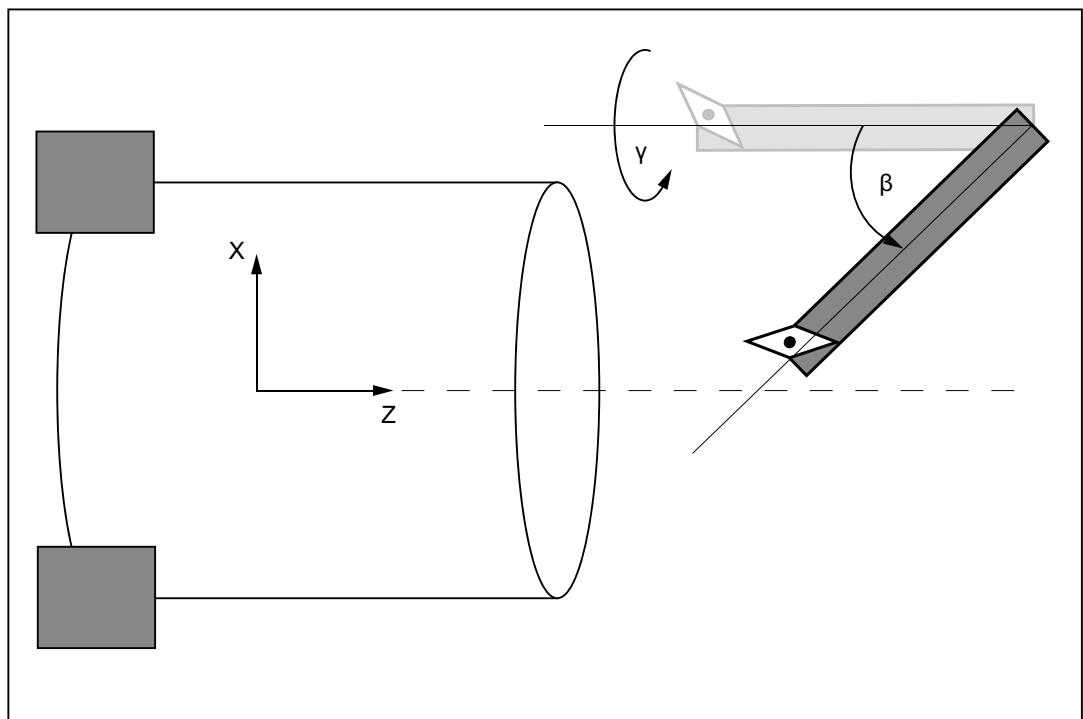
Mit der Funktion "Schneidenden-Modifikation bei drehbaren Werkzeugen" können die veränderten geometrischen Verhältnisse, die sich bei der Drehung von Werkzeugen (vorwiegend Drehwerkzeuge, aber auch Bohr- und Fräswerkzeuge) relativ zum bearbeiteten Werkstück ergeben, berücksichtigt werden.

### 17.7.2 Ermittlung des Drehwinkels

Die aktuelle Drehung des Werkzeugs wird immer aus einem aktuell aktiven orientierbaren Werkzeugträger ermittelt (siehe Kapitel "Orientierbare Werkzeugträger (Seite 1657)").

Die Definition der Drehwinkel des orientierbaren Werkzeugträgers erfolgt dabei in der Regel (aber nicht notwendigerweise) mit dem Befehl `TCOFR` aus einem aktiven Frame. Durch diese Vorgehensweise ist es möglich, die Werkzeugorientierung unabhängig von der tatsächlichen Kinematik, mit der das Werkzeug gedreht wird, immer auf gleiche Art mit Hilfe zweier Winkel zu definieren.

Zur Definition der Werkzeugdrehung werden die beiden maschinenunabhängigen Ausrichtwinkel  $\beta$  (Beta) und  $\gamma$  (Gamma) verwendet.  $\beta$  ist dabei der Drehwinkel um die Applikate (typischerweise eine B-Achse bei  $G_{18}$ ) und  $\gamma$  eine Drehung um die Abszisse (typischerweise eine C-Achse bei  $G_{18}$ ). Zuerst wird die Drehung um  $\gamma$  ausgeführt, anschließend die um  $\beta$ , d. h. die  $\gamma$ -Achse wird von der  $\beta$ -Achse gedreht:



### 17.7.3 Schneidenlage, Schnittrichtung und Winkel bei Drehwerkzeugen

#### Drehwerkzeuge

Unter Drehwerkzeugen werden im Folgenden Werkzeuge verstanden, deren Werkzeugtyp (\$TC\_DP1) Werte im Bereich 500 bis 599 hat. Schleifwerkzeuge (Werkzeugtypen 400 bis 499) sind den Drehwerkzeugen gleichgestellt.

Werkzeuge werden unabhängig vom Werkzeugtyp wie Drehwerkzeuge behandelt, wenn:  
SD42950 \$SC\_TOOL\_LENGTH\_TYPE = 2

#### Schneidenlage und Schnittrichtung

Drehwerkzeuge sind durch ihre Haupt- und Nebenschneiden begrenzt. Der Werkzeug-Parameter "Schneidenlage" wird durch die Lage dieser beiden Schneiden relativ zu den Koordinatenachsen definiert. Die Verhältnisse sind in dem folgenden Bild schematisch dargestellt:

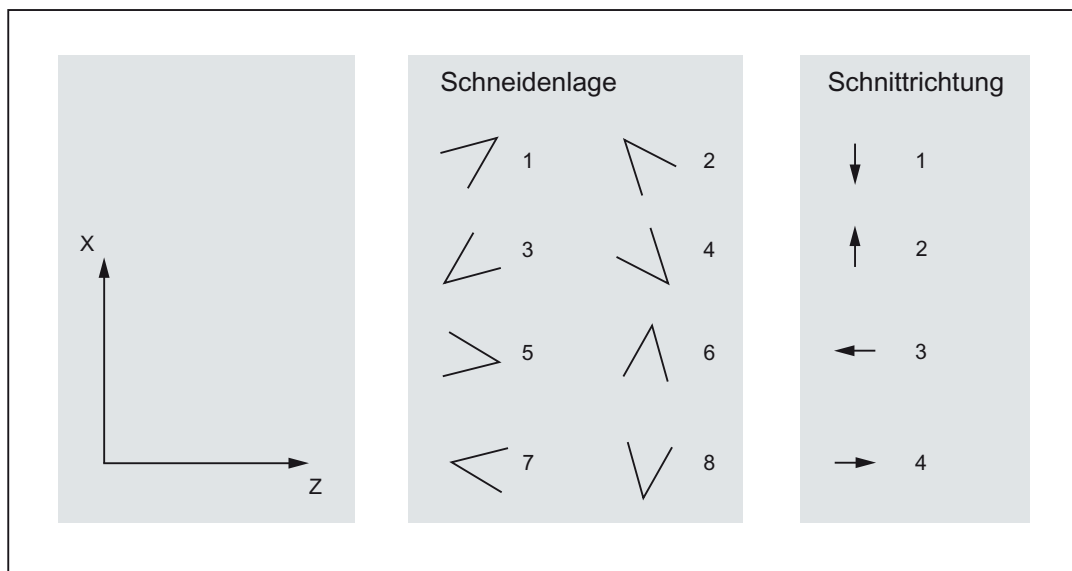


Bild 17-44 Schneidenlage und Schnittrichtung bei Drehwerkzeugen

Die Werte von 1 bis 4 kennzeichnen die Fälle, in denen beide Schneiden im gleichen Quadranten liegen, bei den Werten 5 bis 8 liegen beide Schneiden in benachbarten Quadranten bzw. es liegt eine Koordinatenachse zwischen beiden Schneiden. Die Schneidenlage ist im Werkzeug-Parameter \$TC\_DP2 abgelegt.

Für jedes Drehwerkzeug kann eine Schnittrichtung definiert werden. Diese ist im Werkzeug-Parameter \$TC\_DP11 abgelegt. Sie hat Werte zwischen 1 und 4 und bezeichnet jeweils eine positive oder negative Richtung der Koordinatenachsen:

Wert:	Bedeutung:
1	Ordinate -
2	Ordinate +
3	Abszisse -
4	Abszisse +

Jeder Schneidenlage können zwei verschiedene Schnittrichtungen zugeordnet werden:

Schneidenlage:	1	2	3	4	5	6	7	8
Schnittrichtung:	2, 4	2, 3	1, 3	1, 4	1, 2	3, 4	1, 2	3, 4

### Halterwinkel und Freiwinkel

Im folgenden Bild sind für ein Drehwerkzeug mit Schneidenlage 3 die beiden Winkel (Halterwinkel und Freiwinkel) dargestellt, die zur Beschreibung der Geometrie der Werkzeugschneide notwendig sind. Die Schnittrichtung in diesem Beispiel ist 3, d. h. sie bezeichnet die negative Z-Richtung (= Abszissenrichtung bei G18).

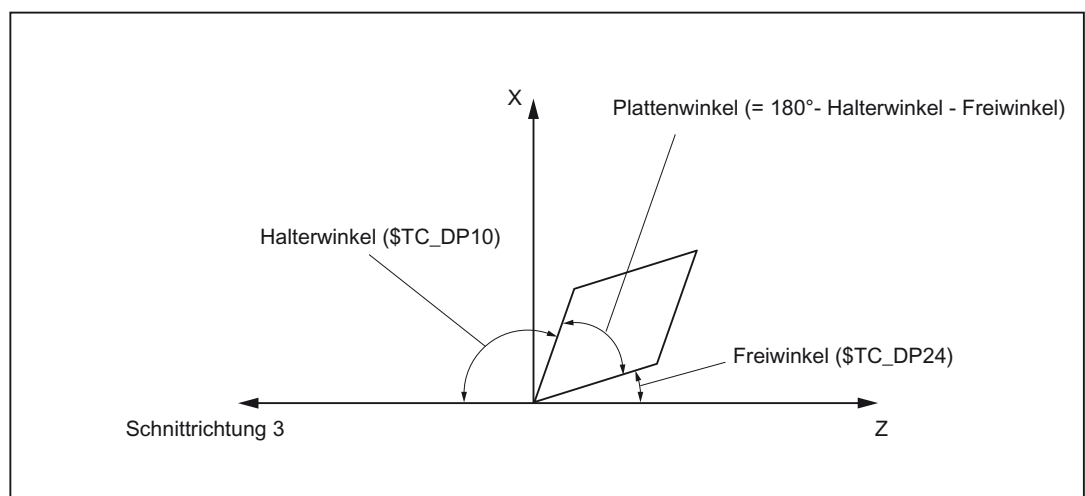


Bild 17-45 Winkel und Schnittrichtung bei einem Drehwerkzeug mit Schneidenlage 3

Die Schnittrichtung gibt die Bezugsrichtung für den Halterwinkel an. Der Freiwinkel ist der zwischen der inversen Schnittrichtung und der benachbarten Schneide (positiv) gemessene Winkel. Halterwinkel und Freiwinkel sind in den Werkzeug-Parametern \$TC\_DP10 bzw. \$TC\_DP24 abgelegt.

---

#### **Hinweis**

Schnittrichtung und Werkzeugwinkel sind nur bei den Schneidenlagen 1 bis 8 von Bedeutung.

---

## 17.7.4 Modifikationen bei der Drehung von Drehwerkzeugen

### Werkzeugorientierung

Drehwerkzeuge sind im Gegensatz zu Fräswerkzeugen nicht rotationssymmetrisch. Das bedeutet, dass zur Beschreibung der Werkzeugorientierung in der Regel 3 Freiheitsgrade bzw. drei rotatorische Achsen notwendig sind. Die Unabhängigkeit von der konkreten Kinematik an der Maschine ist deshalb nur insoweit gegeben, wie die gewünschte Orientierung überhaupt einstellbar ist. Gegebenenfalls kann der dritte Freiheitsgrad durch eine Drehung des Werkstückkoordinatensystems ersetzt werden.

---

#### **Hinweis**

Die Aufteilung der Orientierung in einen Anteil, der durch den orientierbaren Werkzeugträger erzeugt wird, und einen weiteren Anteil, der durch eine Drehung des Koordinatensystems erreicht wird, liegt in der Verantwortung der Applikation. Die Steuerung stellt dafür keine weitere Funktionalität zur Verfügung.

---

### Schneidenform

Wird ein Drehwerkzeug gegenüber der Bearbeitungsebene (d. h. um eine Achse in der Bearbeitungsebene, typischerweise eine C-Achse) um einen Winkel gedreht, der nicht ein Vielfaches von  $180^\circ$  ist, wird die Projektion der (kreisförmigen) Werkzeugschneide in die Bearbeitungsebene zu einer Ellipse. Es wird davon ausgegangen, dass die durch derartige Drehungen entstehenden Abweichungen von der Kreisform so gering sind, dass sie vernachlässigt werden können (Kippwinkel  $< 5^\circ$ ), d. h. die Steuerung geht unabhängig von der Werkzeugorientierung immer von einer kreisförmigen Schneide aus.

Das bedeutet letztlich, dass die Steuerung bezüglich der aktiven Ebene als einzige von der Grundstellung abweichende Einstellung eine Drehung um  $180^\circ$  akzeptiert. Diese Einschränkung gilt nur für die Schneidenform. Die Werkzeuglängen werden bei beliebigen räumlichen Drehungen immer korrekt berücksichtigt.

## Schneidenlage, Schnittrichtung und Winkel

Eine Drehung von  $180^\circ$  um eine Achse in der Bearbeitungsebene bedeutet, dass beim Einsatz des Werkzeugs am gleichen Einsatzort die Spindeldrehrichtung gegenüber dem Einsatz des nicht gedrehten Werkzeugs invertiert werden muss.

Schnittrichtung und Schneidenlage werden ebenso wie der Schneidenbezugspunkt (siehe unten) nicht modifiziert, wenn das Werkzeug um  $\pm 90^\circ$  (mit einer Toleranz von etwa  $1^\circ$ ) aus der Ebene gedreht ist, weil dann die Projektion der Schneide in die aktuelle Ebene nicht definiert ist.

Wird ein Werkzeug in der Ebene gedreht (Drehung um eine Achse senkrecht zur Bearbeitungsebene bzw. um die Y-Achse bei  $G18$ ), wird die Schneidenlage aus den resultierenden Winkeln ermittelt, die sich für Frei- und Halterwinkel ergeben. Sind diese beiden Winkel für das Werkzeug nicht angegeben (d. h.  $\$TC\_DP10$  und  $\$TC\_DP24$  sind beide Null), wird die neue Schneidenlage alleine aus dem Drehwinkel bestimmt. Dabei gilt die Besonderheit, dass sich die Schneidenlage nur in  $90^\circ$ -Schritten ändert, d. h. die Schneidenlage bleibt abhängig vom Grundzustand immer entweder im Wertebereich 1 bis 4 oder im Wertebereich 5 bis 8. Die neue Schneidenlage wird auch dann ausschließlich aus dem Drehwinkel ermittelt, wenn die vorgegebenen Werte für Halterwinkel und Freiwinkel unzulässig sind (negative Werte, resultierender Plattenwinkel negativ oder größer als  $90^\circ$ ). In allen diesen Fällen werden Freiwinkel und Halterwinkel nicht modifiziert.

Die Schnittrichtung wird abhängig von der Drehung so modifiziert, dass der resultierende Freiwinkel kleiner  $90^\circ$  bleibt. Passen originale Schnittrichtung und originale Schneidenlage nicht zusammen wird die Schnittrichtung bei Drehung des Werkzeugs nicht modifiziert (siehe Kapitel "Schneidenlage, Schnittrichtung und Winkel bei Drehwerkzeugen (Seite 1690)").

Der Drehwinkel in der Ebene, wie er aus dem orientierbaren Werkzeugträger ermittelt wurde, steht in den Systemvariablen  $\$P\_CUTMOD\_ANG$  bzw.  $\$AC\_CUTMOD\_ANG$  zur Verfügung. Dieser Winkel ist der Originalwinkel ohne eventuell anschließende Rundungen auf Vielfache von  $45^\circ$  oder  $90^\circ$ .

### Grenzfälle

Haben bei einem Drehwerkzeug Schneidenlage, Schnittrichtung, Frei- und Halterwinkel gültige Werte, so dass sich durch geeignete Drehungen in der Ebene alle Schneidenlagen (1 bis 8) ergeben können, werden in den Fällen, in denen eine der beiden Schneiden (Haupt- bzw. Nebenschneide) um weniger als ein halbes Eingabe-Inkrement ( $0,0005^\circ$  bei einer Eingabe-Feinheit von 3 Nachkommastellen) von einer Koordinatenachse entfernt ist, die Schneidenlagen 1 bis 4 gegenüber den Schneidenlagen 5 bis 8 bevorzugt.

In allen anderen Fällen (Fräswerkzeuge bzw. Drehwerkzeuge ohne gültige Schneidenparameter), in denen Drehungen nur in  $90^\circ$ -Schritten möglich sind, gilt: Ist der Betrag des Drehwinkels kleiner als  $45^\circ + 0,5$  Eingabe-Inkmente (entspricht  $45,0005^\circ$  bei einer Eingabe-Feinheit von 3 Nachkommastellen), werden Schneidenlage und Schnittrichtung nicht verändert, d. h. diese Fälle werden wie Drehungen behandelt, die kleiner  $45^\circ$  sind. Analog dazu werden Drehungen, deren Betrag um weniger als  $45^\circ + 0,5$  Eingabe-Inkmente von  $180^\circ$  abweicht, wie Drehungen im Bereich zwischen  $135^\circ$  und  $225^\circ$  behandelt.

### Schneidenbezugspunkt

Bei Drehwerkzeugen sind einerseits der Schneidenmittelpunkt und andererseits der Schneidenbezugspunkt definiert. Die Lage dieser beiden Punkte zueinander wird dabei durch die Schneidenlage bestimmt.

Der Abstand der beiden Punkte ist für die Schneidenlagen 1 bis 4 gleich dem  $\sqrt{2}$ -fachen des Schneidenradius, für die Schneidenlagen 5 bis 8 gleich dem 1-fachen Schneidenradius. Im ersten Fall liegt der Schneidenbezugspunkt relativ zum Schneidenmittelpunkt in der Bearbeitungsebene auf einer Winkelhalbierenden, im zweiten Fall auf einer Koordinatenachse.

Dreht man das Werkzeug um einen beliebigen Winkel um eine Achse senkrecht zur Bearbeitungsebene, würde sich der Schneidenbezugspunkt mitdrehen, wenn er relativ zum Werkzeug eine feste Position hätte. Die oben genannte Bedingung (Lage auf einer Achse bzw. einer Winkelhalbierenden) wäre dann nicht mehr erfüllt. Dies ist nicht erwünscht. Vielmehr soll der Schneidenbezugspunkt immer so modifiziert werden, dass der Abstandsvektor zwischen Schneidenbezugspunkt und Schneidenmittelpunkt eine der 8 genannten Richtungen hat. Die Schneidenlage muss dazu gegebenenfalls modifiziert werden.

Die Verhältnisse sind beispielhaft im folgenden Bild dargestellt:

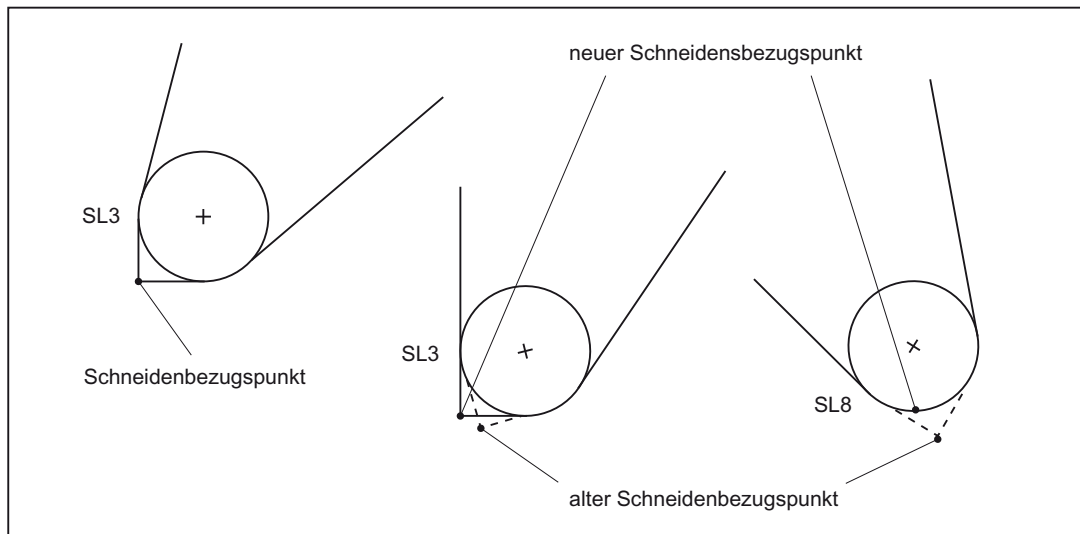


Bild 17-46 Schneidenbezugspunkt und Schneidenlage (SL) bei Werkzeugdrehung

Ein Werkzeug mit der Schneidenlage 3, dem Freiwinkel  $22,5^\circ$  und dem Halterwinkel  $112,5^\circ$  wird gedreht. Bei Drehungen bis  $22,5^\circ$  bleibt die Schneidenlage erhalten, die Lage des Schneidenbezugspunkts relativ zum Werkzeug wird jedoch so korrigiert, dass die relative Lage beider Punkte in der Bearbeitungsebene erhalten bleibt. Bei größeren Drehungen (bis  $67,5^\circ$ ) ändert sich die Schneidenlage auf den Wert 8.

### Hinweis

Da der Schneidenbezugspunkt durch den Werkzeuglängenvektor definiert ist, ändert sich durch die Modifikation des Schneidenbezugspunkts die effektive Werkzeuglänge.

## 17.7.5 Schneidenlage bei Fräs- und Bohrwerkzeugen

### Fräs- und Bohrwerkzeuge

Unter Fräs- und Bohrwerkzeugen werden im Folgenden Werkzeuge verstanden, deren Werkzeugtyp (\$TC\_DP1) Werte im Bereich 100 bis 299 hat.

Werkzeuge werden unabhängig vom Werkzeugtyp wie Fräs- und Bohrwerkzeuge behandelt, wenn:

SD42950 \$SSC\_TOOL\_LENGTH\_TYPE = 1

### Schneidenlage

Für die so definierten Fräs- und Bohrwerkzeuge wird ebenfalls eine Schneidenlage eingeführt, die bei Drehungen entsprechend der nachfolgenden Beschreibung modifiziert wird.

Eine eventuell angegebene Schneidenlage bei Werkzeugen, die keine Fräs- und Bohrwerkzeuge bzw. keine Drehwerkzeuge entsprechend den genannten Definitionen sind, wird nicht ausgewertet.

Die Schneidenlage der Bohr- und Fräswerkzeuge ist wie bei Drehwerkzeugen im Werkzeugparameter \$TC\_DP2 hinterlegt. In Anlehnung an die Definition der Schneidenlage bei Drehwerkzeugen kann dieser Parameter die Werte 5 bis 8 annehmen. Die Schneidenlage gibt dabei die Orientierung (die Richtung der Rotationsachse) des Werkzeugs an:

Schneidenlage	Richtung der Rotationsachse des Werkzeugs
5	Abszisse +
6	Ordinate +
7	Abszisse -
8	Ordinate -

Beispiel:

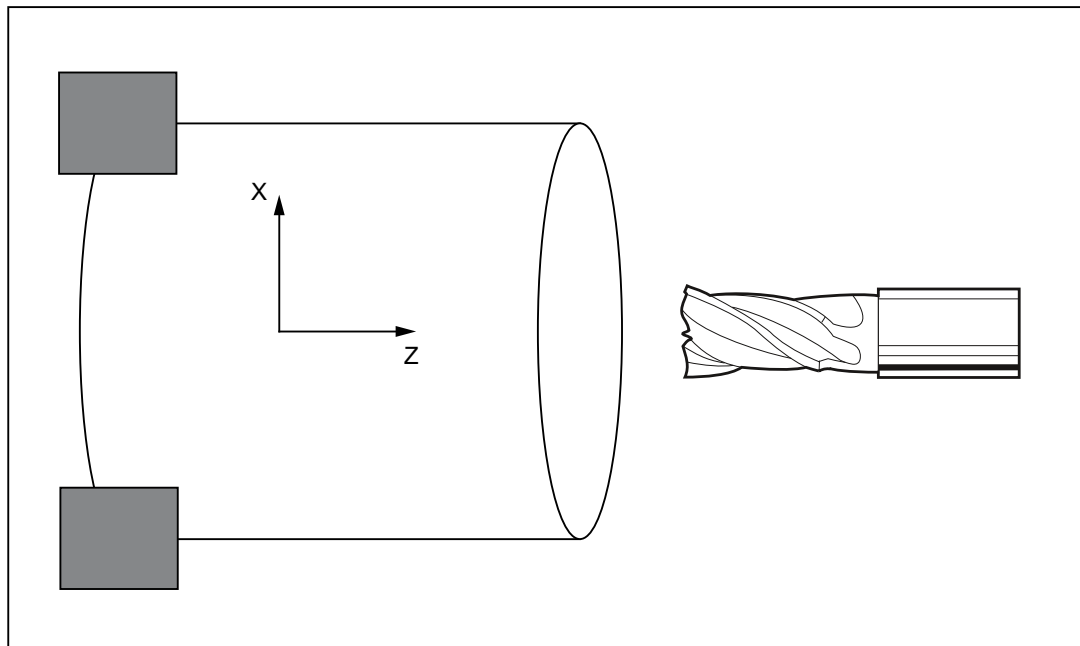


Bild 17-47 Fräs Werkzeug mit Schneidenlage 7

### 17.7.6 Modifikationen bei der Drehung von Fräs- und Bohrwerkzeugen

Bei einer Drehung eines Fräs- oder Bohrwerkzeugs wird die Schneidenlage entsprechend umgerechnet. Schnitttrichtung und Werkzeugwinkel (Freiwinkel bzw. Halterwinkel) sind für Fräs- und Bohrwerkzeuge nicht definiert, so dass die Veränderung der Schneidenlage ausschließlich aus der Drehung abgeleitet wird. Daraus folgt weiter, dass sich bei Fräs- und Bohrwerkzeugen die Schneidenlage immer dann ändert, wenn der Betrag der Drehung gegenüber der Nullstellung größer als  $45^\circ$  wird.



## 17.7.7 Parametrierung

### Verhalten im Fehlerfall

Bei Wirksamwerden der Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen" (durch expliziten Aufruf mit `CUTMOD` oder durch eine Werkzeuganwahl) können verschiedene Fehlerzustände auftreten.

Für jeden dieser möglichen Fehlerzustände kann eingestellt werden, ob der Fehler zu einer Alarmausgabe führen soll, und falls ja, ob ein solcher Alarm nur angezeigt werden soll (Warnhinweis), oder ob zusätzlich die Interpretation des Teileprogramms abgebrochen werden soll. Die Einstellungen erfolgen über das Maschinendatum:

MD20125 \$MC\_CUTMOD\_ERR

Jedem Fehlerzustand sind zwei Bit des Maschinendatums zugeordnet:

Fehlerzustand	Bit	Bedeutung
Für das aktive Werkzeug ist keine gültige Schnitttrichtung definiert.	0	Alarmausgabe bei Fehler "Ungültige Schnitttrichtung"
	1	Programmstopp bei Fehler "Ungültige Schnitttrichtung"
Die Schneidenwinkel (Freiwinkel und Halterwinkel) des aktiven Werkzeugs sind beide Null.	2	Alarmausgabe bei Fehler "Nicht definierte Schneidenwinkel"
	3	Programmstopp bei Fehler "Nicht definierte Schneidenwinkel"
Der Freiwinkel des aktiven Werkzeugs hat einen unzulässigen Wert ( $< 0^\circ$ oder $> 180^\circ$ ).	4	Alarmausgabe bei Fehler "Ungültiger Freiwinkel"
	5	Programmstopp bei Fehler "Ungültiger Freiwinkel"
Der Halterwinkel des aktiven Werkzeugs hat einen unzulässigen Wert ( $< 0^\circ$ oder $> 90^\circ$ ).	6	Alarmausgabe bei Fehler "Ungültiger Halterwinkel"
	7	Programmstopp bei Fehler "Ungültiger Halterwinkel"
Der Plattenwinkel des aktiven Werkzeugs hat einen unzulässigen Wert ( $< 0^\circ$ oder $> 90^\circ$ ).	8	Alarmausgabe bei Fehler "Ungültiger Plattenwinkel"
	9	Programmstopp bei Fehler "Ungültiger Plattenwinkel"
Die Kombination Schneidenlage - Halterwinkel des aktiven Werkzeugs ist unzulässig (bei den Schneidenlagen 1 bis 4 muss der Halterwinkel $\leq 90^\circ$ sein, bei den Schneidenlagen 5 bis 8 muss er $\geq 90^\circ$ sein).	10	Alarmausgabe bei Fehler "Ungültige Kombination Schneidenlage - Halterwinkel"
	11	Programmstopp bei Fehler "Ungültige Kombination Schneidenlage - Halterwinkel"
Unzulässige Drehung des aktiven Werkzeugs (das Werkzeug wurde um $\pm 90^\circ$ (mit einer Toleranz von etwa $1^\circ$ ) aus der aktiven Bearbeitungsebene herausgedreht). Dadurch ist die Schneidenlage in der Bearbeitungsebene nicht mehr definiert.	12	Alarmausgabe bei Fehler "Ungültige Drehung"
	13	Programmstopp bei Fehler "Ungültige Drehung"

## Verhalten bei POWER ON

Die Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen" (CUTMOD) wird bei POWER ON automatisch mit dem Wert initialisiert, der hinterlegt ist im Maschinendatum:

MD20127 \$MC\_CUTMOD\_INIT

Wenn der Wert dieses Maschinendatums "-2" beträgt, dann wird CUTMOD auf den Wert gesetzt, der eingestellt ist im Maschinendatum:

MD20126 \$MC\_TOOL\_CARRIER\_RESET\_VALUE (Wirksamer Werkzeugträger bei RESET)

## 17.7.8 Programmierung

Die Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen" wird mit dem Befehl CUTMOD aktiviert.

### Syntax

CUTMOD=<Wert>

### Bedeutung

CUTMOD	Befehl zum Einschalten der Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen"
<Wert>	Dem CUTMOD-Befehl können folgende Werte zugewiesen werden:
0	Die Funktion ist deaktiviert. Die von den Systemvariablen \$P_AD... gelieferten Werte sind gleich den korrespondierenden Werkzeugparametern.
> 0	Die Funktion wird aktiviert, falls ein orientierbarer Werkzeugträger mit der angegebenen Nummer aktiv ist, d. h. die Aktivierung ist an einen bestimmten orientierbaren Werkzeugträger gebunden. Die von den Systemvariablen \$P_AD... gelieferten Werte sind gegenüber den korrespondierenden Werkzeugparametern abhängig von der aktiven Drehung gegebenenfalls modifiziert. Die Deaktivierung des bezeichneten orientierbaren Werkzeugträgers deaktiviert die Funktion temporär, die Aktivierung eines anderen orientierbaren Werkzeugträgers deaktiviert sie permanent. Im ersten Fall wird die Funktion deshalb bei erneuter Anwahl des gleichen orientierbaren Werkzeugträgers wieder aktiviert, im zweiten Fall ist eine erneute Anwahl notwendig, auch dann wenn zu einem späteren Zeitpunkt der orientierbare Werkzeugträger mit der angegebenen Nummer erneut aktiviert wird. Die Funktion wird durch Reset nicht beeinflusst.
-1	Die Funktion wird immer aktiviert, falls ein orientierbarer Werkzeugträger aktiv ist. Beim Wechsel des Werkzeugträgers oder bei dessen Abwahl und einer späteren erneuten Anwahl muss CUTMOD nicht erneut gesetzt werden.

- 2 Die Funktion wird immer aktiviert, falls ein orientierbarer Werkzeugträger aktiv ist, dessen Nummer gleich der des aktuell aktiven orientierbaren Werkzeugträgers ist.  
Ist kein orientierbarer Werkzeugträger aktiv, ist das gleichbedeutend mit `CUTMOD=0`.  
Ist ein orientierbarer Werkzeugträger aktiv, ist das gleichbedeutend mit der unmittelbaren Angabe der aktuellen Werkzeugträgernummer.
- < -2 Werte kleiner 2 werden ignoriert, d. h. dieser Fall wird so behandelt, als wäre `CUTMOD` nicht programmiert.
- Hinweis:**  
Dieser Wertebereich sollte nicht verwendet werden, weil er für eventuelle spätere Erweiterungen reserviert ist.

---

**Hinweis****SD42984 \$SC\_CUTDIRMOD**

Die über den Befehl `CUTMOD` aktivierbare Funktion ersetzt die über das Settingdatum SD42984 \$SC\_CUTDIRMOD aktivierbare Funktion. Diese Funktion steht jedoch weiterhin unverändert zur Verfügung. Da es aber nicht sinnvoll ist, beide Funktionen parallel zu nutzen, kann sie nur aktiviert werden, wenn `CUTMOD` gleich Null ist.

---

**Wirksamkeit der modifizierten Schneidendaten**

Die modifizierte Schneidenlage und der modifizierte Schneidenbezugspunkt werden bei Programmierung auch für ein bereits aktives Werkzeug sofort wirksam. Eine Werkzeugneuanwahl ist dazu nicht notwendig.

**Einfluss der aktiven Arbeitsebene**

Für die Bestimmung von modifizierter Schneidenlage, Schnittrichtung und Halter- bzw. Freiwinkel ist die Betrachtung der Schneide in der jeweils aktiven Ebene (`G17` - `G19`) maßgebend.

Enthält jedoch das Settingdatum SD42940 \$SC\_TOOL\_LENGTH\_CONST (Wechsel der Werkzeuglängenkomponenten bei Ebenenwechsel) einen gültigen Wert ungleich Null (plus oder minus 17, 18 oder 19), so bestimmt dessen Inhalt die Ebene, in der die relevanten Größen betrachtet werden.

### Systemvariablen

Folgende Systemvariablen stehen zur Verfügung:

Systemvariablen	Bedeutung
\$P_CUTMOD_ANG / \$AC_CUTMOD_ANG	Liefert den (nicht gerundeten) Winkel in der aktiven Bearbeitungsebene, der für die Modifikation der Schneidendaten (Schneidenlage, Schnittrichtung, Freiwinkel und Halterwinkel) bei den mit CUTMOD bzw. \$SC_CUTDIRMOD aktivierten Funktionen zugrunde gelegt wurde. \$P_CUTMOD_ANG bezieht sich auf den aktuellen Zustand im Vorlauf, \$AC_CUTMOD_ANG auf den aktuellen Hauptlaufsatz.
\$P_CUTMOD / \$AC_CUTMOD	Liest den aktuell gültigen Wert, der zuletzt mit dem Befehl CUTMOD programmiert wurde (Nummer des Werkzeugträgers, für den die Schneidendaten-Modifikation aktiviert werden soll). War der letzte programmierte CUTMOD-Wert = -2 (Aktivierung mit dem aktuell aktiven orientierbaren Werkzeugträger), dann wird in \$P_CUTMOD nicht der Wert -2, sondern die Nummer des zum Zeitpunkt der Programmierung aktiven orientierbaren Werkzeugträgers zurückgeliefert. \$P_CUTMOD bezieht sich auf den aktuellen Zustand im Vorlauf, \$AC_CUTMOD auf den aktuellen Hauptlaufsatz.
\$P_CUT_INV / \$AC_CUT_INV	Liefert den Wert TRUE, wenn das Werkzeug so gedreht ist, dass die Spindeldrehrichtung invertiert werden muss. Dazu müssen in dem Satz, auf den sich die jeweilige Leseoperation bezieht, die folgenden vier Bedingungen erfüllt sein: <ol style="list-style-type: none"> <li>1. Es ist ein Dreh- oder Schleifwerkzeug aktiv (Werkzeugtypen 400 bis 599 und / oder SD42950 \$SC_TOOL_LENGTH_TYPE = 2).</li> <li>2. Die Schneidenbeeinflussung wurde mit dem Sprachbefehl CUTMOD aktiviert.</li> <li>3. Es ist ein orientierbarer Werkzeugträger aktiv, der durch den numerischen Wert von CUTMOD bezeichnet wurde.</li> <li>4. Der orientierbare Werkzeugträger dreht das Werkzeug um eine Achse in der Bearbeitungsebene (typischerweise die C-Achse) so, dass die resultierende Normale der Werkzeugschneide gegenüber der Ausgangslage um mehr als 90° (typischerweise 180°) gedreht ist.</li> </ol> Ist mindestens eine der genannten vier Bedingungen nicht erfüllt, ist der Inhalt der Variablen FALSE. Für Werkzeuge, deren Schneidenlage nicht definiert ist, ist der Wert der Variablen immer FALSE. \$P_CUT_INV bezieht sich auf den aktuellen Zustand im Vorlauf und \$AC_CUT_INV auf den aktuellen Hauptlaufsatz.

Alle Hauptlaufvariablen (\$AC\_CUTMOD\_ANG, \$AC\_CUTMOD und \$AC\_CUT\_INV) können in Synchronaktionen gelesen werden. Ein Lesezugriff aus dem Vorlauf generiert einen Vorlaufstopp.

### Modifizierte Schneidendaten

Falls eine Werkzeugdrehung aktiv ist, werden die modifizierten Daten in den folgenden Systemvariablen zur Verfügung gestellt:

Systemvariable	Bedeutung
\$P_AD[2]	Schneidenlage
\$P_AD[10]	Halterwinkel
\$P_AD[11]	Schnittrichtung
\$P_AD[24]	Freiwinkel

### Hinweis

Die Daten sind gegenüber den korrespondierenden Werkzeugparametern (\$TC\_DP2[... ..] usw.) immer dann modifiziert, wenn die Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen" mit dem Befehl `CUTMOD` aktiviert wurde und ein orientierbarer Werkzeugträger aktiv ist, der eine Werkzeugdrehung bewirkt.

## 17.7.9 Beispiel

Das folgende Beispiel bezieht sich auf ein Werkzeug mit der Schneidenlage 3 und einem orientierbaren Werkzeugträger, der das Werkzeug um die B-Achse drehen kann.

Die Zahlenwerte in den Kommentaren geben jeweils die Satzendpositionen in Maschinenkoordinaten (MKS) in der Reihenfolge X, Y, Z an.

Programmcode	Kommentar
N10 \$TC_DP1 [1,1]=500	
N20 \$TC_DP2 [1,1]=3	; Schneidenlage
N30 \$TC_DP3 [1,1]=12	
N40 \$TC_DP4 [1,1]=1	
N50 \$TC_DP6 [1,1]=6	
N60 \$TC_DP10 [1,1]=110	; Halterwinkel
N70 \$TC_DP11 [1,1]=3	; Schnittrichtung
N80 \$TC_DP24 [1,1]=25	; Freiwinkel
N90 \$TC_CARR7 [2]=0 \$TC_CARR8 [2]=1 \$TC_CARR9 [2]=0	; B-Achse
N100 \$TC_CARR10 [2]=0 \$TC_CARR11 [2]=0 \$TC_CARR12 [2]=1	; C-Achse
N110 \$TC_CARR13 [2]=0	
N120 \$TC_CARR14 [2]=0	
N130 \$TC_CARR21 [2]=X	
N140 \$TC_CARR22 [2]=X	
N150 \$TC_CARR23 [2]="M"	

Programmcode	Kommentar		
N160 TCOABS CUTMOD=0			
N170 G18 T1 D1 TCARR=2	X	Y	Z
N180 X0 Y0 Z0 F10000	; 12.000	0.000	1.000
N190 \$TC_CARR13 [2]=30			
N200 TCARR=2			
N210 X0 Y0 Z0	; 10.892	0.000	-5.134
N220 G42 Z-10	; 8.696	0.000	-17.330
N230 Z-20	; 8.696	0.000	-21.330
N240 X10	; 12.696	0.000	-21.330
N250 G40 X20 Z0	; 30.892	0.000	-5.134
N260 CUTMOD=2 X0 Y0 Z0	; 8.696	0.000	-7.330
N270 G42 Z-10	; 8.696	0.000	-17.330
N280 Z-20	; 8.696	0.000	-21.330
N290 X10	; 12.696	0.000	-21.330
N300 G40 X20 Z0	; 28.696	0.000	-7.330
N310 M30			

**Erläuterungen:**

In Satz N180 wird zunächst das Werkzeug bei CUTMOD=0 und nicht gedrehtem orientierbaren Werkzeugträger angewählt. Da alle Offsetvektoren des orientierbaren Werkzeugträgers 0 sind, wird die Position angefahren, die den in \$TC\_DP3 [1, 1] und \$TC\_DP4 [1, 1] angegebenen Werkzeuglängen entspricht.

In Satz N200 wird der orientierbare Werkzeugträger mit einer Drehung von 30° um die B-Achse aktiviert. Da die Schneidenlage wegen CUTMOD=0 nicht modifiziert wird, ist nach wie vor der alte Schneidenbezugspunkt maßgebend. Deshalb wird in Satz N210 die Position angefahren, die den alten Schneidenbezugspunkt im Nullpunkt beibehält (d. h. der Vektor (1, 12) wird in der Z/X-Ebene um 30° gedreht).

In Satz N260 ist im Unterschied zu Satz N200 CUTMOD=2 wirksam. Aufgrund der Drehung des orientierbaren Werkzeugträgers wird die modifizierte Schneidenlage 8. Daraus folgen auch abweichende Achspositionen.

In den Sätzen N220 bzw. N270 wird jeweils die Werkzeugradiuskorrektur (WRK) aktiviert. Die unterschiedliche Schneidenlage in beiden Programmstücken hat auf die Endpositionen der Sätze, in denen die WRK aktiv ist, keinen Einfluss, die entsprechenden Positionen sind deshalb identisch. Erst in den Abwahlsätzen N260 bzw. N300 wirken sich die unterschiedlichen Schneidenlagen wieder aus.

## 17.8 Inkrementell programmierte Korrekturwerte

### 17.8.1 G91 Erweiterung

#### Voraussetzungen

Kettenmaßprogrammierung mit G91 ist so definiert, dass bei Anwahl einer Werkzeugkorrektur der Korrekturwert additiv zum inkrementell programmierten Wert gefahren wird.

#### Anwendungen

Für Anwendungen wie Ankratzen ist es erforderlich, im Kettenmaß nur den programmierten Weg zu fahren. Die aktivierte Werkzeugkorrektur wird nicht gefahren.

#### Ablauf

Anwahl einer Werkzeugkorrektur mit Kettenmaßprogrammierung

- Mit Werkzeugspitze an Werkstück ankratzen.
- Die aktuelle Istposition reduziert um Werkzeugkorrektur in Basisframe übernehmen (Istwertsetzen).
- Ab der Nullposition inkrementell fahren.

#### Aktivierung

Ob bei FRAME und inkrementeller Programmierung einer Achse eine geänderte Werkzeuglänge herausgefahren wird oder nur der programmierte Weg gefahren wird ist einstellbar mit dem Settingdatum:

SD42442 \$SSC\_TOOL\_OFFSET\_INCR\_PROG (Werkzeuglängenkorrekturen)

### Nullpunktverschiebung / Frames G91

Ob beim Wert = 1 standardmäßig Nullpunktverschiebung bei `FRAME` und inkrementeller Programmierung einer Achse herausgefahren wird oder beim Wert = 0 nur der programmierte Wert gefahren wird ist einstellbar mit dem Settingdatum:

SD42440 `$SC_FRAME_OFFSET_INCR_PROG` (Nullpunktverschiebung in Frames)

Für weitere Informationen siehe Kapitel "K2: Achsen, Koordinatensysteme, Frames (Seite 721)".

### Randbedingung

Ist das Verhalten so eingestellt, dass die Verschiebung über Programmende und `RESET` hinweg aktiv bleibt

MD20110 `$MC_RESET_MODE_MASK`, Bit6=1 (Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende)

und wird im 1. Teileprogrammsatz ein inkrementeller Weg programmiert, so wird die Korrektur additiv zum programmierten Weg immer herausgefahren.

---

#### Hinweis

Bei dieser Konfiguration sollten Teileprogramme immer mit Absolut-Programmierung beginnen.

---

## 17.8.2 Bearbeitung in Richtung der Werkzeugorientierung

### Typische Anwendung)

An Maschinen mit orientierbarem Werkzeugträger kann in Werkzeugrichtung verfahren werden (typischerweise beim Bohren), ohne dass ein Frame aktiviert wird (z.B. mittels `TOFRAME` oder `TOROT`), bei dem eine der Achsen in Richtung des Werkzeugs zeigt.

Ebenso an Maschinen, bei denen bei einer Schrägbearbeitung ein Frame aktiv ist, der die schräge Ebene definiert, das Werkzeug aber nicht exakt senkrecht eingestellt werden kann, weil wegen eines indexierten Werkzeugträgers (Hirth-Verzahnung) die Werkzeugorientierung nicht beliebig einstellbar ist.

In diesen Fällen muss dann - abweichend von der eigentlich verlangten Bewegung senkrecht zur Ebene - in Werkzeugrichtung gebohrt werden, da andernfalls der Bohrer nicht in Richtung seiner Längsachse geführt würde, was im Allgemeinen den Bruch des Bohrers zur Folge haben würde.



## MOV<sub>T</sub>

Der Endpunkt einer solchen Bewegung wird mit  $\text{MOV}_T = \dots$  programmiert. Der programmierte Wert wirkt standardmäßig inkrementell in Werkzeugrichtung. Die positive Richtung ist dabei von der Werkzeugspitze zur Werkzeugaufnahme definiert. Der Inhalt von  $\text{MOV}_T$  ist bei der Zustellbewegung (beim Bohren) deshalb in der Regel negativ, bei der Rückzugsbewegung dagegen positiv. Das entspricht den Verhältnissen bei normaler achsparalleler Bearbeitung z.B. mit  $G91Z\dots$

Wird die Bewegung in der Form  $\text{MOV}_T = \text{AC}(\dots)$  programmiert, wirkt  $\text{MOV}_T$  absolut. Dazu wird eine Ebene definiert, die durch den aktuellen Nullpunkt verläuft, und deren Flächennormalenvektor parallel zur Werkzeugorientierung ist.  $\text{MOV}_T$  gibt dann die Lage bezüglich dieser Ebene an:

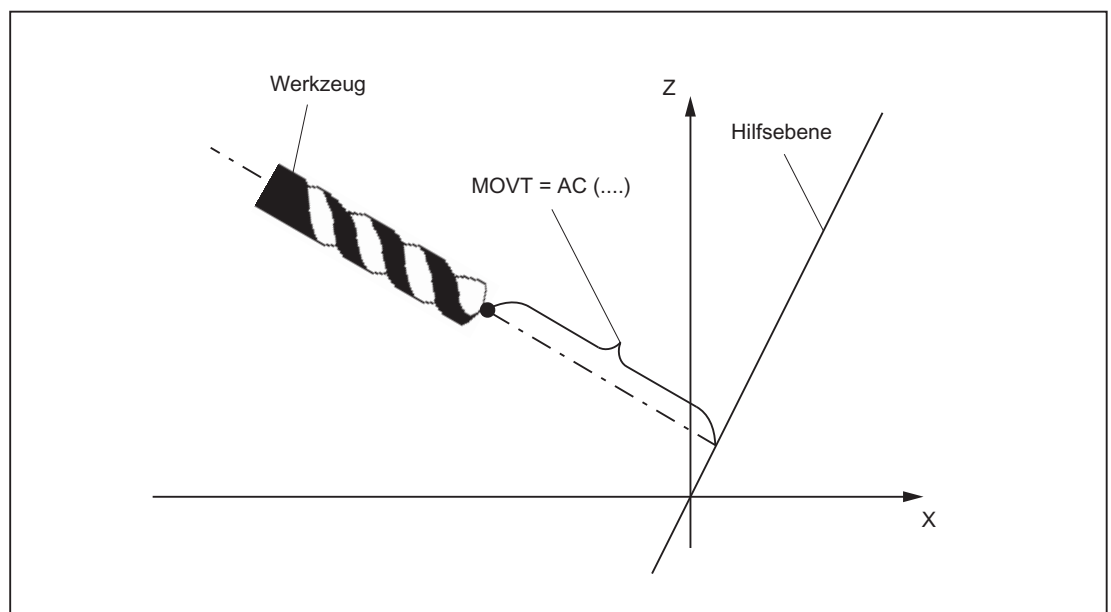


Bild 17-48 Definition der Position bei Absolutprogrammierung einer Bewegung in Werkzeugrichtung

Der Bezug zu dieser Hilfsebene dient nur zur Berechnung der Endposition. Aktive Frames werden durch diese interne Berechnung nicht beeinflusst.

Anstelle von  $\text{MOV}_T = \dots$  kann auch  $\text{MOV}_T = \text{IC}(\dots)$  geschrieben werden, wenn deutlich sichtbar zum Ausdruck gebracht werden soll, dass  $\text{MOV}_T$  inkrementell wirkt. Ein funktioneller Unterschied zwischen beiden Schreibweisen besteht nicht.

## Randbedingungen

Für die Programmierung mit `MOV_T` gibt es folgende Randbedingungen:

- Sie ist unabhängig vom Vorhandensein eines orientierbaren Werkzeugträgers. Die Richtung der Bewegung ist von der aktiven Ebene abhängig. Sie verläuft in Richtung der Applikate, d.h. bei `G17` in Z-Richtung, bei `G18` in Y-Richtung und bei `G19` in X-Richtung. Dies gilt sowohl für den Fall, dass keine orientierbarer Werkzeugträger aktiv ist als auch für den Fall eines orientierbaren Werkzeugträgers ohne drehbares Werkzeug oder mit drehbarem Werkzeug in Grundstellung.
- `MOV_T` wirkt bei aktiver Orientierungstransformation (3-4-5-Achs-Transformation) in gleicher Weise.
- Wird in einem Satz mit `MOV_T` gleichzeitig die Werkzeugorientierung verändert (z.B. bei aktiver 5-Achstransformation durch gleichzeitige Interpolation der Rundachsen), ist die Orientierung am Satzanfang für die Bewegungsrichtung von `MOV_T` maßgebend. Die Bahn der Werkzeugspitze (TCP-Tool Center Point) wird durch die Orientierungsänderung nicht beeinflusst.
- Linear- oder Spline-Interpolation (`G0`, `G1`, `ASPLINE`, `BSPLINE`, `CSPLINE`) muss aktiv sein. Andernfalls wird ein Alarm ausgegeben. Ist eine Spline-Interpolationsart aktiv, ist die resultierende Bahn im Allgemeinen keine Gerade, da der von `MOV_T` ermittelte Endpunkt so behandelt wird, als sei er explizit mit X, Y, Z programmiert worden.
- In einem Satz mit `MOV_T` dürfen keine Geometrieachsen programmiert sein (Alarm 14157).

## 17.9 Werkzeuggrundorientierung

### Anwendung

Im Normalfall hängt die dem Werkzeug selbst zugeordnete Orientierung ausschließlich von der aktiven Bearbeitungsebene ab. So ist z.B. die Werkzeugorientierung bei G17 parallel zu Z, bei G18 parallel zu Y und bei G19 parallel zu X.

Eine hiervon abweichende Werkzeugorientierung kann nur bei der Aktivierung einer 5-Achs-Transformation programmiert werden. Um jeder Werkzeugschneide eindeutig eine Orientierung zuordnen zu können, werden die folgenden Systemvariablen eingeführt:

Systemvariable	Beschreibung der WZ-Orientierung	Format	Vorbelegung
\$TC_DPV[t, d]	Werkzeugschneidenorientierung	INT	0
\$TC_DPV3[t, d]	L1-Komponente der WZ-Orientierung	REAL	0
\$TC_DPV4[t, d]	L2-Komponente der WZ-Orientierung	REAL	0
\$TC_DPV5[t, d]	L3-Komponente der WZ-Orientierung	REAL	0

Indizierung: analog zur WZ-Systemvariablen \$TC\_DPx[t, d]

t: T-Nummer der Schneide

d: D-Nummer der Schneide

Die Bezeichnungen \$TC\_DPV3 bis \$TC\_DPV5 sind in Analogie zu den Bezeichnern \$TC\_DP3 bis \$TC\_DP5 der Werkzeuglängenkomponenten gewählt.

### MD18114

Die Systemvariablen zur Beschreibung der Werkzeugorientierung stehen nur dann zur Verfügung, bei Maschinendatum ungleich Null:

MD18114 \$MN\_MM\_ENABLE\_TOOL\_ORIENT (Werkzeugschneiden Orientierung zuordnen)

MD18114 \$MN_MM_ENABLE_TOOL_ORIENT	
Wert = 1	nur die Systemvariable \$TC_DPV[t, d] steht zur Verfügung
Wert = 2	alle vier Systemvariablen sind vorhanden

### Richtungsvektor definieren

Ist der Inhalt aller vier Systemvariablen 0, wird die Orientierung wie bisher nur durch die aktive Ebene definiert.

Ist die Systemvariable \$TC\_DPV[t, d] gleich Null, definieren die übrigen drei Parameter, falls sie vorhanden sind, einen Richtungsvektor. Der Betrag des Vektors ist dabei unerheblich.

**Beispiel:**

```
$TC_DPV[1, 1] = 0
$TC_DPV3 [1, 1] = 1.0
$TC_DPV4 [1, 1] = 0.0
$TC_DPV5 [1, 1] = 1.0
```

Die Grundorientierung zeigt in diesem Beispiel in Richtung der Winkelhalbierenden in der L1-L3-Ebene, d.h. die Grundorientierung in der Winkelhalbierenden liegt bei einem Fräswerkzeug und aktiver Ebene G17 in der Z-X-Ebene.

### Grundorientierung der Werkzeuge

Grundorientierung von:	bei:
Dreh- und Schleifwerkzeugen	G18
Fräswerkzeugen	G17

Die wirksame Werkzeugorientierung bleibt in diesen Fällen unverändert und entspricht den ursprünglichen Angaben von \$TC\_DPVx[t, d].

Die Grundorientierung ist dabei immer die Richtung, die senkrecht auf der Ebene steht, in der gegebenenfalls eine Werkzeugradiuskorrektur ausgeführt wird. Bei Drehwerkzeugen fällt insbesondere die Werkzeugorientierung in der Regel mit der Werkzeuglängsachse zusammen.

Die nachfolgend genannten Settingdaten sind nur dann wirksam, wenn die Grundorientierung des Werkzeugs durch einen Eintrag in mindestens einem der Systemvariablen \$TC\_DPVx[t, d] definiert ist.

Diese Settingdaten haben keine Wirkung, wenn die Werkzeugorientierung nur durch die Ebenenanwahl G17 - G19 bestimmt wird und ist kompatibel zum bisherigen Verhalten.

Die Ebene der Grundorientierung für eine Werkzeugschneide wird, unabhängig vom Eintrag in \$TC\_DP1, entweder wie ein Fräswerkzeug oder wie ein Drehwerkzeug behandelt, wenn das folgende Settingdatum ungleich Null ist:

SD42950 \$SC\_TOOL\_LENGTH\_TYPE (Zuordnung der Werkzeuglängenkomp. unabh. vom Werkzeugtyp)

## Ebenenwechsel

Ein Ebenenwechsel führt zu einer Orientierungsänderung.

Dabei werden folgende Drehungen ausgeführt:

Beim Wechsel von:	Drehungen
G17 ⇒ G18: G18 ⇒ G19: G19 ⇒ G17:	Drehung von -90 Grad um die Z-Achse und anschließende Drehung von -90 Grad um die X-Achse.
G17 ⇒ G19: G18 ⇒ G17: G19 ⇒ G18:	Drehung von 90 Grad um die X-Achse und anschließende Drehung von 90 Grad um die Z-Achse.

Diese Drehungen sind die gleichen, die auch ausgeführt werden müssen, um bei einem Ebenenwechsel die Komponenten des Werkzeuglängenvektors zu tauschen.

Die Grundorientierung wird bei einer aktiven Adaptertransformation mitgedreht.

Ist das folgende Settingdatum ungleich Null, wird die Werkzeugorientierung bei einem Ebenenwechsel nicht gedreht:

SD42940 \$SC\_TOOL\_LENGTH\_CONST (Wechsel der Werkzeuglängenkomponenten bei Ebenwechsel)

## Komponenten der Werkzeuglänge

Die Komponenten der Werkzeugorientierung werden gleich behandelt wie die Komponenten der Werkzeuglänge, bezüglich der Settingdaten:

SD42910 \$SC\_MIRROR\_TOOL\_LENGTH (Vorzeichenwechsel Werkzeugverschleiß beim Spiegeln)

SD42950 \$SC\_TOOL\_LENGTH\_TYPE (Zuordnung der Werkzeuglängenkomp. unabh. vom Werkzeugtyp)

D.h. die Komponenten werden entsprechend getauscht und den Geometrieachsen zugeordnet.

### Systemvariable \$TC\_DPV[t, d]

Die Systemvariable \$TC\_DPV[t, d] dient dazu, bestimmte, häufig benötigte Grundorientierungen (parallel zu Koordinatenachsen) auf vereinfachte Art und Weise angeben zu können. Die zulässigen Werte sind in der folgenden Tabelle dargestellt. Dabei sind die Angaben in der ersten Spalte und die in der zweiten bzw. dritten Spalte jeweils äquivalent.

\$TC_DPV[t, d]	Grundorientierung	
	Fräswerkzeuge *	Drehwerkzeuge *
≤ 0 oder > 6	(\$TC_DPV5[t, d], \$TC_DPV4[t, d], \$TC_DPV3[t, d],) **	(\$TC_DPV3[t, d], \$TC_DPV5[t, d], \$TC_DPV4[t, d],) **
1	(0, 0, V)	(0, V, 0)
2	(0, V, 0)	(0, 0, V)
3	(V, 0, 0)	(V, 0, 0)
4	(0, 0, -V)	(0, -V, 0)
5	(0, -V, 0)	(0, 0, -V)
6	(-V, 0, 0)	(-V, 0, 0)

\* Als Drehwerkzeuge werden hier alle Werkzeuge verstanden, deren Werkzeugtyp (\$TC\_DP1[t, d]) zwischen 400 und 599 liegt. Alle anderen Werkzeugtypen bezeichnen Fräswerkzeuge.

\*\* Sind hier alle drei Werte \$TC\_DPV3[t, d], \$TC\_DPV4[t, d], \$TC\_DPV5[t, d] gleich Null, wird die Werkzeugorientierung durch die aktive Bearbeitungsebene bestimmt (Default).

V Steht jeweils für einen positiven Wert in der entsprechenden Systemvariablen.

#### Beispiel:

Für Fräswerkzeuge ist:

\$TC\_DPV[t, d] = 2 gleichbedeutend mit:

\$TC\_DPV3[t, d] = 0, \$TC\_DPV4[t, d] = 0, \$TC\_DPV5[t, d] = V.

## Randbedingungen

Wird die Funktion "Ankratzen" im `RESET`-Zustand verwendet, so ist Folgendes bezüglich des Grundstellungswertes zu beachten:

- Für die Bewertung der Verschleißkomponenten ist der Grundstellungswert der G-Code-Gruppe `TOWSTD`, `TOWMCS` und `TOWWCS` maßgebend.
- Wird zur korrekten Berechnung ein anderer als der Grundstellungswert benötigt, darf das Ankratzen nur im Stopp-Zustand erfolgen.

---

### Hinweis

Im Kapitel "Werkzeugkorrektur-Sonderbehandlungen" werden insbesondere Werkzeugkorrekturen mit Bewertung der Vorzeichen für Werkzeuglänge mit Verschleiß und Temperaturveränderungen abgehandelt.

Berücksichtigt werden dabei:

- Der Werkzeugtyp
  - Transformationen für Werkzeugkomponenten
  - Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen sowie unabhängig vom Werkzeugtyp
-

## 17.10 Werkzeugkorrektur-Sonderbehandlungen

### 17.10.1 Relevante Settingdaten

#### SD42900- 42960

Über die Settingdaten SD42900 - SD42940 lassen sich bezüglich der Werkzeugkorrektur folgende Einstellungen vornehmen:

- Vorzeichen der Werkzeuglänge
- Vorzeichen des Verschleiß
- Verhalten der Verschleißkomponenten beim Spiegeln von Geometrieachsen
- Verhalten der Verschleißkomponenten beim Wechsel der Bearbeitungsebene über Settingdaten
- Zuordnung der Werkzeuglängenkomponenten unabhängig vom tatsächlichen Werkzeugtyp
- Transformation der Verschleißkomponenten in ein geeignetes Koordinatensystem zur Beeinflussung der wirksamen Werkzeuglänge

---

#### Hinweis

In der nachfolgenden Beschreibung ist unter Verschleiß die Summe der Werte folgender Komponenten zu verstehen:

- Verschleißwerte: \$TC\_DP12 bis \$TC\_DP20
  - Summenkorrektur, bestehend aus:
    - Verschleißwerte: \$SCPX3 bis \$SCPX11
    - Einrichtewerte: \$ECPX3 bis \$ECPX11
- 

Ausführliche Informationen zu Summen- und Werkzeugkorrekturen finden sich in:

#### Literatur:

Funktionshandbuch Werkzeugverwaltung

Programmierhandbuch Grundlagen; Werkzeugkorrekturen



### Erforderliche Settingdaten

- SD42900 \$SC\_MIRROR\_TOOL\_LENGTH (Spiegeln von Werkzeuglängenkomponenten und Komponenten des Basismaßes)
- SD42910 \$SC\_MIRROR\_TOOL\_WEAR (Spiegeln von Verschleißwerten der Werkzeuglängenkomponenten)
- SD42920 \$SC\_WEAR\_SIGN\_CUTPOS (Vorzeichenbewertung der Verschleißkomponenten)
- SD42930 \$SC\_WEAR\_SIGN (Invertiert die Vorzeichen der Verschleißmaße)
- SD42940 \$SC\_TOOL\_LENGTH\_CONST (Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen)
- SD42950 \$SC\_TOOL\_LENGTH\_TYPE (Zuordnung der Werkzeuglängenkomponenten unabhängig vom Werkzeugtyp)
- SD42935 \$SC\_WEAR\_TRANSFORM (Transformation der Verschleißwerte)
- SD42960 \$SC\_TOOL\_TEMP\_COMP (Werkzeuglängenoffsets)

### 17.10.2 Werkzeuglängen spiegeln (SD42900 \$SC\_MIRROR\_TOOL\_LENGTH)

#### Aktivierung

Die Aktivierung der Werkzeuglängen-Spiegelung erfolgt über das Settingdatum:  
SD42900 \$SC\_MIRROR\_TOOL\_LENGTH <> 0 (TRUE) (Vorzeichenwechsel  
Werkzeuglänge beim Spiegeln)

#### Funktion

Durch Vorzeicheninvertierung werden folgende Komponenten gespiegelt:

- Werkzeuglängen: \$TC\_DP3, \$TC\_DP4, \$TC\_DP5
- Basismaße: \$TC\_DP21, \$TC\_DP22, \$TC\_DP23

Die Spiegelung erfolgt für alle Basismaße, deren zugehörige Achsen gespiegelt sind.  
Verschleißwerte werden **nicht** mitgespiegelt.

### Verschleißwerte spiegeln

Zum Spiegeln der Verschleißwerte ist folgendes Settingdatum zu setzen:

SD42910 \$SC\_MIRROR\_TOOL\_WEAR <> 0 (Vorzeichenwechsel Werkzeugverschleiß beim Spiegeln)

Durch Vorzeicheninvertierung werden die Verschleißwerte der Werkzeuglängenkomponenten, deren zugehörige Achsen gespiegelt sind, gespiegelt.

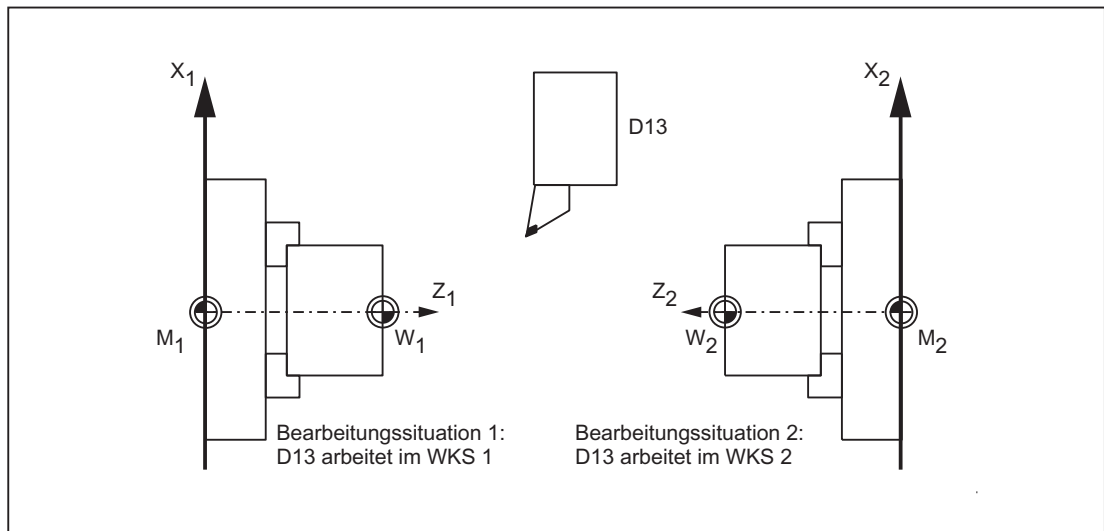


Bild 17-49 Anwendungsbeispiel: Doppelspindel-Drehmaschine

### 17.10.3 Verschleißlängen spiegeln (SD42920 \$SC\_WEAR\_SIGN\_CUTPOS)

#### Aktivierung

Die Aktivierung der Verschleißlängen-Spiegelung erfolgt über:

SD42920 \$SC\_WEAR\_SIGN\_CUTPOS <> 0 (TRUE) (Vorzeichen des Verschleißes bei Werkzeugen mit Schneidenlage)

**Funktion**

Schneidenlage	Länge 1	Länge 2
1	---	---
2	---	invertiert
3	invertiert	invertiert
4	invertiert	---
5	---	---
6	---	---
7	---	invertiert
8	invertiert	---
9	---	---

Bei Werkzeugtypen ohne relevanter Schneidenlage wird keine Verschleißlängen-Spiegelung durchgeführt.

**Hinweis**

Die Spiegelung (Vorzeicheninvertierung) in einer oder mehreren Komponenten kann sich aufheben durch gleichzeitige Aktivierung der Funktionen:

Werkzeuglängen-Spiegelung (SD42900 <> 0)

Und:

Verschleißlängen-Spiegelung (SD42920 <> 0)

**SD42930 \$SC\_WEAR\_SIGN**

Settingdatum ungleich Null:

Das Vorzeichen aller Verschleißmaße wird invertiert. Es wirkt sowohl auf die Werkzeuglängen als auch auf die übrigen Größen wie Werkzeugradius, Verrundungsradius usw.

Wird ein positives Verschleißmaß eingegeben, wird somit das Werkzeug "kürzer" und "dünner".

### Wirksamwerden der veränderten Settingdaten

Die Neubewertung von Werkzeugkomponenten bei einer Änderung der beschriebenen Settingdaten wird erst wirksam, wenn das nächste Mal eine Werkzeugschneide angewählt wird. Ist ein Werkzeug bereits aktiv und die Bewertung der Daten dieses Werkzeugs soll verändert wirksam werden, muss dieses Werkzeug erneut angewählt werden.

#### Beispiel:

```

N10 $SC_WEAR_SIGN = 0           ; Keine Vorzeichenumkehr der Verschleißwerte
N20 $TC_DP1[1,1] = 120         ; Schaftfräser
N30 $TC_DP6[1,1] = 100         ; Werkzeugradius 100 mm
N40 $TC_DP15[1,1] = 1          ; Verschleißmaß des Werkzeugradius 1 mm,
                                resultierender Werkzeugradius 101 mm

N100 T1 D1 G41 X150 Y20
....
N150 G40 X300N10
....
N200 $SC_WEAR_SIGN = 1         ; Vorzeichenumkehr aller Verschleißwerte, durch
                                Neuanwahl (D1) wird der neue Radius von 99 mm
                                wirksam. Ohne D1 wäre der wirksame Radius
                                weiterhin 101 mm.

N300 D1 G41 X350 Y-20
N310 ....

```

Entsprechendes gilt für den Fall, dass sich die resultierende Werkzeuglänge ändert, weil der Spiegelungszustand einer Achse geändert wurde. Das Werkzeug muss nach dem Spiegelbefehl erneut angewählt werden, damit die geänderten Werkzeuglängenkomponenten wirksam werden.

## 17.10.4 Werkzeuglänge und Ebenenwechsel (SD42940 \$SC\_TOOL\_LENGTH\_CONST)

### Ebenenwechsel

Die Zuordnung der Werkzeuglängenkomponenten (Länge, Verschleiß und Basismaß) zu den Geometrieachsen beim Wechsel der Bearbeitungsebene (G17-G19) wird nicht verändert.

## Zuordnung der Werkzeuge

Die Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen für Dreh- und Schleifwerkzeuge (WZ-Typ 400 bis 599) ergibt sich gemäß folgender Tabelle aus dem Wert des Settingdatums:

SD42940 \$SC\_TOOL\_LENGTH\_CONST (Wechsel der Werkzeuglängenkomponenten bei Ebenenwechsel)

Ebene	Länge 1	Länge 2	Länge 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X
*) Jeder Wert ungleich 0, der nicht gleich einem der sechs aufgeführten Werte ist, wird wie der Wert 18 bewertet.			

Folgende Tabelle zeigt die Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen für **alle anderen Werkzeuge** (WZ-Typ < 400 bzw. > 599):

Ebene	Länge 1	Länge 2	Länge 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z
*) Jeder Wert ungleich 0, der nicht gleich einem der sechs aufgeführten Werte ist, wird wie der Wert 17 bewertet.			

### Hinweis

Bei der Darstellung in den Tabellen wird davon ausgegangen, dass die Geometrieachsen 1 bis 3 mit X, Y, Z bezeichnet werden. Für die Zuordnung einer Korrektur zu einer Achse ist nicht der Achsbezeichner, sondern die Achsreihenfolge maßgebend.

Drei WZ-Längenkomponenten können auf obige 6 verschiedene Arten angeordnet werden.

## 17.10.5 Werkzeugtyp (SD42950 \$SC\_TOOL\_LENGTH\_TYPE)

### Merkmale

Die Zuordnung der Werkzeuglängenkomponenten (Länge, Verschleiß und Basismaß) unabhängig zu den Geometrieachsen vom Werkzeugtyp festlegen.

Settingdatum **ungleich** Null: (Festlegung erfolgt standardmäßig)

Es wird zwischen Dreh- und Schleifwerkzeugen (Werkzeugtypen 400 bis 599) und anderen Werkzeugen (Fräswerkzeugen) unterschieden.

Die Werte 0 bis 2 können angenommen werden. Jeder andere Wert wird wie der Wert 0 behandelt.

Die Zuordnung der Werkzeuglängenkomponenten erfolgt unabhängig vom tatsächlichen Werkzeugtyp:

- Beim Wert 1: immer wie bei Fräswerkzeugen
- Beim Wert 2: immer wie bei Drehwerkzeugen

### Orientierbare Werkzeugträger

#### Settingdaten SD42900 - SD42950

Die Settingdaten SD42900 - SD42950 wirken nicht auf die Komponenten eines eventuell aktiven orientierbaren Werkzeugträgers. Ein Werkzeug geht immer mit seiner gesamten resultierenden Länge (Werkzeuglänge + Verschleiß + Basismaß) in die Berechnung mit einem orientierbaren Werkzeugträger ein. Bei der Berechnung der resultierenden Gesamtlänge werden alle Änderungen, die durch die Settingdaten verursacht wurden, berücksichtigt.

---

#### Hinweis

Häufig werden beim Einsatz orientierbarer Werkzeugträger alle Werkzeuge für ein nicht gespiegeltes Grundsystem definiert, auch diejenigen, die nur bei Spiegelbearbeitung verwendet werden. Bei Bearbeitung mit gespiegelten Achsen wird dann der Werkzeugträger so gedreht, dass die tatsächliche Lage des Werkzeugs richtig beschrieben wird. Alle Werkzeuglängenkomponenten wirken dann automatisch in der richtigen Richtung, so dass sich eine Steuerung der Bewertung einzelner Komponenten über Settingdaten abhängig vom Spiegelungszustand einzelner Achsen erübrigt.

Der Einsatz orientierbarer Werkzeugträger ist auch dann sinnvoll, wenn der Maschinentyp es physikalisch nicht zulässt, Werkzeuge die mit verschiedenen Orientierungen fest installiert sind, auch zu drehen. Die Werkzeugvermessung kann dann einheitlich in einer Grundorientierung vorgenommen werden, und die für die Bearbeitung relevanten Maße ergeben sich durch Drehungen eines virtuellen Werkzeugträgers.

---

## 17.10.6 Werkzeuglängen im WKS unter Berücksichtigung der Orientierung

### Wechsel Werkzeug oder Arbeitsebene

Die angezeigten Werte für das Werkzeug entsprechen der Ausdehnung im WKS. Soll ein Werkzeugträger mit schräger Werkzeugaufhängung verwendet werden, so ist darauf zu achten, dass die verwendete Transformation diesen unterstützt. Ist dies nicht der Fall, werden falsche Werkzeugabmessungen angezeigt. Beim Wechseln der Arbeitsebene von G<sub>17</sub> auf G<sub>18</sub> oder G<sub>19</sub> sollte ebenfalls sichergestellt werden, dass die Transformation auch für diese Arbeitsebenen verwendet werden kann. Handelt es sich um eine Transformation, die nur für G<sub>17</sub>-Bearbeitung vorgesehen ist, so werden beim Ebenenwechsel auch weiterhin die Abmessungen für ein Werkzeug in Z-Richtung angezeigt.

Bei abgeschalteter Transformation wird das Basiswerkzeug in X-, Y- oder Z-Richtung, je nach Arbeitsebene, angezeigt. Ein programmierter Werkzeugträger wird berücksichtigt. Diese Abmessungen des Werkzeugs werden beim Verfahren ohne Transformation nicht verändert.

## 17.10.7 Werkzeuglängenoffsets in Werkzeugrichtung

### Temperaturkompensation in Echtzeit

Bei 5-Achs-Maschinen mit beweglichem Werkzeug können in den Bearbeitungsköpfen Temperaturveränderungen auftreten, die wiederum Ausdehnungsänderungen hervorrufen und in Form von Längenausdehnung auf die Werkzeugspindel übertragen. Ein typischer Fall ist bei 5-Achs-Köpfen z.B. die Wärmeausdehnung in Richtung der Spindellängsachse.

Diese Wärmeausdehnung kann auch bei vorhandener Werkzeugorientierung kompensiert werden, indem die Temperaturkompensationswerte nicht mehr den Maschinenachsen, sondern dem Werkzeug zugeordnet werden. Damit sind Längenänderungen bei Änderung der Werkzeugorientierung kompensierbar.

Mit Hilfe der Orientierungstransformation, deren Richtung von der aktuellen Werkzeugorientierung abhängt, können Bewegungen in Echtzeit überlagert und gleichzeitig mitgedreht werden. Dabei werden die Kompensationswerte ständig im Werkzeugkoordinatensystem mitgeführt.

Die Temperaturkompensation wird nur wirksam, wenn die zu kompensierende Achse auch wirklich referiert ist.

## Aktivierung

Die Temperaturkompensation in Werkzeugrichtung wird aktiviert, indem man das folgende Maschinendatum auf einen Wert **ungleich Null** setzt:

MD20390 \$MC\_TOOL\_TEMP\_COMP\_ON (Aktivierung der Temperaturkompensation für Werkzeuglänge)

Zusätzlich muss das Bit 2 gesetzt sein für jede betroffene Kanalachse im Maschinendatum:

MD32750 \$MA\_TEMP\_COMP\_TYPE[<Achsisindex>] (Temperaturkompensationstyp)

Das können mehr als drei Achsen sein, wenn als Folge von Geoachstausch oder Transformationsumschaltungen sukzessive mehr als drei Kanalachsen zeitweise zu Geometrieachsen werden können. Ist dieses Bit für eine bestimmte Kanalachse nicht gesetzt, kann in dieser Achse kein Kompensationswert wirksam werden. Andere Achsen bleiben davon jedoch unbeeinflusst. Ein Alarm wird in diesem Fall nicht ausgegeben.

## Wirkungsbereiche

Die Temperaturkompensation in Werkzeugrichtung wirkt nur bei den generischen 5-Achstransformationen mit:

- Trafotyp 24

Zwei Achsen drehen das Werkzeug.

- Trafotyp 56

Eine Achse dreht das Werkzeug, die andere Achse dreht das Werkstück ohne Temperaturkompensation.

Bei der generischen 5-Achstransformation mit:

- Trafotyp 40

Beim drehbaren Werkstück ist die Werkzeugorientierung konstant, so dass in diesem Fall eine Bewegung der Rundachsen der Maschine keinen Einfluss auf die Richtung der Temperaturkompensation hat.

Die Temperaturkompensation in Werkzeugrichtung wirkt außerdem bei den Orientierungstransformationen (sind keine generischen 5-Achs-Trafos) mit:

- Trafotyp 64 bis 69

Drehbare Linearachse.

---

### Hinweis

Bei allen anderen Transformationstypen ist die Temperaturkompensation aktivierbar. Sie wird aber nicht durch eine Änderung der Werkzeugorientierung beeinflusst. Die Achsen verfahren so, als ob keine Orientierungstransformation mit einer Temperaturkompensation aktiv wäre.

---



## Grenzwerte

Die Kompensationswerte werden auf Maximalwerte begrenzt durch das Maschinendatum:  
MD20392 \$MC\_TOOL\_TEMP\_COMP\_LIMIT[0...2] (Maximale Temperaturkompensation für Werkzeuglänge)

Die Vorbelegung des Grenzwertes ist 1 mm. Wird ein Temperaturkompensationswert vorgegeben, der größer als dieser Grenzwert ist, wird dieser ohne Alarm begrenzt.

## SD42960

Die drei Temperaturkompensationswerte bilden zusammen einen Kompensationsvektor und sind enthalten im Settingdatum:

SD42960 \$SC\_TOOL\_TEMP\_COMP[0...2] (Temperaturkompensation bezogen auf Werkzeuge)

Die Settingdaten werden z.B. mittels Synchronaktionen oder von der PLC aus vom Anwender beschrieben. Die Kompensationswerte können deshalb auch für andere Kompensationszwecke verwendet werden.

Alle drei Kompensationswerte wirken in Grundstellung bzw. bei ausgeschalteter Orientierungstransformation in Richtung der drei Geometrieachsen (in der typischen Reihenfolge X, Y, Z). Die Zuordnung der Komponenten zu den Geometrieachsen ist unabhängig vom Werkzeugtyp (Dreh- / Fräswerkzeuge bzw. Schleifwerkzeuge) und der angewählten Bearbeitungsebene G17 bis G19. Werteveränderungen in den Settingdaten werden sofort wirksam.

## Orientierbarer Werkzeugträger

Ist ein orientierbarer Werkzeugträger aktiv, so wird der Vektor der Temperaturkompensation bei einer Orientierungsänderung mitgedreht. Das gilt unabhängig von einer eventuell aktiven Orientierungstransformation.

Ist ein orientierbarer Werkzeugträger zusammen mit einer generischen 5-Achs-Transformation oder einer Transformation mit drehbarer Linearachse aktiv, so wird der Vektor der Temperaturkompensation beiden Drehungen unterworfen.

---

### Hinweis

Die Transformationen mit drehbaren Linearachsen berücksichtigen zwar die Änderung des Werkzeugvektors (Länge), jedoch **nicht** dessen Orientierungsänderung, die durch einen orientierbaren Werkzeugträger bewirkt wird.

---

Die Temperaturkompensationswerte folgen unmittelbar jeder wirksam werdenden Orientierungsänderung. Dies gilt insbesondere auch dann, wenn eine Orientierungstransformation ein- oder ausgeschaltet wird.

Gleiches gilt bei einer Änderung der Geometrieachsenzuordnung zu den Kanalachsen. Der Temperaturkompensationswert für eine Achse, die z.B. nach einem Transformationswechsel nicht mehr Geometrieachse ist, wird (interpolatorisch) auf Null reduziert. Umgekehrt baut sich in einer Achse, die nach einer Umschaltung zur Geometrieachse wird, ein eventuell vorhandener Temperaturkompensationswert sofort auf.

## Beispiele

### Temperaturkompensation in Werkzeugrichtung

Es ist eine 5-Achs-Maschine mit drehbarem Werkzeug gegeben, bei der das Werkzeug um die C- und um die B-Achse gedreht werden kann.

In Grundstellung ist das Werkzeug parallel zur Z-Achse. Wird die B-Achse um 90 Grad gedreht, zeigt das Werkzeug in X-Richtung.

Deshalb wirkt ein Temperaturkompensationswert im folgenden Settingdatum bei aktiver Transformation ebenfalls in Richtung der Maschinen X-Achse:

SD42960 \$SC\_TOOL\_TEMP\_COMP[2] (Temperaturkompensation bezogen auf Werkzeuge)

Wird in dieser Stellung die Transformation ausgeschaltet, ist die Werkzeugorientierung definitionsgemäß wieder parallel zur Z-Achse und weicht damit von der tatsächlichen Orientierung ab. Der Temperaturoffset in Richtung der X-Achse wird deshalb abgebaut und gleichzeitig in Z-Richtung aufgebaut.

Es sei eine 5-Achs-Maschine mit drehbarem Werkzeug (Trafotyp 24) gegeben. Die wesentlichen Maschinendaten sind im Folgenden dargestellt:

- Die erste Rundachse dreht um Z C-Achse.
- Die zweite Rundachse dreht um Y B-Achse.

Die wesentlichen Maschinendaten sind in der folgenden Tabelle dargestellt:

Maschinendatum	Wert	Anmerkung
MD20390 \$MC_TOOL_TEMP_COMP_ON	= TRUE	Temperaturkompensation aktiv
MD32750 \$MA_TEMP_COMP_TYPE[ AX1 ]	= 4	Korrektur in Werkzeugrichtung
MD32750 \$MA_TEMP_COMP_TYPE[ AX2 ]	= 4	Korrektur in Werkzeugrichtung
MD32750 \$MA_TEMP_COMP_TYPE[ AX3 ]	= 4	Korrektur in Werkzeugrichtung
		Zuordnung des Trafotyps 24:
MD24100 \$MC_TRAFO_TYPE_1	= 24	Trafotyp 24 im 1. Kanal
MD24110 \$MC_TRAFO_AXES_IN_1[0]	= 1	1. Achse der Transformation
MD24110 \$MC_TRAFO_AXES_IN_1[1]	= 2	2. Achse der Transformation
MD24110 \$MC_TRAFO_AXES_IN_1[2]	= 3	3. Achse der Transformation
MD24110 \$MC_TRAFO_AXES_IN_1[3]	= 5	5. Achse der Transformation
MD24110 \$MC_TRAFO_AXES_IN_1[4]	= 4	4. Achse der Transformation
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]	= 1	Geoachse zu Kanalachse 1
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]	= 2	Geoachse zu Kanalachse 2
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]	= 3	Geoachse zu Kanalachse 3
MD24570 \$MC_TRAFO5_AXIS1_1[0]	= 0.0	
MD24570 \$MC_TRAFO5_AXIS1_1[1]	= 0.0	Richtung
MD24570 \$MC_TRAFO5_AXIS1_1[2]	= 1.0	1. Rundachse ist parallel Z
MD24572 \$MC_TRAFO5_AXIS1_2[0]	= 0.0	Richtung
MD24572 \$MC_TRAFO5_AXIS1_2[1]	= 1.0	2. Rundachse ist parallel Y
MD24572 \$MC_TRAFO5_AXIS1_2[2]	= 0.0	
MD25574 \$MC_TRAFO5_BASE_ORIENT_1[0]	= 0.0	
MD25574 \$MC_TRAFO5_BASE_ORIENT_1[1]	= 0.0	Werkzeuggrundorientierung
MD25574 \$MC_TRAFO5_BASE_ORIENT_1[2]	= 1.0	in Z-Richtung

**Temperaturkompensationswerte im NC-Programm**

Die den beiden Achsen X und Z zugeordneten Kompensationswerte sind ungleich Null und werden bei der Temperaturkompensation bezüglich der Werkzeuglänge berücksichtigt. Die jeweils erreichten Maschinenachsenpositionen sind als Kommentare in den Programmzeilen angegeben.

Programmcode	Kommentar
\$SC_TOOL_TEMP_COMP[0] = -0.3	; Kompensationswert in X
\$SC_TOOL_TEMP_COMP[1] = 0.0	;
\$SC_TOOL_TEMP_COMP[2] = -1.0	; Kompensationswert in Z
	; Sollpositionen der Maschinenachsen
N10 G74 X0 Y0 Z0 A0 B0	; X Y Z
N20 X20 Y20 Z20 F10000	; 20.30 20.00 21.00
N30 TRAORI()	; 20.30 20.00 21.00
N40 X10 Y10 Z10 B90	; 11.00 10.00 9.70
N50 TRAF00F	; 10.30 10.00 11.00
N60 X0 Y0 Z0 B0 C0	; 0.30 0.00 1.00
N70 M30	

Mit Ausnahme des Satzes N40 wirkt die Temperaturkompensation immer in den Originalrichtungen, da das Werkzeug in Richtung der Grundorientierung zeigt. Das gilt insbesondere auch im Satz N50. Das Werkzeug zeigt zwar tatsächlich noch in Richtung der X-Achse, weil die B-Achse noch auf 90 Grad steht. Die zu berücksichtigende Orientierung ist aber wegen der bereits ausgeschalteten Transformation wieder parallel zu Z.

Maschinendatum	Wert	Anmerkung
MD20390 \$MC_TOOL_TEMP_COMP_ON	= TRUE	Temperaturkompensation aktiv
MD32750 \$MA_TEMP_COMP_TYPE[ AX1 ]	= 4	Korrektur in Werkzeugrichtung
MD32750 \$MA_TEMP_COMP_TYPE[ AX2 ]	= 4	Korrektur in Werkzeugrichtung
MD32750 \$MA_TEMP_COMP_TYPE[ AX3 ]	= 4	Korrektur in Werkzeugrichtung

**Weitere Literatur**

Weitere Erläuterungen zur "Temperaturkompensation" siehe:

**Literatur:**

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

Informationen zur "Generischen 5-Achstransformation" siehe:

**Literatur:**

Funktionshandbuch Sonderfunktionen; 3- bis 5-Achs-Transformation (F2)

## 17.11 Summen- und Einrichtekorrekturen

### 17.11.1 Allgemeines

#### Summenkorrekturen

Summenkorrekturen können als in der Bearbeitung **programmierbare Prozesskorrekturen** betrachtet werden und setzen sich aus all den Fehlergrößen zusammen (einschließlich dem Verschleiß), die die Abweichung des Werkstücks vom Sollmaß ausmachen.

Summenkorrekturen sind eine verallgemeinerte Form von Verschleiß. Sie sind Bestandteil der Werkzeugschneidendaten. Die Parameter der Summenkorrektur beziehen sich auf die geometrischen Daten einer Schneide.

Die Korrekturdaten einer Summenkorrektur werden über eine **DL-Nummer** angesprochen (**DL**: Location dependent; Korrekturen bezüglich des jeweiligen Einsatzortes).

Die Verschleißwerte einer D-Nummer beschreiben dagegen den physikalischen Verschleiß der Schneide, d.h. im speziellen Fall kann die Summenkorrektur dem Verschleiß der Schneide entsprechen.

Summenkorrekturen können allgemein eingesetzt werden; d.h. mit aktiver oder inaktiver Werkzeugverwaltung sowie auch mit flacher D-Nummernfunktion.

Die Summenkorrekturen werden durch Setzen entsprechender Maschinendaten aufgeteilt in:

- Summenkorrektur fein
- Summenkorrektur grob (Einrichtekorrektur)

#### Einrichtekorrektur

Die Einrichtekorrektur ist die vom Einrichter vor der Bearbeitung einzustellende Korrektur. Diese Werte belegen im NCK eigenen Speicher. Dem Bediener wird dann nur die "Summenkorrekturen fein" über HMI zugänglich gemacht.

"Summenkorrektur fein" und "Summenkorrektur grob" werden NCK-intern addiert. Dieser Wert soll im Folgenden als Summenkorrektur bezeichnet werden.

---

#### Hinweis

Die Funktion wird eingeschaltet über die Maschinendatum-Einstellung:

MD18080 \$MN\_MM\_TOOL\_MANAGEMENT\_MASK, Bit 8=1 (Stufenweise Speicher-Reservierung für die Werkzeugverw.)

---

Bei aktiven kinematischen Transformationen (z.B. 5-Achs-Transformation) wird die Werkzeuglänge zunächst unter Berücksichtigung der verschiedenen Verschleißkomponenten wie beschrieben berechnet. Diese gesamte Werkzeuglänge geht dann in die Transformation ein. Damit werden auch die Verschleißwerte unabhängig vom G-Code der Gruppe 56 - anders als beim orientierbaren Werkzeugträger - immer mittransformiert.

## 17.11.2 Funktionale Beschreibung

### Summenkorrekturen

Pro D-Nummer können mehrere Summenkorrekturen (DL-Nummern) definiert werden. Damit lassen sich z.B. **werkstückortsabhängige** Korrekturwerte ermitteln und einer Schneide zuweisen. Summenkorrekturen wirken wie Verschleiß (d.h. additiv auf die Korrekturwerte der D-Nummer). Die Daten sind fest einer D-Nummer zugeordnet.

### Einstellungen

Über Maschinendaten können Sie folgende Einstellungen vornehmen:

- Summenkorrektur aktivieren
- Anzahl der maximal im NCK anzulegenden DL-Datensätze festlegen
- Anzahl der maximal einer D-Nummer zuzuordnenden DL-Nummern festlegen
- Festlegen, ob die Summenkorrekturen (fein / grob) bei Datensicherung mitgesichert werden sollen
- Festlegen, welche Summenkorrektur aktiviert werden soll, wenn:
  - eine neue Schneidenkorrektur aktiviert wird
  - ein Bedientafelfront-**RESET** erfolgt
  - ein Bedientafelfront-**START** erfolgt
  - das Programmende erreicht wurde

Dabei orientiert sich der Name an der Logik der entsprechenden Maschinendaten für Werkzeuge und Schneiden.

"Einrichtekorrektur" und "Summenkorrektur fein" können über Systemvariablen und entsprechende BTSS-Dienste gelesen und geschrieben werden.

---

#### Hinweis

Bei aktiver WZV kann über ein Maschinendatum festgelegt werden, ob die Summenkorrektur von einem Werkzeug, das während eines programmierten Werkzeugwechsels aktiviert wird, unverändert bleibt oder auf Null gesetzt werden soll.

---

#### Zusammenfassung der Korrekturparameter \$TC\_DPx

Bisher wurden folgende allgemeine Systemvariablen zur Beschreibung einer Schneide definiert:

\$TC_DP1	Werkzeugtyp
\$TC_DP2	Schneidenlage

## Parameter für Geometrie und Verschleiß

Die Systemvariablen \$TC\_DP3 bis \$TC\_DP11 sind die Korrekturen der Werkzeuggeometrie zugeordnet. Die Systemvariablen \$TC\_DP12 bis \$TC\_DP20 erlauben es, für jeden dieser Parameter einen Verschleiß zu benennen.

Geometrie	Verschleiß	Längenkorrekturen
\$TC_DP3	\$TC_DP12	Länge 1
\$TC_DP4	\$TC_DP13	Länge 2
\$TC_DP5	\$TC_DP14	Länge 3
Geometrie	Verschleiß	Radiuskorrektur
\$TC_DP6	\$TC_DP15	Radius
\$TC_DP7	\$TC_DP16	Eckenradius (WZ-Typ 700; Nutsäge)
Geometrie	Verschleiß	weitere Korrekturen
\$TC_DP8	\$TC_DP17	Länge 4 (WZ-Typ 700; Nutsäge)
\$TC_DP9	\$TC_DP18	Länge 5
\$TC_DP10	\$TC_DP19	Winkel 1 (Winkel zwischen Stirnfläche des WZs und Torusfläche)
\$TC_DP11	\$TC_DP20	Winkel 2 (Winkel zwischen WZ-Längsachse und oberem Ende der Torusfläche)

## Basismaß bzw. Adaptermaß

\$TC_DP21	Adapter - Länge 1
\$TC_DP22	Adapter - Länge 2
\$TC_DP23	Adapter - Länge 3

## Technologie

Systemvariable	Freiwinkel
\$TC_DP24	<ul style="list-style-type: none"> <li>Für ManualTurn wird hier der Freiwinkel abgelegt; WZ-Typ 5xx. Gleiche Bedeutung wie in Standardzyklen für Drehwerkzeuge.</li> <li>Für ShopMill wird hier der Spitzenwinkel des Bohrers abgelegt; WZ-Typ 2xx.</li> <li>Verwendung in Standardzyklen für Drehwerkzeuge; WZ-Typ 5xx. Bei diesen Werkzeugen ist das der Winkel an der Nebenschneide.</li> </ul>
\$TC_DP25	<ul style="list-style-type: none"> <li>Für ManualTurn wird hier der Wert für die Schnittgeschwindigkeit abgelegt.</li> <li>Für ShopMill wird hier ein bitcodierter Wert für verschiedene Zustände von Werkzeugen des Typs 1xx und 2xx abgelegt.</li> </ul>

**Parameter der Summen und Einrichtekorrekturen (\$TC\_SCPxy-, \$TC\_ECPxy)**

Die Nummerierung der Parameter orientiert sich an der Nummerierung der Systemvariablen \$TC\_DP3 bis \$TC\_DP11.

Die Wirkung der Parameter ist dem Verschleiß analog (additiv zur WZ-Geometrie). Pro Schneidenparameter sind maximal sechs Summen-/Einrichteparameter definierbar.

WZ-Geometrie-Parameter, auf den die Korrektur addiert wird.	Summen- / Einrichteparameter - Längenkorekturen	WZ-Verschleiß-Parameter
\$TC_DP3	Länge 1 \$TC_SCP13, \$TC_SCP23,\$TC_SCP33, \$TC_SCP43,\$TC_SCP53,\$TC_SCP63 \$TC_ECP13, \$TC_ECP23,\$TC_ECP33, \$TC_ECP43,\$TC_ECP53,\$TC_ECP63 Die fettgedruckten Ziffern 1, 2, ... 6 bezeichnen die Parameter der maximal sechs (ortsabhängigen o.ä.) mit DL =1, ... 6 programmierbaren Korrekturen zum in Spalte eins genannten Parameter.	\$TC_DP12
\$TC_DP4	Länge 2 \$TC_SCP14, \$TC_SCP24,\$TC_SCP34, \$TC_SCP44,\$TC_SCP54,\$TC_SCP64 \$TC_ECP14, \$TC_ECP24,\$TC_ECP34, \$TC_ECP44,\$TC_ECP54,\$TC_ECP64	\$TC_DP13
\$TC_DP5	Länge 3 etc. ...	\$TC_DP14
	<b>Radiuskorrektur</b>	
\$TC_DP6	Radius	\$TC_DP15
\$TC_DP7	Eckenradius	\$TC_DP16
	<b>weitere Korrekturen</b>	
\$TC_DP8	Länge 4	\$TC_DP17
\$TC_DP9	Länge 5	\$TC_DP18
\$TC_DP10	Winkel 1 ...etc.	\$TC_DP19
\$TC_DP11	Winkel 2 \$TC_SCP21, \$TC_SCP31,\$TC_SCP41, \$TC_SCP51,\$TC_SCP61,\$TC_SCP71 \$TC_ECP21, \$TC_ECP31,\$TC_ECP41, \$TC_ECP51,\$TC_ECP61,\$TC_ECP71 Die fettgedruckten Ziffern 2, 3, ... 7 bezeichnen die Parameter der maximal sechs (ortsabhängigen o.ä.) mit DL =1, ... 6 programmierbaren Korrekturen zum in Spalte eins genannten Parameter.	\$TC_DP20



## Randbedingungen

Die maximale Anzahl der DL-Datensätze einer Schneide und die Gesamtanzahl der Summenkorrekturen im NCK werden über Maschinendaten festgelegt. Standardmäßig ist der Wert gleich Null, d.h. es können keine Summenkorrekturen programmiert werden.

Mit aktivierter "Überwachungsfunktion" ist es möglich, ein Werkzeug auf Verschleiß bzw. auch auf "Summenkorrektur" zu überwachen.

Die zusätzlichen Summen-/Einrichtedatensätze belegen zusätzlichen gepufferten Speicher. Pro Parameter werden 8 Byte benötigt.

Ein Summenkorrekturdatensatz benötigt: 8 Byte \* 9 Parameter = 72 Bytes

Ein Einrichtekorrekturdatensatz benötigt ebensoviel. Zusätzlich werden einige Bytes an internen Verwaltungsdaten benötigt.

### 17.11.3 Aktivierung

#### Funktion

Die Funktion muss aktiviert werden über das Maschinendatum:

MD18108 \$MN\_MM\_NUM\_SUMCORR (Summenkorrekturen im TO-Bereich)

Über die Systemvariable \$TC\_ECPx und \$TC\_SCPx bzw. über die BTSS-Schnittstelle definierte Einrichte- und Summenkorrekturen ("fein") können im Teileprogramm aktiviert werden.

Dies erfolgt durch Programmierung des Sprachbefehls `DL="nr"`.

Mit dem Aktivieren einer neuen D-Nummer ist entweder eine neue DL-Nummer zu programmieren, oder es wird die über folgendes Maschinendatum festgelegte DL-Nummer wirksam:

MD20272 \$MC\_SUMCORR\_DEFAULT (Grundstellung Summenkorrektur ohne Programm)

## DL-Programmierung

Die Programmierung der Summenkorrektur ist immer relativ zur aktiven D-Nummer und erfolgt mit dem Befehl:

DL = "n"

Die Summenkorrektur "n" wird zum Verschleiß der aktiven D-Nummer addiert.

---

### Hinweis

Wird mit "Einrichtekorrektur" **und** "Summenkorrektur fein" gearbeitet, werden beide Korrekturen zusammengefasst und dann zum Verschleiß des Werkzeuges addiert.

---

Die Abwahl der Summenkorrektur erfolgt mit dem Befehl:

DL = 0

---

### Hinweis

DL0 ist nicht erlaubt. Mit der Abwahl der Korrektur (D0 und T0) wird die Summenkorrektur ebenfalls unwirksam.

Programmieren einer nicht vorhandenen Summenkorrektur löst einen Alarm aus, analog der Programmierung einer nicht vorhandenen D-Korrektur.

Damit ist nur noch der definierte Verschleiß Bestandteil der Korrektur (in den Systemvariablen \$TC\_DP12 bis \$TC\_DP20 definiert).

Wird bei aktiver D-Korrektur eine Summenkorrektur programmiert (auch bei Abwahl), wirkt sich das auf die Bahn genauso aus wie die Programmierung von D. Damit verliert z.B. eine gerade aktive Radiuskorrektur ihren Bezug zu Nachbarsätzen.

---

## Konfiguration

**MD18112 \$MN\_MM\_KIND\_OF\_SUMCORR, Bit 4=0: (Eigenschaften der Summenkorrekturen im TO-Bereich) Standardeinstellung:**

Es ist nur **ein** Datensatz Summenkorrekturen pro DL-Nummer vorhanden.

Es wird allgemein von Summenkorrektur gesprochen.

Es sind damit die durch \$TC\_SCPx dargestellten Daten gemeint.

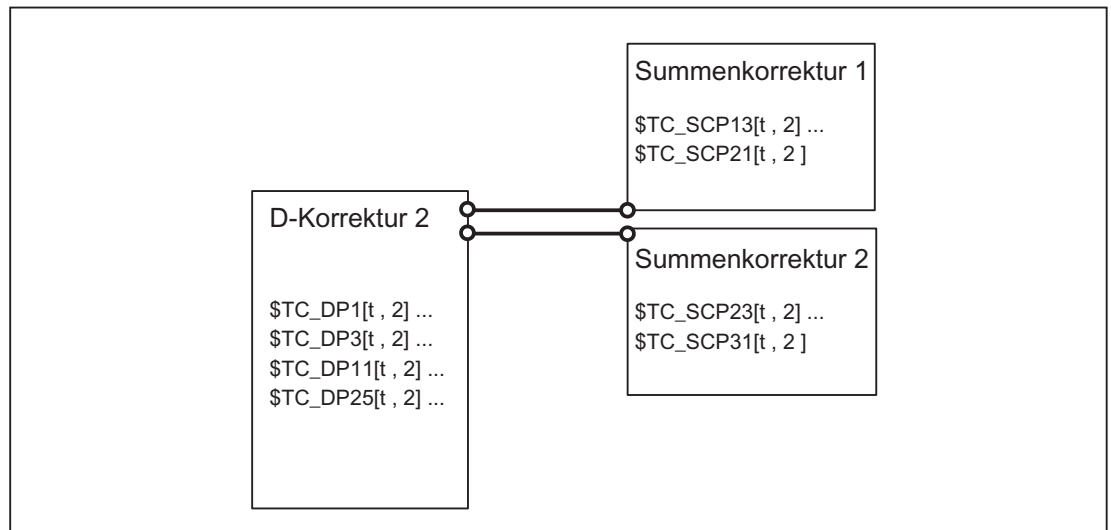


Bild 17-50 MD18112 \$MN\_MM\_KIND\_OF\_SUMCORR, Bit 4 = 0

Das Werkzeug mit T = t ist aktiv. Mit den Daten im Bild wird programmiert:

```

D2                ; Schneiden Korrekturen; d.h. $TC_DP3,... $TC_DP11 + Verschleiß
                  ($TC_DP12,...$TC_DP20) + Adaptermaß
...
DL=1              ; zusätzlich zu den bisherigen Korrekturen von D2 wird die
                  Summenkorrektur 1 addiert d.h. $TC_SCP13,...$TC_SCP21
...
DL=2              ; zur Korrektur D2 wird nicht mehr Summenkorrektur 1 addiert,
                  sondern Summenkorrektur 2 d.h. $TC_SCP23,...$TC_SCP31
...
DL=0              ; Abwahl der Summenkorrektur;
                  nur noch die Daten von D2 sind wirksam

```

**MD18112 \$MN\_MM\_KIND\_OF\_SUMCORR, Bit 4=1: Einrichtekorrekturen stehen zur Verfügung**

Die Summenkorrektur setzt sich nun zusammen aus der "Summenkorrektur fein" (dargestellt durch \$TC\_SCPx) und der Einrichtekorrektur (dargestellt durch \$TC\_ECPx). Damit gibt es zu einer DL-Nummer zwei Datensätze. Die Summenkorrektur ergibt sich durch Addition der entsprechenden Komponenten (\$TC\_ECPx + \$TC\_SCPx).

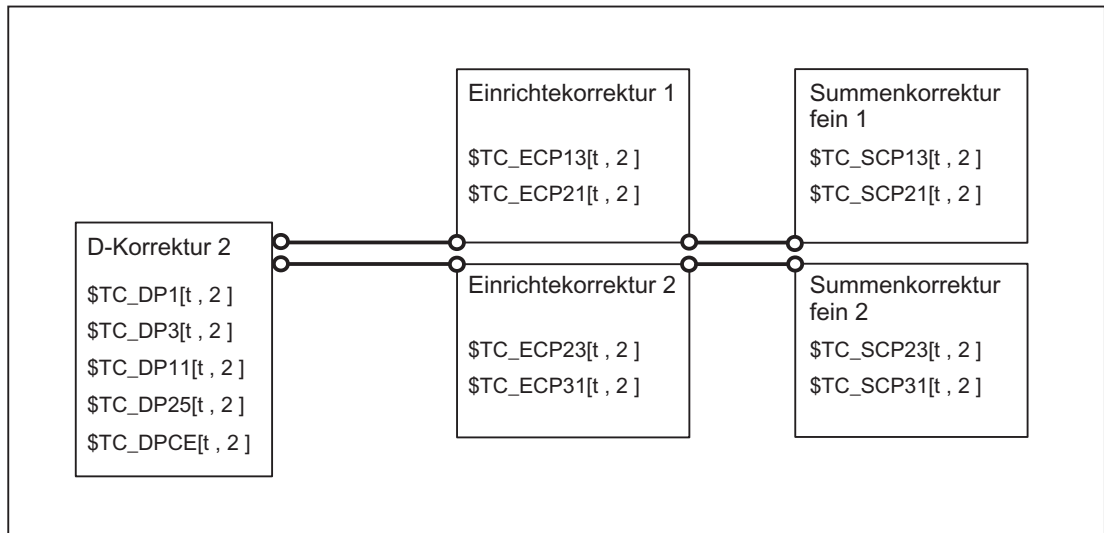


Bild 17-51 MD18112 \$MN\_MM\_KIND\_OF\_SUMCORR, Bit 4 = 1 "Einrichtekorrekturen" + "Summenkorrekturen fein"

Das Werkzeug mit T = t ist aktiv. Mit den Daten im Bild wird programmiert:

```

D2          ; Schneiden Korrekturen; d.h. $TC_DP3, ... $TC_DP11 + Verschleiß
            ($TC_DP12, ... $TC_DP20) + Adaptermaß
...
DL=1       ; zusätzlich zu den bisherigen Korrekturen von D2 wird die
            Summenkorrektur 1 addiert; d.h. $TC_ECP13 +
            $TC_SCP13, ... $TC_ECP21 + $TC_SCP21
...
DL=2       ; zur Korrektur D2 wird nicht mehr Summenkorrektur 1 addiert,
            sondern Summenkorrektur 2; d.h. $TC_ECP23 +
            $TC_SCP23, ... $TC_ECP31 + $TC_SCP31
...
DL=0       ; Abwahl der Summenkorrektur. Nur noch die Daten von D2 sind
            wirksam.
  
```

## Lesen / Schreiben im Teileprogramm

Die einzelnen Summenkorrekturparametersätze werden durch Nummernbereiche der Systemvariablen \$TC\_SCP unterschieden.

Die Bedeutung der einzelnen Variablen ist analog den Geometrie-Variablen \$TC\_DP3 bis \$TC\_DP11. Für die Grundfunktionalität sind nur Länge 1, Länge 2, Länge 3 gesetzt (Variablen \$TC\_SCP13 - \$TC\_SCP15 für die erste Summenkorrektur der Schneide).

```
R5 = $TC_SCP13[ t, d ]           ; Setzt den Wert des R-Parameters auf den Wert der
                                ; ersten Komponente der Summenkorrektur 1 der
                                ; Schneide (d)
                                ; des Werkzeuges (t).
R6 = $TC_SCP21[ t, d ]           ; Setzt den Wert des R-Parameters auf den Wert der
                                ; letzten Komponente der Summenkorrektur 1 der
                                ; Schneide (d) des Werkzeuges (t).
R50 = $TC_SCP23[ t, d ]          ; Setzt den Wert des R-Parameters auf den Wert der
                                ; ersten Komponente der Summenkorrektur 2 der
                                ; Schneide (d) des Werkzeuges (t).
$TC_SCP43[ t, d ] = 1.234        ; Setzt den Wert der ersten Komponente der
                                ; Summenkorrektur 4 der Schneide (d) des
                                ; Werkzeuges (t) auf den Wert 1.234.
```

Die Aussagen für die Einrichtekorrekturen (falls NCK damit konfiguriert ist) sind analog, d.h.:

```
R5 = $TC_ECP13[ t, d ]           ; Setzt den Wert des R-Parameters auf den Wert der
                                ; ersten Komponente der Einrichtekorrektur 1 der
                                ; Schneide (d) des Werkzeuges (t).
R6 = $TC_ECP21[ t, d ]           ; Setzt den Wert des R-Parameters auf den Wert der
                                ; letzten Komponente der Einrichtekorrektur 1 der
                                ; Schneide (d) des Werkzeuges (t).
etc.
```

Beim Arbeiten mit Einrichtekorrekturen werden die "Summenkorrekturen fein" mit den Systemvariablen \$TC\_SCPx beschrieben.

### Neuanlegen - Summenkorrektur

Falls der Korrekturdatensatz (x) noch nicht existiert, wird er mit der ersten Schreiboperation auf einen seiner Parameter (y) angelegt.

```
$TC_SCPxy[ t, d ] = r.r          ; Der Parameter y der Summenkorrektur x erhält den  
                                Wert "r.r.". Die anderen Parameter von x haben  
                                den Wert Null.
```

Beim Arbeiten mit Einrichtekorrekturen werden die "Summenkorrekturen fein" mit den Systemvariablen \$TC\_SCPx beschrieben.

---

#### Hinweis

Beim Arbeiten mit Einrichtekorrekturen wird mit dem Anlegen eines Datensatzes für "Summenkorrektur fein" der zugehörige Datensatz für die Einrichtekorrektur mit angelegt, falls bis dahin zu [t, d] noch kein Datensatz existierte.

---

### Neuanlegen - Einrichtekorrektur

Falls der Korrekturdatensatz (x) noch nicht existiert, wird er mit der ersten Schreiboperation auf einen seiner Parameter (y) angelegt.

```
$TC_ECPxy[ t, d ] = r.r          ; Der Parameter y der Einrichtekorrektur x erhält  
                                den Wert "r.r.". Die anderen Parameter von x  
                                haben den Wert Null.
```

#### Hinweis

Beim Arbeiten mit Einrichtekorrekturen wird mit dem Anlegen eines Datensatzes für Einrichtekorrekturen der zugehörige Datensatz für "Summenkorrektur fein" mit angelegt, falls bis dahin zu [t, d] noch kein Datensatz existierte.

---

## DELDL - Löschen Summenkorrektur

Summenkorrekturen sind meist nur während der Bearbeitung mit einer Schneide zu einem bestimmten Zeitpunkt an einem bestimmten Ort des Werkstückes von Bedeutung. Mit dem NC-Sprachbefehl `DELDL` können die Summenkorrekturen aus Schneiden gelöscht werden (Freigeben von benötigtem Speicher).

```
status = DELDL( t, d )           ; löscht alle Summenkorrekturen der Schneide d des
                                ; Werkzeuges t
                                ; t, d sind optionale Parameter
```

Ist `d` nicht angegeben, werden alle Summenkorrekturen aller Schneiden des Werkzeuges `t` gelöscht.

Sind `d` und `t` nicht angegeben, werden alle Summenkorrekturen der Schneiden aller Werkzeuge der TO-Einheit gelöscht (für den Kanal, in dem der Befehl programmiert wird).

Falls mit Einrichtekorrekturen gearbeitet wird, werden mit dem `DELDL`-Befehl sowohl die Einrichtekorrektur, als auch die "Summenkorrektur fein" der genannten Schneide(n) gelöscht.

---

### Hinweis

Beim "Löschen" wird der Speicher für die Datensätze freigegeben.

Anschließend können die gelöschten Summenkorrekturen nicht mehr aktiviert oder programmiert werden.

---

Summenkorrekturen, Einrichtekorrekturen aktiver Werkzeuge können nicht gelöscht werden (analog zum Löschverhalten von D-Korrekturen bzw. den Werkzeugdaten).

Der Rückgabewert "status" zeigt das Ergebnis des Löschbefehls an:

0:	Löschen erfolgreich durchgeführt
-1:	Löschen nicht (eine Schneide) oder nicht vollständig (mehrere Schneiden) durchgeführt

## Datensicherung

Die Daten werden im Rahmen der allgemeinen Werkzeugdatensicherung (als Bestandteil der D-Nummerndatensätze) gesichert.

Eine Datensicherung der Summenkorrekturen ist sinnvoll, da es möglich sein muss, bei akuten Problemen den momentanen Zustand zu sichern. Über Maschinendaten können die Summenkorrekturen von der Datensicherung ausgeschlossen werden (für "Einrichtekorrekturen" und "Summenkorrekturen fein" getrennt einstellbar).

---

### Hinweis

Summenkorrekturen verhalten sich bzgl. Satzsuchlauf und Repos analog zu D-Korrekturen. Das Verhalten bei Reset und PowerOn kann durch Maschinendaten festgelegt werden.

Falls über das folgende Maschinedatum eingestellt sein, dass nach PowerOn die letzte aktive WZ-Korrekturnummer (D) aktiviert werden soll, ist die zuletzt aktive DL-Nummer jedoch nicht mehr aktiv:

MD20110 \$MC\_RESET\_MODE\_MASK (Festlegung der Steuerungs-Grundstellung nach Reset / TP-Ende)

---



## 17.11.4 Beispiele

### Beispiel 1

Dass keine Korrektur und keine Summenkorrektur wirksam werden, soll beim Werkzeugwechsel festgelegt werden durch die Maschinendaten:

- MD20270 \$MC\_CUTTING\_EDGE\_DEFAULT=0 (Grundstellung der Werkzeugschneide ohne Programmierung)
- MD20272 \$MC\_SUMCORR\_DEFAULT=0 (Grundstellung Summenkorrektur ohne Programm)

```
T5 M06           ; WZ mit Nummer 5 wird eingewechselt - keine Korrektur aktiv
D1 DL=3         ; Korrektur D1 + Summenkorrektur 3 von D1 werden aktiv
X10
DL=2           ; Korrektur D1 + Summenkorrektur 2 werden aktiv
X20
DL=0           ; Summenkorrekturabwahl, nur noch Korrektur D1 ist aktiv
D2            ; Korrektur D2 wird aktiv - kein Beitrag der Summenkorrektur zur
              ; Korrektur
X1
DL=1           ; Korrektur D2 + Summenkorrektur 1 werden aktiv
X2
D0            ; Korrekturabwahl
X3
DL=2           ; ohne Auswirkung - DL2 von D0 ist Null (analog zur
              ; Programmierung T0 D2)
```

## Beispiel 2

Beim Werkzeugwechsel soll festgelegt werden, dass die Korrektur  $D_2$  und die Summenkorrektur  $DL=1$  wirksam werden, durch die Maschinendaten:

MD20270 \$MC\_CUTTING\_EDGE\_DEFAULT=2 (Grundstellung der Werkzeugschneide ohne Programmierung)

MD20272 \$MC\_SUMCORR\_DEFAULT=1 (Grundstellung Summenkorrektur ohne Programm)

```
T5 M06          ; WZ mit Nummer 5 wird eingewechselt - D2 + DL=1 sind aktiv
                  (=Werte der Maschinendaten)
D1 DL=3         ; Korrektur D1 + Summenkorrektur 3 von D1 werden aktiv
X10
DL=2           ; Korrektur D1 + Summenkorrektur 2 werden aktiv
X20
DL=0           ; Summenkorrekturabwahl, nur noch Korrektur D1 ist aktiv
D2             ; Korrektur D2 wird aktiv - Summenkorrektur DL=1 wird aktiviert
X1
DL=2           ; Korrektur D2 + Summenkorrektur 2 werden aktiv
D1             ; Korrektur D1 + Summenkorrektur 1 werden aktiv
```

## 17.11.5 Erweiterungen der Werkzeuglängenbestimmung

### 17.11.5.1 Korrekturen einsatzort-werkstückspezifisch einrechnen

#### Zusammensetzung der wirksamen Werkzeuglänge

Bei einer Werkzeugkorrektur ohne aktive kinematische Transformation setzt sich die wirksame Werkzeuglänge aus maximal 8 Vektoren zusammen:

- Werkzeuglänge (Geometrie) (\$TC\_DP3 - \$TC\_DP5)
- Verschleiß (\$TC\_DP12 - \$TC\_DP14)
- Basismaß (siehe Hinweis) (\$TC\_DP21 - \$TC\_DP23)
- Adaptermaß (siehe Hinweis) (\$TC\_ADPT1 - \$TC\_ADPT3)
- Summenkorrekturen fein (\$TC\_SCPx3 - \$TC\_SCPx5)
- Summenkorrekturen grob bzw. Einrichtkorrekturen (\$TC\_ECPx3 - \$TC\_ECPx5)
- Offsetvektor  $I_1$  des orientierbaren Werkzeugträgers (\$TC\_CARR1 - \$TC\_CARR3)
- Offsetvektor  $I_2$  des orientierbaren Werkzeugträgers (\$TC\_CARR4 - \$TC\_CARR6)
- Offsetvektor  $I_3$  des orientierbaren Werkzeugträgers (\$TC\_CARR15 - \$TC\_CARR17)

---

#### Hinweis

Basismaß und Adaptermaß können nur alternativ wirksam sein.

---

#### Wirkungsweise der einzelnen Vektoren

Die Wirkungsweise der einzelnen Vektoren oder Gruppen von Vektoren hängt dabei von folgenden weiteren Größen ab:

<b>Einflussgröße</b>	<b>Wirkungsweise</b>
G-Codes	aktive Bearbeitungsebene
Werkzeugtyp	Fräswerkzeuge bzw. Dreh- oder Schleifwerkzeuge
Maschinendaten	Werkzeugverwaltung aktiv / nicht aktiv, orientierbarer WZ-Träger vorhanden / nicht vorh.
Settingdaten	Verhalten von Werkzeuglängenkomponenten beim Spiegeln oder bei Ebenenwechsel
Orientierbare Werkzeugträger	Einstellwerte des orientierbaren Werkzeugträgers
Adaptertransformationen	transformierte Werkzeug-Korrekturwerte

### Verteilung auf die Geometrieachskomponenten

Wie die drei Vektorkomponenten von Teilsummen der beteiligten Vektoren auf die drei Geometrieachskomponenten verteilt werden, bestimmen die folgenden Größen:

Einflussgröße	Abhängigkeiten
aktive Bearbeitungsebene: G <sub>17</sub> X/Y-Richtung G <sub>18</sub> Z/X-Richtung G <sub>19</sub> Y/Z-Richtung	Zustellebene: Z Y X
Werkzeugtyp: Fräswerkzeuge, Bohrwerkzeuge, Schleifwerkzeuge, Drehwerkzeuge	Siehe Kapitel "WZ-Parameter 1: Werkzeugtyp (Seite 1598)", Tabelle "Minimal notwendige WZ-Parameter"
SD42900 \$SC_MIRROR_TOOL_LENGTH SD42910 \$SC_MIRROR_TOOL_WEAR SD42920 \$SC_WEAR_SIGN_CUTPOS SD42930 \$SC_WEAR_SIGN SD42940 \$SC_TOOL_LENGTH_CONST SD42950 \$SC_TOOL_LENGTH_TYPE	Siehe Kapitel "Werkzeugkorrektur- Sonderbehandlungen (Seite 1712)" und Kapitel "Settingdaten (Seite 1788)".
Adaptertransformationen	<b>Literatur:</b> Funktionshandbuch Werkzeugverwaltung

Die resultierende Werkzeugorientierung bleibt dabei immer parallel zu einer der drei Achsrichtungen X, Y oder Z und hängt ausschließlich von der aktiven Bearbeitungsebene G<sub>17</sub>-G<sub>19</sub> ab, da dem Werkzeug bisher keine Orientierung zugeordnet werden konnte.

### Stufenlose Veränderung der Werkzeugorientierung

Der orientierbare Werkzeugträger eröffnet zusätzlich zu weiteren Verschiebungen oder Längenänderungen mit Hilfe der Offsetvektoren  $l_1 - l_3$  die Möglichkeit, die Werkzeugorientierung stufenlos zu verändern.

Für weitere Erläuterungen siehe Kapitel "Orientierbare Werkzeugträger (Seite 1657)".

## Kleinere Bediener-Korrekturen

Kleinere Korrekturen müssen hingegen auch im normalen Produktionsbetrieb verändert werden können.

Gründe dafür sind z. B.:

- Werkzeugverschleiß
- Aufspannfehler
- Temperaturngang der Maschine

Diese Korrekturen werden dabei wie folgt definiert:

Definition	Verschleißkomponenten
Verschleiß	\$TC_DP12 - \$TC_DP14,
Summenkorrekturen fein	\$TC_SCPx3 - \$TC_SCPx5,
Summenkorrekturen grob bzw. Einrichtkorrekturen	\$TC_ECPx3 - \$TC_ECPx5

Insbesondere sind Korrekturen, welche die Werkzeuglängenberechnung betreffen, in den Koordinaten einzutragen, in denen sie auch gemessen wurden.

Diese werkstückspezifischen Korrekturen können einfacher vorgenommen werden, mit Hilfe der G-Code Gruppe 56, mit den drei Werten `TOWSTD`, `TOWMCS` und `TOWWCS`, und des Settingdatums:

SD42935 \$SC\_WEAR\_TRANSFORM (Transformation der Werkzeugkomponenten)

## SD42935

Welche der Verschleißkomponenten:

- Verschleiß (\$TC\_DP12 - \$TC\_DP14):
- Einrichtkorrekturen bzw. Summenkorrekturen grob (\$TC\_ECPx3 - \$TC\_ECPx5)
- Summenkorrekturen fein (\$TC\_SCPx3 - \$TC\_SCPx5)

bei den Transformationen:

- Adaptertransformation
- Orientierbarer Werkzeugträger

transformiert bzw. nicht transformiert werden sollen, kann festgelegt werden über das Settingdatum:

SD42935 \$SC\_WEAR\_TRANSFORM (Transformation der Verschleißwerte)

In der Grundstellung des Settingdatums werden alle Verschleißwerte transformiert.

Das Settingdatum wird bei folgenden Funktionen berücksichtigt:

- Verschleißwerte im Maschinenkoordinatensystem  
Teileprogrammmanweisung: TOWMCS
- Verschleißwerte im Werkstückkoordinatensystem  
Teileprogrammmanweisung: TOWWCS

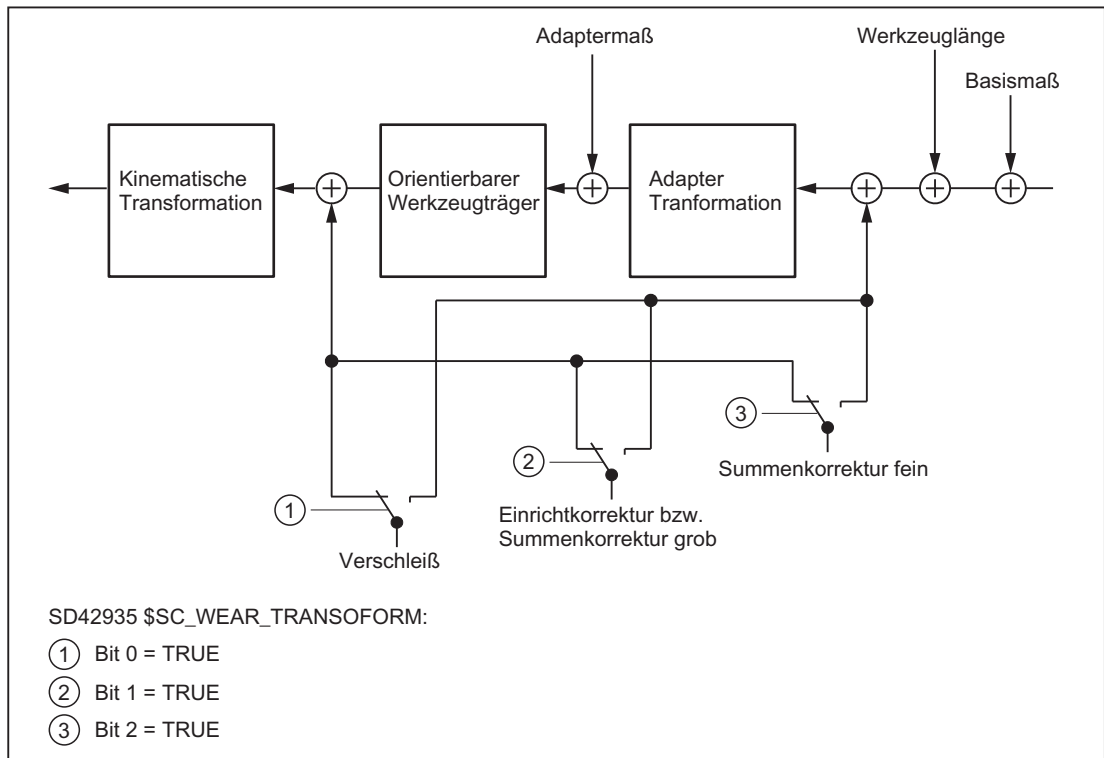


Bild 17-52 Transformation der Verschleißdaten abhängig von SD42935

## Programmierung

Mit der G-Gruppe 56 können folgende Werte festgelegt werden:

Syntax	Korrekturen
TOWSTD	Grundstellungswert für Korrekturen in der Werkzeuglänge
TOWMCS	Verschleißwerte im Maschinenkoordinatensystem (MKS)
TOWWCS	Verschleißwerte im Werkstückkoordinatensystem (WKS)
TOWBCS	Verschleißwerte im Basiskoordinatensystem (BKS)
TOWTCS	Verschleißwerte im Werkzeugkoordinatensystem (TCS) an der Werkzeughalteraufnahme (T Werkzeugträgerbezugspunkt)
TOWKCS	Verschleißwerte im Werkzeugkoordinatensystem der kinematischen Transformation (KCS) des Werkzeugkopfes

## Koordinatensysteme für Offsets der Werkzeuglänge

Mit den G-Codes `TOWMCS`, `TOWWCS`, `TOWBCS`, `TOWTCS` und `TOWKCS` kann z. B. die Werkzeuglängenkomponente Verschleiß in fünf verschiedenen Koordinatensystemen gemessen werden.

- |   |     |
|---|-----|
| 1. Maschinenkoordinatensystem                                 | MKS |
| 1. Basiskoordinatensystem                                     | BKS |
| 1. Werkstückkoordinatensystem                                 | WKS |
| 1. Werkzeugkoordinatensystem der kinematischen Transformation | KCS |
| 1. Werkzeugkoordinatensystem                                  | TCS |

Die berechnete Werkzeuglänge oder auch eine Werkzeuglängenkomponente kann mit der Funktion `GETTCOR` (vordefiniertes Unterprogramm) in einem dieser Koordinatensysteme dargestellt und ausgelesen werden.

Für weitere Erläuterungen siehe Kapitel "Werkzeuglängen, Werkzeuglängenkomponenten lesen (Seite 1754)".

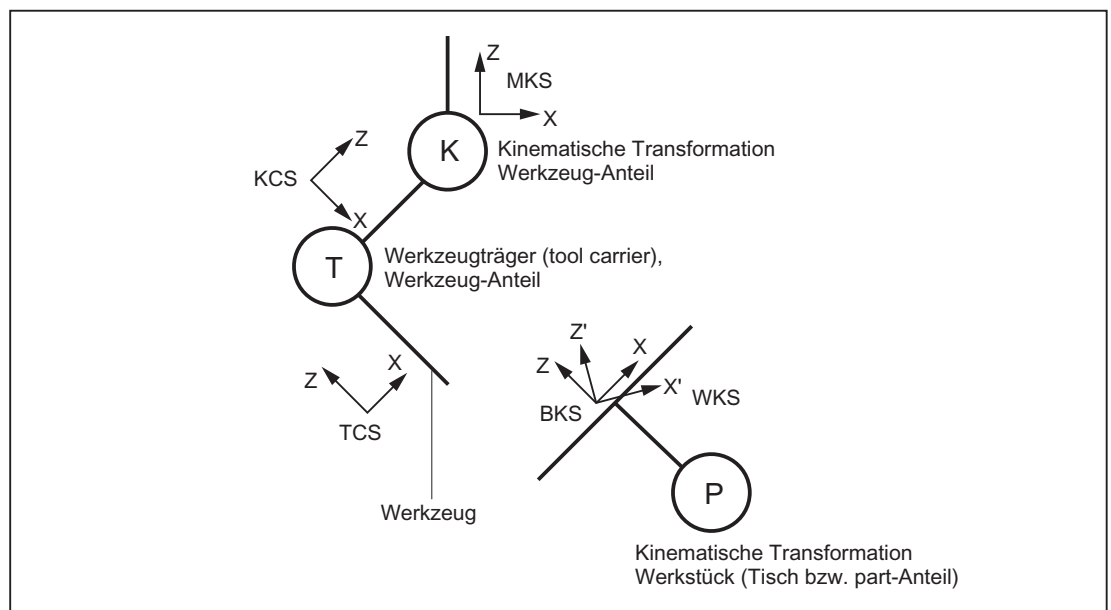


Bild 17-53 Koordinatensystem bei der Bewertung von Werkzeuglängen

### 17.11.5.2 Funktionalität der einzelnen Verschleißwerte

#### TOWSTD

Grundstellungswert (Standardbehandlung):

- Die Verschleißwerte werden zu den übrigen Werkzeuglängenkomponenten addiert. Die daraus resultierende Gesamtwerkzeuglänge geht dann gegebenenfalls in weitere Berechnungen ein.

Für den Fall eines aktiven orientierbaren Werkzeugträgers:

- Die Verschleißwerte werden der entsprechenden Drehung unterworfen.

#### TOWMCS

Verschleißwerte im Maschinenkoordinatensystem (MKS):

Für den Fall einer aktiven Drehung durch einen orientierbaren Werkzeugträger:

- Der Werkzeugträger dreht nur den Vektor der resultierenden Werkzeuglänge ohne Berücksichtigung des Verschleißes.

Anschließend werden der so gedrehte Werkzeuglängenvektor und der Verschleiß addiert. Der Verschleiß wird der Drehung nicht unterworfen.

Wenn **kein** orientierbarer Werkzeugträger aktiv ist oder dieser keine Drehung bewirkt, sind TOWMCS und TOWSTD identisch.

#### Lineare Transformation

Die Werkzeuglänge im MKS ist nur dann eindeutig definiert, wenn das MKS aus dem BKS durch eine lineare Transformation hervorgeht.

Dies wäre ebenso der Fall:

- wenn keine kinematische Transformation aktiv ist
- oder eine Orientierungstransformationen (3-, 4- und 5-Achs-Transformationen) wirksam ist.

#### TOWWCS

Verschleißwerte im Werkstückkoordinatensystem (WKS):

- Bei einem aktiven orientierbaren Werkzeugträger wird der Werkzeugvektor ohne Berücksichtigung des Verschleißes wie bei TOWMCS berechnet.
- Die Angaben des Verschleißes werden im Werkstückkoordinatensystem interpretiert.

Der Verschleißvektor im Werkstückkoordinatensystem wird auf das Maschinenkoordinatensystem umgerechnet und zum Werkzeugvektor addiert.



## TOWBCS

Verschleißwerte im Basiskoordinatensystem (BKS):

- Bei einem aktiven orientierbaren Werkzeugträger wird der Werkzeugvektor ohne Berücksichtigung des Verschleißes wie bei `TOWMCS` berechnet.
- Die Angaben des Verschleißes werden im Werkstückkoordinatensystem interpretiert.

Der Verschleißvektor im Basiskoordinatensystem wird auf das Werkstückkoordinatensystem umgerechnet und zum Werkzeugvektor addiert.

### Nichtlineare Transformation

Ist eine nichtlineare Transformation beispielsweise mit `TRANSMIT` aktiv und wird das MKS als gewünschtem Koordinatensystem angegeben, dann wird an Stelle des MKS automatisch das BKS verwendet.

### Orientierbarer Werkzeugträger

Ein eventuell vorhandener Tischanteil des orientierbaren Werkzeugträgers geht im Gegensatz zu einem Tisch- (bzw. part-) Anteil der kinematischen Transformation nicht unmittelbar in die Koordinatensysteme ein. Eine durch solch einen Anteil beschriebene Drehung wird in einem Basis- oder Systemframe abgebildet und geht damit in den Übergang vom WKS in das BKS mit ein.

### Kinematische Transformation

Der Tisch- (bzw. part-) der kinematischen Transformation wird durch den Übergang vom BKS in das MKS beschrieben.

## TOWTCS

Verschleißwerte im Werkzeugkoordinatensystem (TCS):

- Bei einem aktiven orientierbaren Werkzeugträger wird der Werkzeugvektor ohne Berücksichtigung des Verschleißes wie bei `TOWMCS` berechnet.
- Die Angaben des Verschleißes werden im Werkzeugkoordinatensystem TCS interpretiert.

Der Verschleißvektor im Werkzeugkoordinatensystem (TCS) wird über das Werkzeugkoordinatensystem der kinematischen Transformation (KCS) auf das Maschinenkoordinatensystem umgerechnet und zum Werkzeugvektor addiert.

## TOWKCS

Die Angaben der Verschleißwerte der kinematischen Transformation werden im dazugehörigen Werkzeugkoordinatensystem (KCS) interpretiert.

Der Verschleißvektor wird über das Werkzeugkoordinatensystem der kinematischen Transformation auf das Maschinenkoordinatensystem umgerechnet und zum Werkzeugvektor addiert.

### G-Code Wechsel bei aktivem Werkzeug

Ein Wechsel des G-Codes der Gruppe TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS und TOWKCS beeinflusst ein bereits aktives Werkzeug nicht und wird erst bei der nächsten Anwahl eines Werkzeuges wirksam.

Ein neuer G-Code dieser Gruppe wird auch dann wirksam, wenn er im gleichen Satz programmiert wird, in dem auch ein Werkzeug angewählt wird.

### Bewertung einzelner Verschleißkomponenten

Die Bewertung der einzelnen Verschleißkomponenten (Zuordnung zu den Geometrieachsen, Vorzeichenbewertung) wird beeinflusst durch:

- Die aktive Ebene
- Die Adaptertransformation
- Die in der folgenden Tabelle dargestellten fünf Settingdaten

Settingdatum	Verschleißkomponenten		
	TOWSTD	TOWMCS	TOWWCS
SD42910 \$SC_MIRROR_TOOL_WEAR			
SD42920 \$SC_WEAR_SIGN_CUTPOS	X	X	—
SD42930 \$SC_WEAR_SIGN	X	—	—
SD42940 \$SC_TOOL_LENGTH_CONST	X	X	X
SD42950 \$SC_TOOL_LENGTH_TYPE	X	X	X

#### Hinweis

Verschleißkomponenten, welche einer aktiven Drehung durch eine Adaptertransformation oder einen orientierbaren Werkzeugträger unterworfen werden, werden als nicht-transformierte Verschleißkomponenten bezeichnet.

## Besonderheiten

Ist `TOWMCS` oder `TOWWCS` aktiv, wirkt folgendes Settingdatum nicht auf die nicht-transformierten Verschleißkomponenten:

SD42920 `$SC_WEAR_SIGN_CUTPOS` (Vorzeichen des Verschleißes bei Werkzeugen mit Schneideanlagen)

Nicht auf die nicht-transformierten Verschleißkomponenten wirkt bei `TOWWCS` zusätzlich auch das Settingdatum:

SD42910 `$SC_MIRROR_TOOL_WEAR` (Vorzeichenwechsel Werkzeugverschleiß beim Spiegeln)

In diesem Fall ist eine eventuell aktive Spiegelung in dem Frame, der zur Bewertung der Verschleißkomponenten herangezogen wird, bereits enthalten.

Bei einem Ebenenwechsel bleibt die Zuordnung der nicht-transformierten Verschleißkomponenten zu den Geometrieachsen erhalten, d.h. diese werden nicht wie bei übrigen Längskomponenten getauscht. Die Zuordnung der Komponenten hängt von der aktiven Ebene bei der Werkzeuganwahl ab.

## Beispiel

Es sei ein Fräswerkzeug gegeben, bei dem nur der der Länge L1 zugeordnete Verschleißwert `$TC_DP12` ungleich Null ist.

Ist `G17` aktiv, wirkt diese Länge in Richtung der Z-Achse.

Dieses Maß wirkt auch bei einem Ebenenwechsel nach der Werkzeuganwahl stets in Z-Richtung, wenn `TOWMCS` oder `TOWWCS` aktiv und das Bit 1 gesetzt ist im Settingdatum:

SD42935 `$SC_WEAR_TRANSFORM` (Transformationen für Werkzeugkomponenten)

Ist bei der Werkzeuganwahl z.B. `G18` wirksam, wirkt die Komponente stattdessen unter sonst gleichen Bedingungen stets in Y-Richtung.

## 17.12 Mit Werkzeugumgebungen arbeiten

### 17.12.1 Allgemeines

#### Funktionen

Aktuelle Zustände bezüglich Werkzeugdaten können mit folgenden, allgemein zur Verfügung stehenden Funktionen bearbeitet werden:

- Abspeichern
- Löschen
- Lesen
- Modifizieren

Mit einer weiteren Funktion kann Information über die Zuordnung der Werkzeuglängen des aktiven Werkzeuges zu Abszisse, Ordinate und Applikate ermittelt werden.

### 17.12.2 Abspeichern mit TOOLENV

#### Umfang einer Werkzeugumgebung

Die Speicher-Funktion `TOOLENV` dient dazu, alle aktuellen Zustände zu speichern, die für die Bewertung der im Speicher abgelegten Werkzeugdaten von Bedeutung sind.

Das sind im Einzelnen die folgenden Daten:

- Der aktive G-Code der Gruppe 6 (`G17`, `G18`, `G19`)
- Der aktive G-Code der Gruppe 56 (`TOWSTD`, `TOWMCS`, `TOWWCS`, `TOWBCS`, `TOWTCS`, `TOWKCS`)
- Die aktive Planachse
- Das Maschinendatum:  
MD18112 `$MN_MM_KIND_OF_SUMCORR` (Eigenschaften der Summenkorrekturen im TO-Bereich)
- Das Maschinendatum:  
MD20360 `$MC_TOOL_PARAMETER_DEF_MASK` (Defintion der WZ-Parameter)
- Das Settingdatum:  
SD42900 `$SC_MIRROR_TOOL_LENGTH` (Vorzeichenwechsel Werkzeuglängen beim Spiegeln)
- Das Settingdatum:  
SD42910 `$SC_MIRROR_TOOL_WEAR` (Vorzeichenwechsel Werkzeugverschleiß beim Spiegeln)

- Das Settingdatum:  
SD42920 \$SC\_WEAR\_SIGN\_CUTPOS (Vorzeichen des Verschleißes bei Werkzeugen mit Schneidanlagen)
- Das Settingdatum:  
SD42930 \$SC\_WEAR\_SIGN (Vorzeichen des Verschleißes)
- Das Settingdatum:  
SD42935 \$SC\_WEAR\_TRANSFORM (Transformationen für Werkzeugkomponenten)
- Das Settingdatum:  
SD42940 \$SC\_LENGTH\_CONST (Wechsel der Werkzeugkomponenten bei Ebenenwechsel)
- Das Settingdatum:  
SD42950 \$SC\_TOOL\_LENGTH\_TYPE (Zuordnung der Werkzeuglängenkomp. unabh. vom Werkzeugtyp)
- Der Orientierungsanteil des aktuellen Gesamtframes (Drehungen und Spiegelungen, keine Nullpunktverschiebungen oder Skalierungen)
- Der Orientierungsanteil und die resultierende Länge des aktiven orientierbaren Werkzeugträgers
- Der Orientierungsanteil und die resultierende Länge einer aktiven Transformation
- Außer den genannten Daten, die die Umgebung des Werkzeuges beschreiben, werden auch T-Nummer, D-Nummer und DL-Nummer des aktiven Werkzeugs mit abgelegt, damit später auf dieses Werkzeug in der gleichen Umgebung wie beim Aufruf von `TOOLENV` zugegriffen werden kann, ohne das Werkzeug erneut bezeichnen zu müssen.

## Nicht in der Werkzeugumgebung

Ob die Adapterlänge oder das Basismaß in die Berechnung der Werkzeuglänge eingeht entscheidet der Wert des Maschinendatums:

MD18104 \$MN\_MM\_NUM\_TOOL\_ADAPTER (WZ-Adapter im TO-Bereich)

Da eine Veränderung dieses Maschinendatums nur bei Power On wirksam werden kann, wird es nicht in der Werkzeugumgebung gespeichert.

---

### Hinweis

Resultierende Länge von **orientierbaren Werkzeugträgern** und **Transformationen**:

Sowohl bei orientierbaren Werkzeugträgern als auch bei Transformationen gibt es Systemvariablen bzw. Maschinendaten, die wie zusätzliche Werkzeuglängenkomponenten wirken, und die ganz oder teilweise den von diesen ausführbaren Drehungen unterworfen werden. Die zusätzliche Werkzeuglängenanteile, die sich dadurch ergeben, müssen beim Aufruf von `TOOLENV` mit abgelegt werden, da sie einen Teil der Umgebung bilden, in der das Werkzeug eingesetzt wird.

---

### Adaptertransformation:

Die Adaptertransformation ist eine Eigenschaft des Werkzeugadapters und damit des Gesamtwerkzeugs. Sie ist deshalb nicht Bestandteil einer Werkzeugumgebung, die auf ein anderes Werkzeug angewendet werden kann.

Durch das Abspeichern des gesamten für die Bestimmung der Werkzeuggesamtlänge erforderlichen Daten ist es möglich, zu einem späteren Zeitpunkt die effektive Länge eines Werkzeugs zu berechnen, auch wenn es zu diesem Zeitpunkt nicht mehr aktiv ist bzw. wenn sich die Umgebungsbedingungen (z.B. G-Codes oder Settingdaten) geändert haben. Ebenso kann die effektive Länge eines anderen Werkzeugs berechnet werden mit der Annahme, es würde unter den gleichen Bedingungen eingesetzt wie das Werkzeug, für das der Status abgespeichert wurde.

## Funktion TOOLENV

Abspeichern einer Werkzeugumgebung

Die Funktion `TOOLENV` ist ein vordefiniertes Unterprogramm. Sie muss deshalb alleine in einem Satz stehen.

*Syntax:*

**Status = TOOLENV(\_NAME)**

*Wert/Parameter:*

**Status** INT

- 0:** Funktion o.k.
- 1:** Kein Speicherplatz für Werkzeugumgebungen reserviert:  
MD18116 \$MN\_MM\_NUM\_TOOL\_ENV = 0 (Anzahl Werkzeugumgebungen im TO-Bereich)  
D.h. die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden.
- 2:** Keine freien Speicherplätze für Werkzeugumgebungen mehr vorhanden.
- 3:** Nullstring als Name einer Werkzeugumgebung ist nicht zulässig.
- 4:** Kein Parameter (Name) angegeben.

**\_NAME** STRING

Name, unter dem der aktuelle Datensatz abgelegt wird.

Ist ein Datensatz mit dem gleichen Namen bereits vorhanden, wird dieser überschrieben. Der Status ist in diesem Fall 0.

### 17.12.3 Werkzeugumgebung löschen

#### Funktion DELTOOLENV

Mit dieser Funktion können Datensätze zur Beschreibung von Werkzeugumgebungen gelöscht werden. Löschen bedeutet, dass auf den unter einem bestimmten Namen abgelegten Datensatz nicht mehr zugegriffen werden kann (ein Zugriffsversuch führt zu einem Alarm).

Die Funktion `DELTOOLENV` ist ein vordefiniertes Unterprogramm.

Sie muss deshalb alleine in einem Satz stehen.

*Syntax:*

Es gibt zwei verschiedene Aufrufformen:

**Status = DELTOOLENV()**

**Status = DELTOOLENV(\_NAME)**

*Wert/Parameter:*

**Status**                    INT

**0:**    Funktion o.k.

**-1:**    Kein Speicherplatz für Werkzeugumgebungen reserviert:

MD18116 \$MN\_MM\_NUM\_TOOL\_ENV = 0 (Anzahl Werkzeugumgebungen im TO-Bereich)

D.h. die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden.

**-2:**    Ein Werkzeugumgebung mit dem angegebenen Namen existiert nicht.

**\_NAME**                    STRING

Name des zu löschenden Datensatzes.

Mit der ersten Aufrufform werden alle Datensätze gelöscht.

Mit der zweiten Aufrufform wird der Datensatz mit dem angegebenen Namen gelöscht.

Datensätze können nur mit dem Befehl `DELTOOLENV`, durch einen INITIAL.INI-Download oder durch einen Kaltstart (NCK-Hochlauf mit Standard-Maschinendaten) gelöscht werden. Es gibt keine weiteren automatischen Löschvorgänge (z.B. bei `RESET`).

## 17.12.4 Wieviele und welche Umgebungen sind gespeichert?

### \$P\_TOOLENVN

Diese Systemvariable liefert die Anzahl der verfügbaren Datensätze zur Beschreibung von Werkzeugumgebungen. (Mittels `TOOLENV` definierten und noch nicht gelöschte Datensätze).

Der Wertebereich ist 0 bis zum Maschinendatum:

MD18116 \$MN\_MM\_NUM\_TOOL\_ENV (Anzahl Werkzeugumgebungen im TO-Bereich)

Der Zugriff auf diese Systemvariable ist auch dann zulässig, wenn keine Werkzeugumgebungen möglich sind (MD18116 = 0). In diesem Fall ist der Rückgabewert 0.

*Syntax:*

`_N = $P_TOOLENVN`

*Datentyp:*

`_N` INT

Zahl der definierten `TOOLENV`.

### \$P\_TOOLENV

Diese Systemvariable liefert den Namen des n-ten Datensatzes zur Beschreibung einer Werkzeugumgebung.

Die Zuordnung der Nummern zu den Datensätzen ist nicht fix, sondern kann sich durch das Löschen oder Neuanlegen von Datensätzen verändern. Die Nummerierung der Datensätze wird intern vorgenommen. Sie läuft von 1 bis `$P_TOOLENVN`.

*Syntax:*

`_NAME = $P_TOOLENV[i]`

*Datentyp:*

`_NAME` STRING

Name des Datensatzes mit der Nummer `i`.

`i` INT

Nummer des Datensatzes.

Wird ein Index übergeben, der auf einen nicht definierten Datensatz verweist, wird folgender Alarm ausgegeben:

Alarm "17020 (unerlaubter Array-Index1)"



## 17.12.5 Lesen T, D, DL aus einer Werkzeugumgebung

### Funktion GETTENV

Die Funktion `GETTENV` dient dazu, die in einer Werkzeugumgebung abgelegten T-, D- und DL-Nummern zu lesen.

Die Funktion `GETTENV` ist ein vordefiniertes Unterprogramm. Sie muss deshalb alleine in einem Satz stehen.

*Syntax:*

**Status = GETTENV(\_NAME, \_TDDL)**

*Wert/Parameter:*

**Status** INT

**0:** Funktion o.k.

**-1:** Kein Speicherplatz für Werkzeugumgebungen reserviert:

MD18116 \$MN\_MM\_NUM\_TOOL\_ENV = 0 (Anzahl Werkzeugumgebungen imTO-Bereich)

D.h. die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden.

**-2:** Eine Werkzeugumgebung mit dem in **\_NAME** angegebenen Namen existiert nicht.

**\_NAME** STRING

Name der Werkzeugumgebung, aus der die T-, D- und DL-Nummern gelesen werden.

**\_TDDL[3]** INT

Dieses Integer-Array enthält:

- in "`_TDDL[0]`" die T-Nummer des Werkzeugs,

- in "`_TDDL[1]`" die D-Nummer des Werkzeugs,

- in "`_TDDL[2]`" die DL-Nummer des Werkzeugs,

dessen Werkzeugumgebung im Datensatz mit dem Namen "`_NAME`" abgelegt ist.

Es ist zulässig, beim Aufruf der Funktion `GETTENV` den ersten Parameter wegzulassen (z.B. `GETTENV(, _TDDL)`) oder als ersten Parameter den Nullstring zu übergeben (z.B. `GETTENV('', _TDDL)`). In diesen beiden Sonderfällen werden in `_TDDL` T-, D- und DL-Nummern des **aktiven** Werkzeuges zurückgeliefert.

## 17.12.6 Werkzeuglängen, Werkzeuglängenkomponenten lesen

### Funktion GETTCOR

Die Funktion `GETTCOR` dient dazu, Werkzeuglängen bzw. Werkzeuglängenkomponenten auszulesen.

Dabei kann durch Parametrierung angegeben werden, welche Komponenten berücksichtigt werden sollen, und unter welchen Einsatzbedingungen das Werkzeug betrachtet werden soll.

Die Funktion `GETTCOR` ist ein vordefiniertes Unterprogramm. Sie muss deshalb alleine in einem Satz stehen.

#### *Syntax:*

**Status = GETTCOR(\_LEN, \_COMP, \_STAT, \_T, \_D, \_DL)**

Mit Ausnahme des ersten Parameters (`_LEN`) können alle Parameter auch weggelassen werden.

#### *Wert/Parameter:*

**Status** INT

- 0: Funktion o.k.
- 1: Kein Speicherplatz für Werkzeugumgebungen reserviert:  
MD18116 \$MN\_MM\_NUM\_TOOL\_ENV = 0 (Anzahl Werkzeugumgebungen im TO-Bereich)  
D.h. die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden.
- 2: Eine Werkzeugumgebung mit dem in `_STAT` angegebenen Namen existiert nicht.
- 3: Ungültiger String im Parameter `_COMP`.  
Ursache dieses Fehlers können ungültige oder doppelt programmierte Zeichen sein.
- 4: Ungültige T-Nummer
- 5: Ungültige D-Nummer
- 6: Ungültige DL-Nummer
- 7: Zugriffsversuch auf nicht vorhandenes Speichermodul
- 8: Zugriffsversuch auf eine nicht vorhandene Option (programmierbare Werkzeugorientierung, Werkzeugverwaltung).
- 9: Der String `_COMP` enthält einen Doppelpunkt (Kennung für die Spezifikationen eines Koordinatensystems), es folgt jedoch kein gültiges Zeichen zur Bezeichnung des verlangten Koordinatensystems.

**\_LEN[11]** REAL

Ergebnisvektor

Die Vektorkomponenten sind in folgender Reihenfolge angeordnet:

Werkzeugtyp	(LEN[0])
Schneidenlage	(LEN[1])
Abszisse	(LEN[2])
Ordinate	(LEN[3])
Applikate	(LEN[4])
Werkzeugradius	(LEN[5])

Als Bezugskoordinatensystem für die Längenkomponenten gilt dabei das in **\_COMP** und **\_STAT** definierte Koordinatensystem. Ist in **\_COMP** kein Koordinatensystem definiert, werden die Werkzeuglängen im Maschinenkoordinatensystem dargestellt.

Die Zuordnung von Abszisse, Ordinate und Applikate zu den Geometrieachsen ist dabei abhängig von der in der verwendeten Werkzeugumgebung aktiven Ebene, d.h. bei  $G_{17}$  ist die Abszisse parallel zu X, bei  $G_{18}$  zu Z usw.

Die Komponenten **LEN[6]** bis **LEN[10]** enthalten die zusätzlichen Parameter, die für die Geometriebeschreibung eines Werkzeugs angegeben werden können (z.B. \$TC\_DP7 bis \$TC\_DP11 für die Geometrie bzw. die entsprechenden Komponenten für Verschleiß bzw. Summen- und Einrichtekorrekturen).

Diese 5 zusätzlichen Elemente und der Werkzeugradius sind nur für die Komponenten E, G, S und W definiert. Deren Bewertung hängt nicht von **\_STAT** ab. Die entsprechenden Werte in **LEN[5]** bis **LEN[10]** können deshalb nur dann ungleich Null sein, wenn an der Werkzeuglängenberechnung mindestens eine der vier genannten Komponenten beteiligt ist. Die übrigen Komponenten haben keinen Einfluss auf das Ergebnis. Die Maßangaben beziehen sich auf das Grundsystem der Steuerung (inch bzw. metrisch).

**\_COMP**                      STRING

Dieser String besteht aus zwei Teilstrings, die durch einen **Doppelpunkt** voneinander getrennt sind.

Dabei bezeichnen die einzelnen Zeichen (Buchstaben) des **ersten Teilstrings** die Werkzeuglängenkomponenten, die bei der Werkzeuglängenberechnung berücksichtigt werden sollen.

Der **zweite Teilstring** bezeichnet das Koordinatensystem, in dem die Werkzeuglängen ausgegeben werden soll. Er besteht nur aus einem einzigen relevanten Zeichen.

Die Reihenfolge der Zeichen in den Strings sowie deren Schreibweise (groß oder klein) ist beliebig. Zwischen den Zeichen können beliebig viele Leerzeichen oder Tabulatorzeichen (white spaces) eingefügt werden.

Die Buchstaben in den Teilstrings dürfen **nicht doppelt** programmiert werden. Dabei bedeutet im Einzelnen im **ersten Teilstring**:

- : (Minuszeichen, nur als erstes Zeichen zulässig): Es wird die gesamte Werkzeuglänge abzüglich der Komponenten berechnet, die im nachfolgenden String spezifiziert sind.
- C: Adaptermaß oder Basismaß (diejenige der beiden alternativ existierenden Komponenten, die für das verwendete Werkzeug aktiv ist)
- E: Einrichtkorrekturen
- G: Geometrie
- K: Kinematische Transformation (wird nur bei generischer 3-, 4- und 5-Achs-Transformation ausgewertet)
- S: Summenkorrekturen
- T: Orientierbarer Werkzeugträger
- W: Verschleiß

Ist der erste Teilstring (mit Ausnahme von white spaces) leer, so bedeutet das, dass die gesamte Werkzeuglänge unter Berücksichtigung aller Komponenten berechnet werden soll. Gleiches gilt auch dann, wenn der Parameter **\_COMP** nicht angegeben wird.

Nach einem optional programmierbaren Doppelpunkt muss ein einzelnes Zeichen folgen, das das Koordinatensystem spezifiziert, in dem die Werkzeuglängenkomponenten zu bewerten sind. Wird kein Koordinatensystem angegeben, erfolgt die Bewertung im Maschinenkoordinatensystem (MKS). Die gegebenenfalls zu berücksichtigenden Drehungen werden durch die in **\_STAT** definierte Werkzeugumgebung festgelegt.

Die Zeichen haben die folgende Bedeutung:

- B:** Basiskoordinatensystem (BKS)
- K:** Werkzeugkoordinatensystem der kinematischen Transformation (KCS)
- M:** Maschinenkoordinatensystem (MKS)
- T:** Werkzeugkoordinatensystem (TCS)
- W:** Werkstückkoordinatensystem (WKS)

**\_STAT**                      STRING

Name des Datensatzes zur Beschreibung einer Werkzeugumgebung.

Ist der Wert dieses Parameters der Nullstring (""), oder wird er nicht angegeben, wird der aktuelle Zustand verwendet.

**\_T**                              INT

Interne T-Nummer des Werkzeugs.

Ist dieser Parameter nicht angegeben, oder ist sein Wert 0, wird das in **\_STAT** abgelegte Werkzeug verwendet.

Ist der Wert dieses Parameters -1, wird die T-Nummer des aktiven Werkzeugs verwendet. Es ist auch zulässig, die Nummer des aktiven Werkzeugs explizit anzugeben.

---

### Hinweis

Ist **\_STAT** nicht angegeben, wird als Werkzeugumgebung der aktuelle Zustand verwendet. Da mit **\_T = 0** auf die in der Werkzeugumgebung gespeicherte T-Nummer verwiesen wird, wird in dieser das aktive Werkzeug verwendet, d.h. die Angaben **\_T = 0** und **\_T = -1** sind in diesem Sonderfall gleichbedeutend.

---

**\_D**                              INT

Schneidenummer des Werkzeugs. Ist dieser Parameter nicht angegeben oder ist sein Wert 0, richtet sich die verwendete D-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs.

**\_DL**                            INT

Nummer der ortsabhängigen Korrektur. Ist dieser Parameter nicht angegeben, richtet sich die verwendete DL-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs.

Die Drehungen und Komponentenvertauschungen, die gegebenenfalls von der Adaptertransformation, dem orientierbaren Werkzeugträger und einer kinematischen Transformation ausgeführt werden, sind Bestandteil der Werkzeugumgebung. Sie werden deshalb immer ausgeführt, auch dann, wenn die entsprechende Längskomponente nicht berücksichtigt werden soll. Ist das nicht gewünscht, müssen Werkzeugumgebungen definiert werden, bei denen die entsprechenden Transformationen nicht aktiv sind. In vielen Fällen (nämlich immer dann, wenn an einer Maschine keine Transformationen bzw. orientierbare Werkzeugträger eingesetzt werden), erfüllen die abgespeicherten Datensätze für Werkzeugumgebungen diese Bedingungen automatisch, so dass sie vom Anwender nicht speziell beachtet werden müssen.

### MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK

Mit den beiden niederwertigsten Bits dieses Maschinendatums wird festgelegt, wie bei Dreh- und Schleifwerkzeugen der Verschleiß (Bit 0) bzw. die Werkzeuglänge (Bit 1) in einer eventuell vorhandenen Durchmesserachse zu bewerten ist.

Sind die betreffenden Bits gesetzt, wird der zugehörige Eintrag mit dem Faktor 0.5 bewertet. Diese Bewertung schlägt sich auch in der von GETTCOR gelieferten Werkzeuglänge nieder.

**Beispiel:**

MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK = 3 (Definition der WZ-Parameter)

MD20100 \$MC\_DIAMETER\_AX\_DEF="X" (Geometrieachse mit Planachsfunktion)

X ist Durchmesserachse (Standard-Drehmaschinenkonfiguration):

```

N30      $TC_DP1 [1,1] =      500
N40      $TC_DP2 [1,1] =        2
N50      $TC_DP3 [1,1] =      3.0      ; Geometrie L1
N60      $TC_DP4 [1,1] =      4.0
N70      $TC_DP5 [1,1] =      5.0
N80      $TC_DP12 [1,1] =     12.0      ; Verschleiss L1
N90      $TC_DP13 [1,1] =     13.0
N100     $TC_DP14 [1,1] =     14.0
N110     t1 d1 g18
N120     r1 = GETTCOR(_LEN, "GW")
N130     r3 = _LEN[2]              ; 17.0 (= 4.0 + 13.0)
N140     r4 = _LEN[3]              ; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0)
N150     r5 = _LEN[4]              ; 19.0 (= 5.0 + 14.0)
N160     m30
    
```

### Kinematische Transformation, orientierbarer WZ-Träger

Wird bei der Werkzeuglängenberechnung ein orientierbarer Werkzeugträger berücksichtigt, gehen folgende Vektoren in die Werkzeuglängenberechnung ein:

Typ	Vektoren
<b>M</b>	$l_1$ und $l_2$
<b>T</b>	$l_1$ , $l_2$ und $l_3$
<b>P</b>	Werkzeuglänge wird durch den orientierbaren Werkzeugträger nicht beeinflusst.

Bei der generischen **5-Achstransformation** gehen bei den Trafotypen 24 und 56 folgende Maschinendaten in die Werkzeuglängenberechnung ein:

Trafotyp	Maschinendaten
24	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2
	MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2
	MD24558/24658 \$MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2
	MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2

Der Trafotyp 56 entspricht dem Typ M beim orientierbaren Werkzeugträger.

der Summe der beiden Vektoren  $l_1$  und  $l_3$  beim orientierbaren Werkzeugträger vom Typ M entspricht bei dieser 5-Achs-Transformation in den bisherigen Softwareständen der Vektor:

MD24560/24660 \$MC\_TRAFO5\_JOINT\_OFFSET\_1/2 (Vektor des kinematischen Versatzes der 1./2. 5-Achstransformation im Kanal)

Für die Transformation ist in beiden Fällen nur die Summe relevant. Die Art der Zusammensetzung aus den beiden Einzelkomponenten ist bedeutungslos. Bei der Berechnung der Werkzeuglängen ist es jedoch von Bedeutung, welcher Anteil dem Werkzeug und welcher dem Werkzeugschisch zuzuordnen ist.

Deshalb wurde das neue Maschinendatum eingeführt:

MD24558/24658 \$MC\_TRAFO5\_JOINT\_OFFSET\_PART\_1/2 (Vektor kinematischer Versatz im Tisch)

Es entspricht dem Vektor  $l_3$ .

Das folgende Maschinendatum entspricht jetzt nicht mehr der Summe aus  $l_1$  und  $l_3$ , sondern nur noch dem Vektor  $l_1$ :

MD24560/24660 \$MC\_TRAFO5\_JOINT\_OFFSET\_1/2 (Vektor des kinematischen Versatzes der 1. 5-Achstranf. Im Kanal)

Das neue Verhalten ist identisch zum bisherigen Verhalten, wenn das folgende Maschinendatum gleich Null ist:

MD24558/24658 \$MC\_TRAFO5\_JOINT\_OFFSET\_PART\_1/2 (Vektor kinematischer Versatz im Tisch)

## Beispiele GETTCOR

<b>GETTCOR (_LEN):</b>	Berechnet die Werkzeuglänge des aktuellen aktiven Werkzeugs im Maschinenkoordinatensystem unter Berücksichtigung aller Komponenten.
<b>GETTCOR (_LEN; "CGW : W"):</b>	Es wird die Werkzeuglänge bestehend aus Adapter- bzw. Basismaß, Geometrie und Verschleiß für das aktive Werkzeug berechnet. Weitere Anteile wie z.B. orientierbarer Werkzeugträger oder kinematische Transformationen werden nicht berücksichtigt. Die Ausgabe erfolgt im Werkstückkoordinatensystem.
<b>GETTCOR (_LEN, "-K :B"):</b>	Es wird die gesamte Werkzeuglänge des aktiven Werkzeugs ohne Berücksichtigung der Längenkomponenten einer eventuell aktiven kinematischen Transformation berechnet. Ausgabe im Basiskoordinatensystem.
<b>GETTCOR (_LEN, ":M", "Testenv1",,3):</b>	Es wird die gesamte Werkzeuglänge für das in der Werkzeugumgebung mit dem Namen "Testenv1" abgelegte Werkzeug im Maschinenkoordinatensystem berechnet. Die Berechnung erfolgt jedoch unabhängig von der abgelegten Schneidnummer für die Schneidnummer D3.

## Kompatibilität

Die Funktion `GETTCOR` dient zusammen mit den Funktionen `TOOLENV` und `SETTCOR` unter anderem dazu, Teile der Funktionalität zu ersetzen, die bisher extern in den Messzyklen realisiert war.

In den Messzyklen wurde nur ein Teil der Parameter ausgewertet, die letztlich die effektive Werkzeuglänge bestimmen. Die genannten Funktionen können so parametrisiert werden, dass das Verhalten der Messzyklen bezüglich der Werkzeuglängenberechnung reproduziert werden kann.

## 17.12.7 Werkzeugkomponenten ändern

### Funktion SETTCOR

Die Funktion `SETTCOR` dient dazu, Werkzeugkomponenten unter Berücksichtigung aller Randbedingungen, die in die Bewertung der einzelnen Komponenten eingehen können, zu verändern.



Die Funktion SETTCOR ist ein vordefiniertes Unterprogramm. Sie muss deshalb alleine in einem Satz stehen.

---

### Hinweis

Zur Terminologie: Wenn im Folgenden im Zusammenhang mit der Werkzeuglänge von Werkzeugkomponenten die Rede ist, sind die vektoriell betrachteten Komponenten gemeint, aus denen sich die gesamte Werkzeuglänge zusammensetzt, wie z.B. Geometrie oder Verschleiß. Eine solche Komponente setzt sich deshalb aus drei Einzelwerten (L1, L2, L3) zusammen, die im Folgenden als Koordinatenwerte bezeichnet werden.

Die Werkzeugkomponente "Geometrie" besteht somit beispielsweise aus den drei Koordinatenwerten \$TC\_DP3 bis \$TC\_DP5.

---

### Syntax:

**Status = SETTCOR(\_CORVAL, \_COMP, \_CORCOMP, \_CORMODE, \_GEOAX, \_STAT, \_T, \_D, \_DL)**

Mit Ausnahme der ersten beiden Parameter (\_CORVAL und \_COMP) können alle Parameter auch weggelassen werden.

### Wert/Parameter:

Status	INT
0:	Funktion o.k.
-1:	Kein Speicherplatz für Werkzeugumgebungen reserviert: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 (Anzahl Werkzeugumgebungen im TO-Bereich) D. h. die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden.
-2:	Eine Werkzeugumgebung mit dem in _STAT angegebenen Namen existiert nicht.
-3:	Ungültiger String im Parameter _COMP. Ursache dieses Fehlers können ungültige oder doppelt programmierte Zeichen sein.
-4:	Ungültige T-Nummer.
-5:	Ungültige D-Nummer.
-6:	Ungültige DL-Nummer.
-7:	Zugriffsversuch auf nicht vorhandenes Speichermodul.
-8:	Zugriffsversuch auf eine nicht vorhandene Option (programmierbare Werkzeugorientierung, Werkzeugverwaltung).
-9:	Unzulässiger Zahlenwert für den Parameter _CORCOMP.
-10:	Unzulässiger Zahlenwert für den Parameter _CORMODE.
-11:	Die Inhalte der Parameter _COMP und _CORRCOMP sind widersprüchlich.
-12:	Die Inhalte der Parameter _COMP und _CORRMODE sind widersprüchlich.
-13:	Der Inhalt des Parameters _GEOAX bezeichnet keine Geometrieachse.
-14:	Schreibversuch auf nicht vorhandene Einrichtekorrektur.

**\_CORVAL[3]**      REAL      array

Bezeichnet den Korrekturvektor.

Dabei sind in dem durch **\_STAT** definierten Werkstückkoordinatensystem (WKS) zugeordnet:

- **\_CORVAL[0]** der Abszisse
- **\_CORVAL[1]** der Ordinate
- **\_CORVAL[2]** der Applikate

Soll nur eine Werkzeugkomponente korrigiert werden (d.h. keine vektorielle Korrektur, siehe Parameter **\_CORMODE**), steht der Korrekturwert immer in **\_CORVAL[0]**, gleichgültig auf welche Achse er wirkt. Der Inhalt der beiden übrigen Komponenten wird dann nicht ausgewertet.

Bezieht sich **\_CORVAL** bzw. eine Komponente von **\_CORVAL** auf die Planachse, so wird die Angabe als **Radiusmaß** bewertet. Das bedeutet, dass ein Werkzeug z.B. um das angegebene Maß "länger" wird, was entsprechend zu einer doppelt so großen Änderung des Werkstückdurchmessers führt.

Die Maßangaben beziehen sich auf das **Grundsystem** (inch bzw. metrisch) der Steuerung.

**\_COMP**      STRING

String, der entweder aus einem oder zwei Zeichen besteht. Dabei steht das erste oder einzige Zeichen für die 1. Komponente (Val<sub>1</sub>) und das zweite Zeichen für die 2. Komponente (Val<sub>2</sub>), die entsprechend den nachfolgenden Parametern **\_CORCOMP** und **\_CORMODE** bearbeitet werden.

Die Schreibweise der Zeichen im String (groß oder klein) ist beliebig. Es können beliebig viele Leerzeichen oder Tabulatorzeichen (white spaces) eingefügt werden.

Dabei bedeutet im Einzelnen:

- C:** Adaptermaß oder Basismaß (diejenige der beiden alternativ existierenden Komponenten, die für das verwendete Werkzeug aktiv ist)
- E:** Einrichtkorrekturen
- G:** Geometrie
- S:** Summenkorrekturen
- W:** Verschleiß

**\_CORCOMP** INT

Dieser Parameter spezifiziert die Komponente(n) des Werkzeugdatensatzes, die beschrieben werden sollen. Ist dieser Parameter nicht angegeben, ist sein Wert 0.

Bedeutung der Zahlenwerte:

- 0:** Der Korrekturwert **\_CORVAL[0]** bezieht sich auf die im Parameter **\_GEOAX** übergebene Geometrieachse im Werkstückkoordinatensystem (WKS), d.h. der Korrekturwert muss in die bezeichneten Werkzeugkomponenten so eingerechnet werden, dass sich unter Berücksichtigung aller Parameter, die die Werkzeuglängenberechnung beeinflussen können, als Ergebnis eine Änderung der gesamten Werkzeuglänge in der gegebenen Achsrichtung um den vorgegebenen Wert ergibt.
- Diese Änderung soll erreicht werden durch die Korrektur der in **\_COMP** angegebenen Komponente und der in **\_CORMODE** (siehe nachfolgenden Parameter) angegebenen symbolischen Rechenvorschrift. Die resultierende Korrektur kann sich deshalb in allen drei Achskomponenten auswirken.
- 1:** Wie 0 jedoch vektoriell. Der Inhalt des Vektors **\_CORVAL** bezieht sich auf Abszisse, Ordinate, Applikate im Werkstückkoordinatensystem (WKS). Der nachfolgende Parameter **\_GEOAX** wird nicht ausgewertet.
- 2:** Vektorielle Korrektur, d.h. L1, L2 und L3 können simultan verändert werden. Im Gegensatz zu den Varianten 0 und 1 beziehen sich die in **\_CORVAL** enthaltenen Korrekturwerte jedoch auf die Koordinaten der Val<sub>1</sub>-Komponente (siehe nachfolgenden Parameter **\_CORMODE**) des Werkzeuges. Eine eventuell vorhandene Schrägstellung des Werkzeugs gegenüber dem Werkstückkoordinatensystem hat keinen Einfluss auf die Korrektur.
- 3 - 5:** Korrektur der Werkzeuglänge L1 bis L3 (\$TC\_DP3 bis \$TC\_DP5) bzw. der entsprechenden Werte bei Verschleiß, Einrichte- oder Summenkorrekturen. Der Korrekturwert ist in **\_CORVAL[0]** enthalten. Er ist in den Koordinaten der Val<sub>1</sub>-Komponente (siehe nachfolgenden Parameter **\_CORMODE**) des Werkzeugs gemessen. Eine eventuell vorhandene Schrägstellung des Werkzeugs gegenüber dem Werkstückkoordinatensystem hat keinen Einfluss auf die Korrektur.
- 6:** Korrektur des Werkzeugradius (\$TC\_DP6) bzw. der entsprechenden Werte bei Verschleiß, Einrichte- oder Summenkorrekturen.
- 7 – 11:** Korrektur von \$TC\_DP7 bis \$TC\_DP11 bzw. der entsprechenden Werte bei Verschleiß, Einrichte- oder Summenkorrekturen. Diese Parameter werden wie der Werkzeugradius behandelt.

**\_CORMODE** INT

Dieser Parameter spezifiziert die Art der auszuführenden Schreiboperation.

Ist dieser Parameter nicht angegeben, ist sein Wert 0.

Bedeutung der Einer-Stelle:

$$0: Val_{1neu} = \_CORVAL$$

$$1: Val_{1neu} = Val_{1alt} + \_CORVAL$$

$$2: Val_{1neu} = \_CORVAL$$

$$Val_{2neu} = 0$$

$$3: Val_{1neu} = Val_{1alt} + Val_{2alt} + \_CORVAL$$

$$Val_{2neu} = 0$$

Die Schreibweise  $Val_{1alt} + Val_{2alt}$  ist symbolisch zu verstehen. Werden die beiden Komponenten (aufgrund des Zustandes von **\_STAT**) unterschiedlich bewertet, d.h. ist zwischen beiden Komponenten eine Drehung wirksam, so wird  $Val_{2alt}$  vor der Addition so transformiert, dass die resultierende Werkzeuglänge nach dem Löschen von  $Val_{2neu}$  und vor der Addition von **\_CORVAL** unverändert bleibt.

**\_CORVAL** bezieht sich immer auf  $Val_1$ . **\_CORVAL** ist ein Wert, der stets im Werkstückkoordinatensystem (WKS) gemessen wird. Er ist deshalb gegenüber der Werkzeugkomponente, in die er eingerechnet werden soll, gegebenenfalls bereits transformiert. Er kann somit nicht unmittelbar mit dem abgespeicherten Wert verrechnet werden, sondern muss vor der Addition zu  $Val_1$  bzw.  $Val_2$  zurücktransformiert werden.

Das kann zur Folge haben, dass sich die Korrektur in einer anderen als in der durch **\_CORCOMP** definierten Achse auswirkt, bzw. dass diese auf mehrere Achsen wirkt.

Für den Fall **CORRCOMP = 0**, d.h. dann wenn **\_CORVAL** keinen Vektor, sondern nur einen Einzelwert enthält, werden die beschriebene Operationen in den Koordinaten ausgeführt, in denen **\_CORVAL** gemessen wurde (WKS). Insbesondere gilt das auch für das Nullsetzen von  $Val_{2neu}$  in den Varianten 2 und 3. Dieses Ergebnis wird dann in die Koordinaten des Werkzeugs zurücktransformiert. Das kann zur Folge haben, dass keine der nullzusetzenden Koordinatenwerte ( $L_1$ ,  $L_2$ ,  $L_3$ ) Null wird bzw. dass Koordinatenwerte, die vorher Null waren, nun ungleich Null sind. Werden die entsprechenden Operationen jedoch sukzessive für alle drei Geometrieachsen ausgeführt, so werden stets alle drei Koordinatenwerte der zu löschenden Komponente Null. Ist das Werkzeug gegenüber dem WKS nicht gedreht, oder aber so gedreht, dass alle Werkzeugkomponenten parallel zu den Koordinatenachsen bleiben (Achsvertauschungen), ist gewährleistet, dass sich jeweils auch nur eine Werkzeugkoordinate ändert.

Die sukzessive Ausführung der gleichen Operation (**\_CORRMODE**) mit **\_CORCOMP = 0** für alle drei Koordinatenachsen in beliebiger Reihenfolge ist identisch zu der einmaligen Ausführung der gleichen Operation mit **\_CORCOMP = 1**.

**\_T** INT

Interne T-Nummer des Werkzeugs. Ist dieser Parameter nicht angegeben oder ist sein Wert 0, wird das in **\_STAT** abgelegte Werkzeug verwendet. Ist der Wert dieses Parameters -1, wird die T-Nummer des aktiven Werkzeugs verwendet. Es ist auch zulässig, die Nummer des aktiven Werkzeugs explizit anzugeben.

---

**Hinweis**

Ist **\_STAT** nicht angegeben, wird als Werkzeugumgebung der aktuelle Zustand verwendet. Da mit **\_T = 0** auf die in der Werkzeugumgebung gespeicherte T-Nummer verwiesen wird, wird in dieser das aktive Werkzeug verwendet, d.h. die Angaben **\_T = 0** und **\_T = -1** sind in diesem Sonderfall gleichbedeutend.

---

**\_D** INT

Schneidenummer des Werkzeugs. Ist dieser Parameter nicht angegeben, oder ist sein Wert 0, richtet sich die verwendete D-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs.

**\_DL** INT

Nummer der ortsabhängigen Korrektur. Ist dieser Parameter nicht angegeben, richtet sich die verwendete DL-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs. Spezifizieren T, D und DL ein Werkzeug ohne ortsabhängige Korrekturen, dürfen im Parameter **\_COMP** keine Summen- oder Einrichtkorrekturen spezifiziert werden (Fehlercode in "Status").

Nicht alle möglichen Kombinationen der drei Parameter **\_COMP**, **\_CORCOMP** und **\_CORMODE** sind sinnvoll. Beispielsweise erfordert die Rechenvorschrift 3 in **\_CORCOMP** die Angabe von zwei Zeichen in **\_COMP**. Wird eine unzulässige Parameterkombination angegeben, wird im Status ein entsprechender Fehlercode zurückgegeben.

---

**Hinweis****Berechnung der Werkzeuglänge in Abhängigkeit vom Maschinendatum MD20360**

Mit den beiden niederwertigsten Bits dieses Maschinendatums wird festgelegt, wie bei Dreh- und Schleifwerkzeugen der Verschleiß (Bit 0) bzw. die Werkzeuglänge (Bit 1) in einer eventuell vorhandenen Durchmesserachse zu bewerten ist. Sind die betreffenden Bits gesetzt, wird bei der Werkzeuglängenberechnung der zugehörige Eintrag mit dem Faktor 0.5 bewertet. Die Korrektur mittels **SETTCOR** wird so ausgeführt, dass die gesamte effektive Werkzeuglängenänderung gleich dem in **\_CORVAL** übergebenen Wert ist.

Die Korrektur der Komponente deren Länge, auf Grund des folgenden Maschinendatums bei der Längenberechnung mit dem Faktor 0.5 bewertet wird, muss mit dem doppelten übergebenen Wert erfolgen:

MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK (Definition der WZ-Parameter)

---

**Beispiel 1**

```

N10    def real _CORVAL[3]
N20    $TC_DP1[1,1] =          120          ; Fräswerkzeug
N30    $TC_DP3[1,1] =          10.0        ; Geometrie L1
N40    $TC_DP12[1,1]=          1.0         ; Verschleiss L1
N50    _CORVAL[0] =            0.333
N60    t1 d1 g17 g0
N70    r1 = settcor(_CORVAL, "G", 0, 0, 2)
N80    t1 d1 x0 y0 z0                ; ==>   MCS-Position X0.000 Y0.000
                                           Z1.333
N90    M30

```

**\_CORCOMP** ist 0, deshalb wird der in Z-Richtung wirkende Koordinatenwert der Geometrie-Komponente durch den Korrekturwert 0.333 ersetzt. Die resultierende Werkzeuggesamtlänge ist somit  $L1 = 0.333 + 1.000 = 1.333$

**Beispiel 2**

```

N10    def real _CORVAL[3]
N20    $TC_DP1[1,1] =          120          ; Fräswerkzeug
N30    $TC_DP3[1,1] =          10.0        ; Geometrie L1
N40    $TC_DP12[1,1]=          1.0         ; Verschleiß L1
N50    _CORVAL[0] =            0.333
N60    t1 d1 g17 g0
N70    r1 = settcor(_CORVAL, "W", 0, 1, 2)
N80    t1 d1 x0 y0 z0                ; ==>   MCS-Position X0.000 Y0.000
                                           Z11.333
N90    M30

```

**\_CORCOMP** ist 1, deshalb wird der in Z-Richtung wirkende Korrekturwert von 0.333 zum Verschleißwert von 1.0 addiert.

Die resultierende Werkzeuggesamtlänge ist somit  $L1 = 10.0 + 1.333 = 11.333$ .

## 17.13 Werkzeuglängen L1, L2, L3 Zuordnung: LENTOAX

### Funktion LENTOAX

Die Funktion "LENTOAX" liefert Informationen über die Zuordnung der Werkzeuglängen  $L_1$ ,  $L_2$  und  $L_3$  des **aktiven** Werkzeugs zu Abszisse, Ordinate und Applikate. Die Zuordnung von Abszisse, Ordinate und Applikate zu den Geometrieachsen wird durch Frames und die aktive Ebene ( $G_{17} - G_{19}$ ) beeinflusst.

Betrachtet wird dabei nur der Geometrieanteil eines Werkzeugs ( $\$TC\_DP3[x,y]$  bis  $\$TC\_DP5[x,y]$ ), d.h. eine gegebenenfalls davon abweichende Achszuordnung weiterer Komponenten (z.B. Verschleiß) hat auf das Ergebnis keinen Einfluss.

Die Funktion "LENTOAX" ist ein vordefiniertes Unterprogramm. Sie muss deshalb alleine in einem Satz stehen.

#### Syntax:

**Status = LENTOAX(\_AXIND, \_MATRIX, \_COORD)**

Die ersten beiden Parameter müssen immer vorhanden sein, der letzte Parameter kann auch weggelassen werden.

#### Wert/Parameter:

**Status** INT

- 0: Funktion o.k., Information in **\_AXIND** reicht zur Beschreibung aus (alle Werkzeuglängenkomponenten sind parallel zu Geometrieachsen).
- 1: Funktion ist o.k., zu einer korrekten Beschreibung muss aber der Inhalt von **\_MATRIX** ausgewertet werden (die Werkzeuglängenkomponenten sind nicht parallel zu den Geometrieachsen).
- 1: Ungültiger String im Parameter **\_COORD**.
- 2: Kein Werkzeug aktiv.

**\_AXIND[3]** INT array

Die Indizes 0 bis 2 sind der Abszisse (0), der Ordinate (1) bzw. der Applikate (2) zugeordnet (Bsp.: **\_AXIND[0]** enthält die Nummer der Werkzeuglängenkomponente, die in Richtung der Abszisse wirkt).

Der Inhalt hat folgende Bedeutung:

- 0: Zuordnung vorhanden (Achse existiert nicht)
- 1 bis 3: Nummer der Länge, die in der entsprechende Koordinatenachse wirkt. Das **bzw.** Vorzeichen ist negativ, wenn die Werkzeuglängenkomponente in negative
- 1 bis -3: Koordinatenrichtung zeigt.

**\_MATRIX[3][3]**

REAL

array

Matrix, die den Vektor der Werkzeuglängen ( $L_1=1, L_2=1, L_3=1$ ) in den Vektor der Koordinatenachsen (Abszisse, Ordinate, Applikate) abbildet, d.h. den **Spalten** sind die Werkzeuglängenkomponenten in der Reihenfolge  $L_1, L_2, L_3$  zugeordnet, den **Zeilen** die Achsen in der Reihenfolge Abszisse, Ordinate, Applikate.

In der Matrix sind immer alle Elemente gültig, auch dann, wenn die zu einer Koordinatenachse gehörende Geometrieachse nicht vorhanden ist, d.h. wenn der entsprechende Eintrag in **\_AXIND** 0 ist.

**\_COORD**

STRING

Bezeichnet das Koordinatensystem, für das die Zuordnung gilt.

**MKS** oder **M**: Abbildung der Werkzeuglänge in das Maschinenkoordinatensystem.

**BKS** oder **B**: Abbildung der Werkzeuglänge in das Basiskoordinatensystem.

**WCS** oder **W**: Abbildung der Werkzeuglänge in das Werkstückkoordinatensystem (WCS entspricht WKS; Default).

**KCS** oder **K**: Abbildung der Werkzeuglänge in das Werkzeugkoordinatensystem der **kinematischen Transformation**.

**TCS** oder **T**: Abbildung der Werkzeuglänge in das Werkzeugkoordinatensystem.

Die Schreibweise der Zeichen im String (groß oder klein) ist beliebig.

## Weitere Erklärungen

Sind die Werkzeuglängenkomponenten parallel zu den Geometrieachsen, werden die Achsindizes, die den Längenkomponenten  $L_1$  bis  $L_3$  zugeordnet sind, im Feld **\_AXIND** zurückgeliefert.

Zeigt eine Werkzeuglängenkomponente in negative Achsrichtung, erhält auch der zugehörige Achsindex ein Minuszeichen. Der Rückgabewert (**Status**) ist in diesem Fall 0. Existiert eine Achse nicht, ist der zugehörige Rückgabewert 0. Die Zuordnung kann auch aus dem Parameter **\_MATRIX** gelesen werden. Sechs der neun Matrixelemente sind dann Null, drei Elemente enthalten jeweils den Wert +1 oder -1.

---

### Hinweis

Im TCS sind immer alle Werkzeuglängenkomponenten parallel oder antiparallel zu den Achsen.

Antiparallel können die Komponenten nur dann sein, wenn Spiegeln aktiv ist und das folgende Settingdatum gesetzt ist:

SD42900 \$SC\_MIRROR\_TOOL\_LENGTH (Vorzeichenwechsel Werkzeuglänge beim Spiegeln)

---



Sind nicht alle Längenkomponenten parallel bzw. antiparallel zu den Geometrieachsen, wird in `_AXIND` jeweils der Index der Achse zurückgegeben, die den größten Anteil einer Werkzeuglängenkomponente enthält. In diesem Fall (falls die Funktion nicht aus einem anderen Grund einen Fehler liefert) ist der Returnwert 1. Die Abbildung der Werkzeuglängenkomponenten `L1` bis `L3` auf die Geometrieachsen 1 bis 3 wird dann durch den Inhalt des 3. Parameters `_MATRIX` vollständig beschrieben.

Mit dem Parameter `_COORD` kann angegeben werden, welches Koordinatensystem für die Geometrieachsen verwendet werden soll. Wird der Parameter `_COORD` nicht angegeben (Schreibweise `LENTOAX(_AXIND, _MATRIX)`), wird das WKS verwendet (Default).

**Beispiel:**

Standardfall Fräswerkzeug bei `G17`.

`L1` wirkt in Z (Applikate), `L2` wirkt in Y (Ordinate), `L3` wirkt in X (Abszisse).

Aufruf der Funktion in der Form:

**Status = LENTOAX(\_AXIND, \_MATRIX, "WCS")**

Der Ergebnisparameter `_AXIND` enthält dann die Werte:

`_AXIND[0] = 3`

`_AXIND[1] = 2`

`_AXIND[2] = 1`

oder kurz: ( 3, 2, 1)

Die zugehörige Matrix `_MATRIX` ist in diesem Fall:

$$\text{\_MATRIX} = \begin{matrix} & & 1 \\ & 1 & \\ 1 & & \end{matrix}$$

Ein Wechsel von `G17` nach `G18` oder `G19` ändert am Ergebnis nichts, da die Zuordnung der Längenkomponenten zu den Geometrieachsen sich in gleicher Weise ändert wie die Zuordnung von Abszisse, Ordinate und Applikate.

17.13 Werkzeuglängen L1, L2, L3 Zuordnung: LENTOAX

Es wird nun bei aktivem G17 eine Framedrehung um Z von 60 Grad programmiert, z.B. mit `rot Z60`. Die Richtung der Applikate (Z-Richtung) bleibt unverändert, der Hauptteil von L2 liegt nun in Richtung der neuen X-Achse, der Hauptteil von L1 in Richtung der negativen Y-Achse. Der Return-Status ist deshalb 1, `_AXIND` enthält die Werte (2, -3, 1).

Die zugehörige Matrix `_MATRIX` ist in diesem Fall:

$$\begin{matrix} & \text{sn} ( ) & \text{s} ( ) \\ \_MATRIX = & \text{s} ( ) & \text{sn} ( ) \\ & 1 & \end{matrix}$$

---

**Hinweis**

Weitere Informationen zu den oben genannten Koordinatensystemen siehe:

**Literatur:**

Programmierhandbuch Arbeitsvorbereitung; Werkzeugkorrekturen

---

## 17.14 Randbedingungen

### 17.14.1 Flache D-Nummern-Struktur

#### Schleifwerkzeuge

Mit der einfachen WZV (Flache D-Nr.) können keine Schleifwerkzeuge (WZ-Typen 400-499) definiert werden.

#### Satzsuchlauf

Die T-Nummernausgabe an PLC stößt hier einen Synchronisationsmechanismus in NCK an: bei absoluter, indirekter D-Programmierung liefert PLC die D-Werte über NC/PLC-Nahtstellen. NCK wartet, bis nach der Ausgabe einer T-Nummer eine Reaktion von PLC erfolgt ist "ich habe D-Nummer geschrieben". Bei Satzsuchlauf ohne Berechnung muss dieser Mechanismus der Synchronisation ausgeschaltet sein, solange bis das erste gültige T wieder ausgegeben wurde. Das bedeutet, dass NCK bei D-Programmierung nicht warten darf.

---

#### Hinweis

Wann bei Satzsuchlaufende die Hilfsfunktionen an PLC ausgegeben werden kann gesteuert werden mit dem Maschinendatum:

\$MC\_AUXFU\_AT\_BLOCK\_SEARCH\_END (Hilfsfunktionsausgabe nach Satzsuchlauf)

Automatisch bei End oder bei NC-Start.

---

#### REORG

Die (einzig) schreibbare Variable \$A\_MONIFACT, die hier definiert wird, ist durch ein Hauptlaufdatum hinterlegt. Da der Schreibvorgang hauptlaufsynchron stattfindet, brauchen für Reorg keine speziellen Betrachtungen angestellt werden.

### 17.14.2 SD42935 Erweiterungen

#### SD42935

Welche der Verschleißkomponenten im Zusammenhang mit den Funktionen `TOWMCS` und `TOWWCS` transformiert bzw. nicht transformiert werden sollen, kann festgelegt werden durch das Settingdatum:

SD42935 \$SSC\_WEAR\_TRANSFORM (Transformation der Verschleißwerte)

## 17.15 Beispiele

### 17.15.1 Orientierbare Werkzeugträger

#### 17.15.1.1 Beispiel: Orientierbare Werkzeugträger

##### Anforderung

Im nachfolgenden Beispiel wird ein Werkzeugträger verwendet, der durch eine Drehung um die Y-Achse vollständig beschrieben wird. Es reicht deshalb aus, mit dem Eintrag eines Wertes die Drehachse zu definieren (Satz N20).

In den Sätzen N50 bis N70 wird ein Schaffräser mit der Länge 20 mm und dem Radius 5 mm beschrieben.

Im Satz N90 wird eine Drehung von 37 Grad um die Y-Achse definiert.

Im Satz N120 wird die Werkzeugradiuskorrektur aktiviert, und es werden alle Einstellungen vorgenommen um die in den nachfolgenden Sätzen beschriebene Korrektur unter einer Drehung von 37 Grad um die y-Achse bearbeiten zu können.

```

N10                                     ; Definition des Werkzeugtraegers 1
N20 $TC_CARR8 [1] = 1                   ; Komponente der ersten Drehachse in Y-
                                        ; Richtung
N30
N40                                     ; Definition des
                                        ; Werkzeugkorrekturspeichers T1/D1
N50 $TC_DP1 [1,1] = 120                 ; Schaftfraeser
N60 $TC_DP3 [1,1] = 20                  ; Laenge 1
N70 $TC_DP6 [1,1] = 5                   ; Radius
N80
N90 ROT Y37                             ; 37 Grad-Drehung um y-Achse
N100
N110 X0 Y0 Z0 F10000
N120 G42 CUT2DF TCOFR TCARR = 1 T1 D1 X10
N130 X40
N140 Y40
N150 X0
N160 Y0
N170 M30

```

### 17.15.1.2 Beispiel Orientierbare Werkzeugträger mit drehbarem Tisch

#### Verwendung des MOV-Befehls

Für die Verwendung des `MOV`-Befehls wird angenommen, dass das Programm auf einer 5-Achs-Maschine abläuft, bei der eine Drehung der B-Achse das Werkzeug um die Y-Achse dreht:

```

N10 TRAORI ()
N20 X0 X0 Z0 B45 F2000           ; Einstellen der Werkzeugorientierung
N30 MOV=-10                       ; Zustellbewegung 10mm in
                                   ; Werkzeugrichtung
                                   ; (unter 45 Grad in der Y-Z-Ebene)
N40 MOV=AC(20)                   ; Rueckzug in Werkzeugrichtung auf
                                   ; Abstand
                                   ; 20mm vom Nullpunkt

```

#### Maschine mit drehbarem Tisch

Vollständige Definition für die Verwendung eines orientierbaren Werkzeugträgers mit drehbarem Tisch:

```

N10 $TC_DP1[1,1] = 120
N20 $TC_DP3[1,1] = 13           ; Werkzeuglänge 13mm

; Definition des Werkzeugträgers 1:

N30 $TC_CARR1[1] = 0           ; X-Komponente des 1. Offsetvektors
N40 $TC_CARR2[1] = 0           ; Y-Komponente des 1. Offsetvektors
N50 $TC_CARR3[1] = 0           ; Z-Komponente des 1. Offsetvektors

N60 $TC_CARR4[1] = 0           ; X-Komponente des 2. Offsetvektors
N70 $TC_CARR5[1] = 0           ; Y-Komponente des 2. Offsetvektors
N80 $TC_CARR6[1] = -15        ; Z-Komponente des 2. Offsetvektors

N90 $TC_CARR7[1] = 1           ; X-Komponente der 1. Achse
N100 $TC_CARR8[1] = 0          ; Y-Komponente der 1. Achse
N110 $TC_CARR9[1] = 0          ; Z-Komponente der 1. Achse

N120 $TC_CARR10[1] = 0         ; X-Komponente der 2. Achse
N130 $TC_CARR11[1] = 1         ; Y-Komponente der 2. Achse
N140 $TC_CARR12[1] = 0         ; Z-Komponente der 2. Achse

N150 $TC_CARR13[1] = 30        ; Drehwinkel der 1. Achse
N160 $TC_CARR14[1] = -30       ; Drehwinkel der 2. Achse

```

```

N170 $TC_CARR15 [1] = 0 ; X-Komponente des 3. Offsetvektors
N180 $TC_CARR16 [1] = 0 ; Y-Komponente des 3. Offsetvektors
N190 $TC_CARR17 [1] = 0 ; Z-Komponente des 3. Offsetvektors

N200 $TC_CARR18 [1] = 0 ; X-Komponente des 4. Offsetvektors
N210 $TC_CARR19 [1] = 0 ; Y-Komponente des 4. Offsetvektors
N220 $TC_CARR20 [1] = 15 ; Z-Komponente des 4. Offsetvektors

N230 $TC_CARR21 [1] = A ; Bezug fuer 1. Achse
N240 $TC_CARR22 [1] = B ; Bezug fuer 2. Achse
N250 $TC_CARR23 [1] = "P" ; Typ des Werkzeugtraegers

N260 X0 Y0 Z0 A0 B45 F2000
N270 TCARR=1 X0 Y10 Z0 T1 TCOABS
N280 PAROT
N290 X0 Y0 Z0
N300 G18 MOVT=AC(20)
N310 G17 X10 Y0 Z0
N320 MOVT=-10
N330 PAROTOF
N340 TCOFR
N350 X10 Y10 Z-13 A0 B0
N360 ROTS X-45 Y45
N370 X20 Y0 Z0 D0
N380 Y20
N390 X0 Y0 Z20
N400 M30

```

Die Definition des orientierbaren Werkzeugträgers ist vollständig angegeben. Die Komponenten, die den Wert 0 enthalten, müssten nicht angegeben werden, da sie ohnehin mit Null vorbesetzt sind.

In N270 wird der Werkzeugträger aktiviert.

Da in  $\$TC\_CARR21$  bzw. in  $\$TC\_CARR22$  auf die Maschinenachsen A und B verwiesen wird und TCOABS aktiv ist, werden die Einträge in  $\$TC\_CARR13$  und  $\$TC\_CARR14$  ignoriert, d.h. für die Drehung werden die Achsposition A0 B45 verwendet.

Die Drehung des 4. Offsetvektors (Länge 15 mm in Z-Richtung) um die B-Achse bewirkt eine Verschiebung des Nullpunktes um  $X10.607 [= 15 * \sin(45)]$  und  $Z-4.393 [= -15 * (1. - \cos(45))]$ . Diese Nullpunktverschiebung wird durch einen automatisch beschriebenen Basis- bzw. Systemframe berücksichtigt, so dass die Position X10.607 Y10.000 Z8.607 angefahren wird. In Z-Richtung ergibt sich dabei durch die Werkzeuganwahl eine zusätzliche Verschiebung von 13 mm, die Y-Komponente wird durch die Tischdrehung nicht beeinflusst.

In N280 wird eine Rotation entsprechend der Tischdrehung des orientierbaren Werkzeugträgers definiert. Die neue X-Richtung zeigt deshalb in Richtung der Winkelhalbierenden im 4. Quadranten, die neue Z-Achse in Richtung der Winkelhalbierenden im 1. Quadranten.

In N290 wird der Nullpunkt angefahren, d.h. die Maschinenposition X10.607 Y0 Z-4.393, da die Lage des Nullpunkts durch die Drehung nicht verändert wird.

In N300 wird in Y auf die Position Y33.000 verfahren, da G18 aktiv ist und die Y-Komponenten durch den aktiven Frame nicht beeinflusst werden. X- und Z-Position bleiben unverändert.

In N310 wird die Position X17.678 Y0 Z1.536 angefahren.

In N320 verändert sich als Folge des MOVF-Befehls nur die Z-Position auf den Wert -8.464. Da nur der Tisch drehbar ist, bleibt die Werkzeugorientierung unverändert parallel zur Maschinen-Z-Richtung, auch wenn die Z-Richtung des aktiven Frames um 45 Grad gedreht ist.

N330 löscht den Basis- bzw. Systemframe, die Framedefinition aus N280 wird damit wieder rückgängig gemacht.

In N340 wird mit TCOFR angegeben, dass der orientierbare Werkzeugträger entsprechend dem aktiven Frame ausgerichtet werden soll. Da wegen des PAROTOF-Befehls in N330 keine Drehung mehr aktiv ist, ergibt sich als Resultat die Grundstellung. Die Frameverschiebung wird 0.

In N350 wird deshalb die Position X10 X10 Z0 (= Z-13 + Werkzeuglänge) angefahren. Achtung: Durch die gleichzeitige Programmierung der beiden Rundachsen A und B wird die tatsächliche Lage des orientierbaren Werkzeugträgers mit der in N340 verwendeten in Übereinstimmung gebracht. Die Position, die von den drei Linearachsen angefahren wird ist davon jedoch unabhängig.

In N360 wird mit Hilfe von Raumwinkeln eine Ebene definiert deren Schnittgeraden in der X-Z- und in der Y-Z-Ebene jeweils einen Winkel von +45 Grad bzw. -45 Grad mit der X- bzw. der Y-Achse bilden. Die so definierte Ebene hat damit folgende Lage: die Flächennormale zeigt in Richtung der Raumdiagonalen.

In N370 wird im neuen Koordinatensystem auf die Position X20 Y0 Z0 verfahren. Da gleichzeitig das Werkzeug mit D0 abgewählt wird, gibt es in Z keinen zusätzlichen Offset mehr. Da die neue X-Achse in der alten X-Z-Ebene liegt, wird in diesem Satz die Maschinenposition X14.142 Y0 Z-14.142 erreicht.

In N380 verfährt im gedrehten Koordinatensystem nur die Y-Achse. Das führt zu einer Bewegung aller drei Maschinenachsen. Die Maschinenposition ist X5.977 Y16.330 Z-22.307.

In N390 wird ein Punkt auf der neuen Z-Achse angefahren. Bezogen auf die Maschinenachsen liegt er deshalb auf der Raumdiagonalen. Alle drei Achsen erreichen deshalb die Position 11.547.

### 17.15.1.3 Beispiel Werkzeuggrundorientierung

#### Grundorientierung in der Winkelhalbierenden

Es wird ein Fräs Werkzeug mit der Länge  $L_1=10$  definiert, dessen Grundorientierung jedoch in der Winkelhalbierenden der X-Z-Ebene liegt.

```
N10    $TC_DP1 [1,1]=120
N20    $TC_DP3 [1,1]=10
N30    $TC_DPV [1,1] = 0
N40    $TC_DPV3 [1,1] = 1
N50    $TC_DPV4 [1,1] = 0
N60    $TC_DPV5 [1,1] = 1
N70    g17 f1000 x0 y0 z0 t1 d1
N80    movt=10
N80    m30
```

#### Erläuterungen zum Beispiel:

In  $N_{10}$  bis  $N_{60}$  wird ein Fräs Werkzeug mit der Länge  $L_1=10$  definiert ( $N_{20}$ ). Die Grundorientierung liegt in der Winkelhalbierenden der X-Z-Ebene  $N_{40}$  bis  $N_{60}$ .

In  $N_{70}$  wird das Werkzeug aktiviert und die Nullposition angefahren. Wegen der Werkzeuglänge ergeben sich deshalb in diesem Satz die Maschinenpositionen  $X_0 Y_0 Z_{10}$ .

In  $N_{80}$  wird eine inkrementelle Verfahrbewegung von 10 in Werkzeugrichtung ausgeführt. Die resultierenden Achspositionen sind somit  $X_{7.071} Y_0 Z_{17.071}$ .

### 17.15.1.4 Korrekturen einsatzort- und werkstückspezifisch einrechnen

#### Werkzeug mit Adapter

Im nachfolgenden Programmbeispiel ist ein Werkzeug mit Adapter und orientierbarem Werkzeugträger definiert. Um die Verhältnisse überschaubar zu halten, ist beim Werkzeug selbst, bei den Summen- und Einsatzkorrekturen sowie beim Adapter jeweils nur die Länge  $L_1$  von Null verschieden. Die Offsetvektoren des orientierbaren Werkzeugträgers sind alle Null.

```
N10    $TC_TP2 [1] = "MillingTool"           ; Bezeichner
N20    $TC_TP7 [1] = 9                       ; Platztype
N30    $TC_TP8 [1] = 2                       ; Status : Freigeben und nicht
                                           gesperrt

; D corr. D=1

N40    $TC_DP1 [1,1]=120                     ; WZ-Typ - Fräsen
N50    $TC_DP3 [1,1]=; Längenkorrekturvektor
```



```

N60    $TC_DP12 [1,1]= ; Verschleiß
N70    $TC_SCP13 [1,1]=0.1 ; Summenkoorektur DL=1
N80    $TC_ECP13 [1,1]=0.01 ; Einsatzkorrektur DL=1
N90    $TC_ADPTT [1]=5 ; Adaptertransformation
N100   $TC_ADPT1 [1]=0.001 ; Adaptermaß

; Magazinedaten
N110   $TC_MAP1 [1]=3 ; Art des Magazins: Revolver
N120   $TC_MAP2 [1]="Revolver" ; Bezeichner eines Magazins
N130   $TC_MAP3 [1]=17 ; Zustand des Magazins
N140   $TC_MAP6 [1]=1 ; Dimension - Zeile
N150   $TC_MAP7 [1]=2 ; Dimension - Spalte -> 2 Plätze
N160   $TC_MPP1 [1,1]=1 ; Platzart
N170   $TC_MPP2 [1,1]=9 ; Platztype
N180   $TC_MPP4 [1,1]=2 ; Platzzustand
N190   $TC_MPP7 [1,1]=1 ; Adapter auf Platz bringen
N200   $TC_MPP6 [1,1]=1 ; T-Nummer "MillingTool"
N210   $TC_MAP1 [9999]= 7 ; Art des Magazins: Zw.speicher
N220   $TC_MAP2 [9999]="Zw.speicher" ; Bezeichner eines Magazins
N230   $TC_MAP3 [9999]=17 ; Zustand des Magazins
N240   $TC_MAP6 [9999]=1 ; Dimension - Zeile
N250   $TC_MAP7 [9999]=1 ; Dimension - Spalte -> 1 Platz
N260   $TC_MPP1 [9999,1]=2 ; Platzart
N270   $TC_MPP2 [9999,1]=9 ; Platztype
N280   $TC_MPP4 [9999,1]=2 ; Platzzustand
N290   $TC_MPP5 [9999,1]=1 ; Spindelnr. 1
N300   $TC_MDP2 [1,1]=0 ; Distanz Spindel zu Mag.1

; Definition des Werkzeugträgers 1
N310   $TC_CARR10 [1] = 1 ; Komponente 2. Drehachse in X-
; Richtung
N320   $TC_CARR14 [1] = 45 ; Drehwinkel der 2. Achse
N330   $TC_CARR23 [1] = "T" ; Tool Mode
N340   Stopre
N350   $SSC_WEAR_TRANSFORM = 'B101'
N360   T0 D0 DL=0
N370   ROT X30
N380   G90 G1 G17 F10000 X0 Y0 Z0
N390   T="MillingTool" X0 Y0 Z0 TOWSTD ; X 0.000 Y11.110 Z 0.001
N400   T="MillingTool" X0 Y0 Z0 TOWMCS ; X 0.000 Y10.100 Z 1.011
N410   T="MillingTool" X0 Y0 Z0 TOWWCS ; X 0.000 Y 9.595 Z 0.876
N420   TCARR=1 X0 Y0 Z0 ; X 0.000 Y 6.636 Z 8.017
N430   G18 X0 Y0 Z0 ; X10.100 Y-0.504 Z 0.876
N440   m30

```

## Erläuterungen zum Beispiel oben

Beginnend mit Satz N390 wird in unterschiedlichen Varianten jeweils die Position  $X_0 Y_0 Z_0$  angefahren. Die erreichten Maschinenpositionen sind in den Sätzen jeweils im Kommentar angegeben. Im Anschluss an das Programm wird erläutert, wie die Positionen zu Stande kommen.

N390: Die Adaptertransformation 5 (Satz N90) transformiert die Länge  $L_1$  in die Länge  $L_2$ . Dieser Transformation ist lediglich das Adaptermaß selbst nicht unterworfen. Damit ergibt sich der Y-Wert ( $L_2$  bei G17) aus der Summe aus Werkzeuglänge (10), Werkzeugverschleiß (1), Summenkorrektur (0.1) und Einsatzkorrektur (0.01). Das Adaptermaß (0.001) steht in Z ( $L_1$ ).

N400: In Satz N350 wurden die Bits 0 und 2 gesetzt im Settingdatum:

SD42935 \$SC\_WEAR\_TRANSFORM (Transformationen für Werkzeugkomponenten)

Dies bedeutet, dass wegen TOWMCS in Satz N400 der Werkzeugverschleiß und die Einsatzkorrektur der Adaptertransformation nicht unterworfen werden. Die Summe dieser beiden Korrekturen ist 1.01. Um dieses Maß erhöht sich deshalb die Z-Position und vermindert sich entsprechend die Y-Position gegenüber Satz N390.

In N410 ist TOWMCS aktiv. Die Summe aus Werkzeugverschleiß und Einsatzkorrektur wirkt deshalb im aktiven Werkstückkoordinatensystem. In Satz N370 wurde eine Drehung um die X-Achse um 30 Grad aktiviert. Der ursprüngliche Korrekturwert von 1.01 in Z-Richtung ergibt deshalb eine neue Z-Komponente von 0.875 ( $= 1.01 * \cos(30)$ ) und eine neue Y-Komponente von -0.505 ( $= 1.01 * \sin(30)$ ). Zu der, wie in Satz N390, gebildeten Summe aus Werkzeuglänge, Summenkorrektur und Adaptermaß addiert, ergibt sich das im Programm-Kommentar angegebene Maß.

In N420 wird zusätzlich ein orientierbarer Werkzeugträger aktiviert. Dieser führt eine Drehung um 45 Grad um die X-Achse aus (s. N310 - N330). Da alle Offsetvektoren des Werkzeugträgers Null sind, ergibt sich keine zusätzliche Nullpunktverschiebung. Der orientierbare Werkzeugträger wirkt auf die die Summe aus Werkzeuglänge, Summenkorrektur und Adaptermaß. Der resultierende Teilvektor ist  $X_0 Y_{7.141} Z_{7.142}$ . Darauf wird wie in Satz N410 die im WKS bewertete Summe aus Werkzeugverschleiß und Einsatzkorrektur addiert.

In N430 wird G18 aktiviert. Die Komponenten der Summe aus Werkzeuglänge, Summenkorrektur und Adaptermaß werden entsprechend getauscht. Auf diesen neuen Vektor wirkt der orientierbare Werkzeugträger unverändert (Drehung um X-Achse um 45 Grad). Dabei ergibt sich der resultierende Teilvektor  $X_{10.100} Y_{0.0071} Z_{0.0071}$ . Der aus Werkzeugverschleiß und Einsatzkorrektur gebildete Vektor ( $X_0 Y_{-0.505} Z_{0.875}$ ) wird durch den Wechsel der Ebene nicht beeinflusst. Die Summe beider Vektoren ergibt das im Kommentar von N430 angegebene Maß.

## 17.15.2 Beispiele 3-6: Funktion SETTCOR für WZ-Umgebungen

### Beispiel 3

```

N10    def real _CORVAL[3]
N20    $TC_DP1[1,1] = 120                ; Fräswerkzeug
N30    $TC_DP3[1,1] = 10.0              ; Geometrie L1
N40    $TC_DP12[1,1]= 1.0               ; Verschleiss L1
N50    _CORVAL[0] = 0.333
N60    t1 d1 g17 g0
N70    r1 = settcor(_CORVAL, "GW", 0, 2, 2)
N80    t1 d1 x0 y0 z0                    ; ==> MCS-Position X0.000 Y0.000
                                           Z0.333
N90    M30

```

\_CORCOMP ist 2, deshalb wird die in Z-Richtung wirkende Korrektur in die Geometrie-Komponente eingetragen (der alte Wert wird überschrieben), und der Verschleißwert wird gelöscht. Die resultierende Werkzeugesamtlänge ist somit:

$$L_1 = 0.333 + 0.0 = 0.333.$$

### Beispiel 4

```

N10    def real _CORVAL[3]
N20    $TC_DP1[1,1] = 120                ; Fräswerkzeug
N30    $TC_DP3[1,1] = 10.0              ; Geometrie L1
N40    $TC_DP12[1,1]= 1.0               ; Verschleiss L1
N50    _CORVAL[0] = 0.333
N60    t1 d1 g17 g0
N70    r1 = settcor(_CORVAL, "GW", 0, 3, 2)
N80    t1 d1 x0 y0 z0                    ; ==> MCS-Position X0.000 Y0.000
                                           Z11.333
N90    M30

```

\_CORCOMP ist 3, deshalb werden Verschleißwert und Korrekturwert zur Geometrie-Komponente addiert und die Verschleißkomponente gelöscht. Die resultierende Werkzeugesamtlänge ist somit  $L_1 = 11.333 + 0.0 = 11.333$ .

## Beispiel 5

```

N10   def real _CORVAL[3]
N20   $TC_DP1[1,1] = 120           ; Fräswerkzeug
N30   $TC_DP3[1,1] = 10.0         ; Geometrie L1
N40   $TC_DP12[1,1] = 1.0         ; Verschleiss L1
N50   _CORVAL[0] = 0.333
N60   t1 d1 g17 g0
N70   r1 = settcor(_CORVAL, "GW", 0, 3, 0)
N80   t1 d1 x0 y0 z0              ;==> MCS-Position X0.333 Y0.000
                                       Z11.000
N90   M30

```

\_CORCOMP ist 3 wie im vorigen Beispiel, die Korrektur wirkt jetzt jedoch auf die Geometrieachse mit dem Index 0 (X-Achse), der wegen  $G_{17}$  bei einem Fräswerkzeug die Werkzeugkomponenten  $L_3$  zugeordnet ist. Der Aufruf von SETTCOR beeinflusst deshalb die Werkzeugparameter \$TC\_DP3 und \$TC\_DP12 nicht. Der Korrekturwert wird stattdessen in \$TC\_DP5 eingetragen.

## Beispiel 6

```

N10   def real _CORVAL[3]
N20   $TC_DP1[1,1] = 500           ; Drehwerkzeug
N30   $TC_DP3[1,1] = 10.0         ; Geometrie L1
N40   $TC_DP4[1,1] = 15.0         ; Geometrie L2
N50   $TC_DP12[1,1] = 10.0        ; Verschleiss L1
N60   $TC_DP13[1,1] = 0.0         ; Verschleiss L2
N70   _CORVAL[0] = 5.0
N80   rot y-30
N90   t1 d1 g18 g0
N100  r1 = settcor(_CORVAL, "GW", 0, 3, 1)
N110  t1 d1 x0 y0 z0              ; ==> MCS-Position X24.330
                                       Y0.000 Z17.500
N120  M30

```

Das Werkzeug ist ein Drehwerkzeug. In Satz N80 wird eine Framedrehung aktiviert, so dass das Basiskoordinatensystem (BCS) gegenüber dem Werkstückkoordinatensystem (WKS) gedreht ist. Der Korrekturwert (N70) wirkt im WKS auf die Geometrieachse mit dem Index 1, d.h. da G18 aktiv ist, auf die X-Achse. Da "\_CORRMODE = 3" gilt, muss der Werkzeugverschleiß in Richtung der X-Achse des WCS nach dem Ausführen von N100 Null werden. Der Inhalt der relevanten Werkzeugparameter ist deshalb am Programmende:

```

$TC_DP3 [1,1]           : 21.830           ; Geometrie L1
$TC_DP4 [1,1]           : 21.830           ; Geometrie L2
$TC_DP12 [1,1]          : 2.500            ; Verschleiss L1
$TC_DP13 [1,1]          : -4.330           ; Verschleiss L2

```

Der gesamte Verschleiß einschließlich \_CORVAL wird auf die X'-Richtung im WCS projiziert. Das ergibt den Punkt P2. Die Koordinaten dieses Punktes (gemessen in X-Y-Koordinaten) werden in die Geometriekomponente des Werkzeugs eingetragen. Im Verschleiß bleibt der Differenzvektor  $P_2 - P_1$  übrig. Damit hat der Verschleiß keine Komponente mehr in Richtung von \_CORVAL.

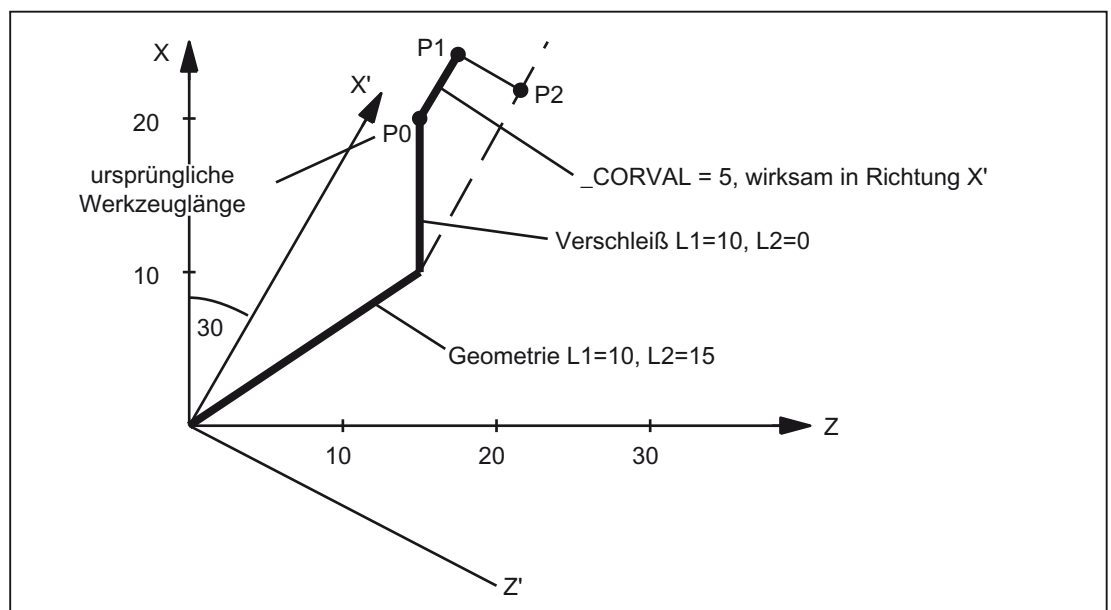


Bild 17-54 Werkzeuglängenkorrektur, Beispiel 6

Setzt man das Beispielprogramm nach N110 mit den folgenden Anweisungen fort:

```

N120  _CORVAL[0] = 0.0
N130  r1 = settcor(_CORVAL, "GW", 0, 3, 0)
N140  t1 d1 x0 y0 z0           ;==> MCS-Position X24.330 Y0.000
                                   Z17.500

```

So wird der restliche Verschleiß vollständig in die Geometrie übernommen, da die Korrektur jetzt in der Z'-Achse wirkt (Parameter `_GEOAX` ist 0). Da der neue Korrekturwert 0 ist, darf sich die Werkzeuggesamtlänge und damit auch die in `N140` angefahrene Position nicht verändern. Wäre `_CORVAL` in `N120` ungleich 0, würde sich eine neue Werkzeuggesamtlänge und damit auch eine veränderte Position in `N140` ergeben, der Verschleißanteil der Werkzeuglänge wäre jedoch in jedem Fall Null, d.h. die gesamte Werkzeuglänge ist anschließend in jedem Fall im Geometrieanteil des Werkzeugs enthalten.

Das gleiche Ergebnis wie mit dem zweimaligen Aufruf der Funktion `SETTCOR` mit dem Parameter `_CORCOMP = 0` erreicht man auch durch einmaligen Aufruf mit `_CORCOMP = 1` (vektorielle Korrektur):

```
N10    def real _CORVAL[3]
N20    $TC_DP1[1,1] = 500                ; Drehwerkzeug
N30    $TC_DP3[1,1] = 10.0              ; Geometrie L1
N40    $TC_DP4[1,1] = 15.0              ; Geometrie L2
N50    $TC_DP12[1,1]= 10.0              ; Verschleiss L1
N60    $TC_DP13[1,1]= 0.0               ; Verschleiss L2
N70    _CORVAL[0] = 0.0
N71    _CORVAL[1] = 5.0
N72    _CORVAL[2] = 0.0
N80    rot y-30
N90    t1 d1 g18 g0
N100   r1 = settcor(_CORVAL, "GW", 1, 3, 1)
N110   t1 d1 x0 y0 z0                   ; ==> MCS-Position X24.330 Y0.000
                                           Z17.500
N120   M30
```

In diesem Fall sind alle Verschleißkomponenten des Werkzeugs sofort nach dem ersten Aufruf von `SETTCOR` in `N100` gleich Null.

### Beispiel 7

```
N10    def real _CORVAL[3]
N20    $TC_DP1[1,1] = 500                ; Drehwerkzeug
N30    $TC_DP3[1,1] = 10.0              ; Geometrie L1
N40    $TC_DP4[1,1] = 15.0              ; Geometrie L2
N50    $TC_DP12[1,1]= 10.0              ; Verschleiss L1
N60    $TC_DP13[1,1]= 0.0               ; Verschleiss L2
N70    _CORVAL[0] = 5.0
N80    rot y-30
N90    t1 d1 g18 g0
N100   r1 = settcor(_CORVAL, "GW", 3, 3)
N110   t1 d1 x0 y0 z0                   ; ==> MCS-Position X25.000 Y0.000
                                           Z15.000
```

Gegenüber Beispiel 6 ist hier der Parameter `_CORCOMP = 3`, deshalb kann die Angabe des Parameters `_GEOAX` entfallen. Der in `_CORVAL[0]` enthaltene Wert wirkt jetzt unmittelbar auf die Werkzeugkomponente L1, die Drehung in `N80` hat auf das Ergebnis keinen Einfluss, die Verschleißkomponenten in `$TC_DP12` wird zusammen mit `_CORVAL[0]` in den Geometrieanteil übernommen, so dass wegen `$TC_DP13` die gesamte Werkzeuglänge bereits nach dem ersten Aufruf von `SETTCOR` in `N100` im Geometrieteil des Werkzeugs steht.

### Beispiel 8

```

N10    def real _CORVAL[3]
N20    $TC_DP1[1,1] = 500                ; Drehwerkzeug
N30    $TC_DP3[1,1] = 10.0              ; Geometrie L1
N40    $TC_DP4[1,1] = 15.0              ; Geometrie L2
N50    $TC_DP5[1,1] = 20.0              ; Geometrie L3
N60    $TC_DP12[1,1] = 10.0             ; Verschleiss L1
N70    $TC_DP13[1,1] = 0.0              ; Verschleiss L2
N80    $TC_DP14[1,1] = 0.0              ; Verschleiss L3
N90    $SSC_WEAR_SIGN = TRUE
N100   _CORVAL[0] = 10.0
N110   _CORVAL[1] = 15.0
N120   _CORVAL[2] = 5.0
N130   rot y-30
N140   t1 d1 g18 g0
N150   r1 = settcor(_CORVAL, "W", 1, 1)
N160   t1 d1 x0 y0 z0                    ; ==> MCS-Position X7.990 Y25.000
                                           Z31.160

```

In `N90` ist das Settingdatum gesetzt:

`SD42930 $SSC_WEAR_SIGN` (Vorzeichen des Verschleißes)

D.h. der Verschleiß muss mit negativem Vorzeichen bewertet werden.

Die Korrektur ist vektoriell ( $\_CORCOMP = 1$ ), und der Korrekturvektor muss zum Verschleiß addiert werden ( $\_CORMODE = 1$ ). Die geometrischen Verhältnisse in der Z-X-Ebene sind in dem folgenden Bild dargestellt:

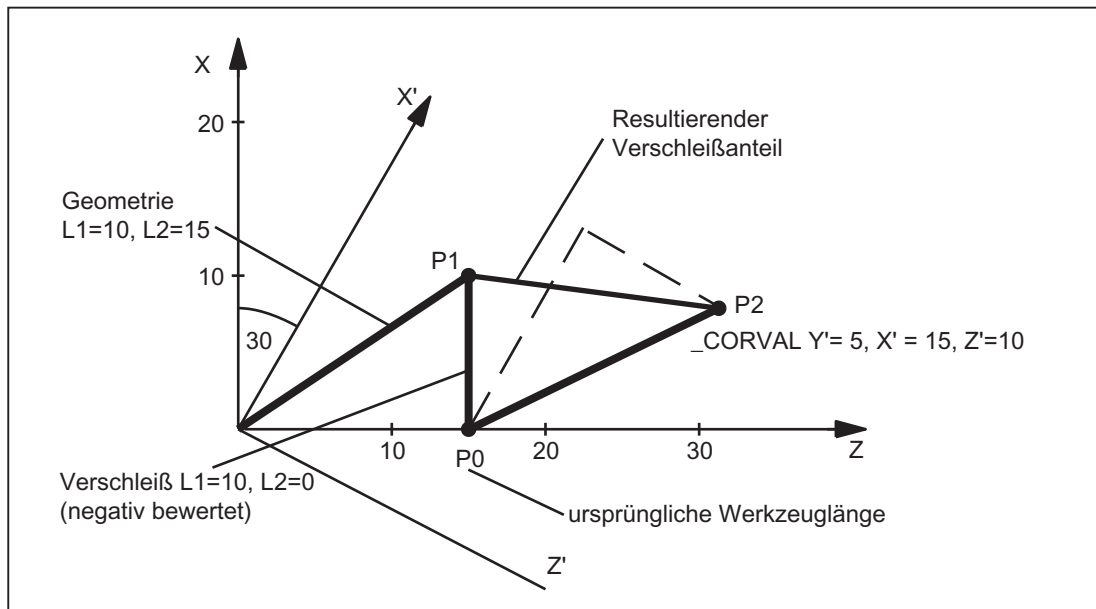


Bild 17-55 Werkzeuglängenkorrektur, Beispiel 8

Wegen  $\_CORMODE = 1$  bleibt der Geometrieanteil des Werkzeugs unverändert. Der im WKS (Drehung um y-Achse) definierte Korrekturvektor muss so in den Verschleißanteil übernommen werden, dass die gesamte Werkzeuglänge im Bild auf den Punkt  $P_2$  verweist. Der resultierende Verschleißanteil des Werkzeugs ist deshalb durch die Distanz der beiden Punkte  $P_1$  und  $P_2$  gegeben.

Da der Verschleiß wegen des Settingdatums SD42930 aber negativ bewertet wird, muss die so ermittelte Korrektur mit negativem Vorzeichen in den Korrekturspeicher eingetragen werden. Der Inhalt der relevanten Werkzeugparameter ist am Programmende deshalb:

```

$TC_DP3 [1,1]      : 10.000          ; Geometrie L1 (unverändert)
$TC_DP4 [1,1]      : 15.000          ; Geometrie L2 (unverändert)
$TC_DP5 [1,1]      : 10.000          ; Geometrie L3 (unverändert)

$TC_DP12 [1,1]     : 2.010           ; Verschleiss L1
                    ; (= 10 -15*cos(30) + 10*sin(30))
$TC_DP13 [1,1]     : -16.160        ; Verschleiss L2
                    ; (= -15*sin(30) - 10*cos(30))
$TC_DP14 [1,1]     : -5.000          ; Verschleiss L3
    
```

An der  $L_3$ -Komponente in Y-Richtung lässt sich die Wirkung des Settingdatums SD42930 ohne die zusätzliche Komplizierung durch die Framedrehung erkennen.



**Beispiel 9:**

2 (Werkzeuglänge muss in der Durchmesserachse mit dem Faktor 0.5 bewertet werden) sei der Wert des Maschinendatums:

MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK (Definition der WZ-Parameter)

X sei Durchmesserachse:

```

N10    def real _LEN[11]
N20    def real _CORVAL[3]
N30    $TC_DP1[1,1] = 500
N40    $TC_DP2[1,1] = 2
N50    $TC_DP3[1,1] = 3.
N60    $TC_DP4[1,1] = 4.
N70    $TC_DP5[1,1] = 5.
N80    _CORVAL[0] = 1.
N90    _CORVAL[1] = 1.
N100   _CORVAL[2] = 1.
N110   t1 d1 g18 g0 x0 y0 z0                ; ==> MCS-Position X1.5 Y5 Z4
N120   r1 = settcor(_CORVAL, "g", 1, 1)
N130   t1 d1 x0 y0 z0                       ; ==> MCS-Position X2.5 Y6 Z5
N140   r3 = $TC_DP3[1,1]                    ; = 5. = (3.000 + 2. * 1.000)
N150   r4 = $TC_DP4[1,1]                    ; = 5. = (4.000 + 1.000)
N160   r5 = $TC_DP5[1,1]                    ; = 6. = (5.000 + 1.000)
N170   m30

```

Die Korrektur der Werkzeuglänge soll in jeder Achse 1 mm sein (N80 bis N100).

In den Längen  $L_2$  und  $L_3$  wird deshalb zur Originallänge jeweils 1 mm addiert.

Zur Originalwerkzeuglänge in  $L_1$  wird dagegen der doppelte Korrekturwert (2 mm) addiert, damit sich die Gesamtwerkzeuglänge wie verlangt um 1 mm ändert. Im Vergleich der angefahrenen Positionen in den Sätzen N110 und N130 erkennt man, dass sich jede Achsposition um 1 mm verändert hat.

## 17.16 Datenlisten

### 17.16.1 Maschinendaten

#### 17.16.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
18082	MM_NUM_TOOL	Anzahl der Werkzeuge, die NCK verwalten kann (SRAM)
18088	MM_NUM_TOOL_CARRIER	Maximale Anzahl der definierter Werkzeugträger
18094	MM_NUM_CC_TDA_PARAM	Anzahl der Werkzeugdaten (SRAM)
18096	MM_NUM_CC_TOA_PARAM	Anzahl der Daten, pro Werkzeugschneide für Compilezyklen (SRAM)
18100	MM_NUM_CUTTING_EDGES_IN_TOA	Werkzeugkorrekturen im TOA-Bereich (SRAM)
18102	MM_TYPE_OF_CUTTING_EDGE	Art der D-Nummer Programmierung (SRAM)
18105	MM_MAX_CUTTING_EDGE_NO	Maximaler Wert der D-Nummer
18106	MM_MAX_CUTTING_EDGE_PERTOOL	Maximale Anzahl der D-Nummern pro Werkzeug
18108	MM_NUM_SUMCORR	Anzahl aller Summenkorrekturen im NCK
18110	MM_MAX_SUMCORR_PER_CUTTEDGE	Anzahl von Summenkorrekturen pro Schneide
18112	MM_KIND_OF_SUMCORR	Eigenschaften der Summenkorrekturen im TO-Bereich (SRAM)
18114	MM_ENABLE_TOOL_ORIENT	Werkzeugschneiden Orientierung zuordnen
18116	MM_NUM_TOOL_ENV	Werkzeugumgebungen im TO-Bereich (SRAM)

#### 17.16.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20096	T_M_ADDRESS_EXT_IS_SPINO	Bedeutung der Adresserweiterung bei T, M Werkzeugwechsel
20110	RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach RESET/Teileprogrammende
20120	TOOL_RESET_VALUE	Werkzeug Längenkorrektur im Hochlauf (Reset/TP-Ende)
20121	TOOL_PRESEL_RESET_VALUE	Vorgewähltes Werkzeug bei RESET
20125	CUTMOD_ERR	Fehlerbehandlung für die Funktion CUTMOD
20126	TOOL_CARRIER_RESET_VALUE	Wirksamer Werkzeugträger bei RESET
20127	CUTMOD_INIT	CUTMOD bei POWER ON initialisieren
20130	CUTTING_EDGE_RESET_VALUE	Werkzeugschneide Längenkorrektur im Hochlauf (Reset/TP-Ende)
20132	SUMCORR_RESET_VALUE	Wirksame Summenkorrektur bei RESET

Nummer	Bezeichner: \$MC_	Beschreibung
20140	TRAFO_RESET_VALUE	Transformationsdatensatz im Hochlauf (Reset/TP-Ende)
20180	TOCARR_ROT_ANGLE_INCR[i]	Rundachseninkrement des orientierbaren Werkzeugträgers
20182	TOCARR_ROT_ANGLE_OFFSET[i]	Rundachsoffset des orientierbaren Werkzeugträgers
20184	TOCARR_BASE_FRAME_NUMBER	Nummer des Basisframes für Aufnahme des Tischoffsets
20188	TOCARR_FINE_LIM_LIN	Limit inere Feinverschiebung TCARR
20190	TOCARR_FINE_LIM_ROT	Limit der rotatorische Feinverschiebung TCARR
20202	WAB_MAXNUM_DUMMY_BLOCKS	Max. Satzanzahl ohne Verfahrbewegung bei WAB
20204	WAB_CLEARANCE_TOLERANCE	Richtungsumkehr bei WAB
20210	CUTCOM_CORNER_LIMIT	Maximalwinkel für Ausgleichssätze bei WRK
20220	CUTCOM_MAX_DISC	Maximaler Wert für DISC
20230	CUTCOM_CURVE_INSERT_LIMIT	Maximalwert für Schnittpunktberechnung bei WRK
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	Sätze für vorausschauende Konturberechnung bei WRK
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	Maximale Satzanzahl ohne Verfahrbewegung bei WRK
20252	CUTCOM_MAXNUM_SUPPR_BLOCKS	Maximale Satzanzahl mit Korrekturunterdrückung
20256	CUTCOM_INTERS_POLY_ENABLE	Schnittpunktverfahren für Polynome möglich
20270	CUTTING_EDGE_DEFAULT	Grundstellung der Werkzeugschneide ohne Programmierung
20272	SUMCORR_DEFAULT	Grundstellung Summenkorrektur ohne Programm
20360	TOOL_PARAMETER_DEF_MASK	Definition der Werkzeug-Parameter
20390	TOOL_TEMP_COMP_ON	Aktivierung der Temperaturkompensation für Werkzeuglänge
20392	TOOL_TEMP_COMP_LIMIT	Maximale Temperaturkompensation für Werkzeuglänge
20610	ADD_MOVE_ACCEL_RESERVE	Beschleunigungsreserve für überlagerte Bewegungen
21080	CUTCOM_PARALLEL_ORI_LIMIT	Minimaler Winkel (Bahntangente und Werkzeugorientierung) bei 3D-Werkzeugradiuskorrektur
22530	TOCARR_CHANGE_M_CODE	M-Code bei Werkzeugträgerwechsel
22550	TOOL_CHANGE_MODE	Neue WZK bei M-Funktion
22560	TOOL_CHANGE_M_CODE	M-Funktion für Werkzeugwechsel
22562	TOOL_CHANGE_ERROR_MODE	Verhalten bei Fehlern im Werkzeugwechsel
24558	TRAFO5_JOINT_OFFSET_PART_1	Vektor kinematischer Versatz im Tisch, Trafo 1
24658	TRAFO5_JOINT_OFFSET_PART_2	Vektor kinematischer Versatz im Tisch, Trafo 2
28085	MM_LINK_TOA_UNIT	Zuordnung der TO-Einheit zu einem Kanal (SRAM)

## 17.16.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32750	TEMP_COMP_TYPE	Temperaturkompensationstyp

## 17.16.2 Settingdaten

## 17.16.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42442	TOOL_OFFSET_INCR_PROG	Werkzeuglängenkorrekturen
42470	CRIT_SPLINE_ANGLE	Ecken-Grenzwinkel für Kompressor
42480	STOP_CUTCOM_STOPRE	Alarmreaktion bei Werkzeugradiuskorrektur und Vorlaufstopp
42494	CUTCOM_ACT_DEACT_CTRL	An-/Abfahrverhalten bei Wz.-Radiuskorrektur
42496	CUTCOM_CLSDT_CONT	Verhalten der Werkzeugradiuskorrektur bei geschlossener Kontur
42900	MIRROR_TOOL_LENGTH	Vorzeichenwechsel Werkzeuglängen beim Spiegeln
42910	MIRROR_TOOL_WEAR	Vorzeichenwechsel Werkzeugverschleiß beim Spiegeln
42920	WEAR_SIGN_CUTPOS	Vorzeichen des Verschleißes bei Werkzeugen mit Schneidenlage
42930	WEAR_SIGN	Vorzeichen des Verschleißes
42935	WEAR_TRANSFORM	Transformationen für Werkzeugkomponenten
42940	TOOL_LENGTH_CONST	Wechsel der Werkzeuglängenkomponenten bei Ebenenwechsel
42950	TOOL_LENGTH_TYPE	Zuordnung der Werkzeuglängenkompensation unabhängig vom Werkzeugtyp
42960	TOOL_TEMP_COMP	Temperaturkompensationswert bezogen auf das Werkzeug
42974	TCARR_FINE_CORRECTION	Feinverschiebung TCARR ein/aus
42984	CUTDIRMOD	Modifikation von \$P_AD[2] bzw. \$P_AD[11]

**17.16.3 Signale****17.16.3.1 Signale von Kanal**

<b>Signalname</b>	<b>SINUMERIK 840D sl</b>	<b>SINUMERIK 828D</b>
T-Funktion 1 Änderung	DB21, ... .DBX61.0	-
D-Funktion 1 Änderung	DB21, ... .DBX62.0	-
T-Funktion 1	DB21, ... .DBB116-119	DB2500.DBD2000
D-Funktion 1	DB21, ... .DBB128-129	DB2500.DBD5000
Aktive G-Funktion der Gruppe 7	DB21, ... .DBB214	DB3500.DBB6
Aktive G-Funktion der Gruppe 16	DB21, ... .DBB223	DB3500.DBB15
Aktive G-Funktion der Gruppe 17	DB21, ... .DBB224	DB3500.DBB16
Aktive G-Funktion der Gruppe 18	DB21, ... .DBB225	DB3500.DBB17
Aktive G-Funktion der Gruppe 23	DB21, ... .DBB230	DB3500.DBB22



## Z1: NC/PLC-Nahtstellensignale

### 18.1 Diverse Nahtstellensignale und Funktionen (A2)

#### 18.1.1 Signale von PLC an NC (DB10)

<b>DB10</b> <b>DBX56.4 - DBX56.7</b>	<b>Schlüsselschalter-Stellung 0 bis 3</b>				
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch			
Signalbedeutung	Abhängig von der Schlüsselschalter-Stellung kann der Zugriff auf bestimmte Elemente im NCK gesperrt bzw. freigegeben werden.				
	<ul style="list-style-type: none"> <li>• Schlüsselschalter-Stellung 0 hat die geringsten Zugriffsrechte</li> <li>• Schlüsselschalter-Stellung 3 hat die höchsten Zugriffsrechte</li> </ul>				
	Die NC/PLC-Nahtstellensignale der Schlüsselschalter-Stellungen 1 bis 3 können entweder direkt vom Schlüsselschalter der Maschinensteuertafel oder vom PLC-Anwenderprogramm vorgeben werden.				
	Es darf jeweils nur ein Bit gesetzt sein. Sind gleichzeitig mehrere Bits gesetzt, wird steuerungsintern Schalterstellung 3 aktiviert.				
	Schlüsselschalter-Stellung	DBX56.7	DBX56.6	DBX56.5	DBX56.4
0	0	0	0	1	
1	0	0	1	0	
2	0	1	0	0	
3	1	0	0	0	
korrespondierend mit ...	Maschinendaten für Schutzstufen: MD11612, MD51044 - MD51064, MD51070 - MD51073, MD51199 - MD51211, MD51215 - MD51225, MD51235 Verriegelung über Kennwort				

#### 18.1.2 Anwahl-/Statussignale von HMI an PLC (DB10)

<b>DB10</b> <b>DBX103.0</b>	<b>Ferndiagnose aktiv</b>
Flankenauswertung: nein	
Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Ferndiagnose (Option!) ist aktiv, d. h. die Bedienung der Steuerung erfolgt über einen externen PC.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Ferndiagnose ist nicht aktiv.

<b>DB10 DBX103.5</b>	<b>AT-Box ready</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Es ist die AT-Box für Erweiterungsbaugruppen betriebsbereit.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die AT-Box ist nicht betriebsbereit. Eine Erweiterungsbaugruppe gemäß AT-Spezifikation hat entweder keine oder nur eingeschränkte Funktionalität.

<b>DB10 DBX103.6</b>	<b>HMI-Temperaturgrenze</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Umgebungsbedingungen der Grenzwerte befinden sich im zulässigen Toleranzbereich von 5 bis 55 Grad C.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Temperaturgrenze wurde entweder unterschritten oder überschritten. Die Temperaturüberwachung hat angesprochen und die PCU außer Betrieb gesetzt.

<b>DB10 DBX103.7</b>	<b>HMI Batteriealarm</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Batterieüberwachung hat angesprochen. Bei Netzausfall können die zuletzt geänderten Daten und eine korrekte Gerätekonfiguration verloren gehen. Ein entsprechender Alarm wird gemeldet. Die Pufferbatterie ist zu überprüfen. Eine zu geringe Batteriespannung beeinflusst auch die aktuelle Uhrzeit an der Bedienoberfläche.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Es steht kein HMI-Batteriealarm an.
weiterführende Literatur	Gerätehandbuch Bedienkomponenten (PCU)

### 18.1.3 Signale von NC an PLC (DB10)

<b>DB10 DBX104.7</b>	<b>NCK-CPU ready</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die NCK-CPU ist betriebsbereit und meldet sich zyklisch bei der PLC. Nach ordnungsgemäßem Anlauf und erstem vollständigen OB1-Zyklus (Grundstellungszyklus) tauschen PLC und NCK fortlaufend Lebenszeichen aus. Das PLC-Grundprogramm setzt das Nahtstellensignal "NCK-CPU-Ready" auf 1.



<b>DB10 DBX104.7</b>	<b>NCK-CPU ready</b>
Signalzustand 0	Die NCK-CPU ist nicht betriebsbereit. Bleibt das Lebenszeichen der NCK aus, wird die Nahtstelle PLC/NCK vom PLC-Grundprogramm neutralisiert und das Nahtstellensignal "NCK-CPU-Ready" wird auf 0 gesetzt. Vom PLC-Grundprogramm werden folgende Maßnahmen eingeleitet: <ul style="list-style-type: none"> <li>• Statussignale von NCK an PLC (Anwendernahtstelle) werden gelöscht</li> <li>• Änderungssignale bei den Hilfsfunktionen werden gelöscht</li> <li>• zyklische Bearbeitung der Anwendernahtstelle PLC an NCK wird beendet.</li> </ul>
weiterführende Literatur	<ul style="list-style-type: none"> <li>• Diagnosehandbuch</li> <li>• Funktionshandbuch Grundfunktionen; PLC-Grundprogramm</li> </ul>

<b>DB10 DBX108.3</b>	<b>SINUMERIK Operate an BTSS ready</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	SINUMERIK Operate ist betriebsbereit und meldet sich zyklisch bei der NCK.
Signalzustand 0	SINUMERIK Operate ist nicht betriebsbereit.
weiterführende Literatur	Diagnosehandbuch

<b>DB10 DBX108.5</b>	<b>Antriebe im zyklischen Betrieb</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Bei allen Maschinenachsen der NC befinden sich die entsprechenden Antriebe im zyklischen Betrieb, d.h. sie tauschen zyklisch mit der NC PROFIdrive-Telegramme aus.
Signalzustand 0	Bei mindestens einer Maschinenachse der NC befindet sich der entsprechende Antrieb nicht im zyklischen Betrieb, d.h. er tauscht zyklisch keine PROFIdrive-Telegramme mit der NC aus.

<b>DB10 DBX108.6</b>	<b>Antrieb ready</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Für <b>alle</b> Maschinenachsen der NC steht bei den entsprechenden Antrieben das Betriebsbereitschaft an: DB31, ... DBX93.5 == 1 (DRIVE ready)
Signalzustand 0	Bei mindestens einer Maschinenachsen der NC steht beim entsprechenden Antrieb das Betriebsbereitschaft <b>nicht</b> an: DB31, ... DBX93.5 == 0 (DRIVE ready)
korrespondierend mit ...	DB31, ... DBX93.5 (DRIVE ready)

<b>DB10 DBX108.7</b>	<b>NC ready</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	<p>Die Steuerung ist betriebsbereit. Dieses Nahtstellensignal ist ein Abbild des Relaiskontakts "NC-Ready". Dieses Signal wird gesetzt, wenn:</p> <ul style="list-style-type: none"> <li>• Relaiskontakt "NC-Ready" ist geschlossen</li> <li>• Alle steuerungsinternen Spannungen sind aufgebaut</li> <li>• Die Steuerung ist im zyklischen Betrieb</li> </ul>
Signalzustand 0	<p>Die Steuerung ist nicht betriebsbereit. Der Relaiskontakt "NC-Ready" ist geöffnet. Folgende Störungen bewirken die Wegnahme von NC-Ready:</p> <ul style="list-style-type: none"> <li>• Unter- bzw. Überspannungs-Überwachung hat angesprochen</li> <li>• Einzelkomponenten sind nicht betriebsbereit (NCK-CPU ready)</li> <li>• Watchdog der NCK-CPU</li> </ul> <p>Wird das Signal "NC ready" zu 0, so werden sofern noch möglich steuerungsintern folgende Maßnahmen eingeleitet:</p> <ul style="list-style-type: none"> <li>• Wegnahme der Reglerfreigaben (damit Stillsetzen der Antriebe)</li> <li>• Vom PLC-Grundprogramm werden folgende Maßnahmen eingeleitet: <ul style="list-style-type: none"> <li>– Statussignale von NCK an PLC (Anwendernahtstelle) werden gelöscht</li> <li>– Änderungssignale bei den Hilfsfunktionen werden gelöscht</li> <li>– Die zyklische Bearbeitung der Anwendernahtstelle PLC an NCK wird beendet</li> </ul> </li> </ul> <p>Weitere Informationen siehe Literatur! Die Steuerung ist erst nach POWER ON wieder betriebsbereit.</p>
korrespondierend mit ...	Relaiskontakt "NC-Ready"
weiterführende Literatur	<ul style="list-style-type: none"> <li>• Diagnosehandbuch</li> <li>• Funktionshandbuch Grundfunktionen; PLC-Grundprogramm</li> </ul>

<b>DB10 DBX109.0</b>	<b>NCK-Alarm steht an</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	<p>Es steht mindestens ein NCK-Alarm an. Dies ist ein Sammelsignal für die Nahtstellensignale aller vorhandenen Kanäle: DB21, ... DBX36.6 (Kanalspezifischer NCK-Alarm steht an).</p>
Signalzustand 0	Es steht kein NCK-Alarm an.
korrespondierend mit ...	DB21, ... DBX36.6 (Kanalspezifischer NCK-Alarm steht an) DB21, ... DBX36.7 (NCK-Alarm mit Bearbeitungsstillstand steht an)
weiterführende Literatur	Diagnosehandbuch

<b>DB10 DBX109.5</b>	<b>Kühlkörpertemperatur Alarm NCU</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Bei der NCU wurden die Grenzwerte der Kühlkörpertemperatur überschritten. Die Kühlkörperüberwachung hat angesprochen und ein dauerhafter Betrieb der Bedientafel ist nicht mehr sichergestellt.	
Signalzustand 0	Die Kühlkörperüberwachung der NCU hat nicht angesprochen.	

<b>DB10 DBX109.6</b>	<b>Lufttemperatur-Alarm</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	<p>Die Umgebungstemperatur- oder die Lüfterüberwachung hat angesprochen. Folgende Ursachen können vorliegen:</p> <ul style="list-style-type: none"> <li>Die Temperatur-Überwachung hat eine zu hohe Umgebungstemperatur (ca. 60°C) festgestellt. Alarm 2110 "NCK-Temperaturalarm" wird gemeldet.</li> <li>Die Überwachung der Drehzahl des zur Kühlung der Baugruppe eingesetzten 24V-DC-Lüfters hat angesprochen. Alarm 2120 "NCK-Lüfteralarm" wird gemeldet.</li> </ul> <p>Maßnahmen: Austausch des Lüfters vornehmen bzw. für zusätzliche Lüftung sorgen. Bei Ansprechen eines Temperatur- bzw. Lüfterfehlers wird in der Ein-/Rückspeise-Einheit ein Relaiskontakt (Klemme 5.1, 5.2 bzw. 5.1, 5.3) betätigt, der vom Kunden ausgewertet werden kann.</p>	
Signalzustand 0	Weder Temperatur-Überwachung noch die Lüfterüberwachung haben angesprochen.	
Anwendungsbeispiel(e)	Vom PLC-Anwenderprogramm können bei Ansprechen der Temperatur- oder Lüfterüberwachung entsprechende Maßnahmen veranlasst werden.	
korrespondierend mit ...	Bei Ansprechen eines Temperatur- bzw. Lüfterfehlers wird in der Ein-/Rückspeise-Einheit ein Relaiskontakt (Klemme 5.1, 5.2 bzw. 5.1, 5.3) betätigt. Dieser kann ausgewertet werden.	
weiterführende Literatur	Diagnosehandbuch	

<b>DB10 DBX109.7</b>	<b>NCK-Batteriealarm</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	<p>Die Überwachung der NCK-Batteriespannung hat angesprochen. Folgende Ursachen können vorliegen:</p> <ul style="list-style-type: none"> <li>Die Batteriespannung ist innerhalb des Vorwarnbereiches (ca. 2,7 bis 2,9 V). Der Alarm 2100 "NCK-Batterie Warnschwelle erreicht" wird gemeldet. Auswirkungen und Maßnahme siehe bei weiterführender Literatur.</li> <li>Die Batteriespannung ist unterhalb des Vorwarnbereiches (<math>\leq 2,6</math> V). Der Alarm 2101 "NCK-Batteriealarm" wird im zyklischen Betrieb gemeldet. Auswirkungen: Ein Netzspannungsausfall - z.B. durch Ausschalten der Steuerung - würde einen Datenverlust bei den gepufferten Daten (z.B. Teileprogrammspeicher, Variablen, Maschinendaten, ...) zur Folge haben. Maßnahme: siehe bei weiterführender Literatur.</li> <li>Beim Hochlauf der Steuerung wird festgestellt, dass die Batteriespannung unterhalb des Vorwarnbereiches (<math>\leq 2,6</math> V) liegt. Der Alarm 2102 "NCK-Batteriealarm" wird gemeldet; NC-Ready und BAG-Ready werden nicht erteilt. Auswirkungen: Bei den gepufferten Daten kann bereits ein Datenverlust aufgetreten sein! Maßnahme: siehe bei weiterführender Literatur.</li> </ul>
Signalzustand 0	Die Batteriespannung liegt über dem unteren Grenzwert (Normalfall).
Sonderfälle, Fehler, ...	Die NCK-Batterien sollten nur bei eingeschalteter NC gewechselt werden, um den Datenverlust wegen fehlender Speicherpufferung zu vermeiden.
weiterführende Literatur	<ul style="list-style-type: none"> <li>Diagnosehandbuch</li> <li>Gerätehandbuch NCU</li> </ul>

### 18.1.4 Signale an Bedientafelfront (DB19)

<b>DB19 DBX0.0</b>	<b>Bildschirm hell</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Dunkelschaltung des Bildschirms wird außer Kraft gesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Dunkelschaltung des Bildschirms bleibt erhalten.
korrespondierend mit ...	DB19 DBX0.1 (Bildschirm dunkelsteuern)

<b>DB19 DBX0.1</b>	<b>Bildschirm dunkelsteuern</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der Bildschirm ist "dunkel" geschaltet durch das PLC-Anwenderprogramm. Die automatische Hell-/Dunkelschaltung des Bildschirms ist somit unwirksam: d.h. bei Betätigung der Tastatur wird der Bildschirm nicht automatisch hell geschaltet.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Bildschirm ist durch das PLC-Anwenderprogramm "hell" geschaltet. Bei diesem Signalzustand kann die Hell-/Dunkelschaltung des Bildschirms von der Steuerung automatisch über die Tastatur abgeleitet werden. Der Bildschirm wird dunkel geschaltet falls an der Tastatur für eine über das folgende Maschinendatum festgelegte Zeitdauer kein Tastendruck erfolgt: MD9006 \$MM_DISPLAY_BLACK_TIME (Zeit für Bildschirm-Dunkelsteuerung) Die Hellschaltung erfolgt mit dem ersten Tastendruck an der Bedientafelfront.	
Anwendungs- beispiel(e)	Schonung des Bildschirms.	
Sonderfälle, Fehler, ...	Achtung: Die Tastatur der Bedientafelfront ist weiterhin wirksam wenn das Nahtstellensignal: DB19 DBX0.1 (Bildschirm dunkelsteuern) = 1 Es wird daher empfohlen, diese ebenfalls zu verriegeln mit dem Nahtstellensignal: DB19 DBX0.2 (Tastensperre)	
korrespondierend mit ...	DB19 DBX0.2 (Tastensperre) MD9006 \$MM_DISPLAY_BLACK_TIME (Zeit für Bildschirm-Dunkelsteuerung)	

<b>DB19 DBX0.2</b>	<b>Tastensperre</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Tastatur ist für den Bediener gesperrt.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Tastatur ist für den Bediener freigegeben.	
Anwendungs- beispiel(e)	Falls der Bildschirm mit dem Nahtstellensignal: DB19 DBX0.1 (Bildschirm dunkelsteuern) dunkel geschaltet wird, soll gleichzeitig die Tastatur mit dem Nahtstellensignal: DB19 DBX0.2 (Tastensperre) verriegelt werden, um damit eine unbeabsichtigte Bedienhandlung auszuschließen.	
korrespondierend mit ...	DB19 DBX0.1 (Bildschirm dunkelsteuern)	

<b>DB19 DBX0.3</b>	<b>Cancelalarme löschen</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Fehlerlösch Taste auf der Maschinensteuertafel ist gedrückt. Daraufhin werden alle Cancelalarme des NCKs und der Bedientafel quittiert. Die PLC-Alarme quittiert die Applikation selbst. Beim NCK bleiben PowerOn und Reset-Alarme bestehen, bis die Fehlerursache behoben ist.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Fehlerlösch Taste auf der Maschinensteuertafel ist nicht gedrückt.
Funktionalität	Nur gültig für HMI Advanced.
korrespondierend mit ...	DB19 DBX20.3 (Cancelalarm gelöscht)

<b>DB19 DBX0.4</b>	<b>Recallalarme löschen</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Fehlerlösch Taste auf der Maschinensteuertafel ist gedrückt. Daraufhin werden alle Cancelalarme des NCKs und der Bedientafel quittiert. Die PLC-Alarme quittiert die Applikation selbst. Beim NCK bleiben PowerOn und Reset-Alarme bestehen, bis die Fehlerursache behoben ist.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Fehlerlösch Taste auf der Maschinensteuertafel ist nicht gedrückt.
Anwendungsbeispiel(e)	gültig nur für HMI Advanced
korrespondierend mit ...	DB19 DBX20.4 (Recallalarm gelöscht)

<b>DB19 DBX0.7</b>	<b>Istwerte in WKS, 0 = MKS</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Von der PLC wird die Anzeige der Istwerte im Werkstückkoordinatensystem (WKS) angewählt. Damit wird bei Anwahl des Maschinenbereichs stets die WKS-Anzeige aktiviert; d.h. im Fenster "Position" werden die Maschinen- und die Zusatzachsen sowie ihre Istpositionen und Restwege im WKS angezeigt.  Da das Nahtstellensignal nur bei Eintritt in das Maschinengrundbild ausgewertet wird, kann der Bediener innerhalb des Maschinenbereichs mit den Softkeys "Istwerte MKS" und "Istwerte WKS" beliebig zwischen den jeweiligen Koordinatensystemen umschalten.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Damit wird bei Anwahl des Maschinenbereichs das jeweils zuvor angewählte Koordinatensystem (WKS oder MKS) wieder aktiviert und angezeigt.
Anwendungsbeispiel(e)	Mit dem Nahtstellensignal DB19 DBX0.7 (Istwert in WKS) = 1 wird jeweils bei erneuter Anwahl des Maschinenbereichs die für den Bediener häufig benötigte Werkstückkoordinatensystem-Anzeige (WKS) angewählt.

<b>DB19 DBX0.7</b>	<b>Istwerte in WKS, 0 = MKS</b>
korrespondierend mit ...	DB19 DBX20.7 (MKS/WKS umschalten)
weiterführende Literatur	Bedienungsanleitung HMI (entsprechend der verwendeten Software)

<b>DB19 DBB6</b>	<b>Analog Spindel 1, Auslastung in Prozent</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch

<b>DB19 DBB7</b>	<b>Analog Spindel 2, Auslastung in Prozent</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch

<b>DB19 DBB8</b>	<b>Kanalnummer der Maschinensteuertafel an HMI</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch

<b>DB19 DBB10</b>	<b>PLC Hardkeys (Wertebereich 1 ...255, 0 ist Grundstellung)</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Anwahl Programmierbereich aktiv
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Anwahl Programmierbereich inaktiv

<b>DB19 DBX13.5</b>	<b>Teileprogramm Entladen</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Entladen aktiv
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Entladen inaktiv
Anwendungsbeispiel(e)	Es kann ein File-Transfer mit der Festplatte initiiert werden.

<b>DB19 DBX13.6</b>	<b>Teilprogramm Laden</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Laden aktiv	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Laden inaktiv	
Anwendungs- beispiel(e)	Es kann ein File-Transfer mit der Festplatte initiiert werden.	

<b>DB19 DBX13.7</b>	<b>Teilprogramm Anwahl</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Anwahl aktiv	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Anwahl inaktiv	
Anwendungs- beispiel(e)	Es kann ein File-Transfer mit der Festplatte initiiert werden.	

<b>DB19 DBX14.0 - DBX14.6</b>	<b>PLC-Index</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Beschreibung	Dieses Byte zur Steuerung der V24-Schnittstelle beschreibt für die Standard-Steuerdatei den PLC-Index, der die Achs-, Kanal- oder TO-Nr. spezifiziert.	
Anwendungs- beispiel(e)	Abhängig von: DB19 DBX14.7=0 → Akt. FS: PLC-Index, der die Achs-, Kanal- oder TO-NR. spezifiziert DB19 DBX14.7=1 → Pas.FS: PLC-Index für die Anwender-Steuerdatei	

<b>DB19 DBX14.7</b>	<b>Aktives oder passives Filesystem</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Passives Filesystem	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Aktives Filesystem	



<b>DB19 DBB15</b>	<b>PLC-Zeilenoffset</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Beschreibung	Dieses Byte zur Steuerung der V24-Schnittstelle gibt an, in welcher Zeile der Standard- bzw. Anwender-Steuerdatei die zu übertragende Datei steht.
Anwendungs- beispiel(e)	Abhängig von: DB19 DBX14.7=0 → Akt. FS: PLC-Zeilenoffset in eine Standard-Steuerdatei DB19 DBX14.7=1 → Pas. FS: PLC-Zeilenoffset in eine Anwender-Steuerdatei

<b>DB19 DBX16.0 - DBX16.6</b>	<b>PLC-Index für die Anwender-Steuerdatei</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Beschreibung	Dieses Byte zur Steuerung des Datei-Transfers über Festplatte beschreibt den Index zur Steuerdatei (Jobliste).
Anwendungs- beispiel(e)	Abhängig von: DB19 DBX14.7=0 → Akt. FS: PLC-Index, für die Standard-Steuerdatei DB19 DBX14.7=1 → Pas.FS: PLC-Index für die Anwender-Steuerdatei

<b>DB19 DBX16.7</b>	<b>Aktives oder passives Filesystem</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Passives Filesystem
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Aktives Filesystem
Anwendungs- beispiel(e)	bei HMI Advanced immer 1

<b>DB19 DBB17</b>	<b>PLC-Zeilenoffset in der Anwender-Steuerdatei</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Beschreibung	Dieses Byte zur Steuerung des Datei-Transfers über Festplatte gibt an, in welcher Zeile der Anwender-Steuerdatei die zu übertragende Steuerdatei steht.
Anwendungs- beispiel(e)	Abhängig von: DB19 DBX14.7=0 → Akt. FS: PLC-Zeilenoffset in eine Standard-Steuerdatei DB19 DBX14.7=1 → Pas. FS: PLC-Zeilenoffset in eine Anwender-Steuerdatei

<b>DB19 DBX44.0</b>	<b>BA-Wechselsperre</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	BA-Wechselsperre aktiv	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	BA-Wechselsperre inaktiv	

<b>DB19 DBX45.0</b>	<b>FC9 Out: Active</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	

<b>DB19 DBX45.1</b>	<b>FC9 Out: Done</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	

<b>DB19 DBX45.2</b>	<b>FC9 Out: Error</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	

<b>DB19 DBX45.3</b>	<b>FC9 Out: StartError</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	

## 18.1.5 Signale von Bedientafelfront (DB19)

<b>DB19 DBX20.1</b>	<b>Bildschirm dunkel</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Der Bildschirm ist dunkel geschaltet.
Signalzustand 0	Der Bildschirm ist nicht dunkel geschaltet.
Anwendungs- beispiel(e)	Mit diesem NST kann die PLC feststellen, ob der Bildschirm über das Nahtstellensignal: DB19, DBX0.1 (Bildschirm dunkelsteuern) oder über das Maschinendatum: MD9006 \$MM_DISPLAY_BLACK_TIME (Zeit für Bildschirm-Dunkelschaltung) dunkel geschaltet wurde.
korrespondierend mit ...	MD9006 \$MM_DISPLAY_BLACK_TIME (Zeit für Bildschirm-Dunkelschaltung)

<b>DB19 DBX20.3</b>	<b>Cancelalarm gelöscht</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Cancelalarm gelöscht aktiv
Signalzustand 0	Cancelalarm gelöscht inaktiv Hinweis: Das Signal wird nicht automatisch zurückgesetzt, sondern muss vom Anwender über das PLC-Anwenderprogramm zurückgesetzt werden.
Anwendungs- beispiel(e)	Gilt nur für HMI Advanced

<b>DB19 DBX20.4</b>	<b>Recallalarm gelöscht</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Recallalarm gelöscht aktiv
Signalzustand 0	Recallalarm gelöscht inaktiv Hinweis: Das Signal wird nicht automatisch zurückgesetzt, sondern muss vom Anwender über das PLC-Anwenderprogramm zurückgesetzt werden.
Anwendungs- beispiel(e)	Gilt nur für HMI Advanced

<b>DB19 DBX20.6</b>	<b>Simulation aktiv</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Beim Eintritt in die Simulation = 1	
Signalzustand 0	Beim Verlassen der Simulation = 0	
Anwendungs- beispiel(e)	Vom Maschinenhersteller auswertbar, um mit NC-Start gleichzeitig den Test zu aktivieren.	

<b>DB19 DBX20.7</b>	<b>MKS/WKS umschalten</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Das Koordinatensystem wird über die Bedienung vom Werkstück-Koordinaten-System (WKS) in das Maschinen-Koordinaten-System (MKS) bzw. vom MKS in das WKS umgeschaltet. Das Signal steht nach Betätigung 1 PLC-Zyklus lang an.	
Signalzustand 0	Keine Wirkung	
Anwendungs- beispiel(e)	Das Nahtstellensignal: DB19, DBX20.7 (MKS/WKS umschalten) muss auf das Nahtstellensignal: DB19, DBX0.7 (Istwert in WKS) übertragen werden, damit die Umschaltung wirksam wird.	
korrespondierend mit ...	DB19, DBX0.7 (Istwert in WKS)	

<b>DB19 DBB22</b>	<b>angezeigte Kanalnummer von HMI</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	

<b>DB19 DBB24</b>	<b>aktuelle Bildnummer von JobShop-Oberfläche</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	

<b>DB19 DBX26.1</b>	<b>O.K. (Teileprogramm-Handling-Status)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Übertragung korrekt beendet	
Signalzustand 0	Übertragung fehlerhaft beendet	
weiterführende Literatur	Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "PLC-Funktionen" > "Programmanwahl"	
korrespondierend mit ...	DB19.DBB13 (Programmanwahl: Anforderung) DB19.DBB16 (Programmanwahl: Steuerdatei-Index) DB19.DBB17 (Programmanwahl: Programmlisten-Index) DB19.DBB27 (Programmanwahl: Errorhandling)	

<b>DB19 DBX26.2</b>	<b>Error (Teileprogramm-Handling-Status)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Übertragung fehlerhaft beendet	
Signalzustand 0	Übertragung korrekt beendet	
weiterführende Literatur	Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "PLC-Funktionen" > "Programmanwahl"	
korrespondierend mit ...	DB19.DBB13 (Programmanwahl: Anforderung) DB19.DBB16 (Programmanwahl: Steuerdatei-Index) DB19.DBB17 (Programmanwahl: Programmlisten-Index) DB19.DBB27 (Programmanwahl: Errorhandling)	

<b>DB19 DBX26.3</b>	<b>Aktiv (Teileprogramm-Handling-Status)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Auftrag in Arbeit	
Signalzustand 0	Kein Auftrag in Arbeit	
weiterführende Literatur	Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "PLC-Funktionen" > "Programmanwahl"	
korrespondierend mit ...	DB19.DBB13 (Programmanwahl: Anforderung) DB19.DBB16 (Programmanwahl: Steuerdatei-Index) DB19.DBB17 (Programmanwahl: Programmlisten-Index) DB19.DBB27 (Programmanwahl: Errorhandling)	

<b>DB19 DBX26.5</b>	<b>Entladen (Teileprogramm-Handling-Status)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Entladen aktiv	
Signalzustand 0	Entladen inaktiv	
weiterführende Literatur	Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "PLC-Funktionen" > "Programmanwahl"	
korrespondierend mit ...	DB19.DBB13 (Programmanwahl: Anforderung) DB19.DBB16 (Programmanwahl: Steuerdatei-Index) DB19.DBB17 (Programmanwahl: Programmlisten-Index) DB19.DBB27 (Programmanwahl: Errorhandling)	

<b>DB19 DBX26.6</b>	<b>Laden (Teileprogramm-Handling-Status)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Laden aktiv	
Signalzustand 0	Laden inaktiv	
weiterführende Literatur	Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "PLC-Funktionen" > "Programmanwahl"	
korrespondierend mit ...	DB19.DBB13 (Programmanwahl: Anforderung) DB19.DBB16 (Programmanwahl: Steuerdatei-Index) DB19.DBB17 (Programmanwahl: Programmlisten-Index) DB19.DBB27 (Programmanwahl: Errorhandling)	

<b>DB19 DBX26.7</b>	<b>Anwahl (Teileprogramm-Handling-Status)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Anwahl aktiv	
Signalzustand 0	Anwahl inaktiv	
weiterführende Literatur	Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "PLC-Funktionen" > "Programmanwahl"	
korrespondierend mit ...	DB19.DBB13 (Programmanwahl: Anforderung) DB19.DBB16 (Programmanwahl: Steuerdatei-Index) DB19.DBB17 (Programmanwahl: Programmlisten-Index) DB19.DBB27 (Programmanwahl: Errorhandling)	

<b>DB19 DBB27</b>	<b>Error Programhandling</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Beschreibung	Fehlerkennungen:aaa	
	<b>Wert</b>	<b>Bedeutung</b>
	0	Kein Fehler
	1	Ungültige Nummer für Steuerdatei (Wert im DB19.DBB16 < 127 oder ungültig)
	3	Steuerdatei "plc_proglist_main.ppl" nicht gefunden (Wert im DB19.DBB16 ungültig)
	4	Ungültiger Index in Steuerdatei (Wert im DB19.DBB17 ist falsch)
	5	Jobliste im angewählten Werkstück konnte nicht geöffnet werden
	6	Fehler in der Jobliste (Joblisten-Interpreter meldet Fehler)
	7	Joblisten-Interpreter meldet leere Auftragsliste
weiterführende Literatur	Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "PLC-Funktionen" > "Programmanwahl"	
korrespondierend mit ...	DB19.DBB13 (Programmanwahl: Anforderung) DB19.DBB16 (Programmanwahl: Steuerdatei-Index) DB19.DBB17 (Programmanwahl: Programmlisten-Index) DB19.DBB26 (Programmanwahl: Quittierung)	

18.1.6 Signale an Kanal (DB21, ...)

<b>DB21, ... DBX6.2</b>	<b>Restweg löschen (kanalspezifisch)</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Restweg löschen (kanalspezifisch):</p> <p>Das NST "Restweg löschen (kanalspezifisch)" wirkt nur in der Betriebsart AUTOMATIK für Bahnachsen.</p> <p>Die steigende Flanke des Nahtstellensignals wirkt nur bei den Achsen, die sich im Geometrieverband befinden. Diese werden ebenfalls durch Rampenstopp stillgesetzt sowie deren Restweg (Soll-Ist-Differenz) gelöscht. Ein eventuell vorhandener Schleppabstand wird noch abgebaut. Anschließend wird der nächste Programmsatz eingeleitet.</p> <p>Bei Positionierachsen wird somit das NST "Restweg löschen (kanalspezifisch)" ignoriert.</p> <p>Anmerkung:</p> <p>Das NST "Restweg löschen" hat bei einem Programmsatz mit Verweilzeit keinen Einfluss auf die ablaufende Verweilzeit.</p>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Wirkung
Signal irrelevant bei ...	Positionierachsen
Anwendungsbeispiel(e)	Beenden der Verfahrbewegung aufgrund eines externen Signals (z.B. Messtaster)
Sonderfälle, Fehler, ...	<p>Restweg löschen (kanalspezifisch)</p> <p>Nach Stillsetzen von Achsen mit NST "Restweg löschen" erfolgt für den nachfolgenden Programmsatz die Satzaufbereitung mit den neuen Positionen. Die Geometrieachsen fahren somit nach "Restweg löschen" eine andere Kontur als ursprünglich im Teileprogramm festgelegt.</p> <p>Durch Programmierung von G90 im Folgesatz nach "Restweg löschen" kann erreicht werden, dass zumindest die programmierte absolute Position angefahren wird. Dagegen wird mit G91 die ursprünglich im Teileprogramm festgelegte Position im Folgesatz nicht mehr erreicht.</p>
korrespondierend mit ...	DB31, ... DBX2.2 (Restweg löschen (achs-spezifisch))



## 18.1.7 Signale von Kanal (DB21, ...)

<b>DB21, ... DBX36.6</b>	<b>Kanalspezifischer NCK-Alarm steht an</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Für diesen Kanal steht mindestens ein NCK-Alarm an. Damit wird auch das folgende Sammel-Nahtstellensignal gesetzt: DB10 DBX109.0 (NCK-Alarm steht an) Vom PLC-Anwenderprogramm kann abgefragt werden, ob die Bearbeitung für den betroffenen Kanal aufgrund eines NCK-Kanals unterbrochen ist: DB21, ... DBX36.7 (NCK-Alarm mit Bearbeitungsstillstand steht an).
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Für diesen Kanal steht kein NCK-Alarm an.
korrespondierend mit ...	DB21, ... DBX36.7 (NCK-Alarm mit Bearbeitungsstillstand steht an). DB10 DBX109.0 (NCK-Alarm steht an).
weiterführende Literatur	Diagnoseanleitung

<b>DB21, ... DBX36.7</b>	<b>NCK-Alarm mit Bearbeitungsstillstand steht an</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Es steht mindestens ein NCK-Alarm an, der einen Bearbeitungsstillstand des in diesen Kanal ablaufenden Teileprogramms bewirkt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Für diesen Kanal steht kein NCK-Alarm an, der einen Bearbeitungsstillstand bewirkt.
Anwendungs- beispiel(e)	Damit kann die Programmunterbrechung aufgrund eines NCK-Alarms vom PLC-Anwenderprogramm sofort erkannt werden und somit Folgeschritte eingeleitet werden.
korrespondierend mit ...	DB21, ... DBX36.6 (Kanalspezifischer NCK-Alarm steht an)
weiterführende Literatur	Diagnoseanleitung

### 18.1.8 Signale an Achse/Spindel (DB31, ...)

<b>DB31, ... DBX1.0</b>	<b>Antriebstest Fahrfreigabe</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	<p>Verfahrbewegungen der Achse sind für den Antriebstest freigegeben.                  Rückmeldesignal des Sicherheits-Handshake bei Start des Funktionsgenerators der NC.                  Die NC hat bei Start des Funktionsgenerators mit dem folgenden Nahtstellensignal die Verfahrfreigabe für die Achse angefordert:                  DB31, ... DBX61.0 = 1 (Antriebstest Fahranforderung)                  Als Rückmeldung an die NC, dass die Achse verfahren darf, setzt das PLC-Anwenderprogramm das aktuelle Nahtstellensignal:                  DB31, ... DBX1.0 = 1 (Antriebstest Fahrfreigabe)                  Die Entscheidungs-Hoheit über die Fahrfreigabe einer Achse, liegt ausschließlich bei der PLC.</p>	
Signalzustand 0	Verfahrbewegungen der Achse sind für den Antriebstest gesperrt.	
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb	

<b>DB31, ... DBX1.3</b>	<b>Achsen-/Spindelsperre</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	<p><b>Achse</b>                  Es werden keine Sollwerte an den Lageregler ausgegeben. Die Verfahrbewegung der Achse ist gesperrt. Die Achse ist aber weiterhin in Lageregelung und ein verbliebener Schleppabstand wird ausgeregelt.                  Wird die Achse verfahren, werden an der Bedienoberfläche als Istposition und -geschwindigkeit die Sollposition und -geschwindigkeit angezeigt. Es werden keine Sollwerte an den Antrieb übertragen, die Maschinenachse verfährt daher nicht. Mit RESET wird die Anzeige der Istposition auf den Istwert der Maschineachse gesetzt.                  Fahrbefehle werden an die NC/PLC-Nahtstelle ausgegeben.                  Wird das Nahtstellensignal bei einer verfahrenen Achse gesetzt, wird diese unter Einhaltung der Beschleunigungskennlinie stillgesetzt und ein Alarm angezeigt.</p> <p><b>Spindel</b>                  Bei Steuerbetrieb werden an den Drehzahlregler keine Drehzahlsollwerte ausgegeben.                  Bei Positionierbetrieb werden an den Lageregler keine Lageteilsollwerte ausgegeben.                  Die Verfahrbewegung der Spindel ist gesperrt. An der Bedienoberfläche wird als Drehzahlwert der Drehzahlsollwert angezeigt.                  Wird das Nahtstellensignal bei einer verfahrenen Spindel gesetzt, wird diese unter Einhaltung der Beschleunigungskennlinie stillgesetzt und ein Alarm angezeigt.</p>	

<b>DB31, ... DBX1.3</b>	<b>Achsen-/Spindelsperre</b>
Signalzustand 0	<p>Das Rücksetzen des Nahtstellensignals wird erst wirksam, wenn die Achse/Spindel steht, d.h. kein Interpolationssollwert mehr ansteht.</p> <p><b>Achse</b> Lagesollwerte werden zyklisch an den Lageregler übergeben.</p> <p>War das Nahtstellensignal gesetzt, kann nach dem Rücksetzen die Achse sofort wieder normal verfahren werden.</p> <p><b>Spindel</b> Drehzahlsollwerte werden zyklisch an den Drehzahlregler übergeben.</p> <p>War das Nahtstellensignal gesetzt, kann nach dem Rücksetzen die wirksame Spindelsperre nur durch RESET bzw. M2 und erneutem Programmstart aufgehoben werden.</p>

<b>DB31, ... DBX1.3</b>	<b>Achsen-/Spindelsperre</b>																																							
Anwendungs- beispiel(e)	Unterbinden der Verfahrbewegungen der Achse bzw. Spindel beim Einfahren und Testen eines neu erstellen Teileprogramms.																																							
Sonderfälle, Fehler, ...	<p>Ist das Nahtstellensignal gesetzt, sind folgende Nahtstellensignale bezüglich des Bremsens der Achse/Spindel unwirksam:</p> <ul style="list-style-type: none"> <li>• DB31, ... DBX2.1 (Reglerfreigabe)</li> <li>• DB2 ... (Vorschub-/Spindel Halt)</li> <li>• DB31, ... DBX12.0-12.1 (Hardwareendschalter)</li> </ul> <p>Die Achse/Spindel kann weiterhin in den Zustand "Halten" bzw. "Nachführen" versetzt werden (siehe DB31, ... DBX1.4 (Nachführbetrieb)).</p> <p><b>Hinweise</b></p> <ul style="list-style-type: none"> <li>• Mit dem Nahtstellensignal wird die Sollwertausgabe am Antrieb gesperrt.</li> <li>• Bei fahrender Achse bringt ein kurzer Impuls die Achse zum Stehen. In diesem Satz verfährt diese Achse nicht wieder, sondern erst im nächsten Satz.</li> <li>• Ein neues Synchronisieren erfolgt mit dem nächsten Fahrbefehl dieser Achse automatisch, d.h. der nicht gefahrene Weg wird dann herausgefahren.</li> </ul> <p>Beispiel:</p> <pre>N10 G0 X0 Y0  N20 G1 F1000 X100  N30 Y100  N40 X200</pre> <p>Zu N20: bei Position X20 wird kurzzeitig das Nahtstellensignal "Achsen Sperre" gesetzt. Die X-Achse wird bis zum Stillstand abgebremst.</p> <p>Zu N40: Die X-Achse fährt von der letzten Position (ca. 20mm) auf die programmierte Position 200mm.</p> <p>Auswirkungen des Nahtstellensignals bei Spindel- bzw. Achskopplungen:</p> <table border="1"> <thead> <tr> <th>LS/LA<sup>1)</sup></th> <th>FS/FA<sup>1)</sup></th> <th>Koppl.<sup>2)</sup></th> <th>Auswirkung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Aus</td> <td>Sollwerte der Achsen werden ausgegeben</td> </tr> <tr> <td>0</td> <td>1</td> <td>Aus</td> <td>Keine Sollwertausgabe für FS/FA</td> </tr> <tr> <td>1</td> <td>0</td> <td>Aus</td> <td>Keine Sollwertausgabe für LS/LA</td> </tr> <tr> <td>1</td> <td>1</td> <td>Aus</td> <td>Keine Sollwertausgabe für FS/FA und LS/LA</td> </tr> <tr> <td>0</td> <td>0</td> <td>Ein</td> <td>Sollwerte der Achsen werden ausgegeben</td> </tr> <tr> <td>0</td> <td>1</td> <td>Ein</td> <td>Sperre wirkt nicht für FS/FA</td> </tr> <tr> <td>1</td> <td>0</td> <td>Ein</td> <td>Sperre wirkt auch für FS/FA</td> </tr> <tr> <td>1</td> <td>1</td> <td>Ein</td> <td>Keine Sollwertausgabe für FS/FA und LS/LA</td> </tr> </tbody> </table> <p>1) Status des Nahtstellensignals; LS/LA: Leitspindel/-achse; FS/FA: Folgespindel/-achse 2) Status der Kopplung zwischen Leit- und Folgespindel/-achse</p>				LS/LA <sup>1)</sup>	FS/FA <sup>1)</sup>	Koppl. <sup>2)</sup>	Auswirkung	0	0	Aus	Sollwerte der Achsen werden ausgegeben	0	1	Aus	Keine Sollwertausgabe für FS/FA	1	0	Aus	Keine Sollwertausgabe für LS/LA	1	1	Aus	Keine Sollwertausgabe für FS/FA und LS/LA	0	0	Ein	Sollwerte der Achsen werden ausgegeben	0	1	Ein	Sperre wirkt nicht für FS/FA	1	0	Ein	Sperre wirkt auch für FS/FA	1	1	Ein	Keine Sollwertausgabe für FS/FA und LS/LA
LS/LA <sup>1)</sup>	FS/FA <sup>1)</sup>	Koppl. <sup>2)</sup>	Auswirkung																																					
0	0	Aus	Sollwerte der Achsen werden ausgegeben																																					
0	1	Aus	Keine Sollwertausgabe für FS/FA																																					
1	0	Aus	Keine Sollwertausgabe für LS/LA																																					
1	1	Aus	Keine Sollwertausgabe für FS/FA und LS/LA																																					
0	0	Ein	Sollwerte der Achsen werden ausgegeben																																					
0	1	Ein	Sperre wirkt nicht für FS/FA																																					
1	0	Ein	Sperre wirkt auch für FS/FA																																					
1	1	Ein	Keine Sollwertausgabe für FS/FA und LS/LA																																					
korrespondierend mit ...	DB21, ... DBX33.7 (Programmtest aktiv)																																							
weiterführende Literatur	Verhalten bei Synchronbetrieb: Funktionshandbuch Erweiterungsfunktionen; Synchronspindel (S3)																																							

<b>DB31, ... DBX1.4</b>	<b>Nachführbetrieb</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	<p>Nachführbetrieb ist aktiv.</p> <ul style="list-style-type: none"> <li>• Der Lagesollwert wird kontinuierlich nachgeführt: Lagesollwert = Lageistwert</li> <li>• Rückmeldung für den aktiven Nachführbetrieb: DB31, ... DBX61.3 = 1 (Nachführen aktiv)</li> <li>• Stillstands- oder Klemmungsüberwachung sind nicht aktiv.</li> <li>• Beim Wiedereinschalten der Regelung erfolgt steuerungsintern ein Rückpositionieren (REPOSA: Anfahren auf einer Geraden mit allen Achsen) auf die zuletzt programmierte Position, wenn ein Teileprogramm aktiv ist.</li> </ul>
Signalzustand 0	<p>Nachführbetrieb ist nicht aktiv.</p> <p>Bei Wegnahme der "Reglerfreigabe" wird steuerungsintern der alte Lagesollwert erhalten. Wird während dessen die Achse/Spindel aus der Position gedrückt, entsteht ein Schleppabstand zwischen Lagesollwert und Lageistwert. Diese Lagedifferenz wird mit Erteilen der "Reglerfreigabe" sofort ausgeregelt, so dass die alte Sollposition wieder eingenommen wird. Anschließend beginnen alle weiteren Achsbewegungen bei der Sollposition, die vor Wegnahme der "Reglerfreigabe" bestand.</p> <p>Die Achse erhält somit bei Wiedereinschalten der Lageregelung ggf. einen Drehzahlsollwertsprung. Stillstands- oder Klemmungsüberwachung sind aktiv.</p> <p>Um die Stillstandsüberwachung auszuschalten, sollte beim Klemmen einer Achse folgendes Nahtstellensignal gesetzt werden: DB31, ... DBX2.3 = 1(Klemmvorgang läuft)</p> <p>Im Zustand "Halten" wird das Nahtstellensignal: DB31, ... DBX61.3 (Nachführen aktiv) auf 0-Signal gesetzt.</p>
Sonderfälle, Fehler, ...	<p>Wird aufgrund von Störungen steuerungsintern die Reglerfreigabe weggenommen, sollte vor NC-Start nach erfolgreichem Löschen der anstehenden Alarme und dem damit einhergehenden steuerungsintern Setzen der Reglerfreigabe, für die Achse der Zustand "Halten" aktiviert werden: DB31, ... DBX1.4 = 0 (Nachführbetrieb).</p> <p>Ansonsten würde bei NC-Start und aktivem Nachführbetrieb durch das interne Restweglöschen der Verfahrenweg des vorhergehenden NC-Satzes nicht ausgeführt werden.</p> <p><b>Achtung</b> Beim Übergang vom Zustand "Nachführen" in den Zustand "Halten" bzw. mit Erteilung der Reglerfreigabe in Lageregelung, wird steuerungsintern Restweglöschen aktiviert. Dies hat unter anderem zur Folge, dass ein Satz, in dem nur diese Achse verfahren soll, direkt beendet ist.</p>
korrespondierend mit ...	<p>DB31, ... DBX2.1 (Reglerfreigabe)</p> <p>DB31, ... DBX2.3 (Klemmvorgang läuft)</p> <p>DB31, ... DBX61.3 (Nachführen aktiv)</p>

<b>DB31, ... DBX1.5 - DBX1.6</b>	<b>Lagemesssystem 1 (LMS1) / Lagemesssystem 2 (LMS2)</b>		
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch		
Signalzustand 1	Das Lagemesssystem ist freigegeben		
Signalzustand 0	Das Lagemesssystem ist gesperrt		
Signalzustand - Überblick	<b>LMS1</b>	<b>LMS2</b>	<b>Auswirkung</b>
	1	0	Lagemesssystem 1 ist aktiv: <ul style="list-style-type: none"> <li>• Lagemesssystem 1 wird zur Lageregelung verwendet.</li> <li>• Falls Lagemesssystem 2 ebenfalls vorhanden ist (MD30200 \$MA_NUM_ENCS == 2), wird dessen Lageistwert ebenfalls erfasst.</li> </ul>
	0	1	Lagemesssystem 2 ist aktiv: <ul style="list-style-type: none"> <li>• Lagemesssystem 2 wird zur Lageregelung verwendet.</li> <li>• Falls Lagemesssystem 1 ebenfalls vorhanden ist (MD30200 \$MA_NUM_ENCS == 2), wird dessen Lageistwert ebenfalls erfasst.</li> </ul>
	1	1	<ul style="list-style-type: none"> <li>• Lagemesssystem 1 wird zur Lageregelung verwendet.</li> <li>• Falls Lagemesssystem 2 ebenfalls vorhanden ist (MD30200 \$MA_NUM_ENCS == 2), wird dessen Lageistwert ebenfalls erfasst.</li> </ul>
	0	0	Lagemesssystem 1 und 2 sind inaktiv: <ul style="list-style-type: none"> <li>• Es erfolgt keine Istwerterfassung.</li> <li>• Die Überwachungen der Lagemesssysteme sind abgeschaltet</li> <li>• Folgende Nahtstellensignale werden zurückgesetzt:                             <ul style="list-style-type: none"> <li>– DB31, ... DBX60.4/5 == 0 (Referiert/Synchronisiert)</li> <li>– DB31, ... DBX61.5 (Lageregler aktiv),</li> <li>– DB31, ... DBX61.6 (Drehzahlregler aktiv)</li> <li>– DB31, ... DBX61.7 (Stromregler aktiv)</li> </ul> </li> </ul> Nach Beendigung der Parkstellung muss die Achse neu referenziert werden.
<b>Hinweise</b>			
<ul style="list-style-type: none"> <li>• Werden bei einer fahrenden Achse das Nahtstellensignal des aktiven Lagemesssystem zurückgesetzt, wird die Achse mit Rampenstopp stillgesetzt, ohne dass steuerungsintern die Reglerfreigabe weggenommen wird.</li> <li>• Hat eine drehzahlgeregelte Spindel kein Lagemesssystem, muss Nahtstellensignal "Reglerfreigabe" gesetzt sein: DB31, ... DBX2.1 == 1(Reglerfreigabe)</li> </ul>			

<b>DB31, ... DBX1.5 - DBX1.6</b>	<b>Lagemesssystem 1 (LMS1) / Lagemesssystem 2 (LMS2)</b>
Anwendungs- beispiel(e)	<p>1. Umschaltung von Lagemesssystem 1 auf Lagemesssystem 2 (und umgekehrt):</p> <p>Falls die Achse in beiden Lagemesssystemen referenziert war und zwischenzeitlich die Grenzfrequenz des verwendeten Messwertgebers nicht überschritten wurde, d.h. DB31, ... DBX60.4 und 60.5 == 1 (Referiert/Synchronisiert 1 und 2), ist nach der Umschaltung ein erneutes Referenzpunktfahren nicht erforderlich.</p> <p>Bei der Umschaltung wird die aktuelle Abweichung zwischen Lagemesssystem 1 und 2 sofort verfahren.</p> <p>Mit folgendem Maschinendatum kann ein Toleranzband vorgegeben werden, in dem die Abweichung zwischen den beiden Istwerten bei der Umschaltung liegen darf: MD36500 \$MA_ENC_CHANGE_TOL (Maximale Toleranz bei Lageistwertumschaltung) Ist die Istwertdifferenz größer als die Toleranz, wird nicht umgeschaltet und der Alarm 25100 "Meßsystemumschaltung nicht möglich" angezeigt.</p> <p>2. Parkende Achse (d.h. kein Lagemesssysteme ist aktiv):</p> <p>Zur Entfernung des Messwertgebers wird die Überwachung der Lagemesssysteme ausgeschaltet.</p> <p>3. Lagemesssystem ausschalten:</p> <p>Mit dem Ausschalten des Lagemesssystem 1 oder 2 wird das zugehörige Nahtstellensignal: DB31, ... DBX60.4/60.5 (Referiert/Synchronisiert 1/2) zurückgesetzt.</p> <p>4. Referenzpunktfahren:</p> <p>Das Referenzpunktfahren der Achse wird mit dem angewählten Lagemesssystem durchgeführt. Jedes LMS muss separat referiert werden.</p>
Sonderfälle, Fehler, ...	Ist der Zustand "parkende Achse" aktiv, wird bei NC-Start für diese Achse folgendes Nahtstellensignal ignoriert: DB31, ... DBX60.4/60.5 (Referiert / Synchronisiert 1 / 2).
korrespondierend mit ...	DB31, ... DBX60.4 / .5 (Referiert/Synchronisiert 1 / 2) DB31, ... DBX61.6 (Drehzahlregler aktiv) DB31, ... DBX2.1 (Reglerfreigabe) MD36500 \$MA_ENC_CHANGE_TOL (Maximale Toleranz bei Lageistwertumschaltung) MD30200 \$MA_NUM_ENCS (Anzahl der Geber)
weiterführende Literatur	Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-Istwertssysteme, Regelung (G2)

<b>DB31, ... DBX2.1</b>	<b>Reglerfreigabe</b>
Flankenauswertung: nein   Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Reglerfreigabe erteilt. Der Lageregelkreis ist geschlossen und die Achse/Spindel ist in Regelung. Rückmeldung: DB31, ... DBX61.5 = 1 (Lageregler aktiv) War die Achse/Spindel vor dem Rücksetzen des Nahtstellensignals referenziert, muss nach dem erneuten Setzen des Nahtstellensignals die Achse/Spindel nicht erneut referenziert werden. Randbedingung: Die Grenzfrequenz des aktiven Messsystems darf zwischenzeitlich nicht überschritten worden sein. <b>Hinweis</b> Wurde die Achse/Spindel während der Zeit in welcher die Reglerfreigabe nicht gesetzt war, aus ihrer Position bewegt, ist das Verhalten beim Setzen der Reglerfreigabe abhängig vom Nahtstellensignal "Nachführbetrieb": <ul style="list-style-type: none"> <li>• DB31, ... DBX1.4 == 1 (Nachführbetrieb) Die Lageregelung erfolgt auf der aktuellen Position</li> <li>• DB31, ... DBX1.4 == 0 (Nachführbetrieb) Die Lageregelung erfolgt auf der letzten Position vor dem Rücksetzen der Reglerfreigabe</li> </ul>
Signalzustand 0	Reglerfreigabe nicht erteilt. Das Verhalten bei Wegnahme der "Reglerfreigabe" ist abhängig davon, ob die Achse/Spindel zu diesem Zeitpunkt steht oder verfährt: <ul style="list-style-type: none"> <li>• Achse/ Spindel steht:                         <ul style="list-style-type: none"> <li>– Der Lageregelkreis der Achse wird geöffnet.</li> <li>– Bei DB31, ... DBX1.4 == 1 (Nachführbetrieb) ⇒ Lagesollwert = Lageistwert</li> <li>– Die Reglerfreigabe am Antriebs wird zurückgesetzt</li> <li>– Folgende Nahtstellensignale werden zurückgesetzt:                                  DB31, ... DBX61.5 = 0 (Lageregler aktiv)                                  DB31, ... DBX61.6 = 0 (Drehzahlregler aktiv)                                  DB31, ... DBX61.7 = 0 (Stromregler aktiv)</li> </ul> </li> <li>• Achse/Spindel verfährt                         <ul style="list-style-type: none"> <li>– Die Achse wird bis zum Stillstand mit Schnellstopp abgebremst.</li> <li>– Alarm 21612 "VDI-Signal Reglerfreigabe während der Bewegung zurückgesetzt"</li> <li>– Der Lageregelkreis der Achse/Spindel wird geöffnet.</li> <li>– Unabhängig vom Nahtstellensignal DB31, ... DBX1.4 (Nachführbetrieb) wird am Ende des Bremsvorganges der Lagesollwert nachgeführt (Lagesollwert = Lageistwert) und das Rückmeldesignal DB31, ... DBX61.3 = 1 (Nachführbetrieb) gesetzt.</li> <li>– Folgende Nahtstellensignale werden zurückgesetzt:                                  DB31, ... DBX61.5 (Lageregler aktiv)                                  DB31, ... DBX61.6 (Drehzahlregler aktiv)                                  DB31, ... DBX61.7 (Stromregler aktiv)</li> </ul> </li> </ul>



<b>DB31, ... DBX2.1</b>	<b>Reglerfreigabe</b>
Anwendungs- beispiel(e)	<p>Mechanisches Klemmen einer Achse</p> <p>Ist die Achse auf der Klemmposition positioniert, wird die Klemmung geschlossen. Anschließend wird die Reglerfreigabe zurückgesetzt. Ansonsten würde der Lageregler ständig gegen die Klemmung arbeiten, falls die Achse während des Klemmvorgangs mechanisch aus ihrer vorgegebenen Position bewegt wurde.</p> <p>Beim Aufheben der Klemmung wird zuerst die Reglerfreigabe gesetzt und anschließend die mechanische Klemmung gelöst.</p>
Sonderfälle, Fehler, ...	<p>Fahranforderung für eine Achse/Spindel ohne Reglerfreigabe:</p> <ul style="list-style-type: none"> <li>• Die Achse/Spindel wird nicht verfahren</li> <li>• Der Fahrbefehl wird an die Nahtstelle ausgegeben</li> <li>• Solange die Fahranforderung besteht, wird mit Setzen der Reglerfreigabe die Achse/Spindel sofort verfahren.</li> </ul> <p>Das Rücksetzen der Reglerfreigabe für eine fahrende Geometrieachse führt immer zu einer Konturverletzung.</p>
korrespondierend mit ...	<p>DB31, ... DBX61.3 (Nachführen aktiv)</p> <p>DB31, ... DBX1.4 (Nachführbetrieb)</p> <p>DB31, ... DBX61.5 (Lageregler aktiv)</p> <p>DB31, ... DBX61.6 (Drehzahlregler aktiv)</p> <p>DB31, ... DBX61.7 (Stromregler aktiv)</p> <p>MD36620 \$MA_SERVO_DISABLE_DELAY_TIME (Abschaltverzögerung Reglerfreigabe)</p> <p>MD36610 \$MA_AX_EMERGENCY_STOP_TIME (Zeitdauer der Bremsrampe bei Fehlerzuständen)</p>

<b>DB31, ... DBX2.2</b>	<b>Restweg löschen (achsspezifisch) / Spindel-Reset</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Flankenwechsel 0 → 1	<p>Restweg löschen (achsspezifisch) wird angefordert.</p> <p>Es ergibt sich, abhängig von der Betriebsart, unterschiedliches Verhalten:</p> <ul style="list-style-type: none"> <li>• Betriebsart JOG: <p>Wird das Nahtstellensignal für eine Achse gegeben (Flankenwechsel 0 → 1), so wird diese durch Rampenstopp stillgesetzt und deren Restweg (Soll-Ist-Differenz) gelöscht. Ein eventuell vorhandener Schleppabstand wird noch abgebaut.</p> </li> <li>• Betriebsart AUTOMATIK und MDA: <p>Die steigende Flanke des Nahtstellensignals wirkt nur bei den Achsen, die sich nicht im Geometrieverband befinden. Diese werden durch Rampenstopp stillgesetzt und deren Restweg (Soll-Ist-Differenz) gelöscht. Anschließend kann der nächste Programmsatz eingeleitet werden. Bei Geometrieachsen wird somit das NST "Restweg löschen axial" ignoriert.</p> <p>Anmerkung: "Restweg löschen" hat bei einem Teileprogrammsatz mit Verweilzeit keinen Einfluss auf die ablaufende Verweilzeit.</p> </li> </ul>
Flankenwechsel 1 → 0	keine Wirkung
Anwendungs- beispiel(e)	Beenden der Verfahrbewegung aufgrund eines externen Signals (z.B. Messtaster)

<b>DB31, ... DBX2.2</b>	<b>Restweg löschen (achsspezifisch) / Spindel-Reset</b>
Sonderfälle, Fehler, ...	"Restweg löschen (achsspezifisch)" Nach Stillsetzen von Achsen mit "Restweg löschen" erfolgt für den nachfolgenden Programmsatz die Satzaufbereitung mit den neuen Positionen. Die Achsen fahren somit nach "Restweg löschen" eine andere Kontur als ursprünglich im Teileprogramm festgelegt. Durch Programmierung von G90 im Folgesatz nach "Restweg löschen " kann erreicht werden, dass zumindest die programmierte absolute Position angefahren wird. Dagegen wird mit G91 die ursprünglich im Teileprogramm festgelegte Position im Folgesatz nicht mehr erreicht.
korrespondierend mit ...	DB21, ... DBX6.2 (Restweg löschen, kanalspezifisch)
weiterführende Literatur	Funktionshandbuch Grundfunktionen; Spindeln (S1)

<b>DB31, ... DBX9.0 - DBX9.2</b>	<b>Lagereglerparametersatz-Anwahl A, B, C</b>			
Flankenauswertung: nein	Signal(e) aktualisiert: auf Anforderung			
Bedeutung	Mit den Nahtstellensignalen A, B und C können 6 unterschiedliche Lagereglerparametersätze ausgewählt werden. Dabei gilt folgende Zuordnung:			
	<b>Parametersatz</b>	<b>C</b>	<b>B</b>	<b>A</b>
	1	0	0	0
	2	0	0	1
	3	0	1	0
	4	0	1	1
	5	1	0	0
	6	1	0	1
	6	1	1	0
	6	1	1	1
Signal irrelevant bei ...	MD35590 \$MA_PARAMSET_CHANGE_ENABLE = 0			
korrespondierend mit ...	DB31, ...DBX69.0, .1, ..2 (Rückmeldung: aktiver Lagereglerparametersatz)			
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb, Kapitel "Inbetriebnahme NCK" >			
Flankenauswertung: nein	Signal(e) aktualisiert: auf Anforderung			

<b>DB 31, ... DBX9.3</b>	<b>Parametersatzvorgaben von NC gesperrt</b>
Flankenauswertung: nein	Signal(e) aktualisiert: auf Anforderung
Signalzustand 1	Die Parametersatzumschaltung ist gesperrt.
Signalzustand 0	Die Parametersatzumschaltung ist freigegeben.
korrespondierend mit ...	DB31, ... DBX9.0, .1, .2

<b>DB31, ... DBX20.1</b>	<b>Hochlaufgebersperre</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Für den Antrieb wird ein Schnellstop mit Drehzahlsollwert 0 angefordert. Dabei wird der Antrieb ohne Hochlaufgeberrampe stillgesetzt (generatorisches Bremsen). Rückmeldung für das Auslösen des Schnellstops im Antrieb erfolgt über: DB31, ... DBX92.1 == 1 (Hochlaufgebersperre aktiv)
Signalzustand 0	Für den Antrieb wird kein Schnellstop mit Drehzahlsollwert 0 angefordert.
korrespondierend mit ...	DB31, ... DBX92.1 (Hochlaufgebersperre aktiv)
Anmerkung	Im Antrieb muss ein dem Interface Mode "SIMODRIVE 611 universal" kompatibles PROFIDrive-Telegramm eingestellt sein. Siehe Antriebsparameter: p0922 und p2038
weiterführende Literatur	NC: Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb Antrieb: SINAMICS S120/S150 Listenhandbuch

<b>DB31, ... DBX21.0 - DBX21.2</b>	<b>Antriebsparametersatz-Anwahl A, B, C</b>																																				
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch																																				
Bedeutung	Mit den Nahtstellensignalen A, B und C können 8 unterschiedliche Antriebsparametersätze ausgewählt werden. Dabei gilt folgende Zuordnung: <table border="1" data-bbox="399 1187 1468 1534"> <thead> <tr> <th>Parametersatz</th> <th>C</th> <th>B</th> <th>A</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>4</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>5</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>6</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>7</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>8</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Parametersatz	C	B	A	1	0	0	0	2	0	0	1	3	0	1	0	4	0	1	1	5	1	0	0	6	1	0	1	7	1	1	0	8	1	1	1
Parametersatz	C	B	A																																		
1	0	0	0																																		
2	0	0	1																																		
3	0	1	0																																		
4	0	1	1																																		
5	1	0	0																																		
6	1	0	1																																		
7	1	1	0																																		
8	1	1	1																																		
Anwendungsbeispiel(e)	Die Umschaltung der Antriebsparameter kann beispielsweise verwendet werden bei: <ul style="list-style-type: none"> <li>• Getriebeumschaltung</li> <li>• Messkreisumschaltung</li> </ul>																																				
Sonderfälle, Fehler, ...	Prinzipiell ist die Umschaltung der Antriebsparametersätze zu jedem Zeitpunkt möglich. Da jedoch insbesondere bei Umschaltung der Drehzahlreglerparameter und der Motordrehzahlnormierung Momentensprünge auftreten können, sollte die Umschaltung nur bei stationären Zuständen, insbesondere Achsstillstand, erfolgen.																																				
korrespondierend mit ...	DB31, ... DBX93.0, .1, .2 (Rückmeldung: aktiver Antriebsparametersatz)																																				
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb																																				

<b>DB31, ... DBX21.3 - DBX21.4</b>	<b>Motoranwahl A, B</b>				
Flankenbewertung: nein		Signal(e) aktualisiert: zyklisch			
Bedeutung	Mit den Nahtstellensignalen A, B kann zwischen 4 verschiedenen Motoren bzw. Motorbetriebsarten umgeschaltet werden. Dabei gilt folgende Zuordnung:				
		<b>Motoranwahl</b>	<b>Anwendung</b>	<b>B</b>	<b>A</b>
		Motor 1	Betriebsart 1	0	0
		Motor 2	Betriebsart 2	0	1
		Motor 3	Betriebsart 3	1	0
	Motor 4	Betriebsart 4	1	1	
Sobald im Antrieb die Anforderung zum Umschalten auf einen neuen Motor bzw. -betriebsart erkannt wird, wird die Impulsfreigabe weggenommen.					
Anwendungsbeispiel(e)	Über die Motor-Anwahl kann z.B. bei einem Hauptspindeltrieb (HSA) zwischen der Betriebsart 1 (Sternbetrieb) und Betriebsart 2 (Dreieckbetrieb) umgeschaltet werden.				
Sonderfälle, Fehler, ...	<b>Achtung</b> Vor einer neuen Anwahl muss folgendes Nahtstellensignal zurückgesetzt worden sein: DB31, ... DBX21.5 = 0 (Motoranwahl erfolgt)				
korrespondierend mit ...	DB31, ... DBX93.3 und .4 (Rückmeldung: Aktiver Motor) DB31, ... DBX21.5 (Motoranwahl erfolgt)				
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb				

<b>DB31, ... DBX21.5</b>	<b>Motoranwahl erfolgt</b>			
Flankenbewertung: nein		Signal(e) aktualisiert: zyklisch		
Signalzustand 1	Die notwendigen elektrischen (z.B. Schützumschaltung bei Stern-/Dreieckumschaltung) und/oder mechanischen Umschaltungen sind abgeschlossen. Die Achse kann wieder verfahren werden. Vom Antrieb werden daraufhin die Impulse freigegeben.			
	<b>Achtung</b> Vor einer neuen Motoranwahl über DB31, ... DBX21.3 und .4 muss das Nahtstellensignal zurückgesetzt werden.			
Signalzustand 0	Die notwendigen elektrischen (z.B. Schützumschaltung bei Stern-/Dreieckumschaltung) und/oder mechanischen Umschaltungen sind noch <b>nicht</b> abgeschlossen. Die Achse darf nicht verfahren. Vom Antrieb werden die Impulse nicht freigegeben.			
korrespondierend mit ...	DB31, ... DBX93.3 und 4 (Aktiver Motor) DB31, ... DBX21.3 und 4 (Motor-Anwahl A, B)			
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb			

<b>DB31, ... DBX21.6</b>	<b>Integratorsperre Drehzahlregler</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Der Integrator (I-Anteil) des Drehzahlreglers ist gesperrt bzw. soll gesperrt werden (P- statt PI-Verhalten). <b>Hinweis:</b> Bei Aktivierung der Integratorsperre des Drehzahlreglers können, je nach Anwendungsfall, Ausgleichsvorgänge auftreten (z.B. wenn der Integrator zuvor stationär eine Last hält). Vom Antrieb wird mit dem Nahtstellensignal: DB31, ... DBX93.6 (Integrator n-Regler gesperrt) zur PLC die erfolgte Integratorsperre quittiert.
Signalzustand 0	Der Integrator (I-Anteil) des Drehzahlreglers ist freigegeben (PI-Verhalten).
korrespondierend mit ...	DB31, ... DBX93.6 (Rückmeldung: Der Integrator des Drehzahlreglers ist gesperrt)
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb

<b>DB31, ... DBX21.7</b>	<b>Impulsfreigabe</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Für den Antrieb sind die Impulse freigegeben. Die Impulsfreigabe im Antrieb erfolgt aber nur, wenn der Antrieb Bereitschaft signalisiert: DB31, ... DBX93.5 == 1 (Rückmeldung: Antrieb Ready)
Signalzustand 0	Für den Antrieb sind die Impulse gesperrt.
Anwendungs- beispiel(e)	Sicherheitsrelevantes Signal
Sonderfälle, Fehler, ...	Bei Wegnahme der Impulsfreigabe während der Bewegung (z.B. Not-Halt), wird die Achse/Spindel nicht mehr geführt gebremst. Die Achse/Spindel trudelt aus.
korrespondierend mit ...	DB31, ... DBX93.5 (Rückmeldung: Antrieb Ready) DB31, ... DBX93.7 (Rückmeldung: Impulse sind freigegeben)
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb

### 18.1.9 Signale von Achse/Spindel (DB31, ...)

**Hinweis**

**SINAMICS S120: Meldungswort (MELDW)**

Das Meldungswort (MELDW) ist nur in SIMODRIVE 611u kompatiblen PROFIDrive Telegrammen enthalten, z.B. Telegramm 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139

**Literatur**

SINAMICS Listenhandbuch, Funktionsplan 2419 und 2420

**SINAMICS S120: Zustandswort 1 / 2 (ZSW1/2)**

Die Zustandsworte ZSW1 bzw. ZSW2 beziehen sich nachfolgend nur auf SIMODRIVE 611u kompatible PROFIDrive Telegramme (Interface Mode SIMODRIVE 611u, p2038 = 1)

<b>DB31, ... DBX61.0</b>	<b>Antriebstest Fahranforderung</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Die Steuerung meldet, dass alle Verfahrbedingungen für die Achse erfüllt sind. Voraussetzungen dafür sind: <ul style="list-style-type: none"> <li>• Die mechanische Bremse der betreffenden Achse wurde vorher gelöst und alle weiteren Achsverfahrbedingungen sind erfüllt.</li> </ul> Mit: DB31, ... DBX61.0 (Antriebstest Fahranforderung) = 1-Signal können die entsprechenden Achsen verfahren werden. <ul style="list-style-type: none"> <li>• Die Achsensperre:                      DB31, ... DBX1.3 (Achsen-/Spindelsperre) = 1-Signal                      ist nicht aktiv.</li> </ul>	
Signalzustand 0	Die Steuerung meldet, dass die Achsen nicht verfahren werden können. Achsen können nicht verfahren werden bei: <ul style="list-style-type: none"> <li>• DB31, ... DBX61.0 (Antriebstest Fahranforderung) = 0-Signal</li> <li>• Steuerungsintern bei Störungen</li> </ul> Damit sind die dafür erforderlichen o.g. Voraussetzungen nicht erfüllt.	
Weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb	

<b>DB31, ... DBX61.3</b>	<b>Nachführen aktiv (Rückmeldung)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	<p>Die Steuerung meldet, dass der Nachführbetrieb für die Achse/Spindel aktiv ist. Voraussetzungen dafür sind:</p> <ul style="list-style-type: none"> <li>Die Reglerfreigabe für den Antrieb ist weggenommen (entweder von der PLC mit "Reglerfreigabe" = 0-Signal oder steuerungsintern bei Störungen; siehe Literatur)</li> <li>Nachführbetrieb ist angewählt (entweder von PLC mit NST "Nachführbetrieb" = 1-Signal oder steuerungsintern z.B. bei Wegnahme der Reglerfreigabe von einer fahrenden Achse)</li> </ul> <p>Während Nachführbetrieb wirksam ist, wird der Lagesollwert dem Istwert ständig nachgeführt. Die Stillstands- und Klemmungsüberwachung ist nicht wirksam.</p>	
Signalzustand 0	<p>Die Steuerung meldet, dass der Nachführbetrieb für die Achse/Spindel nicht aktiv ist. Die Stillstands- und Klemmungsüberwachung ist wirksam. Damit sind die dafür erforderlichen o.g. Voraussetzungen nicht erfüllt. Im Zustand "Halten" ist das Nahtstellensignal: DB31, ... DBX61.3 (Nachführen aktiv) auf 0-Signal.</p>	
Sonderfälle, Fehler, ...	<p>Achtung: Beim Übergang von "Nachführen" in den Zustand "Halten" (mit NST "Nachführbetrieb" = 0 setzen) oder in den Regelungsbetrieb (mit NST "Reglerfreigabe" = 1) wird steuerungsintern ein Restweglöschen ausgelöst.</p>	
korrespondierend mit ...	<p>DB31, ... DBX2.1 (Reglerfreigabe) DB31, ... DBX1.4 (Nachführbetrieb)</p>	
weiterführende Literatur	Diagnosehandbuch	

<b>DB31, ... DBX61.4</b>	<b>Achse/Spindel steht (<math>n &lt; n_{min}</math>) (Status)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	<p>Die aktuelle Geschwindigkeit der Achse bzw. die Istdrehzahl der Spindel liegt unter der mit dem Maschinendatum: MD36060 \$MA_STANDSTILL_VELO_TOL (Maximale Geschwindigkeit/Drehzahl für Signal "Achse/Spindel steht") festgelegten Grenze.</p>	
Signalzustand 0	<p>Die aktuelle Geschwindigkeit der Achse bzw. die Istdrehzahl der Spindel ist größer als der im MD angegebene Wert (Stillstandsbereich). Steht ein Fahrbefehl an, z.B. bei einer Spindel, so ist das Signal immer = 0, auch wenn die aktuelle Drehzahl unterhalb von MD36060 liegt. Wenn das Nahtstellensignal: DB31, ... DBX61.4 (Achse/ Spindel steht) gemeldet wird und für die Spindel keine Lageregelung aktiv ist, dann wird an der Bedienoberfläche die Istdrehzahl mit Null angezeigt und mit der Systemvariablen \$AA_S[n] wird Null gelesen.</p>	

<b>DB31, ... DBX61.4</b>	<b>Achse/Spindel steht (<math>n &lt; n_{min}</math>) (Status)</b>
Anwendungs- beispiel(e)	Freigabe zum Öffnen der Schutzvorrichtung (z.B. Tür auf). Öffnen des Werkstückfutters bzw. der Werkzeug-Spanneinrichtung nur, wenn die Spindel steht. Beim Getriebestufenwechsel kann der Pendelbetrieb eingeschaltet werden, nachdem die Spindel auf Stillstand abgebremst hat. Vor Hochdrehen der Spindel muss ein Schließen der Werkzeug-Spannvorrichtung erfolgt sein.
korrespondierend mit ...	MD36060 \$MA_STANDSTILL_VELO_TOL (Maximale Geschwindigkeit/Drehzahl für Signal "Achse/Spindel steht")

<b>DB31, ... DBX61.5</b>	<b>Lageregler aktiv (Status)</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die Steuerung meldet, dass der Lageregler für die Achse bzw. Spindel geschlossen ist.
Signalzustand 0	Die Steuerung meldet, dass der Lageregler für die Achse bzw. Spindel offen ist. Mit Wegnahme der "Reglerfreigabe" aufgrund einer Störung bzw. vom PLC-Anwenderprogramm wird der Lageregler geöffnet und somit das Nahtstellensignal: DB31, ... DBX61.5 (Lageregler aktiv) auf 0-Signal gesetzt. Spindel ohne Lageregelung: Signal "Lageregler aktiv" ist immer "0". Weitere Auswirkungen siehe weiterführende Literatur.
Anwendungs- beispiel(e)	Bei aktiver Lageregelung wird die Achse/Spindel durch den Lageregler in Position gehalten. Somit können eventuell vorhandene Bremsen oder Klemmungen geöffnet sein. Das Nahtstellensignal: DB31, ... DBX61.5 (Lageregler aktiv) kann als Rückmeldung für das Nahtstellensignal: DB31, ... DBX2.1 (Reglerfreigabe) verwendet werden. Bei einer hängenden Achse ist die Haltebremse zu aktivieren, sobald die Lageregelung nicht mehr aktiv ist. Sofern eine Spindel technisch dafür ausgelegt ist, kann sie im Teileprogramm in den lagegeregelten Betrieb als Spindel bzw. als Achse umgeschaltet werden (mit SPCON bzw. M70). In diesen Fällen wird das Nahtstellensignal "Lageregler aktiv" gesetzt.
Sonderfälle, Fehler, ...	Sonderfall bei Simulationsachsen (MD30350 \$MA_SIMU_AX_VDI_OUTPUT = "1"): Das NST "Lageregler aktiv" wird auch bei Simulationsachsen gesetzt, sobald das MD= "1" ist.
korrespondierend mit ...	DB31, ... DBX2.1 (Reglerfreigabe) DB31, ... DBX1.4 (Nachführbetrieb) DB31, ... DBX1.5 und 1.6 (Lagemesssystem 1 und 2)
weiterführende Literatur	Diagnosehandbuch



<b>DB31, ... DBX61.6</b>	<b>Drehzahlregler aktiv (Status)</b>		
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch		
Signalzustand 1	Die Steuerung meldet, dass der Drehzahlregler für die Achse bzw. Spindel geschlossen ist.		
Signalzustand 0	Die Steuerung meldet, dass der Drehzahlregler für die Achse bzw. Spindel offen ist. Der Drehzahlreglerausgang wird gelöscht.		
Anwendungs- beispiel(e)	Bei Spindel ohne Lageregelung kann das Nahtstellensignal als Rückmeldung für das Nahtstellensignal DB31, ... DBX2.1 (Reglerfreigabe) verwendet werden.		
Sonderfälle, Fehler, ...	Das Nahtstellensignal wird auch bei Simulationsachsen gesetzt: Simulationsachse: MD30350 \$MA_SIMU_AX_VDI_OUTPUT == 1		
korrespondierend mit ...	DB31, ... DBX61.5 (Lageregler aktiv)		

<b>DB31, ... DBX61.7</b>	<b>Stromregler aktiv (Status)</b>		
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch		
Signalzustand 1	Der Stromregler für die Achse/Spindel ist aktiv.		
Signalzustand 0	Der Stromregler für die Achse/Spindel ist nicht aktiv. Der Stromreglerausgang, einschließlich der Aufschaltgrößen auf die Stellspannung, wird gelöscht.		
korrespondierend mit ...	DB31, ... DBX61.5 (Lageregler aktiv) DB31, ... DBX61.6 (Drehzahlregler aktiv)		

<b>DB31, ... DBX69.0 - DBX69.2</b>	<b>Aktiver Lagereglerparametersatz A, B, C (Rückmeldung)</b>			
Flankenauswertung: nein	Signal(e) aktualisiert: nach Umschaltung			
Bedeutung	Mit den Nahtstellensignalen A, B und C wird der aktive Lagereglerparametersätze angezeigt. Dabei gilt folgende Zuordnung:			
	<b>Parametersatz</b>	<b>C</b>	<b>B</b>	<b>A</b>
	1	0	0	0
	2	0	0	1
	3	0	1	0
	4	0	1	1
	5	1	0	0
	6	1	0	1
	6	1	1	0
	6	1	1	1
Signal irrelevant bei ...	MD35590 \$MA_PARAMSET_CHANGE_ENABLE == 0 Ist die Umschaltung des Lagereglerparametersatzes ausgeschaltet, ist immer der 1. Parametersatz aktiv.			
korrespondierend mit ...	DB31, ...DBX9.0 - DBX9.2 (Lagereglerparametersatz-Anwahl A, B, C)			

<b>DB31, ... DBX76.0</b>	<b>Schmierimpuls</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Flankenwechsel 0 → 1 bzw. 1 → 0	Sobald die Achse/Spindel die im Maschinendatum: MD33050 \$MA_LUBRICATION_DIST (Verfahrstrecke für Schmierung von PLC) eingestellte Verfahrstrecke zurückgelegt hat, wird das Nahtstellensignal invertiert. Hinweis: Die Wegmessung bei jedem Hochlauf der Steuerung neu gestartet.
Anwendungs- beispiel(e)	Mit dem Nahtstellensignal "Schmierimpuls" kann die Schmiermittelpumpe der Achse/Spindel angesteuert werden. Damit kann die Bettschmierung in Abhängigkeit von dem jeweils verfahrenen Weg erfolgen.
korrespondierend mit ...	MD33050 \$MA_LUBRICATION_DIST (Schmierimpulsdistanz)

<b>DB31, ... DBX92.1</b>	<b>Hochlaufgebersperre aktiv (Rückmeldung)</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Für den Antrieb ist der Hochlaufgeber-Schnellstopp aktiv.
Signalzustand 0	Für den Antrieb ist der Hochlaufgeber-Schnellstopp nicht aktiv.
Anwendungs- beispiel(e)	Lageregler-seitige Umgehung des Hochlaufgebers.
korrespondierend mit ...	DB31, ... DBX20.1 (Hochlaufgeber Schnellstopp)
Anmerkung	Im SINAMICS kann ein Protokoll eingestellt werden, das in einem sogenannten 611U- Kompatibilitätsmodus läuft.
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb

<b>DB31, ... DBX92.4</b>	<b>Antriebsautarke Bewegung aktiv</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Eine antriebsautarke Bewegung ist aktiv. Die Achse verfährt aufgrund von Sollwerten, die von antriebsinternen Funktionen erzeugt werden. Der Antrieb reagiert noch auf Steuersignale der NC, z.B. Reglerfreigabe. Sollwertvorgaben der NC werden ignoriert.
Signalzustand 0	Eine antriebsautarke Bewegung ist nicht aktiv.
Anwendungs- beispiel(e)	Antriebsinterne Funktionen: <ul style="list-style-type: none"> <li>• Rotor- bzw. Pollageidentifikation</li> <li>• Funktionsgenerator</li> </ul>
Anmerkung	DBX92.4 = 1 <b>WENN</b> MELDW.11 == 1 (Reglerfreigabe) <b>UND</b> ZSW1.2 == 0 (Betrieb freigegeben)
weiterführende Literatur	SINAMICS S120 Funktionshandbuch; Pollageidentifikation: Kapitel "Servoregelung" > "Pollageidentifikation" Funktionsgenerator: Kapitel "Servoregelung" > "Optimierung des Strom- und Drehzahlreglers"

<b>DB31, ... DBX93.0 - DBX93.2</b>	<b>Aktiver Antriebsparametersatz A, B, C (Rückmeldung)</b>				
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch				
Bedeutung	Mit den Nahtstellensignalen A, B und C wird der aktive Antriebsparametersatz angezeigt. Dabei gilt folgende Zuordnung:				
		<b>Parametersatz</b>	<b>C</b>	<b>B</b>	<b>A</b>
		1	0	0	0
		2	0	0	1
		3	0	1	0
		4	0	1	1
		5	1	0	0
		6	1	0	1
		7	1	1	0
	8	1	1	1	
korrespondierend mit ...	DB31, ... DBX21.0 - DBX21.2 (Antriebsparametersatz-Anwahl)				
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb				

<b>DB31, ... DBX93.3 - DBX93.4</b>	<b>Aktiver Motor A, B (Rückmeldung)</b>				
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch				
Bedeutung	Rückmeldung von Antrieb, welche Motoranwahl wirksam ist.				
	Mit den Nahtstellensignalen A, B wird der aktive Motoren bzw. Motorbetriebsart angezeigt. Dabei gilt folgende Zuordnung:				
		<b>Motoranwahl</b>	<b>Bedeutung</b>	<b>B</b>	<b>A</b>
		Motor 1	Stern-Betrieb aktiv	0	0
		Motor 2	Dreieck-Betrieb aktiv	0	1
	Motor 3	reserviert	1	0	
	Motor 4	reserviert	1	1	
korrespondierend mit ...	DB31, ... DBX21.3 und .4 (Motoranwahl) DB31, ... DBX21.5 (Motoranwahl erfolgt)				
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb				

<b>DB31, ... DBX93.5</b>	<b>Antrieb bereit</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Der Antrieb ist betriebsbereit.
Signalzustand 0	Der Antrieb ist nicht betriebsbereit. Wird das Signal im Betrieb zurückgesetzt, wird der Antrieb stillgesetzt (Impulssperre oder Schnellstopp). Im Hochlauf wird Impulssperre beibehalten. Zusätzlich werden folgende NC/PLC-Nahtstellensignale zurückgesetzt: DB10, DBX108.6 = 0 (Antrieb ready) DB31, ... DBX61.7 = 0 (Stromregler aktiv) DB31, ... DBX61.6 = 0 (Drehzahlregler aktiv)
Anmerkung	DB31, ... DBX93.5 = MELDW.12
korrespondierend mit ...	<b>SINUMERIK</b> DB10, DBX108.6 (Antriebe-Ready) DB31, ... DBX61.7 (Stromregler aktiv) DB31, ... DBX61.6 (Drehzahlregler aktiv)
weiterführende Literatur	SINAMICS S120S150 Listenhandbuch Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb

<b>DB31, ... DBX93.6</b>	<b>Integrator n-Regler gesperrt (Rückmeldung)</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die angeforderte Abschaltung des Integrators des Drehzahlreglers ist im Antrieb wirksam. Der Drehzahlregler wurde von PI- auf P-Verhalten umgeschaltet.
Signalzustand 0	Der Integrator des Drehzahlreglers ist freigegeben. Der Drehzahlregler wirkt als PI-Regler.
korrespondierend mit ...	DB31, ... DBX21.6 (Integratorsperre n-Regler)
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb

<b>DB31, ... DBX93.7</b>	<b>Impulse freigegeben (Rückmeldung)</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Für die Achse/Spindel sind die Impulse des zugeordneten Antriebs freigegeben.
Signalzustand 0	Die Impulse des zugeordneten Antriebs sind nicht freigegeben. Sind die Impulse am zugeordneten Antrieb gelöscht, werden folgende Nahtstellensignale zurückgesetzt: <ul style="list-style-type: none"> <li>• DB31, ... DBX61.7 (Stromregler aktiv)</li> <li>• DB31, ... DBX61.6 (Drehzahlregler aktiv)</li> <li>• DB31, ... DBX61.5 (Lageregler aktiv)</li> </ul>
Anmerkung	DB31, ... DBX93.7 = MELDW.13
korrespondierend mit ...	DB31, ... DBX21.7 (Impulsfreigabe)
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb

<b>DB31, ... DBX94.0</b>	<b>Temperaturvorwarnung Motor</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die Motortemperatur hat die im Antrieb projektierte Warnschwelle (p0604) überschritten. <b>Hinweis</b> Bleibt die Motortemperatur länger als die parametrisierte Zeitstufe (p0606) zu hoch, wird eine Störung ausgegeben, der Antrieb stillgesetzt und die Impulsfreigabe weggenommen. Sinkt die Motortemperatur vor Ablauf der Zeitstufe (p0606) wieder unter die Warnschwelle (p0604), wird das Nahtstellensignal wieder zurückgesetzt.
Signalzustand 0	Die Motortemperatur liegt unterhalb der Warnschwelle (p0604).
Anmerkung	Die aktuelle Motortemperatur wird auf der Bedienoberfläche angezeigt unter: Bedienbereich "Diagnose" > "Service-Anzeige: Achse/Spindel"
korrespondierend mit ...	DB31, ... DBX94.1 (Temperaturvorwarnung Kühlkörper)
weiterführende Literatur	Siehe DB31, ... DBX94.1 (Temperaturvorwarnung Kühlkörper)

<b>DB31, ... DBX94.1</b>	<b>Temperaturvorwarnung Kühlkörper</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die Kühlkörpertemperatur der Leistungshalbleiter hat die parametrisierte Warnschwelle (p0294) überschritten. <b>Hinweis</b> Im Antrieb erfolgt die parametrisierte Reaktion (p0290). Bleibt die Temperaturüberschreitung bestehen, wird nach ca. 20 s eine Störung ausgegeben, der Antrieb stillgesetzt und die Impulsfreigabe weggenommen.
Signalzustand 0	Die Kühlkörper-Temperatur-Überwachung des Antriebsmoduls hat nicht angesprochen, d. h. die Temperatur ist unterhalb der Warnschwelle.
Anmerkung	Die Nahtstellensignale DB31, ... DBX94.0 und .1 werden von folgenden Signalen des zyklischen Antriebstelegramms abgeleitet: <ul style="list-style-type: none"> <li>• Fall 1: Temperaturwarnung im Meldungswort <ul style="list-style-type: none"> <li>– DB31, ... DBX94.0 <math>\triangleq</math> MELDW, Bit 6 (Keine Warnung Übertemperatur Motor)</li> <li>– DB31, ... DBX94.1 <math>\triangleq</math> MELDW, Bit 7 (Keine Warnung thermische Überlast Leistungsteil)</li> </ul> </li> <li>• Fall 2: Warnung der Warnungsklasse B (Interface Mode "SIMODRIVE 611u", p2038=1) DB31, ... DBX94.0 == 1 und DBX94.1 == 1, wenn gilt: zykl. Antriebstelegramm, ZSW1: Bit 11 == 0 und 12 == 1 (Warnungsklasse B)</li> </ul> Die Nahtstellensignale werden aus der Warnung der Warnklasse B abgeleitet, wenn keine spezifische Information aus dem Meldewort vorliegt. Es wird ein Alarm angezeigt: Alarmnummer = 200.000 + Warnwert (r2124)
weiterführende Literatur	<ul style="list-style-type: none"> <li>• S120 Inbetriebnahmehandbuch, Kapitel "Inbetriebnahme" &gt; "Temperatursensoren bei SINAMICS-Komponenten"</li> <li>• S120 Funktionshandbuch, Kapitel "Überwachungs- und Schutzfunktionen"</li> <li>• S120 Listenhandbuch <ul style="list-style-type: none"> <li>– MELDW, Bit 6 <math>\triangleq</math> BO: r2135.14 → Funktionsplan: 2548, 8016</li> <li>– MELDW, Bit 7 <math>\triangleq</math> BO: r2135.15 → Funktionsplan: 2548, 2452, 2456, 8016</li> </ul> </li> <li>• SINUMERIK Diagnosehandbuch</li> </ul>

<b>DB31, ... DBX94.2</b>	<b>Hochlaufvorgang beendet</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Der Drehzahlwert hat nach neuer Drehzahlsollwertvorgabe das über p2164 festgelegte Drehzahltoleranzband erreicht und es für die Zeitdauer von p2166 nicht verlassen. Nachfolgende Drehzahlschwankungen, auch außerhalb des Toleranzbandes z. B. infolge Belastungsänderungen, haben auf das Nahtstellensignal keinen Einfluss.
Signalzustand 0	Der Hochlaufvorgang ist nach einer Veränderung des Drehzahlsollwerts noch aktiv.
korrespondierend mit ...	DB31, ... DBX94.6 ("n <sub>ist</sub> = n <sub>soll</sub> ") DB31, ... DBX94.3 (" M <sub>D</sub>   = M <sub>dx</sub> ")
weiterführende Literatur	<ul style="list-style-type: none"> <li>• SINUMERIK Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb</li> <li>• SIMATIC S120 Listenhandbuch</li> </ul>

<b>DB31, ... DBX94.3</b>	$ M_d  < M_{dx}$	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Die aktuelle Momentenausnutzung liegt unter der Momentenausnutzungsschwelle (Momentenschwellwert 2, p2194). Der Hochlaufvorgang ist abgeschlossen, der Antrieb befindet sich im stationären Zustand und der Momentensollwert $ M_d $ Antrieb überschreitet das Schwellenmoment $M_{dx}$ nicht. Der Verlauf der Momentenschwelle ist drehzahlabhängig. Während des Hochlaufvorgangs ist $DB31, \dots, DBX94.3 ( M_d  < M_{dx}) == 1$ . Das Nahtstellensignal wird erst aktualisiert, nachdem der Hochlaufvorgang beendet ist ( $DB31, \dots, DBX94.2 == 1$ ) und die Meldungsverriegelungszeit für das Schwellenmoment abgelaufen ist.	
Signalzustand 0	Der Momentensollwert $ M_d $ ist größer als das Schwellenmoment $M_{dx}$ . Über das Nahtstellensignal kann eine Überlastung des Motors festgestellt werden. Im PLC-Anwenderprogramm kann daraufhin eine entsprechende Reaktion eingeleitet werden.	
Anmerkung	DB31, ... DBX94.3 = MELDW.1	
weiterführende Literatur	<ul style="list-style-type: none"> <li>• SINUMERIK Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb</li> <li>• SIMATIC S120 Listenhandbuch</li> </ul>	

<b>DB31, ... DBX94.4</b>	$ n_{ist}  < n_{min}$	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Der Drehzahlwert $n_{ist}$ ist kleiner als $n_{min}$ (Drehzahlschwellwert 3, p2161).	
Signalzustand 0	Der Drehzahlwert ist größer als die Schwellenminimaldrehzahl $n_{min}$ .	
Anmerkung	DB31, ... DBX94.4 = MELDW.2	
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb	

<b>DB31, ... DBX94.5</b>	$ n_{ist}  < n_x$	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Der Drehzahlwert $n_{ist}$ ist kleiner als $n_x$ (Drehzahlschwellwert 2, p2155).	
Signalzustand 0	Der Drehzahlwert $n_{ist}$ ist größer als die Schwellendrehzahl $n_x$ .	
Anmerkung	DB31, ... DBX94.5 = MELDW.3	
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb	

<b>DB31, ... DBX94.6</b>	<b>n<sub>ist</sub> = n<sub>soll</sub></b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Der Drehzahlwert liegt mindestens die parametrisierte Zeit (Einschaltverzögerung n_ist = n_soll, p2167) innerhalb des Toleranzbandes um den Drehzahlsollwert (Drehzahlschwellwert 4, p2163).	
Signalzustand 0	Der Drehzahlwert liegt außerhalb des Toleranzbandes um den Drehzahlsollwert (Drehzahlschwellwert 4, p2163).	
weiterführende Literatur	Inbetriebnahmehandbuch IBN CNC: NCK, PLC, Antrieb	

<b>DB31, ... DBX94.7</b>	<b>Variable Meldefunktion</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Die parametrisierte Antriebsgröße hat den vorgegebene Schwellwert, einschließlich Hysterese, überschritten.	
Signalzustand 0	Die parametrisierte Antriebsgröße hat den vorgegebene Schwellwert, einschließlich Hysterese, unterschritten.	
Anmerkung	Mit der Funktion "Variable Meldefunktion" können im Antrieb BICO-Verschaltungen und Parameter, die das Attribut traceable haben, überwacht werden. DB31, ... DBX94.7 = MELDW.5	
weiterführende Literatur	SINAMICS S120 Funktionshandbuch, Kapitel "Servoregelung" > "Variable Meldefunktion"	

<b>DB31, ... DBX95.7</b>	<b>Warnung der Warnungsklasse C steht an</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Der Antrieb meldet, dass eine Warnung der Warnungsklasse C ansteht.	
Signalzustand 0	Der Antrieb meldet, dass keine Warnung der Warnungsklasse C ansteht.	
Anmerkung	Eine Warnung ist im Antrieb die Reaktion auf einen erkannten möglichen oder zu erwartenden Fehlerzustand, der nicht zum Abschalten des Antriebs führt und nicht quittiert werden muss.	
weiterführende Literatur	SINAMICS S120 Listenhandbuch, Kapitel "Störungen und Warnungen"	



## 18.2 Achsüberwachungen, Schutzbereiche (A3)

### 18.2.1 Signale an Kanal (DB21, ...)

<b>DB21, ... DBX1.1</b>	<b>Schutzbereiche freigeben</b>	
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bei einer positiven Flanke dieses Signals wird ein Schutzbereich freigegeben und der anstehende Alarm gelöscht. Danach kann ein Start der Bewegung in den gleichen Schutzbereich erfolgen. Durch den Start der Bewegung wird der Schutzbereich freigegeben, die NST "maschinen- bzw. kanalspezifischer Schutzbereich verletzt" gesetzt und die Bewegung der Achse begonnen. Wird eine Bewegung gestartet, die nicht in den freigegebenen Schutzbereich führt, wird die Freigabe hinfällig.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	keine Wirkung	
Anwendungs- beispiel(e)	Damit können Schutzbereiche freigegeben werden: <ul style="list-style-type: none"> <li>• Wenn die aktuelle Position im Schutzbereich liegt (Alarm 2 steht an).</li> <li>• Wenn auf der Schutzbereichsgrenze eine Bewegung gestartet werden soll (Alarm 1 oder 2 steht an).</li> </ul>	

<b>DB21, ... DBX8.0 - DBX9.1</b>	<b>maschinenbezogenen Schutzbereich 1 (... 10) aktivieren</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der voraktivierte, maschinenbezogene Schutzbereich 1 (...10) wird durch das PLC-Anwenderprogramm wirksam gesetzt. Der Schutzbereich ist sofort wirksam. Wirksam gesetzt werden können nur Schutzbereiche, die im Teileprogramm voraktiviert wurden.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der voraktivierte, maschinenbezogene Schutzbereich 1 (...10) wird durch das PLC-Anwenderprogramm unwirksam gesetzt. Der Schutzbereich ist sofort unwirksam. Unwirksam gesetzt werden können nur Schutzbereiche, die durch die PLC wirksam gesetzt und im NC-Teileprogramm voraktiviert wurden.	
Anwendungs- beispiel(e)	Damit kann z. B. vor dem Einschwenken eines Messtasters in den Arbeitsraum der zugehörige maschinenbezogene Schutzbereich wirksam gesetzt werden.	

<b>DB21, ... DBX10.0 - DBX11.1</b>	<b>kanalspezifischen Schutzbereich 1 (...10) aktivieren</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der voraktivierte, kanalspezifische Schutzbereich 1 (...10) wird durch das PLC-Anwenderprogramm wirksam gesetzt. Der Schutzbereich ist sofort wirksam. Wirksam gesetzt werden können nur Schutzbereiche, die im Teileprogramm voraktiviert wurden.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der voraktivierte, kanalspezifische Schutzbereich 1 (...10) wird durch das PLC-Anwenderprogramm unwirksam gesetzt. Der Schutzbereich ist sofort unwirksam. Unwirksam gesetzt werden können nur Schutzbereiche, die durch die PLC wirksam gesetzt und im NC-Teileprogramm voraktiviert wurden.	
Anwendungs- beispiel(e)	Damit kann z.B. vor dem Einfahren einer Synchronspindel in den Arbeitsraum der zugehörige kanalspezifische Schutzbereich wirksam gesetzt werden.	

### 18.2.2 Signale von Kanal (DB21, ...)

<b>DB21, ... DBX272.0 – DBX273.1</b>	<b>maschinenbezogener Schutzbereich 1 (...10) voraktiviert</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der maschinenbezogene Schutzbereich 1 (...10) ist im aktuellen Satz voraktiviert. (Die Voraktivierung erfolgte im Teileprogramm.) Der Schutzbereich kann damit im PLC-Anwenderprogramm über das Nahtstellensignal: DB21, ... DBX8.0 - DBX9.1 (maschinenbezogenen Schutzbereich 1 (...10) aktivieren) wirksam bzw. unwirksam gesetzt werden.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der maschinenbezogene Schutzbereich 1 (...10) ist im aktuellen Satz deaktiviert. (Die Deaktivierung erfolgte im Teileprogramm). Der Schutzbereich kann damit im PLC-Anwenderprogramm über das Nahtstellensignal: DB21, ... DBX8.0 bis DBX9.1 (maschinenbezogenen Schutzbereich 1 (...10) aktivieren) nicht wirksam bzw. unwirksam gesetzt werden.	
Korrespondierend mit ....	DB21, ... DBX8.0 - DBX9.1 (maschinenbezogenen Schutzbereich 1 (...10) aktivieren)	

<b>DB21, ... DBX274.0 – DBX275.1</b>	<b>kanalspezifischer Schutzbereich 1 (...10) voraktiviert</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der kanalspezifische Schutzbereich 1 (...10) ist im aktuellen Satz voraktiviert. (Die Voraktivierung erfolgte im Teileprogramm.) Der Schutzbereich kann damit im PLC-Anwenderprogramm über das Nahtstellensignal: DB21, ... DBX10.0 - DBX11.1 (kanalspezifischen Schutzbereich 1 (...10) aktivieren)) wirksam bzw. unwirksam gesetzt werden.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der kanalspezifische Schutzbereich 1 (...10) ist im aktuellen Satz deaktiviert. (Die Deaktivierung erfolgte im Teileprogramm.) Der Schutzbereich kann damit im PLC-Anwenderprogramm über das Nahtstellensignal: DB21, ... DBX10.0 - DBX11.1 (kanalspezifischen Schutzbereich 1 (...10) aktivieren)) nicht wirksam bzw. unwirksam gesetzt werden.	
Korrespondierend mit ....	DB21, ... DBX10.0 - DBX11.1 (kanalspezifischen Schutzbereich 1 (...10) aktivieren))	

<b>DB21, ... DBX276.0 – DBX277.1</b>	<b>maschinenbezogener Schutzbereich 1 (...10) verletzt</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der aktivierte, maschinenbezogene Schutzbereich 1 (...10) wird im aktuellen Satz, bzw. in der aktuellen JOG-Bewegung verletzt. Der voraktivierte, maschinenbezogene Schutzbereich 1 (...10) würde im aktuellen Satz verletzt werden, wenn er durch die PLC wirksam gesetzt wäre.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der aktivierte, maschinenbezogene Schutzbereich 1 (...10) wird im aktuellen Satz nicht verletzt. Der voraktivierte, maschinenbezogene Schutzbereich 1 (...10) würde im aktuellen Satz nicht verletzt werden, wenn er durch die PLC wirksam gesetzt wäre	
Anwendungs- beispiel(e)	Mit dem NST kann vor dem Einschwenken von Teilen in den Arbeitsraum überprüft werden, ob sich das Werkzeug bzw. Werkstück in dem maschinenbezogenen Schutzbereich des einzuschwenkenden Teiles befindet.	

<b>DB21, ... DBX278.0 - DBX279.1</b>	<b>kanalspezifischer Schutzbereich 1 (...10) verletzt</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der aktivierte, kanalspezifische Schutzbereich 1 (...10) wird im aktuellen Satz verletzt. Der voraktivierte, kanalspezifische Schutzbereich 1 (...10) würde im aktuellen Satz verletzt werden, wenn er durch die PLC wirksam gesetzt wäre.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der aktivierte, kanalspezifische Schutzbereich 1 (...10) wird im aktuellen Satz nicht verletzt. Der voraktivierte, kanalspezifische Schutzbereich 1 (...10) würde im aktuellen Satz nicht verletzt werden, wenn er durch die PLC wirksam gesetzt wäre.
Anwendungs- beispiel(e)	Mit dem NST kann vor dem Einschwenken von Teilen in den Arbeitsraum überprüft werden, ob sich das Werkzeug bzw. Werkstück in dem kanalspezifischen Schutzbereich des einzuschwenkenden Teiles befindet.

### 18.2.3 Signale an Achse/Spindel (DB31, ...)

<b>DB31, ... DBX2.3</b>	<b>Klemmvorgang läuft</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Klemmvorgang läuft. Die Klemmungsüberwachung wird aktiviert.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Klemmvorgang beendet. Die Klemmungsüberwachung wird von der Stillstandsüberwachung abgelöst.
korrespondierend mit ...	MD36050 \$MA_CLAMP_POS_TOL (Klemmungstoleranz)

<b>DB31, ... DBX3.6</b>	<b>Geschwindigkeits-/Spindeldrehzahlbegrenzung</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die NCK begrenzt die Geschwindigkeit/Spindeldrehzahl auf den Grenzwert, der im Maschinendatum: MD35160 \$MA_SPIND_EXTERN_VELO_LIMIT eingetragen ist.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Begrenzung aktiv.
korrespondierend mit ...	MD35100 \$MA_SPIND_VELO_LIMIT (max. Spindeldrehzahl) SD43220 \$SA_SPIND_MAX_VELO_G26 (prog. Spindeldrehzahlbegrenzung G26) MD43230 \$SA_SPIND_MAX_VELO_LIMIT (prog. Spindeldrehzahlbegrenzung G96/G961)

<b>DB31, ... DBX12.0 - DBX12.1</b>	<b>Hardwareendschalter plus und minus</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Am Ende der beiden Seiten des Verfahrbereiches einer Maschinenachse kann jeweils ein Schalter angebracht sein, der beim Anfahren über die PLC ein Signal "Hardwareendschalter plus oder minus" an die NC gibt.</p> <p>Wird das Signal als gesetzt erkannt, so wird der Alarm 021614 "Hardwareendschalter + bzw. -" ausgegeben und die Achse sofort abgebremst.</p> <p>Auf welche Art, wird mit dem Maschinendatum: MD36600 \$MA_BRAKE_MODE_CHOICE (Bremsverhalten bei Hardwareendschalter) festgelegt.</p> <p>Wird zum Signal "Hardwareendschalter" noch die Reglerfreigabe weggenommen, so reagiert die Achse wie im Kapitel A2 ("Diverse Nahtstellensignale") erläutert.</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Normalzustand, kein HW-Endschalter angesprochen.	
korrespondierend mit ...	MD36600 \$MA_BRAKE_MODE_CHOICE (Bremsverhalten bei Hardwareendschalter)	

<b>DB31, ... DBX12.2 - DBX12.3</b>	<b>2. Softwareendschalter plus bzw. minus</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>2. Softwareendschalter für die Plus- bzw. Minus-Richtung ist wirksam. 1. Softwareendschalter für die Plus- bzw. Minus-Richtung ist unwirksam.</p> <p>Zusätzlich zu den 1. Softwareendschaltern (Plus bzw. Minus) können über diese Nahtstellensignale die 2. Softwareendschalter (Plus bzw. Minus) aktiviert werden.</p> <p>Die Position (Lage) wird durch die Maschinendaten: MD36130 \$MA_POS_LIMIT_PLUS2 (2. Softwareendschalter Plus) und MD36120 \$MA_POS_LIMIT_MINUS2 (2. Softwareendschalter Minus) festgelegt.</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>1. Softwareendschalter für die Plus- bzw. Minus-Richtung ist wirksam. 2. Softwareendschalter für die Plus- bzw. Minus-Richtung ist unwirksam.</p>	
korrespondierend mit ...	<p>MD36110 \$MA_POS_LIMIT_PLUS (1. Softwareendschalter Plus) MD36130 \$MA_POS_LIMIT_PLUS2 (2. Softwareendschalter Plus) MD36100 \$MA_POS_LIMIT_MINUS (1. Softwareendschalter Minus) MD36120 \$MA_POS_LIMIT_MINUS2 (2. Softwareendschalter Minus)</p>	

### 18.2.4 Signale von Achse/Spindel (DB31, ...)

<b>DB31, ... DBX60.2 - DBX60.3</b>	<b>Gebergrenzfrequenz überschritten 1</b> <b>Gebergrenzfrequenz überschritten 2</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die im Maschinendatum: MD36300 \$MA_ENC_FREQ_LIMIT (Gebergrenzfrequenz) eingestellte Grenzfrequenz ist überschritten.  Der Referenzpunkt für das betreffende Lagemesssystem ist verloren (NST: Referiert/Synchronisiert hat Signalzustand 0). Eine Lageregelung ist nicht mehr möglich. Spindeln laufen mit Drehzahlregelung weiter. Achsen werden mit Schnellstopp (mit offenem Lageregelkreis) über eine Drehzahlsollwerttrampe stillgesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die im Maschinendatum: MD36300 \$MA_ENC_FREQ_LIMIT eingestellte Grenzfrequenz ist nicht mehr überschritten.  Für den Flankenwechsel 1 → 0 muss die Geberfrequenz den Wert von Maschinendatum: MD36302 \$MA_ENC_FREQ_LIMIT_LOW (Gebergrenzfrequenz für Geber-Neusynchronisation) unterschritten haben.

## 18.3 Bahnsteuerbetrieb, Genauhalt und LookAhead (B1)

### 18.3.1 Signale von Kanal (DB21, ...)

<b>DB21, ... DBX36.3</b>	<b>Alle Achsen stehen</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Alle Achsen des Kanals stehen mit Interpolator-Ende. Es stehen keine weiteren Verfahrbewegungen an.	

### 18.3.2 Signale von Achse/Spindel (DB31, ...)

<b>DB31, ... DBX60.6</b>	<b>Position erreicht mit Genauhalt grob</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse ist im entsprechenden Genauhalt und für die Achse ist kein Interpolator aktiv und: <ul style="list-style-type: none"> <li>• die Steuerung ist im Reset-Zustand (Reset-Taste bzw. Programmende).</li> <li>• die Achse ist zuletzt als Positionierachse bzw. positionierende Spindel programmiert worden (Grundeinstellung der Zusatzachse: Positionierachse).</li> <li>• die Bahnbewegung ist mit NC-Stop beendet worden.</li> <li>• die Spindel ist im lagegeregelten Modus (<i>SPCON</i>/<i>SPOS</i>-Anweisung) und steht.</li> <li>• die Achse wird vom drehzahlgeregelten in den lagegeregelten Modus mit dem NST "Lagemesssystem" umgeschaltet.</li> </ul>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Achse ist nicht im entsprechenden Genauhalt oder für die Achse ist der Interpolator aktiv oder: <ul style="list-style-type: none"> <li>• die Bahnbewegung ist mit NC-Stop beendet worden.</li> <li>• die Spindel ist im drehzahlgeregeltem Modus (<i>SPCOF</i>/<i>SPOSA</i>-Anweisung).</li> <li>• der Modus "Nachführbetrieb" ist für die Achse aktiv.</li> <li>• der Modus "Parken" ist für die Achse aktiv.</li> <li>• die Achse wird vom lagegeregelten in drehzahlgeregelten Modus mit dem NST "Lagemesssystem" umgeschaltet.</li> </ul>	
Signal irrelevant bei ...	Rundachsen, die als Rundungsachsen definiert sind.	
korrespondierend mit ...	MD36000 \$MA_STOP_LIMIT_COARSE (Genauhalt grob)	

<b>DB31, ... DBX60.7</b>	<b>Position erreicht mit Genauhalt fein</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Siehe DB31, ... DBX60.6 (Position erreicht mit Genauhalt grob).	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Siehe DB31, ... DBX60.6 (Position erreicht mit Genauhalt grob).	
Signal irrelevant bei ...	Rundachsen, die als Rundungsachsen definiert sind.	
korrespondierend mit ...	MD36010 \$MA_STOP_LIMIT_FINE (Genauhalt fein)	



## 18.4 Fahren auf Festanschlag (F1)

### 18.4.1 Signale an Achse/Spindel (DB31, ...)

<b>DB31, ... DBX1.1</b>	<b>Festanschlag erreicht quittieren</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Bedeutung nach dem Erreichen des Festanschlags:          DB31, ... DBX62.5 (Festanschlag erreicht) = 1</p> <p>→ Die Achse drückt mit dem Klemmmoment gegen den Festanschlag:          → Das Festanschlags-Überwachungsfenster ist aktiviert.          → Es wird ein Satzwechsel durchgeführt.</p>
Signalzustand 0  Flankenwechsel 1 → 0	<p>Bedeutung nach dem Erreichen des Festanschlags:          DB31, ... DBX62.5 (Festanschlag erreicht) = 1</p> <p>→ Die Achse drückt mit dem Klemmmoment gegen den Festanschlag.          → Das Festanschlag-Überwachungsfenster ist aktiviert.          → Es wird kein Satzwechsel durchgeführt und es wird die Kanalmeldung          "Warten: HiFu- Quittierung fehlt" angezeigt.</p> <p>Bedeutung nach dem Erreichen des Festanschlags:          NST "Festanschlag erreicht" DB31, ... DBX62.5 = 1</p> <p>→ Die Funktion wird abgebrochen, es wird der Alarm "20094 Achse %1 Funktion wurde          abgebrochen" angezeigt.</p> <p>Bedeutung bei Abwahl der Funktion <math>F_{XS}=0</math> über Teileprogramm:          → Die Momentenbegrenzung und die Überwachung des Festanschlags-Überwachungsfensters wird          aufgehoben.</p>
NST irrelevant bei ...	MD37060 \$MA_FIXED_STOP_ACKN_MASK (Beachtung von PLC-Quittierungen für Fahren auf Festanschlag) = 0 oder 1
korrespondierend mit ...	MD37060 \$MA_FIXED_STOP_ACKN_MASK (Beachtung von PLC-Quittierungen für Fahren auf Festanschlag) DB31, ... DBX62.5 (Festanschlag erreicht)

18.4 Fahren auf Festanschlag (F1)

<b>DB31, ... DBX1.2</b>	<b>Sensor Festanschlag</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Festanschlag ist erreicht.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Festanschlag ist nicht erreicht.
korrespondierend mit ...	Das Signal ist nur wirksam, wenn: MD37040 \$MA_FIXED_STOP_BY_SENSOR = 1

<b>DB31, ... DBX3.1</b>	<b>Fahren auf Festanschlag freigeben</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bedeutung bei Anwahl der Funktion $F_{XS}$ über Teileprogramm (NST "Fahren auf Festanschlag aktivieren" = 1): → Fahren auf Festanschlag wird freigegeben und die Achse fährt von der Startposition mit der programmierten Geschwindigkeit auf die programmierte Zielposition.
Signalzustand 0  Flankenwechsel 1 → 0	Bedeutung bei der Anwahl der Funktion $F_{XS}$ über Teileprogramm (NST "Fahren auf Festanschlag aktivieren" = 1): → Fahren auf Festanschlag ist gesperrt. → Die Achse steht mit reduziertem Moment auf der Startposition. → Die Kanalmeldung "Warten: HiFu-Quittierung fehlt" wird angezeigt.  Bedeutung vor dem Erreichen des Festanschlages (NST "Festanschlag erreicht" = 0): → Fahren auf Festanschlag wird abgebrochen. → Der Alarm "20094: Achse%1 Funktion wurde abgebrochen" wird angezeigt.  Bedeutung nach dem Erreichen des Festanschlages NST "Festanschlag erreicht" = 1): → Die Momentenbegrenzung und die Überwachung des Festanschlag-Überwachungsfensters werden aufgehoben. Abwahl: DB31, ...DBX1.1 (Festanschlag erreicht quittieren)
NST irrelevant bei ...	MD37060 \$MA_FIXED_STOP_ACKN_MASK (Beachtung von PLC-Quittierungen für Fahren auf Festanschlag) = 0 oder 2
korrespondierend mit ...	MD37060 \$MA_FIXED_STOP_ACKN_MASK (Beachtung von PLC-Quittierungen für Fahren auf Festanschlag) DB31, ... DBX62.4 (Fahren auf Festanschlag aktivieren)

## 18.4.2 Signale von Achse/Spindel (DB31, ...)

<b>DB31, ... DBX62.4</b>	<b>Fahren auf Festanschlag aktivieren</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Funktion "Fahren auf Festanschlag" ist aktiv. Dieses Signal wird bei analogen Antrieben verwendet, um z.B. die im Steller parametrisierte Strom- bzw. Momentenbegrenzung zu aktivieren.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Funktion "Fahren auf Festanschlag" ist nicht aktiv.

<b>DB31, ... DBX62.5</b>	<b>Festanschlag erreicht</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Nach Anwahl der Funktion $F_{XS}$ wurde der Festanschlag erreicht. Dieses Signal wird bei analogen Antrieben verwendet, um z.B. den Steller vom drehzahlgeregelten in den strom- bzw. momentengeregelten Betrieb zu schalten, damit ein programmierbares Klemmmoment vorgegeben werden kann.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Nach Anwahl der Funktion $F_{XS}$ wurde der Festanschlag noch nicht erreicht.

## 18.5 Hilfsfunktionsausgaben an PLC (H2)

### 18.5.1 Signale an Kanal (DB21, ...)

<b>DB21, ... DBX30.5</b>	<b>Assoziiertes M01 aktivieren</b>	
Flankenauswertung: nein	Signal(e) aktualisiert:	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	PLC meldet an NCK, dass assoziiertes M01 (Hilfsfunktion) aktiviert werden soll.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Assoziiertes M01 (Hilfsfunktion) deaktivieren.	
korrespondierend mit ...	DB21, ... DBX 318.5 (assoziiertes M01 aktiv)	

### 18.5.2 Signale von Kanal (DB21, ...)

<b>DB21, ... DBB58, DBB60 - DBB65</b>	<b>M-, S-, T-, D-, H-, F-Fkt. Änderung</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Eine M-, S-, T-, D-, H- oder F-Information wurde mit einem neuen Wert zusammen mit dem zugehörigem Änderungssignal zu Beginn eines OB1-Zyklus auf die Nahtstelle ausgegeben. Dabei zeigt das Änderungssignal an, dass der entsprechende Wert gültig ist.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Änderungssignale werden zu Beginn des nächsten OB1-Zyklus durch das PLC-Grundprogramm zurückgesetzt. Der Wert der jeweiligen Information ist nicht gültig.	

<b>DB21, ... DBX59.0 - DBX59.4</b>	<b>M-Fkt. 1-5 nicht decodiert</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>M-Funktion größer 99 (bei erweiterter Adresse = 0) oder bei erweiterter Adresse &gt; 0 nicht in der Dekodierliste enthalten. Dieses Signal steht zusammen mit dem zugehörigen M-Änderungssignal einen OB1-Zyklus an. Ursache:</p> <ul style="list-style-type: none"> <li>• Falsche M-Funktion programmiert</li> <li>• M-Funktion in der Dekodierliste der PLC nicht projiziert</li> </ul> <p>Abhilfe z.B.:</p> <ul style="list-style-type: none"> <li>• PLC setzt Einleesperre</li> <li>• Ausgabe eines PLC-Alarmes</li> </ul>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	M-Funktion kleiner 99 (bei erweiterter Adresse = 0) oder bei erweiterter Adresse > 0 in der Dekodierliste enthalten.	

<b>DB21, ... DBB60 - DBB64, DBB66 - DBB67</b>	<b>M-, S-, T-, D-, H-, F-Fkt. Zusatzinformation "Quick" (schnelle Quittung)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Eine M-, S-, T-, D-, H- oder F-Information wurde mit einem neuen Wert zusammen mit dem zugehörigen Änderungssignal zu Beginn eines OB1-Zyklus auf die Nahtstelle ausgegeben. Dabei zeigt die Zusatzinformation "Quick" die schnelle Hilfsfunktion an.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Änderungssignale werden zu Beginn des nächsten OB1-Zyklus durch das PLC-Grundprogramm zurückgesetzt. Der Wert der jeweiligen Information ist nicht gültig.	

<b>DB21, ... DBB68 - DBB97</b>	<b>M-Funktion 1 bis 5</b> <b>Erweiterte Adresse M-Funktion 1 bis 5</b>
Flankenauswertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Hier werden bis zu 5 in einem NC-Satz programmierte M-Funktionen gleichzeitig bereitgestellt, sobald die M-Änderungssignale anstehen. Wertebereich der M-Funktionen: 0 bis 9999 9999; ganzzahlig Wertebereich der erweiterten Adresse: 0 bis 99; ganzzahlig Die M-Funktionen bleiben stehen, bis sie durch neue M-Funktionen überschrieben werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<ul style="list-style-type: none"> <li>• Nach Hochlauf der PLC.</li> <li>• Vor Eintrag einer neuen Hilfsfunktion werden alle anderen gelöscht.</li> </ul>
Anwendungs- beispiel(e)	Dekodierung bzw. Auswertung von M-Funktionen, die nicht standardmäßig oder über eine Liste dekodiert werden. Über die erweiterte Adresse kann die M-Funktion einem anderen Kanal zugeordnet werden, der nicht dem entspricht, in dem das Programm läuft.
Sonderfälle, Fehler, .....	Bei M00 bis M99 ist die erweiterte Adresse = 0.

<b>DB21, ... DBB98 - DBB115</b>	<b>S-Funktion 1 bis 3</b> <b>Erweiterte Adresse S-Funktion 1 bis 3</b>
Flankenauswertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Hier werden bis zu 3 in einem NC-Satz programmierte S-Funktionen gleichzeitig bereitgestellt, sobald die S-Änderungssignale anstehen. Wertebereich der Spindeldrehzahl: 0 bis 999 999; ganzzahlig Wertebereich der erweiterten Adresse: 0 bis 6; ganzzahlig Die S-Funktionen bleiben stehen, bis sie durch neue S-Funktionen überschrieben werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<ul style="list-style-type: none"> <li>• Nach Hochlauf der PLC.</li> <li>• Vor Eintrag einer neuen Hilfsfunktion werden alle anderen gelöscht.</li> </ul>
Anwendungs- beispiel(e)	Spindeldrehzahlsteuerung durch die PLC. Über die erweiterte Adresse wird programmiert, für welche Spindel das S-Wort gültig ist. Z. B.: S2=500

DB21, ... DBB118	T-Funktion 1	
Flankenauswertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Hier wird die in einem NC-Satz programmierte T-Funktion bereitgestellt, sobald das T-Änderungssignal ansteht. Wertebereich der T-Funktion: 0 bis 99 999 999; ganzzahlig Die T-Funktion bleibt stehen, bis sie durch eine neue T-Funktion überschrieben wird.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<ul style="list-style-type: none"> <li>• Nach Hochlauf der PLC.</li> <li>• Vor Eintrag einer neuen Hilfsfunktion werden alle anderen gelöscht.</li> </ul>	
Anwendungs- beispiel(e)	Steuerung der automatischen Werkzeugauswahl.	
Sonderfälle, Fehler, .....	Mit T <sub>0</sub> wird das aktuelle Werkzeug aus der Werkzeughalterung entfernt und kein neues eingewechselt (Standardprojektierung des Maschinenherstellers).	
Anmerkung	8 dekadige T-Nr. sind nur als T-Funktion 1 verfügbar.	

DB21, ... DBB129	D-Funktion 1	
Flankenauswertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Hier wird die in einem NC-Satz programmierte D-Funktion bereitgestellt, sobald das D-Änderungssignal ansteht. Wertebereich der D-Funktion: 0 bis 999; ganzzahlig Die D-Funktion bleibt stehen, bis sie durch eine neue D-Funktion überschrieben wird.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<ul style="list-style-type: none"> <li>• Nach Hochlauf der PLC.</li> <li>• Vor Eintrag einer neuen Hilfsfunktion werden alle anderen gelöscht.</li> </ul>	
Anwendungs- beispiel(e)	Realisierung von Schutzfunktionen.	
Sonderfälle, Fehler, .....	D <sub>0</sub> ist für die Abwahl der aktuellen Werkzeugkorrektur reserviert.	

<b>DB21, ... DBB140 - DBB157</b>	<b>H-Funktion 1 bis 3 Erweiterte Adresse H-Funktion 1 bis 3</b>
Flankenauswertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Hier werden bis zu 3 in einem NC-Satz programmierte H-Funktionen gleichzeitig bereitgestellt, sobald die H-Änderungssignale anstehen. Wertebereich der H-Funktion: Gleitpunkt (entsprechend MC5+Format) Wertebereich der erweiterten Adresse: 0 bis 99; ganzzahlig Die H-Funktionen bleiben stehen, bis sie durch neue H-Funktionen überschrieben werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<ul style="list-style-type: none"> <li>Nach Hochlauf der PLC.</li> <li>Vor Eintrag einer neuen Hilfsfunktion werden alle anderen gelöscht.</li> </ul>
Anwendungs- beispiel(e)	Schaltfunktionen an der Maschine.

<b>DB21, ... DBB158 - DBB193</b>	<b>F-Funktion 1 bis 6 Erweiterte Adresse F-Funktion 1 bis 6</b>						
Flankenauswertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK						
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Hier werden bis zu 6 in einem NC-Satz programmierte F-Funktionen (ein Bahnvorschub und bis zu 5 achsspezifische Vorschübe für Positionierachsen) gleichzeitig bereitgestellt, sobald die F-Änderungssignale anstehen. Wertebereich der F-Funktion: Gleitpunkt (entsprechend MC5+Format) Wertebereich der erweiterten Adresse: 0 bis 18; ganzzahlig Die erweiterte Adresse der F-Funktion wird aus der Vorschubart (Bahnvorschub oder achsspezifischer Vorschub) und dem Achsnamen gebildet. Sie ist folgendermaßen codiert: <table border="1" style="margin: 10px auto; width: 80%;"> <tr> <td style="padding: 5px;">0:</td> <td style="padding: 5px;">Bahnvorschub; z.B.: F=1000</td> <td style="padding: 5px;">z.B.: F=1000</td> </tr> <tr> <td style="padding: 5px;">1 bis 18:</td> <td style="padding: 5px;">Maschinenachsnummer der Positionierachse bei achsspezifischem Vorschub</td> <td style="padding: 5px;">z.B.: FA[X1]=500</td> </tr> </table>	0:	Bahnvorschub; z.B.: F=1000	z.B.: F=1000	1 bis 18:	Maschinenachsnummer der Positionierachse bei achsspezifischem Vorschub	z.B.: FA[X1]=500
0:	Bahnvorschub; z.B.: F=1000	z.B.: F=1000					
1 bis 18:	Maschinenachsnummer der Positionierachse bei achsspezifischem Vorschub	z.B.: FA[X1]=500					
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<ul style="list-style-type: none"> <li>Nach Hochlauf der PLC.</li> <li>Vor Eintrag einer neuen Hilfsfunktion werden alle anderen gelöscht.</li> </ul>						
Anwendungs- beispiel(e)	Beeinflussen des programmierten F-Wortes durch die PLC z.B. durch Überschreiben der eingestellten Vorschubkorrektur.						
korrespondierend mit ...	MD22240 \$MC_AUXFU_F_SYNC_TYPE (Ausgabezeitpunkt der F-Funktionen)						



<b>DB21, ... DBB194 - DBB206</b>	<b>Dynamische M-Funktionen: M0 - M99</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: auftragsgesteuert von NCK	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die dynamischen M-Signalbits werden durch dekodierte M-Funktionen gesetzt.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die dynamischen M-Signalbits werden bei einer allgemeinen Hilfsfunktionsausgabe durch das PLC-Grundprogramm quittiert, nach dem der OB1 einmal komplett durchlaufen wurde. Bei einer schnellen Hilfsfunktionsausgabe werden sie, nachdem die PLC die Hilfsfunktionen erkannt hat, im gleichen OB40-Zyklus quittiert.	
Anwendungs- beispiel(e)	Spindel Rechts-/Linkslauf, Kühlmittel ein-/ausschalten.	

<b>DB21, ... DBX318.5</b>	<b>Assoziiertes M01/M00 aktiv</b>	
Flankenbewertung: nein	Signal(e) aktualisiert:	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit dem Bit wird angezeigt, dass bei entsprechender vorheriger Freischaltung/Aktivierung von assoziierten M00 und M01 (Hilfsfunktionen), eine M00 oder M01 Hilfsfunktion aktiv ist.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine assoziierten M00/M01 Hilfsfunktionen aktiv.	
korrespondierend mit ...	DB21, ... DBX30.5 (assoziiertes M01 aktivieren)	

### 18.5.3 Signale von Achse/Spindel (DB31, ...)

<b>DB31, ... DBD78</b>	<b>F-Hilfsfunktion für Positionierachse</b>
Flankenbewertung: nein	Signal(e) aktualisiert: auftragsgesteuert
	Die Werte der F-Hilfsfunktion für Positionierachsen werden hier abgelegt. Die jeweilige Achse bestimmt sich aus der erweiterten Adresse

<b>DB31, ... DBD86</b>	<b>M-Hilfsfunktion für Spindel</b>
Flankenbewertung: nein	Signal(e) aktualisiert: auftragsgesteuert
	Die Werte für die Hilfsfunktionen M3, M4, M5 werden in die zugehörige Nahtstelle für die adressierte Spindel gelegt.

<b>DB31, ... DBD88</b>	<b>S-Hilfsfunktion für Spindel</b>
Flankenbewertung: nein	Signal(e) aktualisiert: auftragsgesteuert
	Die Werte für die S-Hilfsfunktionen werden in die zugehörige Nahtstelle für die adressierte Spindel gelegt.

## 18.6 BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

### 18.6.1 Signale an BAG (DB11)

DB11 DBX0.0	Betriebsart AUTOMATIK
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart AUTOMATIK ist vom PLC-Programm angewählt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart AUTOMATIK ist nicht vom PLC-Programm angewählt.
Signal irrelevant bei ...	DB11 DBX0.4 (Betriebsart Wechselsperre) = 1
korrespondierend mit ...	DB11 DBX6.0 (aktive Betriebsart AUTOMATIK)

DB11 DBX0.1	Betriebsart MDA
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart MDA ist vom PLC-Programm angewählt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart MDA ist nicht vom PLC-Programm angewählt.
Signal irrelevant bei ...	DB11 DBX0.4 (Betriebsart Wechselsperre) = 1
korrespondierend mit ...	DB11 DBX6.1 (aktive Betriebsart MDA)

<b>DB11 DBX0.2</b>	<b>Betriebsart JOG</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart JOG ist vom PLC-Programm angewählt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart JOG ist nicht vom PLC-Programm angewählt.
Signal irrelevant bei ...	DB11 DBX0.4 (Betriebsart Wechselsperre) = 1
korrespondierend mit ...	DB11 DBX6.2 (aktive Betriebsart JOG)

<b>DB11 DBX0.4</b>	<b>Betriebsarten-Wechselsperre</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die momentan aktive Betriebsart (JOG, MDA oder Automatik) der BAG kann nicht gewechselt werden. Die innerhalb einer Betriebsart anwählbare Maschinenfunktionen können gewechselt werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Betriebsart in der BAG kann gewechselt werden.
	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>Betriebsarten-Anwahl</p> <pre> graph LR     subgraph "Betriebsarten-Anwahl"         A[Betriebsart AUTOMATIK]         B[Betriebsart MDA]         C[Betriebsart JOG]     end     A --- S     B --- S     C --- S     S --- SW[Betriebsarten-wechsel-Sperre]     SW --- NC[NC]     DBX0.4[DBX0.4 = 0] --- SW             </pre> </div>

<b>DB11 DBX0.5</b>	<b>BAG-Stop</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Für alle Kanäle der BAG wird ein NC-Stop ausgelöst. Der Kanalzustand aller aktiven Kanäle wechselt auf den Kanalzustand unterbrochen. Alle Kanäle im Kanalzustand "Reset" bleiben im Kanalzustand "Reset". Laufende Programme werden sofort (an frühest möglicher Stelle, auch im laufenden Satz) unterbrochen und der Programmzustand wechselt auf "Angehalten". Alle laufenden Achsen der BAG werden entlang ihrer Beschleunigungskennlinien ohne Konturverletzung auf Stillstand abgebremst. Das Programm kann mit NC-Start wieder fortgesetzt werden. Alle Spindeln der BAG werden nicht beeinflusst.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Kanalzustand und Programmablauf wird nicht beeinflusst.	
Sonderfälle, Fehler, .....	Alle Achsen der BAG, die nicht durch ein Programm oder einen Programmsatz angestoßen werden (z.B. Achsen laufen aufgrund Verfahrtasten auf der MSTT), bremsen mit BAG-Stop auf Stillstand ab.	

<b>DB11 DBX0.6</b>	<b>BAG-Stopp Achsen plus Spindeln</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Für alle Kanäle der BAG wird ein NC-Stop ausgelöst. Der Kanalzustand aller aktiven Kanäle wechselt auf den Kanalzustand "Unterbrochen". Alle Kanäle im Kanalzustand "Reset" bleiben im Kanalzustand "Reset". Laufende Programme werden sofort (an frühest möglicher Stelle, auch im laufenden Satz) unterbrochen und der Programmzustand wechselt auf "Angehalten". Alle laufenden Achsen und Spindeln der BAG werden entlang ihrer Beschleunigungskennlinien ohne Konturverletzung auf Stillstand abgebremst. Das Programm kann mit NC-Start wieder fortgesetzt werden.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Kanalzustand und Programmablauf wird nicht beeinflusst.	
Sonderfälle, Fehler, .....	Alle Achsen und Spindeln der BAG, die nicht durch ein Programm oder einen Programmsatz angestoßen werden (z.B. Achsen laufen aufgrund Verfahrtasten auf der MSTT, Spindeln sind PLC-gesteuert), bremsen mit "BAG-Stop plus Spindeln" auf Stillstand ab.	

<b>DB11 DBX0.7</b>	<b>BAG-Reset</b>	
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Für alle Kanäle der BAG wird ein Reset ausgelöst. Alle Kanäle befinden sich dann im Kanalzustand "Reset". Alle laufenden Programme befinden sich dann im Programmzustand "Abgebrochen". Alle laufende Achsen und Spindeln werden entlang ihrer Beschleunigungskennlinien ohne Konturverletzung auf Stillstand abgebremst. Die Grundstellungen werden eingestellt (z. B. G-Funktionen). Die Alarmer der BAG werden gelöscht, sofern sie nicht POWER ON-Alarmer sind.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Kanalzustand und Programmablauf wird nicht durch dieses Signal beeinflusst.	
korrespondierend mit ...	DB21, ... DBX7.7 (Kanal-Reset) DB11 DBX6.7 (alle Kanäle im Reset-Zustand)	
Sonderfälle, Fehler, .....	Ein Alarm, der das Nahtstellensignal: DB11 DBX6.3 (BAG betriebsbereit) wegnimmt, sorgt dafür, dass sich alle Kanäle der BAG nicht mehr im Reset-Zustand befinden. Um dann die Betriebsart umschalten zu können, muss ein BAG-Reset (DB11 DBX0.7) ausgelöst werden.	

<b>DB11 DBX1.0</b>	<b>Maschinenfunktion TEACH IN</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion TEACH IN wird innerhalb der Betriebsart JOG für die BAG aktiviert.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion TEACH IN wird nicht aktiviert.	
Signal irrelevant bei ...	wenn Betriebsart JOG nicht aktiv ist.	
weiterführende Literatur	Bedienungsanleitung HMI (entsprechend verwendeter Software)	

<b>DB11 DBX1.1</b>	<b>Maschinenfunktion REPOS</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion REPOS wird innerhalb der Betriebsart JOG für die BAG aktiviert.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion REPOS wird nicht aktiviert.	
Signal irrelevant bei ...	Betriebsart JOG nicht aktiv.	
Anwendungs- beispiel(e)	Bei einem Fehler bei der Teileprogrammbearbeitung (z.B. Werkzeugbruch) wird in JOG über Handfahren von der Fehlerstelle weggefahren, um das Werkzeug austauschen zu können. Über die Maschinenfunktion REPOS kann dann wieder über Handfahren genau auf die vorherige Position gefahren werden, um danach das Programm in der Betriebsart Automatik weiterlaufen lassen zu können.	
weiterführende Literatur	Bedienungsanleitung HMI (entsprechend verwendeter Software)	

<b>DB11 DBX1.2</b>	<b>Maschinenfunktion REF</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion REF wird innerhalb der Betriebsart JOG für die BAG aktiviert.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion REF wird nicht aktiviert.	
Signal irrelevant bei ...	wenn Betriebsart JOG nicht aktiv ist.	
weiterführende Literatur	siehe Kapitel "R1: Referenzieren (Seite 1319)"	

DB11 DBX1.6		Einzelsatz Typ B	
Flankenbewertung: nein		Signal(e) aktualisiert:	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bit gesetzt und DB11 DBX1.7 nicht gesetzt: BAG übergreifendes Verhalten <ul style="list-style-type: none"> <li>• Alle Kanäle sind gestoppt.</li> <li>• Alle Kanäle erhalten einen Start.</li> <li>• Der Kanal KS stoppt am Satzende</li> <li>• Die Kanäle KA erhalten eine STOPATEND. (Vergleichbar mit DB21, ... DBX7.2 (NC-Stop an Satzgrenze).)</li> <li>• Alle Kanäle sind (irgendwann) an einer Satzgrenze gestoppt.</li> </ul> (Wenn DB11 DBX1.6 und DB11 DBX1.7 zugleich gesetzt sind, kann nicht entschieden werden, welcher Einzelsatztyp gewünscht ist. Die Steuerung nimmt dann an: Kein BAG übergreifender Einzelsatz.)		
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Wenn Bit DB11 DBX1.6 nicht gesetzt ist und Bit DB11 DBX1.7 gesetzt ist, liegt Einzelsatz Typ A vor. (Wenn DB11 DBX1.6 und DB11 DBX1.7 nicht gesetzt sind, kann nicht entschieden werden, welcher Einzelsatztyp gewünscht ist. Die Steuerung nimmt dann an: Kein BAG übergreifender Einzelsatz.)		
korrespondierend mit ...	Einzelsatz Typ A		

DB11 DBX1.7		Einzelsatz Typ A	
Flankenbewertung: nein		Signal(e) aktualisiert:	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	DB11 DBX1.7 gesetzt und DB11 DBX1.6 nicht gesetzt: BAG übergreifendes Verhalten <ul style="list-style-type: none"> <li>• Alle Kanäle sind gestoppt.</li> <li>• Alle Kanäle erhalten einen Start (Start-Taste).</li> <li>• Der Kanal KS stoppt am Satzende (wg. Einzelsatz)</li> <li>• Die Kanäle KA erhalten einen STOP. (Vergleichbar mit der STOP-Taste).</li> <li>• Alle Kanäle sind gestoppt. (Bremsphase von allen KA's)</li> </ul> (Wenn DB11 DBX1.6 und DB11 DBX1.7 zugleich gesetzt sind, kann nicht entschieden werden, welcher Einzelsatztyp gewünscht ist. Die Steuerung nimmt dann an: Kein BAG übergreifender Einzelsatz.)		
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Wenn DB11 DBX1.7 nicht gesetzt und DB11 DBX1.6 gesetzt ist, liegt Einzelsatz Typ B vor. (Wenn DB11 DBX1.6 und DB11 DBX1.7 nicht gesetzt sind, kann nicht entschieden werden, welcher Einzelsatztyp gewünscht ist. Die Steuerung nimmt dann an: Kein BAG übergreifender Einzelsatz.)		
korrespondierend mit ...	Einzelsatz Typ B		



## 18.6.2 Signale von BAG (DB11)

<b>DB11</b> <b>DBX4.0</b>	<b>angewählte Betriebsart AUTOMATIK</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart AUTOMATIK ist von HMI angewählt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart AUTOMATIK ist von HMI nicht angewählt.

<b>DB11</b> <b>DBX4.1</b>	<b>angewählte Betriebsart MDA</b>
Flankenbewertung:	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart MDA ist von HMI angewählt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart MDA ist von HMI nicht angewählt.

<b>DB11</b> <b>DBX4.2</b> <b>Datenbaustein</b>	<b>angewählte Betriebsart JOG</b> <b>Signal(e) von BAG (HMI → PLC)</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart JOG ist von HMI angewählt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart JOG ist vom HMI nicht angewählt.

<b>DB11 DBX5.0</b>	<b>angewählte Maschinenfunktion TEACH IN</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion TEACH IN ist innerhalb der BAG vom HMI angewählt.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion TEACH IN ist vom HMI nicht angewählt.	
weiterführende Literatur	Bedienungsanleitung HMI (entsprechend verwendeter Software)	

<b>DB11 DBX5.1</b>	<b>angewählte Maschinenfunktion REPOS</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion REPOS ist innerhalb der BAG vom HMI angewählt.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion REPOS ist vom HMI nicht angewählt.	
Anwendungs- beispiel(e)	Bei einem Fehler bei der Teileprogrammbearbeitung (z.B. Werkzeugbruch) wird in JOG über Handfahren von der Fehlerstelle weggefahren, um das Werkzeug austauschen zu können. Über die Maschinenfunktion REPOS kann dann wieder über Handfahren genau auf die vorherige Position gefahren werden, um danach das Programm in der Betriebsart Automatik weiterlaufen lassen zu können.	
weiterführende Literatur	Bedienungsanleitung HMI (entsprechend verwendeter Software)	

<b>DB11 DBX5.2</b>	<b>angewählte Maschinenfunktion REF</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion REF ist innerhalb der BAG vom HMI angewählt.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion REF ist vom HMI nicht angewählt.	
weiterführende Literatur	siehe Kapitel "R1: Referenzieren (Seite 1319)"	

<b>DB11 DBX6.0 Datenbaustein</b>	<b>aktive Betriebsart AUTOMATIK Signal(e) von BAG (NCK → PLC)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart AUTOMATIK ist aktiv.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart AUTOMATIK ist nicht aktiv.	

<b>DB11 DBX6.1</b>	<b>aktive Betriebsart MDA</b>	
Flankenbewertung:	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart MDA ist aktiv.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart MDA ist nicht aktiv.	

<b>DB11 DBX6.2</b>	<b>aktive Betriebsart JOG</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Betriebsart JOG ist aktiv	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Betriebsart JOG ist nicht aktiv	

<b>DB11 DBX6.3</b>	<b>BAG betriebsbereit</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Nach Netz-Ein und Aufbau aller Spannungen wird dieses Signal gesetzt. Die Betriebsartengruppe ist nun betriebsbereit und es können in den einzelnen Kanälen Teileprogramme abgearbeitet bzw. Achsen verfahren werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Betriebsartengruppe ist nicht betriebsbereit. Mögliche Ursachen dafür sind: <ul style="list-style-type: none"> <li>• Vorliegen eines schweren Achs- oder Spindelalarms</li> <li>• Hardwarefehler</li> <li>• Falsch konfigurierte Betriebsartengruppe (Maschinendaten)</li> </ul> Wechselt BAG-Betriebsbereit auf Signalzustand "0" so werden: <ul style="list-style-type: none"> <li>• die Achs- und Spindeltriebe mit max. Bremsstrom auf Stillstand abgebremst.</li> <li>• die Signale von der PLC an den NCK in den inaktiven Zustand versetzt (Löschstellung).</li> </ul>
Sonderfälle, Fehler, .....	Ein Alarm, der das Nahtstellensignal: DB11 DBX6.3 (BAG betriebsbereit) wegnimmt, sorgt dafür, dass sich alle Kanäle der BAG nicht mehr im Reset-Zustand befinden. Um dann die Betriebsart umschalten zu können, muss ein BAG-Reset (DB11 DBX 0.7) erfolgen.

<b>DB11 DBX6.7</b>	<b>alle Kanäle im Reset-Zustand</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Alle Kanäle, die dieser BAG zugehörig sind, haben den "Kanalzustand Reset" (DB21, ... DBX7.7).
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Mindestens ein Kanal der BAG ist nicht im "Kanalzustand Reset" (DB21, ... DBX7.7).
korrespondierend mit ...	DB21, ... DBX7.7 (Kanalzustand Reset)

<b>DB11 DBX7.0</b>	<b>aktive Maschinenfunktion TEACH IN</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion TEACH IN ist innerhalb der BAG aktiv.

<b>DB11 DBX7.0</b>	<b>aktive Maschinenfunktion TEACH IN</b>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion TEACH IN ist nicht aktiv.
weiterführende Literatur	Bedienungsanleitung HMI (entsprechend verwendeter Software)

<b>DB11 DBX7.1</b>	<b>aktive Maschinenfunktion REPOS</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion REPOS ist innerhalb der BAG aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion REPOS ist nicht aktiv.
Anwendungs- beispiel(e)	Bei einem Fehler bei der Teileprogrammbearbeitung (z.B. Werkzeugbruch) wird in JOG über Handfahren von der Fehlerstelle weggefahren, um das Werkzeug austauschen zu können. Über die Maschinenfunktion REPOS kann dann wieder über Handfahren genau auf die vorherige Position gefahren werden, um danach das Programm in der Betriebsart Automatik weiterlaufen lassen zu können.
Weiterführende Literatur	Bedienungsanleitung HMI (entsprechend verwendeter Software)

<b>DB11 DBX7.2</b>	<b>aktive Maschinenfunktion REF</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion REF ist innerhalb der BAG aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion REF ist nicht aktiv.
Weiterführende Literatur	siehe Kapitel "R1: Referenzieren (Seite 1319)"

### 18.6.3 Signale an Kanal (DB21, ...)

<b>DB21, ... DBX0.4</b>	<b>Einzelsatz aktivieren</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	In den Betriebsarten AUTOMATIK und MDA muss für das im Kanal angewählte Teileprogramm durch den Bediener die Freigabe zur Abarbeitung jedes einzelnen Teileprogrammsatzes durch erneutes Auslösen von NC-Start erfolgen.
Signalzustand 0	Keine Auswirkung.
Sonderfälle, Fehler, .....	<ul style="list-style-type: none"> <li>Bei aktiver Werkzeugkorrektur werden gegebenenfalls Zwischensätze eingefügt, für die ebenfalls eine Freigabe über NC-Start erfolgen muss.</li> <li>Bei einer Serie von G33-Sätzen ist Einzelsatz nur dann wirksam, wenn "Probelaufvorschub" angewählt ist.</li> <li>Rechensätze werden bei Dekodier-Einzelsatz nicht im Einzelschritt bearbeitet.</li> </ul>
korrespondierend mit ...	DB21, ... DBX35.3 (Programmzustand unterbrochen)

<b>DB21, ... DBX0.5</b>	<b>M01 aktivieren</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die Aktivierung der Programmbeeinflussung "bedingter Halt" M01 ist angefordert.
Signalzustand 0	Die Aktivierung der Programmbeeinflussung "bedingter Halt" M01 ist nicht angefordert.
korrespondierend mit ...	DB21, ... DBX24.5 (M01 angewählt) DB21, ... DBX32.5 (M0/M01 aktiv)

<b>DB21, ... DBX1.6</b>	<b>PLC-Aktion beendet</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
	<p>Am Ende des Satzsuchlaufes werden abschließend Aktionssätze ausgeführt:                  DB21, ... DBX32.3 (Aktionssatz aktiv) == 1 UND                  DB21, ... DBX32.6 (letzter Aktionssatz aktiv) == 1</p> <p>Über Alarm "10208 Kanal &lt;Kanalnummer&gt; Zur Programmfortsetzung NC-Start geben" wird der Anwender darauf aufmerksam gemacht, dass er zur Fortsetzung des Teileprogramms ab dem Zielsatz erneut NC-Start auslösen muss.</p> <p>Sollen vor dem NC-Start durch das PLC-Anwenderprogramm noch Aktionen ausgeführt werden (z.B. Werkzeugwechsel) kann durch Parametrierung des Suchlauf-Modus:                  MD11450 \$MN_SEARCH_RUN_MODE = 1                  die Ausgabe des Alarms bis zum erneuten Setzen des vorliegenden Signals verzögert werden.</p>
Signalzustand 1	Die PLC-Aktion ist beendet.
Signalzustand 0	Die PLC-Aktion ist noch nicht beendet.
korrespondierend mit ...	DB21, ... DBX32.3 (Aktionssatz aktiv) DB21, ... DBX32.6 (letzter Aktionssatz aktiv) DB21, ... DBX33.4 (Satzsuchlauf aktiv)

<b>DB21, ... DBX1.7</b>	<b>Programmtest aktivieren</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die Aktivierung des Programmtest ist angefordert. Während des Programmtests erfolgen alle Verfahrbewegungen der Achsen (nicht Spindeln) unter "Achsensperre". <b>Achtung!</b> Aufgrund der Achsensperre wird die Belegung eines Werkzeug-Magazins beim Programmtest nicht verändert. Der Anwender/Maschinenhersteller muss über ein geeignetes PLC-Anwenderprogramm sicherstellen, dass die Konsistenz zwischen der NCK-internen Werkzeugverwaltung und die tatsächliche Belegung der Werkzeugmagazin erhalten bleibt. Siehe dazu das in der PLC-Toolbox enthaltene Programmbeispiel.
Signalzustand 0	Die Aktivierung des Programmtest ist nicht angefordert.
korrespondierend mit ...	DB21, ... DBX25.7 (Programmtest angewählt) DB21, ... DBX33.7 (Programmtest aktiv)

<b>DB21, ... DBB2</b>	<b>Satz ausblenden aktivieren</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die im Teileprogramm mit einem Schrägstrich (/) gekennzeichneten Ausblendsätze werden nicht abgearbeitet. Bei mehreren aufeinander folgenden Ausblendsätzen wird das Signal nur dann wirksam, wenn es vor der Dekodierung des ersten Ausblendsatzes dieser Folge von Ausblendsätzen ansteht.  Hinweis Das Signal sollte vor dem Start des Teileprogramms anstehen.
Signalzustand 0	Die im Teileprogramm mit einem Schrägstrich (/) gekennzeichneten Ausblendsätze werden abgearbeitet.
korrespondierend mit ...	DB21, ... DBX26.0 (Satz ausblenden angewählt) DB21, ... DBX35.2 (Programmmzustand angehalten)

<b>DB21, ... DBX6.1</b>	<b>Einleesperre</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Der Hauptlauf liest keine durch den Vorlauf aufbereiteten Teileprogrammsätze mehr ein.  Hinweis Das Signal ist nur in den Betriebsarten AUTOMATIK und MDA wirksam.
Signalzustand 0	Der Hauptlauf liest die durch den Vorlauf aufbereiteten Teileprogrammsätze ein.
korrespondierend mit ...	DB21, ... DBX35.0 (Programmmzustand läuft)

<b>DB21, ... DBX6.4</b>	<b>Programmebenenabbruch</b>	
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch	
Flankenwechsel 0 → 1	Mit jedem Flankenwechsel 0 → 1 wird die momentan bearbeitete Programmebene (Unterprogrammebene, ASUP-Ebene, Rettroutine) sofort abgebrochen. Das Teileprogramm wird auf der nächsthöheren Programmebene ab dem Aussprungpunkt weiterbearbeitet.	
Flankenwechsel 1 → 0	Keine Wirkung.	
Sonderfälle, Fehler, .....	Die Hauptprogrammebene kann nicht mit diesem NST sondern nur mit dem NST "Reset" abgebrochen werden.	

<b>DB21, ... DBX7.0</b>	<b>NC-Start-Sperre</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Durch NC-Start-Sperre wird der Start eines Teileprogramms durch das NC-Start Signal DB21, ... DBX7.1 (NC-Start) == 1 verhindert.	
Signalzustand 0	Die NC-Start-Sperre ist nicht aktiv.	
Sonderfälle, Fehler, .....	Der Start eines im Kanal angewählten Teileprogramms durch den Teileprogrammbehehl <i>START</i> in einem anderen Kanal (Programmkoordinierung), wird durch das Nahtstellensignal: DB21, ... DBX7.0 (NC-Start-Sperre) == 1 nicht verhindert.	
korrespondierend mit ...	DB21, ... DBX7.1 (NC-Start)	

<b>DB21, ... DBX7.1</b>	<b>NC-Start</b>	
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch	
Flankenwechsel 0 → 1	Betriebsart AUTOMATIK: Das angewählte NC-Programm wird gestartet bzw. fortgesetzt, bzw. die während der Programmunterbrechung nachgespeicherten Hilfsfunktionen werden ausgegeben. Werden beim Programmzustand "Programm unterbrochen" Daten von der PLC in die NC übergeben, so werden diese mit NC-Start sofort verrechnet. Betriebsart MDA: Die eingegebenen Satzinformationen bzw. Teileprogrammätze werden zur Ausführung freigegeben.	
Flankenwechsel 1 → 0	Keine Wirkung.	
korrespondierend mit ...	DB21, ... DBX7.0 (NC-Start-Sperre)	



<b>DB21, ... DBX7.2</b>	<b>NC-Stop an Satzgrenze</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das laufende NC-Programm wird nach Fertigbearbeiten des laufenden Teileprogrammsatzes angehalten. Sonst wie DB21, ... DBX7.3 (NC-Stop).
Signalzustand 0	Keine Wirkung.
korrespondierend mit ...	DB21, ... DBX7.3 (NC-Stop) DB21, ... DBX7.4 (NC-Stop Achsen plus Spindeln) DB21, ... DBX35.2 (Programmzustand angehalten) DB21, ... DBX35.6 (Kanalzustand unterbrochen)

<b>DB21, ... DBX7.3</b>	<b>NC-Stop</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	<p>Betriebsart AUTOMATIK bzw. MDA: Die Abarbeitung des im Kanal aktiven Teileprogramms wird angehalten. Die Achsen (nicht Spindeln) werden, unter Einhaltung der parametrisierten Beschleunigungen, bis zum Stillstand abgebremst.</p> <ul style="list-style-type: none"> <li>• Programmzustand: Angehalten</li> <li>• Kanalzustand: Unterbrochen</li> </ul> <p>Betriebsart JOG: Noch nicht vollständig verfahrene inkrementelle Wege (INC...) werden in der Betriebsart JOG mit dem nächsten NC-Start abgefahren.</p> <p>Hinweis Das Signal muss mindestens einen PLC-Zyklus (OB1) anstehen.</p>
Signalzustand 0	Keine Wirkung.
Signal irrelevant bei ...	<ul style="list-style-type: none"> <li>• Programmzustand: Abgebrochen</li> <li>• Kanalzustand: Reset</li> </ul>
Sonderfälle, Fehler, .....	<ul style="list-style-type: none"> <li>• Werden nach NC-Stop Daten an den NCK übergeben (z.B. Werkzeugkorrektur), werden diese mit dem nächsten NC-Start verrechnet.</li> </ul>
korrespondierend mit ...	DB21, ... DBX7.2 (NC-Stop an Satzgrenze) DB21, ... DBX7.4 (NC-Stop Achsen plus Spindeln) DB21, ... DBX35.2 (Programmzustand angehalten) DB21, ... DBX35.6 (Kanalzustand unterbrochen)

<b>DB21, ... DBX7.4</b>	<b>NC-Stop Achsen plus Spindeln</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
	Siehe DB21, ... DBX7.3 (NC-Stop). Zusätzlich werden auch die Spindeln des Kanals angehalten.

<b>DB21, ... DBX7.7</b>	<b>Reset</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Der Kanal wird zurückgesetzt. Die Löschstellungen werden eingestellt (z.B. G-Funktionen). Die Alarmer des Kanals werden gelöscht, sofern sie nicht POWER ON-Alarmer sind. Das Signal Reset muss von der PLC gegeben werden (z.B. durch die Verknüpfung mit der Reset-Taste auf der MSTT). Das Signal wird nur vom angewählten Kanal ausgewertet. Der Programmzustand wechselt auf "Abgebrochen", der Kanalzustand wechselt auf "Kanalzustand Reset".
Signalzustand 0	Keine Auswirkung.
korrespondierend mit ...	DB11, ... DBX0.7 (BAG-Reset) DB21, ... DBX35.7 (Kanalzustand Reset)

<b>DB21, ... DBX31.0 - DBX31.2</b>	<b>REPOS-Mode (A, B, C)</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
korrespondierend mit ...	DB21, ...DBX25.0-2 (REPOS-Mode) DB31, ... DBX10.0 (REPOS_DELAY)

<b>DB21, ... DBX31.4</b>	<b>REPOS-Mode-Änderung</b>
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Flankenwechsel 0 → 1	Der REPOS-Mode hat sich geändert.
Flankenwechsel 1 → 0	Der REPOS-Mode hat sich nicht geändert.
korrespondierend mit ...	DB21, ... DBX31.0-2 (REPOS-Mode) DB21, ... DBX319.0 (Quittierung der REPOS-Mode-Änderung)

### 18.6.4 Signale von Kanal (DB21, ...)

<b>DB21, ... DBX32.3</b>	<b>Aktionssatz aktiv</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Der Aktionssatz wird bearbeitet.	
Signalzustand 0	Kein Aktionssatz aktiv.	
weiterführende Literatur	Bedienungsanleitung HMI (entsprechend verwendeter Software)	

<b>DB21, ... DBX32.4</b>	<b>Anfahrssatz aktiv</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Der Anfahrssatz zur Fortsetzung des Programms bei "Satzsuchlauf mit Berechnung an Kontur" ist aktiv, da bei "Satzsuchlauf mit Berechnung an Satzendpunkt" kein eigener Anfahrssatz erzeugt wird. Die Achsen werden automatisch auf die aufgesammelte Suchlaufposition positioniert, wenn bei "Satzsuchlauf mit Berechnung an Kontur" das ASUP mit REPOSA verlassen wird.	
Signalzustand 0	Suchziel wurde bei "Satzsuchlauf mit Berechnung an Kontur" gefunden.	
Weiterführende Literatur	Programmierhandbuch Arbeitsvorbereitung	



<b>DB21, ... DBX32.6</b>	<b>letzter Aktionssatz aktiv</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Der letzte Aktionssatz wird bearbeitet. Dies bedeutet, dass alle Aktionssätze seitens der NC abgearbeitet sind und Aktionen seitens der PLC (ASUP, FC) oder des Bedieners wie z.B. Überspeichern, Betriebsartenwechsel nach JOG/REPOS möglich sind. So kann z.B. die PLC vor dem Start der Bewegung noch einen Werkzeugwechsel durchführen.
Signalzustand 0	Der letzte Aktionssatz ist nicht in Bearbeitung. Aktionssätze beinhalten die während "Satzsuchlauf mit Berechnung" aufgesammelten Aktionen, wie <ul style="list-style-type: none"> <li>• Hilfsfunktionsausgaben H, M00, M01, M..</li> <li>• Werkzeugprogrammierung T, D, DL</li> <li>• Spindelprogrammierung S-Wert, M3/M4/M5/M19, SPOS</li> <li>• Vorschubprogrammierung, F</li> </ul>
Weiterführende Literatur	siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)"

<b>DB21, ... DBX33.4</b>	<b>Satzsuchlauf aktiv</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die Funktion Satzsuchlauf ist aktiv. Sie wurde über die Bedienoberfläche angewählt und mit dem Nahtstellensignal: DB21, ... DBX7.1 (NC-Start) gestartet.
Signalzustand 0	Suchziel wurde gefunden.
Anwendungs- beispiel(e)	Mit der Funktion Satzsuchlauf ist es möglich, auf einen bestimmten Satz in einem Teileprogramm zu springen und erst ab diesem Satz die Bearbeitung des Teileprogramms zu starten.
weiterführende Literatur	siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 489)"

<b>DB21, ... DBX33.5</b>	<b>M02 / M30 aktiv</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	<ul style="list-style-type: none"> <li>• NC-Satz mit M02 bzw. M30 (bzw. M17 wenn ein Unterprogramm gestartet wurde) ist vollständig abgearbeitet; sind in diesem Satz auch Verfahrenbewegungen programmiert, wird das Signal erst mit Erreichen der Zielposition ausgegeben.</li> <li>• Das Programm wurde durch Reset abgebrochen, der Programmzustand wechselt auf abgebrochen.</li> <li>• Bei Anwahl der Betriebsart MDA bzw. der Maschinenfunktion REF bzw. PRESET</li> <li>• Nach DB10 DBX56.2 (Not-Halt quittieren)</li> </ul>
Signalzustand 0	<ul style="list-style-type: none"> <li>• Kein Programmende oder Programmabbruch</li> <li>• Zustand nach dem Einschalten der Steuerung</li> <li>• Start eines NC-Programmes</li> </ul>
Bild	<p>         ① Datentransfer in Arbeitsspeicher      ④ M-Änderungssignal (1 PLC-Zykluszeit)          ② Satz abgearbeitet                      ⑤ NST "M02/M30 aktiv"          ③ NC-Satz mit M02       </p>
Anwendungs- beispiel(e)	Die PLC kann mit diesem Signal das Ende der Programmbearbeitung erkennen und darauf reagieren.
Sonderfälle, Fehler, .....	<ul style="list-style-type: none"> <li>• Die Funktionen M02 und M30 sind gleichwertig.</li> <li>• Das Nahtstellensignal: DB21, ... DBX33.5 (M02/M30 aktiv) steht nach Programmende statisch an.</li> <li>• Nicht für automatische Folgefunktionen geeignet wie Werkstückzählung, Stangenvorschub u.a.m. Für diese Funktionen ist M02/M30 in einem eigenen Satz zu schreiben und das Wort M02/M30 oder das ausdecodierte M-Signal zu verwenden.</li> <li>• Im letzten Satz eines Programms dürfen keine Hilfsfunktionen geschrieben werden, die zu Einlesehalt führen und keine S-Werte, die über M02/M30 hinaus wirken sollen.</li> </ul>

<b>DB21, ... DBX33.6</b>	<b>Transformation aktiv</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Im NC-Teilprogramm ist der NC-Befehl <code>TRAORI</code> (Transformation einschalten) programmiert. Dieser Satz wurde von der NC abgearbeitet und die Transformation ist nun aktiviert.
Signalzustand 0	Keine Transformation ist aktiv.
weiterführende Literatur	Programmierhandbuch Arbeitsvorbereitung

<b>DB21, ... DBX33.7</b>	<b>Programmtest aktiv</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Die Programmbeeinflussung Programmtest ist aktiv. Es wird für alle Achsen (nicht Spindeln) intern Achsensperre gegeben. Bei der Abarbeitung eines Teilprogrammsatzes oder Teilprogramms bewegen sich deshalb die Maschinenachsen nicht. Die Achsbewegungen werden aber auf der Bedienoberfläche durch sich verändernde Achspositionswerte simuliert. Die Achspositionswerte für die Anzeige werden aus den rechnerischen Sollwerten generiert. Die Abarbeitung des Teilprogramms läuft sonst ganz normal ab.
Signalzustand 0	Die Programmbeeinflussung "Programmtest" ist nicht aktiv.
korrespondierend mit ...	DB21, ... DBX1.7 (Programmtest aktivieren) DB21, ... DBX25.7 (Programmtest angewählt)

<b>DB21, ... DBX35.0</b>	<b>Programmzustand läuft</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das Teilprogramm wurde mit dem Nahtstellensignal: DB21, ... DBX7.1 (NC-Start) gestartet und läuft. Das laufende Programm wurde mit dem Nahtstellensignal: DB21, ... DBX6.1 (Einlesesperre) angehalten.
Signalzustand 0	<ul style="list-style-type: none"> <li>• Programm angehalten durch <code>M00/M01</code> oder NC-Stop oder Betriebsartenwechsel.</li> <li>• Bei Einzelsatzbetrieb ist der Satz abgearbeitet.</li> <li>• Programmende erreicht (<code>M02/M30</code>).</li> <li>• Programmabbruch durch Reset.</li> <li>• Kein aktueller Satz im Speicher (z.B. bei MDA).</li> <li>• Aktueller Satz ist nicht abarbeitbar.</li> </ul>
Signal irrelevant bei ...	Das Teilprogramm wurde mit dem Nahtstellensignal: DB21, ... DBX7.1 (NC-Start) gestartet und läuft.

<b>DB21, ... DBX35.0</b>	<b>Programmzustand läuft</b>
Sonderfälle, Fehler, .....	Das Nahtstellensignal: DB21, ... DBX35.0 (Programmzustand läuft) wechselt nicht nach 0 wenn die Werkstückbearbeitung durch folgende Ereignisse angehalten wird: <ul style="list-style-type: none"> <li>• Ausgabe von Vorschubsperrung oder Spindelstop</li> <li>• DB21, ... DBX6.1 (Einlesesperre)</li> <li>• Vorschubkorrektur auf 0%</li> <li>• Ansprechen der Spindel- und Achsüberwachungen</li> <li>• Vorgabe von Lage-Sollwerten im NC-Programm für Achsen im "Nachführbetrieb", für Achsen ohne "Reglerfreigabe" oder für "Parkende Achsen"</li> </ul>
korrespondierend mit ...	DB21, ... DBX6.1 (Einlesesperre)

<b>DB21, ... DBX35.1</b>	<b>Programmzustand warten</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das laufende Programm ist in einem NC-Satz auf den Programmbefehl <code>WAIT_M</code> oder <code>WAIT_E</code> gestoßen. Die im <code>WAIT</code> -Befehl angegebene Wartebedingung für den Kanal oder die Kanäle ist noch nicht erfüllt.
Signalzustand 0	Programmzustand warten liegt nicht vor.
Weiterführende Literatur	Programmierhandbuch Grundlagen

<b>DB21, ... DBX35.2</b>	<b>Programmzustand angehalten</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das NC-Teilprogramm ist angehalten worden durch: <ul style="list-style-type: none"> <li>• DB21, ... DBX7.3 (NC-Stop)</li> <li>• DB21, ... DBX7.4 (NC-Stop Achsen plus Spindeln)</li> <li>• DB21, ... DBX7.2 (NC-Stop an Satzgrenze)</li> <li>• Programmierem <code>M00</code> bzw. <code>M01</code></li> </ul> <p>oder</p> <ul style="list-style-type: none"> <li>• Einzelsatzbetrieb</li> </ul>
Signalzustand 0	"Programmzustand angehalten" liegt nicht vor.
korrespondierend mit ...	DB21, ... DBX7.3 (NC-Stop) DB21, ... DBX7.4 (NC-Stop Achsen plus Spindeln) DB21, ... DBX7.2 (NC-Stop an Satzgrenze)



<b>DB21, ... DBX35.3</b>	<b>Programmzustand unterbrochen</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Beim Wechsel der Betriebsart von AUTOMATIK bzw. MDA (bei angehaltenem Programmzustand) nach JOG wechselt der Programmzustand auf "Unterbrochen". Das Programm kann nachher in AUTOMATIK oder MDA durch Betätigen von NC-Start ab der Unterbrechungsstelle weiter abgearbeitet werden.
Signalzustand 0	"Programmzustand unterbrochen" liegt nicht vor.
Sonderfälle, Fehler, .....	Das Nahtstellensignal: DB21, ... DBX35.3 (Programmzustand unterbrochen) zeigt an, dass das Teileprogramm durch erneuten Start weiter bearbeitet werden kann.

<b>DB21, ... DBX35.4</b>	<b>Programmzustand abgebrochen</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das Programm ist angewählt aber nicht gestartet oder das laufende Programm wurde mit Reset abgebrochen.
Signalzustand 0	Programmzustand abgebrochen liegt nicht vor.
korrespondierend mit ...	DB21, ... DBX7.7 (Reset)

<b>DB21, ... DBX35.5</b>	<b>Kanalzustand aktiv</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	In diesem Kanal: <ul style="list-style-type: none"> <li>• läuft momentan in der Betriebsart Automatik oder MDA eine Teileprogrammabarbeitung bzw.</li> <li>• wird in der Betriebsart JOG mindestens eine Achse verfahren.</li> </ul>
Signalzustand 0	DB21, ... DBX35.3 (Kanalzustand unterbrochen) oder DB21, ... DB35.7 (Kanalzustand Reset) liegt vor.

<b>DB21, ... DBX35.6</b>	<b>Kanalzustand unterbrochen</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das NC-Teilprogramm in AUTOMATIK oder MDA bzw. eine Verfahrbewegung in JOG kann unterbrochen werden durch: <ul style="list-style-type: none"> <li>• DB21, ... DBX7.3 (NC-Stop)</li> <li>• DB21, ... DBX7.4 (NC-Stop Achsen plus Spindeln)</li> <li>• DB21, ... DBX7.2 (NC-Stop an Satzgrenze)</li> <li>• Programmierem M00 bzw. M01</li> </ul> <p>oder</p> <ul style="list-style-type: none"> <li>• Einzelsatzbetrieb</li> </ul> <p>Nach NC-Start kann das Teilprogramm bzw. die unterbrochene Verfahrbewegung weiter abgearbeitet werden.</p>
Signalzustand 0	DB21, ... DBX35.5 (Kanalzustand aktiv) oder DB21, ... DB35.7 (Kanalzustand Reset) liegt vor.

<b>DB21, ... DBX35.7</b>	<b>Kanalzustand Reset</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das Signal wird auf 1 gesetzt, sobald sich der Kanal im Resetzustand befindet, also keine Bearbeitung aktiv ist.
Signalzustand 0	Das Signal wird auf 0 gesetzt, sobald eine Bearbeitung im Kanal stattfindet, z.B.: <ul style="list-style-type: none"> <li>• Abarbeitung eines Teilprogramms</li> <li>• Satzsuchlauf</li> <li>• TEACH IN aktiv</li> <li>• Überspeichern aktiv</li> </ul>

<b>DB21, ... DBX36.4</b>	<b>Interruptbehandlung aktiv</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Ein oder mehrere Kanäle der BAG befinden sich als Folge einer aktiven Interruptroutine <b>nicht</b> in der gewünschten Betriebsart. <b>Hinweis:</b> Das Signal wird nicht gesetzt, wenn eine Interruptbehandlung in einer Programmbetriebsart durchgeführt wird.
Signalzustand 0	Alle Kanäle haben die angeforderte Betriebsart.
korrespondierend mit ...	MD11600 \$MN_BAG_MASK

<b>DB21, ... DBX36.5</b>	<b>Kanal betriebsbereit</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Ein Kanal ist für eine Teileprogrammbearbeitung von Maschinenachsen, Geometrieachsen und Positionierachsen bereit. Diese sind entsprechend der Maschinenkonfiguration und dem aktuellen Programmzustand den betreffenden Kanälen bereits zugeordnet.
Signalzustand 0	Der betroffene Kanal ist für eine Teileprogrammbearbeitung von Maschinenachsen, Geometrieachsen und Positionierachsen nicht betriebsbereit.
korrespondierend mit ...	MD11600 \$MN_BAG_MASK

<b>DB21, ... DBX37.6</b>	<b>Einlesesperre wird ignoriert</b>
Flankenauswertung: nein	Signal(e) aktualisiert:
	Das Ignorieren der Einlesesperre (DB21, ... DBX6.1) wird über folgende Maschinendaten eingestellt: <ul style="list-style-type: none"> <li>• MD11602 \$MN_ASUP_START_MASK, Bit 2 = 1 (Starten auch erlaubt, wenn Einlesesperre aktiv ist)</li> <li>• MD20116 \$MC_IGNORE_INHIBIT_ASUP (Interruptprogramm trotz Einlesesperre abarbeiten)</li> <li>• MD20107 \$MC_PROG_EVENT_IGN_INHIBIT (Prog-Events ignorieren Einlesesperre)</li> </ul> Teileprogrammsätze bei denen die Einlesesperre ignoriert wird, werden als Einlesesperre-unwirksam bezeichnet.
Signalzustand 1	Einlesesperre ist aktiv (DB21, ... DBX6.1==1) UND Teileprogrammsatz ist Einlesesperre-unwirksamen.
Signalzustand 0	Einlesesperre ist nicht aktiv (DB21, ... DBX6.1 == 0) ODER (Einlesesperre ist aktiv (DB21, ... DBX6.1 == 1) UND Teileprogrammsatz ist Einlesesperre-wirksamen)
korrespondierend mit ...	DB21, ... DBX37.7 (Stopp am Satzende während Einzelsatz (SBL) wird ignoriert)

<b>DB21, ... DBX37.7</b>	<b>Stopp am Satzende während Einzelsatz (SBL) wird ignoriert</b>
Flankenauswertung:	Signal(e) aktualisiert:
	Das Ignorieren des Stops am Satzende während Einzelsatz (DB21, ... DBX0.4 == 1) wird über folgende Maschinendaten und Teileprogrammbefehle eingestellt: <ul style="list-style-type: none"> <li>• MD10702 \$MN_IGNORE_SINGLESBLOCK_MASK (Einzelsatzstopp verhindern)</li> <li>• MD20117 \$MC_IGNORE_SINGLESBLOCK_ASUP (Interruptprogramm trotz Einzelsatz komplett abarbeiten)</li> <li>• MD20106 \$MC_PROG_EVENT_IGN_SINGLESBLOCK (Prog-Events ignorieren Einzelsatz)</li> <li>• SBLOF (Einzelsatz unterdrücken), SBLON (Einzelsatzunterdrückung aufheben)</li> </ul> Teileprogrammsätze bei denen Stops am Satzende während Einzelsatz ignoriert wird, werden als Einzelsatz-unwirksam bezeichnet.
Signalzustand 1	Einzelsatz ist aktiv (DB21, ... DBX0.4 == 1) UND Teileprogrammsatz ist Einzelsatz-unwirksam.

<b>DB21, ... DBX37.7</b>	<b>Stopp am Satzende während Einzelsatz (SBL) wird ignoriert</b>
Signalzustand 0	Einzelsatz ist nicht aktiv (DB21, ... DBB0.4 == 0) ODER (Einzelsatz ist aktiv (DB21, ... DBB0.4 ==1) UND Teileprogrammsatz ist Einzelsatz-wirksam)
korrespondierend mit ...	Einlesesperre wird ignoriert. DB21, ... DBX37.6 (Einlesesperre wird ignoriert)

<b>DB21, ... DBB208 - DBB271</b>	<b>Aktive G-Funktion der Gruppe 1 bis 60</b>										
Flankenbewertung: nein					Signal(e) aktualisiert: zyklisch						
Signalzustand <> 0	Die im BCD-Format angezeigte G-Funktion oder ihr mnemotechnischer Bezeichner ist in der G-Gruppe ist aktiv.										
		<b>Wert</b>	<b>Bedeutung</b>								
		1	1. G-Funktion der G-Gruppe								
		2	2. G-Funktion der G-Gruppe								
	n	n. G-Funktion der G-Gruppe									
Signalzustand = 0	Keine G-Funktion oder mnemotechnischer Bezeichner der G-Gruppe aktiv.										
Anwendungs- beispiel(e)	<b>Bit</b>	7	6	5	4	3	2	1	0		
	<b>Wert</b>	128	64	32	16	8	4	2	1		
	Beispiel:										
<b>G90</b>	0	1	0	1	1	0	1	0			
Sonderfälle, Fehler, .....	G-Funktionen werden im Gegensatz zu Hilfsfunktionen nicht quittungsgesteuert an die PLC ausgegeben, d.h. das Teileprogramm wird nach der G-Funktionsausgabe sofort weiter abgearbeitet.										
weiterführende Literatur	Eine vollständige Auflistung der G-Gruppen und G-Funktionen findet sich in: <b>Literatur:</b> Programmieranleitung Grundlagen.										

<b>DB21, ... DBB317.1</b>	<b>Werkstück-Soll erreicht</b>										
Flankenbewertung: nein					Signal(e) aktualisiert: zyklisch						
Signalzustand = 1	Die Anzahl der gefertigten Werkstücke (Ist-Werkstückzahl) ist <b>gleich</b> der Anzahl der zu fertigenden Werkstücke (Soll-Werkstückzahl): \$AC_ACTUAL_PARTS == \$AC_REQUIRED_PARTS										
Signalzustand = 0	Die Anzahl der gefertigten Werkstücke (Ist-Werkstückzahl) ist <b>ungleich</b> der Anzahl der zu fertigenden Werkstücke (Soll-Werkstückzahl): \$AC_ACTUAL_PARTS <> \$AC_REQUIRED_PARTS										
korrespondierend mit ...	MD27880 \$MC_PART_COUNTER (Aktivierung der Werkstückzähler)										
weiterführende Literatur	Funktionshandbuch Grundlagen, Kapitel " K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten" > " Programmlaufzeit / Werkstückzähler" > "Werkstückzähler"										

<b>DB21, ... DBX318.0</b>	<b>ASUP ist angehalten</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Das Signal wird auf 1 gesetzt, wenn die Steuerung selbsttätig vor dem ASUP-Ende stoppt. Das NST DB21, ... DBX318.0 (ASUP ist angehalten) wird nur für den Fall "Interrupt in einer Programmbetriebsart und Kanalzustand gestoppt" versorgt.
Signalzustand 0	Das NST DB21, ... DBX318.0 (ASUP ist angehalten) wird mit Start und Reset auf 0 gesetzt.
	<p>Typischer Ablauf ASUP mit REPOSA:</p> <p>ASUP mit REPOSA Ereignis, das die Steuerung in den "angehalten" Zustand bringt, tritt auf. Beispiel: Stopp-Taste</p> <p>Bahnbewegung eines ASUPs mit REPOSA</p> <p>Steuerung ist selbsttätig angehalten.</p> <p>NST "ASUP angehalten" ist gesetzt.</p> <p>FC9 Baustein meldet "ASUP Done"</p> <p>Steuerung hat angehalten, danach wird vom FC das ASUP gestartet.</p> <p>Bahn des Teilprogramms</p>
ASUP mit REPOSA wird im Zustand Automatik angehalten ausgelöst	<p>Wird das Interruptprogramm mit REPOSA abgeschlossen, so ist folgender Ablauf typisch:</p> <ul style="list-style-type: none"> <li>• Mit Stop-Taste, Stop-All-Taste, oder durch Alarm wird das Teilprogramm angehalten.</li> <li>• Die Steuerung nimmt den Programmzustand "Angehalten" ein.</li> <li>• Der PLC löst über den Baustein FC9 ein ASUP aus.</li> <li>• Vor dem Wiederaufahren an die Kontur hält die Steuerung an und geht in den Programmzustand "Angehalten". NST DB21, ... DBX318.0 (ASUP ist angehalten) wird gesetzt.</li> <li>• Der Benutzer drückt Start. Das NST DB21, ... DBX318.0 (ASUP ist angehalten) wird zurückgesetzt, die Wiederaufahrbewegung wird begonnen.</li> <li>• Mit dem Ende der Wiederaufahrbewegung wird das FC9-Signal "ASUP Done" gesetzt und die Bahn des unterbrochenen Teilprogramms wird fortgesetzt.</li> </ul>

<b>DB21, ... DBX318.1</b>	<b>Satzsuchlauf via Programmtest ist aktiv (SERUPRO)</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Während der Bearbeitung der Teileprogrammätze im Rahmen des Satzsuchlaufes (interner Kanalzustand: "Programmtest"), bis zum Einwechseln des Zielsatzes im Hauptlauf (Programmzustand: "Angehalten").
Signalzustand 0	Mit dem Einwechseln des Zielsatzes in den Hauptlauf (interner Kanalzustand: "Programmtest" ist abgewählt; Stopp-Bedingung: "Suchziel gefunden" wird angezeigt).
Sonderfälle, Fehler, .....	Satzsuchlauf (SERUPRO) kann nur in der Betriebsart AUTOMATIK im Programmzustand "Abgebrochen" aktiviert werden.

<b>DB21, ... DBX319.0</b>	<b>Quittierung der REPOS-Mode-Änderung</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Flankenwechsel 0 → 1	Das von der NC erkannte Nahtstellensignal: DB21, ... DBX31.4 (REPOS-Mode-Änderung) wird mit dem vorliegenden Nahtstellensignal quittiert, wenn der angeforderte REPOS-Mode: DB21, ... DBX31.0-2 (REPOS-Mode) und das Verzögerungssignal: DB31, ... DBX10.0 (REPOSDELAY) in der NC übernommen wurden. Die Signalzustände beziehen sich auf den aktuellen Hauptlaufsatz
Flankenwechsel 1 → 0	SERUPRO-ASUP stoppt selbsttätig vor dem REPOS und DB21, ... DBX31.4 (REPOS-Mode-Änderung) wirkt nicht auf SERUPRO-Anfahren.
korrespondierend mit ...	DB21, ... DBX31.4 (REPOS-Mode-Änderung)

<b>DB21, ... DBX319.1 - DBX319.3</b>	<b>Aktiver REPOS-Mode (A, B, C)</b>																								
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch																								
Bedeutung	Über die Nahtstellensignale A, B, C wird der aktive REPOS-Mode angezeigt:																								
	<table border="1"> <thead> <tr> <th>Aktiver REPOS-Mode</th> <th>C</th> <th>B</th> <th>A</th> </tr> </thead> <tbody> <tr> <td>0 = kein REPOS-Mode aktiv</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1 = Wiederanfahren an Satzanfangspunkt <small>RMBBL</small></td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>2 = Wiederanfahren an Unterbrechungspunkt <small>RMIBL</small></td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>3 = Wiederanfahren an Satzende <small>RMEBL</small></td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>4 = Wiederanfahren an nächstliegenden Bahnpunkt <small>RMNBL</small></td> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Aktiver REPOS-Mode	C	B	A	0 = kein REPOS-Mode aktiv	0	0	0	1 = Wiederanfahren an Satzanfangspunkt <small>RMBBL</small>	0	0	1	2 = Wiederanfahren an Unterbrechungspunkt <small>RMIBL</small>	0	1	0	3 = Wiederanfahren an Satzende <small>RMEBL</small>	0	1	1	4 = Wiederanfahren an nächstliegenden Bahnpunkt <small>RMNBL</small>	1	0	0
Aktiver REPOS-Mode	C	B	A																						
0 = kein REPOS-Mode aktiv	0	0	0																						
1 = Wiederanfahren an Satzanfangspunkt <small>RMBBL</small>	0	0	1																						
2 = Wiederanfahren an Unterbrechungspunkt <small>RMIBL</small>	0	1	0																						
3 = Wiederanfahren an Satzende <small>RMEBL</small>	0	1	1																						
4 = Wiederanfahren an nächstliegenden Bahnpunkt <small>RMNBL</small>	1	0	0																						

<p>DB21, ... DBX319.1 - DBX319.3</p>	<p><b>Aktiver REPOS-Mode (A, B, C)</b></p>
	<p>Beispielhafter Ablauf von REPOS-Quittierungen im Teileprogramm und zeitlicher Signalverlauf des Quittierungsvorgangs von der NC:</p> <p>Legend:</p> <ul style="list-style-type: none"> <li>① Start des Teileprogramms</li> <li>② Stopp des Teileprogramms</li> <li>③ RMNBL vorwählen</li> <li>④ ASUP auslösen</li> <li>⑤ Befehl mit ASUP wird begonnen</li> <li>⑥ Wiederanfahrbewegung von REPOS ist beendet. Der Restsatz beginnt.</li> <li>⑦ Der Restsatz ist zu Ende</li> </ul>
<p>korrespondierend mit ...</p>	<p>DB21, ... DBX31.0-2 (REPOS-Mode) DB21, ... DBX31.4 (REPOS-Mode-Änderung) DB21, ... DBX319.0 (Quittierung der REPOS-Mode-Änderung) DB31, ... DBX70.2 (Quittierung der REPOS-Verzögerung)</p>
<p>weiterführende Literatur</p>	<p>siehe Kapitel "Satzsuchlauf Typ 5 SERUPRO (Seite 537)"</p>

<b>DB21, ... DBX319.5</b>	<b>Repos DEFERAL Chan</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1	Alle Achsen die von diesem Kanal aktuell kontrolliert werden, haben entweder keine REPOS-Verschiebung oder ihre REPOS-Verschiebungen sind ungültig.
Signalzustand 0	Sonstige.
korrespondierend mit ...	DB31, ... DBX70.0 (Repos Verschiebung)

<b>DB21, ... DBB376</b>	<b>ProgEventDisplay</b>														
Flankenauswertung: nein	Signal(e) aktualisiert: ereignisgesteuert <b>Hinweis:</b> Signaldauer minimal ein kompletter PLC-Zyklus														
	Alle Bits == 0: Es ist kein ereignisgesteuerter Programmaufruf aktiv Bedeutung der einzelnen Bits:														
	<table border="1"> <thead> <tr> <th>Bit</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Teileprogramm-Start aus Kanalzustand "Reset"</td> </tr> <tr> <td>1</td> <td>Teileprogramm-Ende</td> </tr> <tr> <td>2</td> <td>Bedientafel-Reset</td> </tr> <tr> <td>3</td> <td>Hochlauf</td> </tr> <tr> <td>4</td> <td>1.Start nach Suchlauf</td> </tr> <tr> <td>5 - 7</td> <td>reserviert, aktuell immer 0</td> </tr> </tbody> </table>	Bit	Bedeutung	0	Teileprogramm-Start aus Kanalzustand "Reset"	1	Teileprogramm-Ende	2	Bedientafel-Reset	3	Hochlauf	4	1.Start nach Suchlauf	5 - 7	reserviert, aktuell immer 0
Bit	Bedeutung														
0	Teileprogramm-Start aus Kanalzustand "Reset"														
1	Teileprogramm-Ende														
2	Bedientafel-Reset														
3	Hochlauf														
4	1.Start nach Suchlauf														
5 - 7	reserviert, aktuell immer 0														
Signalzustand 1	Das dem Bit zugeordnete Ereignis hat die Funktion "Ereignisgesteuerter Programmaufruf" aktiviert.														
Signalzustand 0	Das dem Bit zugeordnete Ereignis hat die Funktion "Ereignisgesteuerter Programmaufruf" nicht aktiviert oder das ereignisgesteuerte Anwenderprogramm ist abgelaufen oder wurde mit RESET abgebrochen.														

<b>DB21, ... DBX378.0</b>	<b>ASUP aktiv</b>
Flankenauswertung: nein	Signal(e) aktualisiert: ereignisgesteuert
Signalzustand 1	Ein ASUP ist aktiv. <b>Hinweis:</b> Durch DB21, ... DBX378.0 bekommt der Anwender auch außerhalb vom Baustein FC9 eine Rückmeldung über ein laufendes ASUP.
Signalzustand 0	Es ist kein ASUP aktiv.



<b>DB21, ... DBX378.1</b>	<b>Stilles ASUP aktiv</b>
Flankenauswertung: nein	Signal(e) aktualisiert: ereignisgesteuert
Signalzustand 1	Ein ASUP mit unterdrückter Anzeigenaktualisierung (siehe MD20191 \$MC_IGN_PROG_STATE_ASUP) ist aktiv.
Signalzustand 0	Es ist kein ASUP mit unterdrückter Anzeigenaktualisierung aktiv.

<b>DB21, ... DBX384.0</b>	<b>Programmverzweigung steuern</b>
Flankenauswertung: nein	Signal(e) aktualisiert:
Signalzustand 1	GOTOS im Teileprogramm löst einen Rücksprung an den Programmanfang aus. Das Programm wird daraufhin erneut abgearbeitet.
Signalzustand 0	GOTOS löst keinen Rücksprung aus. Die Programmbearbeitung wird mit dem nächsten Teileprogrammsatz nach GOTOS fortgesetzt.
korrespondierend mit ...	MD27860 \$MC_PROCESSTIMER_MODE (Aktivierung der Programmlaufzeit-Messung) MD27880 \$MC_PART_COUNTER (Aktivierung der Werkstück-Zähler)

### 18.6.5 Signale an Achse/Spindel (DB31, ...)

<b>DB31, ... DBX10.0</b>	<b>REPOSDELAY</b>
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die REPOS-Verschiebung der Achse wird erst mit ihrer nächsten Programmierung herausfahren.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Es liegt keine REPOS-Verschiebung vor.
Sonderfälle, Fehler, .....	Für Bahnachsen ist das Signal nicht relevant.
korrespondierend mit ...	DB21, ... DBX31.0 - DBX31.2 (REPOSPATHMODE0-2) DB31, ... DBX70.2 (REPOS Delay Quit) DB31, ... DBX72.0 (REPOSDELAY)

### 18.6.6 Signale von Achse/Spindel (DB31, ...)

<b>DB31, ... DBX70.0</b>	<b>REPOS Verschiebung</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Eine REPOS-Verschiebung muss für die entsprechende Achse herausgefahren werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Es muss keine REPOS-Verschiebung für die Achse herausgefahren werden.
korrespondierend mit ...	DB31, ... DBX70.1 (REPOS Verschiebung gültig)

<b>DB31, ... DBX70.1</b>	<b>REPOS Verschiebung gültig</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der Gültigkeitsbereich von der REPOS-Verschiebung wird mit dem Wert 1 angezeigt. Die REPOS-Verschiebung wurde gültig berechnet.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Mit dem Wert Null wird angezeigt, dass die REPOS-Verschiebung ungültig berechnet wurde.
Anwendungs- beispiel(e)	REPOS-Verschiebung im Gültigkeitsbereich aktualisieren: Zwischen SERUPRO-Ende und Start kann mit Modewechsel die Achse in JOG verfahren werden. Der Anwender fährt die REPOS-Verschiebung auf den Wert Null.
korrespondierend mit ...	DB31, ... DBX70.0 (REPOS Verschiebung)

<b>DB31, ... DBX70.2</b>	<b>REPOS Delay Quitt</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse wurde innerhalb eines Verfahrssatzes programmiert und die REPOS-Verschiebung dabei herausgefahren. Hinweis Für die Achse stand eine REPOS-Verschiebung an und REPOSDELAY war aktiv: DB31, ... DBX10.0 (REPOSDELAY) == 1 Dieses Signal verhält sich analog zu: DB21, ... DBX319.1 - DBX319.3 (Repos Path Mode Quitt)	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Mit dem Wert Null wird quittiert, dass die REPOS-Verschiebung für diese Achse nicht aktiv ist. Dieses Signal wird mit dem Einwechseln des Restsatzes zurückgenommen.	
korrespondierend mit ...	DB31, ... DBX10.0 (REPOSDELAY) DB31, ... DBX72.0 (REPOSDELAY)	

<b>DB31, ... DBX72.0</b>	<b>REPOSDELAY</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Nach Satzsuchlauf wird eine REPOS-Verschiebung für diese Achse nicht mit dem Anfahrssatz herausgefahren, sondern erst mit dem nächsten Verfahrssatz in dem die Achse programmiert ist.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die REPOS-Verschiebung für diese Achse ist nicht aktiv.	
Sonderfälle, Fehler, .....	Für Bahnachsen ist das Signal nicht relevant.	
korrespondierend mit ...	DB21, ... DBX31.0 - DBX31.2 (REPOSPATHMODE) DB31, ... DBX10.0 (REPOSDELAY) DB31, ... DBX70.2 (REPOS Delay Quitt)	

<b>DB31, ... DBX76.4</b>	<b>Bahnachse</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse ist an der Bahn beteiligt (Bahnachse). Hinweis Im Zusammenhang mit SERUPRO im Zustand: "Zielsatz gefunden", bezieht sich das Signal auf den Zustand der Achse im Zielsatz.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Achse ist nicht an der Bahn beteiligt.	

## 18.7 Achsen, Koordinatensysteme, Frames (K2)

### 18.7.1 Signale an Achse/Spindel (DB31, ...)

<b>DB 31, ... DBX3.0</b>	<b>Externe Nullpunktverschiebung</b>	
Flankenauswertung: nein	Signal(e) aktualisiert:	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der achsspezifisch hinterlegte, d.h. vorgewählte Wert der Externen Nullpunktverschiebung einer Achse wird für die Berechnung der Gesamt-Nullpunktverschiebung zwischen Basis- und Werkstück-Koordinatensystem neu übernommen.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der achsspezifisch hinterlegte, d.h. vorgewählte Wert der Externen Nullpunktverschiebung einer Achse wird für die Berechnung der Gesamt-Nullpunktverschiebung zwischen Basis- und Werkstück-Koordinatensystem nicht übernommen. Es wirkt weiterhin der bisherige Wert.	
Signal irrelevant bei ...	\$AA_ETRANS[Achse] gleich Null für alle Achsen.	
Sonderfälle, Fehler, .....	Signal Null nach Hochlauf (Power On).	
korrespondierend mit ...	\$AA_ETRANS[Achse]	

## 18.8 Not-Halt (N2)

### 18.8.1 Signale an NC (DB10)

<b>DB10 DBX56.1</b>	<b>Not-Halt</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die NC wird in den Not-Halt-Zustand versetzt und der Not-Halt-Ablauf in der NC wird gestartet.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die NC befindet sich nicht im Not-Halt-Zustand. Der Not-Halt-Zustand ist (noch) aktiv, kann aber mit den Nahtstellensignalen: DB10 DBX56.2 (Not-Halt quittieren) und DB11 DBX0.7 (BAG-Reset) zurückgesetzt werden.	
korrespondierend mit ...	DB10 DBX56.2 (Not-Halt quittieren) DB10 DBX106.1 (Not-Halt aktiv)	

<b>DB10 DBX56.2</b>	<b>Not-Halt quittieren</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Der Not-Halt-Zustand wird nur dann wieder rückgesetzt, wenn zuerst das Nahtstellensignal: DB10 DBX56.2 (Not-Halt quittieren) und anschließend das Nahtstellensignal: DB11, ... DBX0.7 (BAG-Reset) gesetzt wird.</p> <p>Dabei ist zu beachten, dass das NST "Not-Halt quittieren" und das NST "Reset" gemeinsam mindestens so lange gesetzt sein müssen, bis das Nahtstellensignal: DB10 DBX106.1 (Not-Halt aktiv) rückgesetzt wurde.</p> <p>Durch Rücksetzen des Not-Halt-Zustands wird:</p> <ul style="list-style-type: none"> <li>• die Reglerfreigabe zugeschaltet.</li> <li>• für alle Achsen der Nachführbetrieb aufgehoben und die Lageregelung aufgenommen.</li> <li>• DB31, ... DBX61.5 (Lageregler aktiv) gesetzt.</li> <li>• DB11, ... DBX6.3 (BAG betriebsbereit) gesetzt.</li> <li>• DB10 DBX106.1 (Not-Halt aktiv) rückgesetzt.</li> <li>• der Alarm 3000 gelöscht.</li> <li>• die Teileprogrammabarbeitung für alle Kanäle abgebrochen.</li> </ul>	

<b>DB10 DBX56.2</b>	<b>Not-Halt quittieren</b>
	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>NST "Not-Halt"</p> <p>NST "Not-Halt quittieren"</p> <p>NST "Not-Halt aktiv"</p> <p>NST "Reset"</p> </div> <div style="flex: 1;"> </div> </div> <p>         ① Das NST "Not-Halt quittieren" ist wirkungslos          ② Das NST "Reset" ist wirkungslos          ③ Die NST "Not-Halt quittieren" und "Reset" setzen "Not-Halt aktiv" zurück     </p>
Sonderfälle, Fehler, .....	Mit dem Nahtstellensignal: DB21, ... DBX7.7 (Reset) kann der Not-Halt-Zustand nicht rückgesetzt werden.
korrespondierend mit ...	DB10 DBX56.1 (Not-Halt) DB10 DBX106.1 (Not-Halt aktiv) DB11 DBX0.7 (BAG-Reset)

### 18.8.2 Signale von NC (DB10)

<b>DB10 DBX106.1</b>	<b>Not-Halt aktiv</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die NC befindet sich im Not-Halt-Zustand.	
korrespondierend mit ...	DB10 DBX56.1 (Not-Halt) DB10 DBX56.2 (Not-Halt quittieren)	

## **18.9 PLC-Grundprogramm (P3)**

Zur Beschreibung der NC/PLC-Nahtstellensignale siehe:

**Literatur:**

Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

Kapitel: "Signal-/Datenbeschreibungen"

## 18.10 Referenzpunktfahren (R1)

### 18.10.1 Signale an Kanal (DB21, ...)

<b>DB21, ... DBX1.0</b>	<b>Referieren aktivieren</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Das kanalspezifische Referieren wird mit dem Nahtstellensignal: DB21, ... DBX1.0 (Referieren aktivieren) gestartet.</p> <p>Die Steuerung quittiert den erfolgreichen Start mit dem Nahtstellensignal: DB21, ... DBX33.0 (Referieren aktiv)</p> <p>Mit dem kanalspezifischen Referieren kann jede Maschinenachse, die dem Kanal zugeordnet ist, referiert werden (steuerungsintern werden dazu die Verfahrtasten plus/minus simuliert).</p> <p>Mit dem achsspezifischen Maschinendatum: MD34110 \$MA_REFP_CYCLE_NR (Achsenreihenfolge beim kanalspez. Referieren) kann festgelegt werden, in welcher Reihenfolge die Maschinenachsen referiert werden.</p> <p>Haben alle in REFP_CYCLE_NR eingetragenen Achsen ihren Referenzpunkt erreicht, wird das Nahtstellensignal: DB21, ... DBX36.3 (alle Achsen stehen) gesetzt.</p>
Anwendungs- beispiel(e)	<p>Sollen die Maschinenachsen in einer bestimmten Reihenfolge referiert werden, gibt es folgende Möglichkeiten:</p> <ul style="list-style-type: none"> <li>• Der Bediener muss beim Starten die Reihenfolge selbst einhalten.</li> <li>• Die PLC muss die Reihenfolge beim Starten kontrollieren oder selbst festlegen.</li> <li>• Die Funktion kanalspezifisches Referieren wird verwendet.</li> </ul>
korrespondierend mit ...	<p>DB21, ... DBX33.0 (Referieren aktiv)</p> <p>DB21, ... DBX36.2 (alle referenzpunktspflichtigen Achsen sind referiert)</p>



<b>DB21, ... DBX33.0</b>	<b>Referieren aktiv</b>	
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Das kanalspezifische Referieren wurde mit dem Nahtstellensignal: DB21, ... DBX1.0 (Referieren aktivieren) gestartet und der erfolgreiche Start wurde mit dem Nahtstellensignal: DB21, ... DBX33.0 (Referieren aktiv) quittiert. Das kanalspezifische Referieren läuft.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<ul style="list-style-type: none"> <li>• Kanalspezifisches Referieren ist abgeschlossen</li> <li>• Achsspezifisches Referieren läuft</li> <li>• Kein Referieren aktiv</li> </ul>	
Signal irrelevant bei ...	Spindeln	
korrespondierend mit ...	DB21, ... DBX1.0 (Referieren aktivieren)	

## 18.10.2 Signale von Kanal (DB21, ...)

<b>DB21, ... DBX36.2</b>	<b>alle referenzpunktpflichtigen Achsen sind referenziert</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Alle referenzpunktpflichtigen Achsen (Linearachsen und Rundachsen) des Kanals sind referenziert. Das Maschinendatum: MD20700 \$MC_REFP_NC_START_LOCK (NC-Startsperre ohne Referenzpunkt) ist Null.</p> <p>Sind an einer Achse zwei Lagemesssysteme angeschlossen, die einen NC-Start verhindern würden, muss das Aktive referenziert sein, damit die Achse als referenziert gilt. Erst bei Vorhandensein dieses Signals wird ein NC-Start für die Teileprogrammbearbeitung angenommen.</p> <p>Referenzpunktpflichtig sind die Achsen, wenn: MD34110 \$MA_REFP_CYCLE_NR_ = -1 und die Achse nicht in Parkstellung ist (Lagemesssysteme inaktiv und Reglerfreigabe weggenommen).</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Eine oder mehrere referenzpunktpflichtige Achsen des Kanals sind nicht referenziert.	
Sonderfälle, Fehler, .....	Die Spindeln des Kanals haben auf dieses Nahtstellensignal keine Auswirkung.	
korrespondierend mit ...	DB31, ... DBX60.4 (Referenziert/Synchronisiert 1) DB31, ... DBX60.5 (Referenziert/Synchronisiert 2)	

### 18.10.3 Signale an Achse/Spindel (DB31, ...)

<b>DB31, ... DBX2.4 - DBX2.7</b>	<b>Referenzpunktwert 1 bis 4</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit Erreichen des Referenznockens wird der NCK mitgeteilt, welcher codierte Referenznocken angefahren ist. Das Nahtstellensignal: DB31, ... DBX2.4 - DBX2.7 (Referenzpunktwert 1 bis 4) muss so lange gesetzt bleiben, bis der Referenzpunkt erreicht ist, oder bis ein neuer codierter Referenznocken angefahren ist. Ist die Maschinenachse auf dem Referenzpunkt angekommen (Achse steht), wird der über NST "Referenzpunktwert 1 bis 4" vorgewählte Referenzpunktwert aus dem Maschinendatum: MD34100 \$MA_REFP_SET_POS (Referenzpunktwert) als neue Bezugsposition in die Steuerung übernommen.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Wirkung.
Signal irrelevant bei ...	Längenmesssystemen mit abstandscodierten Referenzmarken
Anwendungsbeispiel(e)	An einer Werkzeugmaschine mit großen Verfahrwegen können durch vier codierte Referenznocken, die auf den gesamten Verfahrweg der Achse verteilt sein können, vier unterschiedliche Referenzpunkte angefahren werden, und so die Zeit bis zum Erreichen eines gültigen Referenzpunktes verringert werden.
Sonderfälle, Fehler, .....	Ist die Maschinenachse auf dem Referenzpunkt angekommen und keines der vier NST "Referenzpunktwert 1 bis 4" gesetzt, gilt automatisch Referenzpunktwert 1.
korrespondierend mit ...	MD34100 \$MA_REFP_SET_POS (Referenzpunktwert)

<b>DB31, ... DBX12.7</b>	<b>Verzögerung Referenzpunktfahren</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenachse befindet sich auf dem Referenznocken.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse befindet sich vor dem Referenznocken. Durch einen entsprechend langen Referenznocken (bis zum Verfahrbereichsende) sollte ausgeschlossen werden, dass sich die Maschinenachse hinter dem Referenznocken befinden kann.
korrespondierend mit ...	DB31, ... DBX2.4 - DBX2.7 (Referenzpunktwert 1 bis 4)

### 18.10.4 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX60.4	Referenziert/Synchronisiert 1	
Flankenbewertung:	Signal(e) aktualisiert:	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p><b>Achsen:</b> Ist die Maschinenachse beim Referenzpunktfahren auf dem Referenzpunkt (inkrementelle Messsysteme) bzw. Zielpunkt (Längenmesssystem mit abstandscodierten Referenzmarken) angekommen, ist die Maschinenachse referenziert und das Nahtstellensignal: DB31, ... DBX60.4 (Referenziert/synchronisiert 1) (je nachdem, welches Lagemesssystem bei Referenzieren aktiv ist) wird gesetzt.</p> <p><b>Spindeln:</b> Eine Spindel ist nach Netz ein spätestens nach einer Spindelumdrehung (360 Grad) synchronisiert (Nullmarke überfahren oder Bero angesprochen).</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse/Spindel mit dem Lagemesssystem 1 ist nicht referenziert/synchronisiert.	
korrespondierend mit ...	DB31, ... DBX1.5 (Lagemesssystem 1)	

DB31, ... DBX60.5	Referenziert/Synchronisiert 2	
Flankenbewertung:	Signal(e) aktualisiert:	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p><b>Achsen:</b> Ist die Maschinenachse beim Referenzpunktfahren auf dem Referenzpunkt (inkrementelle Messsysteme) bzw. Zielpunkt (Längenmesssystem mit abstandscodierten Referenzmarken) angekommen, ist die Maschinenachse referenziert und das Nahtstellensignal: DB31, ... DBX60.5 (Referenziert/synchronisiert 2) (je nachdem, welches Lagemesssystem bei Referenzieren aktiv ist) wird gesetzt.</p> <p><b>Spindeln:</b> Eine Spindel ist nach Netz ein spätestens nach einer Spindelumdrehung (360 Grad) synchronisiert (Nullmarke überfahren oder Bero angesprochen).</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>Die Maschinenachse/Spindel mit dem Lagemesssystem 2 ist nicht referenziert/synchronisiert.</p> <p><b>Achsen:</b> Alarm 21610 wurde ausgelöst.</p> <p><b>Spindeln:</b> Gebergrenzfrequenz wurde überschritten.</p>	
korrespondierend mit ...	DB31, ... DBX1.6 (Lagemesssystem 2) MD34102 \$MA_REFP_SYNC_ENCS (Messsystemabgleich) = 0	

<b>DB31, ... DBX71.4</b>	<b>POS_RESTORED 1</b>
Flankenbewertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Wird MD34210 \$MA_ENC_REFP_STATE auf den Wert 3 gesetzt, so wird bei abstandscodierten inkrementellen Messsystemen die letzte vor dem Ausschalten gepufferte Achsposition restauriert. Es erfolgt kein automatisches Referenzieren. Das Lagemesssystem 1 befindet sich im Zustand "Position restauriert".
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse/Spindel mit dem Lagemesssystem 1 ist nicht restauriert.
korrespondierend mit ...	DB31, ... DBX60.4 (Referenziert/Synchronisiert 1)

<b>DB31, ... DBX71.5</b>	<b>POS_RESTORED 2</b>
Flankenbewertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Wird MD34210 \$MA_ENC_REFP_STATE auf den Wert 3 gesetzt, so wird bei abstandscodierten inkrementellen Messsystemen die letzte vor dem Ausschalten gepufferte Achsposition restauriert. Es erfolgt kein automatisches Referenzieren. Das Lagemesssystem 2 befindet sich im Zustand "Position restauriert".
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse/Spindel mit dem Lagemesssystem 2 ist nicht restauriert.
korrespondierend mit ...	DB31, ... DBX60.5 (Referenziert/Synchronisiert 2)

## 18.11 Spindeln (S1)

### 18.11.1 Signale an Achse/Spindel (DB31, ...)

DB31, ... DBX2.2	Spindel-Reset/Restweg löschen
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Flankenwechsel 0 → 1	<p>Unabhängig vom Maschinendatum: MD35040 \$MA_SPIND_ACTIVE_AFTER_RESET wählt Spindel-Reset für die verschiedenen Spindel-Betriebsarten in folgender Weise:</p> <ul style="list-style-type: none"> <li>• Steuerbetrieb: <ul style="list-style-type: none"> <li>– Spindel stoppt</li> <li>– Programm läuft weiter</li> <li>– Spindel läuft mit folgendem M- und S-Programmbefehlen weiter</li> </ul> </li> <li>• Pendelbetrieb: <ul style="list-style-type: none"> <li>– Pendeln wird abgebrochen</li> <li>– Achsen laufen weiter</li> <li>– Programm wird mit aktueller Getriebestufe fortgesetzt</li> <li>– mit nachfolgendem M-Wert und größeren S-Wert wird gegebenenfalls das NST: DB31, ... DBX83.1 (programmierte Drehzahl zu hoch) gesetzt.</li> </ul> </li> <li>• Positionierbetrieb: <ul style="list-style-type: none"> <li>– wird gestoppt</li> </ul> </li> <li>• Achsbetrieb: <ul style="list-style-type: none"> <li>– wird gestoppt</li> </ul> </li> </ul>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Wirkung.
korrespondierend mit ...	MD35040 \$MA_SPIND_ACTIVE_AFTER_RESET (eigener Spindel-Reset) DB21, ... DBX7.7 (Reset) DB31, ... DBX2.2 (Restweg löschen): ist ein anderer Name für das gleiche Signal

<b>DB31, ... DBX16.0 - DBX16.2</b>	<b>Istgetriebestufe A bis C</b>																					
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch																					
Signalzustand 1 (zustandsgesteuert)	<p>Ist die neue Getriebestufe eingelegt, werden vom PLC-Anwender die Nahtstellensignale: DB31, ... DBX16.2 - DBX16.0 (Istgetriebestufe A bis C) und DB31, ... DBX16.3 (Getriebe ist umgeschaltet) gesetzt.</p> <p>Damit wird der NCK mitgeteilt, dass die richtige Getriebestufe erfolgreich eingelegt wurde. Der Getriebestufenwechsel gilt als beendet (Spindelbetriebsart Pendelbetrieb ist abgewählt), die Spindel dreht in der neuen Getriebestufe auf die letzte programmierte Spindeldrehzahl hoch und der nächste Satz im Teileprogramm kann zur Ausführung kommen.</p> <p>Die Istgetriebestufe wird codiert angegeben. Für jede der 5 Getriebestufen gibt es einen Parametersatz, der folgendermaßen zugeordnet ist:</p> <table border="1" data-bbox="368 804 1426 1223"> <thead> <tr> <th>Parametersatz-Nr</th> <th>NC/PLC-Nahtstelle</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>–</td> <td>Daten für Achsbetrieb</td> </tr> <tr> <td>1</td> <td>000 001</td> <td>Daten für 1. Getriebestufe</td> </tr> <tr> <td>2</td> <td>010</td> <td>Daten für 2. Getriebestufe</td> </tr> <tr> <td>3</td> <td>011</td> <td>Daten für 3. Getriebestufe</td> </tr> <tr> <td>4</td> <td>100</td> <td>Daten für 4. Getriebestufe</td> </tr> <tr> <td>5</td> <td>101 110 111</td> <td>Daten für 5. Getriebestufe</td> </tr> </tbody> </table>	Parametersatz-Nr	NC/PLC-Nahtstelle	Bedeutung	0	–	Daten für Achsbetrieb	1	000 001	Daten für 1. Getriebestufe	2	010	Daten für 2. Getriebestufe	3	011	Daten für 3. Getriebestufe	4	100	Daten für 4. Getriebestufe	5	101 110 111	Daten für 5. Getriebestufe
Parametersatz-Nr	NC/PLC-Nahtstelle	Bedeutung																				
0	–	Daten für Achsbetrieb																				
1	000 001	Daten für 1. Getriebestufe																				
2	010	Daten für 2. Getriebestufe																				
3	011	Daten für 3. Getriebestufe																				
4	100	Daten für 4. Getriebestufe																				
5	101 110 111	Daten für 5. Getriebestufe																				
Sonderfälle, Fehler, .....	Wird vom PLC-Anwender eine andere Istgetriebestufe an die NCK rückgemeldet, als von der NCK als Sollgetriebestufe an die PLC gemeldet wurde, gilt der Getriebestufenwechsel trotzdem als erfolgreich abgeschlossen und die Istgetriebestufe A bis C wird aktiviert.																					
korrespondierend mit ...	DB31, ... DBX82.0 - DBX82.2 (Sollgetriebestufe A bis C) DB31, ... DBX82.3 (Getriebe umschalten) DB31, ... DBX16.3 (Getriebe ist umgeschaltet) DB31, ... DBX18.5 (Pendeldrehzahl) Parametersätze für Getriebestufen																					

<b>DB31, ... DBX16.3</b>	<b>Getriebe ist umgeschaltet</b>	
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Ist die neue Getriebestufe eingelegt, werden vom PLC-Anwender die Nahtstellensignale: DB31, ... DBX16.0 - DBX16.2 (Istgetriebestufe A bis C) und DB31, ... DBX16.3 (Getriebe ist umgeschaltet) gesetzt.</p> <p>Damit wird dem NCK mitgeteilt, dass die richtige Getriebestufe erfolgreich eingelegt wurde. Der Getriebestufenwechsel gilt als beendet (Spindelbetriebsart Pendelbetrieb wird abgewählt), die Spindel dreht in der neuen Getriebestufe auf die letzte programmierte Spindeldrehzahl hoch und der nächste Satz im Teileprogramm kann zur Ausführung kommen.</p> <p>Das Nahtstellensignal: DB31, ... DBX82.3 (Getriebe umschalten) wird durch den NCK rückgesetzt, worauf der PLC-Anwender das Nahtstellensignal: (Getriebe ist umgeschaltet) rücksetzt.</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Wirkung.	
Signal irrelevant bei ...	... anderen Spindelbetriebsarten als Pendelbetrieb	
Sonderfälle, Fehler, .....	Wird vom PLC-Anwender eine andere Istgetriebestufe an die NCK rückgemeldet, als von der NCK als Sollgetriebestufe an die PLC gemeldet wurde, gilt der Getriebestufenwechsel trotzdem als erfolgreich abgeschlossen und die Istgetriebestufe A bis C wird aktiviert.	
korrespondierend mit ...	DB31, ... DBX16.2 - DBX16.0 (Istgetriebestufe A bis C) DB31, ... DBX82.2 - DBX82.0 (Sollgetriebestufe A bis C) DB31, ... DBX82.3 (Getriebe umschalten) DB31, ... DBX18.5 (Pendeldrehzahl)	

<b>DB31, ... DBX16.4 - DBX16.5</b>	<b>Spindel neu synchronisieren 1 und 2</b>	
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch	
Flankenwechsel 0 → 1	Die Spindel soll neu synchronisiert werden, da die Synchronisation zwischen Lagemesssystem der Spindel und der 0-Grad-Position verloren gegangen ist.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Wirkung.	
Signal irrelevant bei ...	... anderen Spindelbetriebsarten als Steuerbetrieb.	
Anwendungs- beispiel(e)	Die Maschine hat eine Umschaltung zwischen vertikaler und horizontaler Spindel. Dabei werden zwei verschiedene Lagemessgeber (einer für die vertikale Spindel und einer für die horizontale Spindel), aber nur ein Istwerteingang an der Steuerung verwendet. Wird zwischen vertikaler und horizontaler Spindel umgeschaltet, muss neu synchronisiert werden. Diese Synchronisation wird mit dem NST "Spindel neu synchronisieren 1 oder 2" angestoßen.	
korrespondierend mit ...	DB31, ... DBX60.4 (Referenziert/Synchronisiert 1) DB31, ... DBX60.5 (Referenziert/Synchronisiert 2)	

<b>DB31, ... DBX16.7</b>	<b>S-Wert löschen</b>	
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch	
Flankenwechsel 0 → 1	<b>Steuerbetrieb:</b> <ul style="list-style-type: none"> <li>• Spindel stoppt</li> <li>• Programm läuft weiter</li> <li>• Spindel läuft mit folgendem S-Wert weiter, wenn M3 oder M4 aktiv waren</li> </ul> <b>Pendelbetrieb, Achsbetrieb, Positionierbetrieb:</b> <ul style="list-style-type: none"> <li>• Signal ist für die entsprechende Funktion unwirksam. Wird jedoch wieder in den Steuerbetrieb umgeschaltet, muss ein neuer S-Wert programmiert werden.</li> </ul>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Wirkung.	
Anwendungs- beispiel(e)	Beenden einer Verfahrbewegung aufgrund eines externen Signals (z.B. Messtaster).	



<b>DB31, ... DBX17.4 - DBX17.5</b>	<b>Spindel neu synchronisieren beim Positionieren 1 und 2</b>	
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Die Spindel soll beim Positionieren neu synchronisiert werden.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Wirkung.	
Signal irrelevant bei ...	... anderen Spindelbetriebsarten als Positionierbetrieb.	
Anwendungs- beispiel(e)	Die Spindel besitzt ein indirektes Messsystem und zwischen Motor und dem Spannmittel kann ein Schlupf auftreten. Beim Start des Positioniervorgangs wird, wenn das Signal=1 ist, die alte Referenz gelöscht und die Nullmarke neu gesucht, bevor die Endposition angefahren wird.	
korrespondierend mit ...	DB31, ... DBX60.4 (Referenziert/Synchronisiert 1) DB31, ... DBX60.5 (Referenziert/Synchronisiert 2)	

<b>DB31, ... DBX17.6</b>	<b>M3 / M4 invertieren</b>	
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Spindelmotordrehrichtung ändert sich bei folgenden Funktionen: <ul style="list-style-type: none"> <li>• M3</li> <li>• M4</li> <li>• M5</li> <li>• SPOS/M19/SPOSA aus der Bewegung; nicht wirksam bei SPOS/M19/SPOSA aus dem Stillstand.</li> </ul>	
Anwendungs- beispiel(e)	Die Maschine hat eine Umschaltung zwischen vertikaler und horizontaler Spindel. Dabei ist die mechanische Konstruktion so ausgeführt, dass bei der horizontalen Spindel ein Zahnrad mehr als bei der vertikalen Spindel im Eingriff ist. Dadurch muss bei vertikaler Spindel die Drehrichtung geändert werden, wenn sich die Spindel mit M3 immer nach rechts drehen soll.	

<b>DB31, ... DBX18.4</b>	<b>Pendeln durch die PLC</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Ist das Nahtstellensignal: DB31, ... DBX18.4 (Pendeln durch die PLC) nicht gesetzt, wird mit dem Nahtstellensignal: DB31, ... DBX18.5 (Pendeldrehzahl) ein automatisches Pendeln in der NCK durchgeführt. Die beiden Zeiten für die Drehrichtungen werden in den Maschinendaten: MD35440 \$MA_SPIND_OSCILL_TIME_CW (Pendelzeit für M3-Richtung) und MD35450 \$MA_SPIND_OSCILL_TIME_CCW (Pendelzeit für M4-Richtung) eingegeben.</p> <p>Ist das NST "Pendeln durch die PLC" gesetzt, wird mit dem NST "Pendeldrehzahl" nur in Verbindung mit den Nahtstellensignalen: DB31, ... DBX18.6 - DBX18.7 (Solldrehrichtung links und rechts) eine Drehzahl ausgegeben. Das Pendeln, also das ständige Wechseln der Drehrichtung, wird durch den PLC-Anwender mit den Nahtstellensignalen "Solldrehrichtung links und rechts" durchgeführt (Pendeln durch die PLC).</p>
Anwendungs- beispiel(e)	Kann die neue Getriebestufe trotz mehrmaligem Versuch beim Pendeln durch die NCK nicht eingelegt werden, kann auf Pendeln durch die PLC umgeschaltet werden. Dabei können die beiden Zeiten für die Drehrichtungen dann beliebig durch den PLC-Anwender verändert werden. Somit kann sichergestellt werden, dass auch bei ungünstigen Zahnradstellungen ein sicheres Umschalten der Getriebestufe möglich ist.
korrespondierend mit ...	<p>MD35440 \$MA_SPIND_OSCILL_TIME_CW (Pendelzeit für M3-Richtung) MD35450 \$MA_SPIND_OSCILL_TIME_CCW (Pendelzeit für M4-Richtung) DB31, ... DBX18.5 (Pendeldrehzahl) DB31, ... DBX18.7 (Solldrehrichtung links) DB31, ... DBX18.6 (Solldrehrichtung rechts)</p>

<b>DB31, ... DBX18.5</b>	<b>Pendelfreigabe</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Soll ein Getriebestufenwechsel durchgeführt werden (DB31, ... DBX82.3 (Getriebe umschalten) ist gesetzt), wechselt die Spindelbetriebsart in den Pendelbetrieb.</p> <p>Je nachdem, zu welchem Zeitpunkt das Nahtstellensignal: DB31, ... DBX18.5 (Pendeldrehzahl) gesetzt wird, bremst die Spindel mit unterschiedlichen Beschleunigungen auf Stillstand ab:</p> <p>Das NST "Pendeldrehzahl" ist gesetzt bevor das Nahtstellensignal: DB31, ... DBX82.3 (Getriebe umschalten) durch die NCK gesetzt wird.</p> <p>Die Spindel wird mit der Beschleunigung beim Pendeln: MD35410 \$MA_SPIND_OSCILL_ACCEL auf Stillstand abgebremst.</p> <p>Steht die Spindel, wird sofort mit dem Pendeln begonnen.</p> <p>Das NST "Pendeldrehzahl" wird gesetzt nachdem das NST "Getriebe umschalten" durch die NCK gesetzt wurde und nachdem die Spindel steht. Die Lageregelung wird abgeschaltet. Die Spindel wird mit der Beschleunigung im Drehzahlregelbetrieb abgebremst.</p> <p>Nachdem das NST "Pendeldrehzahl" gesetzt wurde, beginnt die Spindel mit der Pendelbeschleunigung (MD35410) zu pendeln.</p> <p>Ist das Nahtstellensignal: DB31, ... DBX18.4 (Pendeln durch die PLC) nicht gesetzt, wird mit dem NST "Pendeldrehzahl" ein automatisches Pendeln in der NCK durchgeführt.</p> <p>Die beiden Zeiten für die Drehrichtungen werden in den Maschinendaten: MD35440 \$MA_SPIND_OSCILL_TIME_CW (Pendelzeit für M3-Richtung) und MD35450 \$MA_SPIND_OSCILL_TIME_CCW (Pendelzeit für M4-Richtung) eingegeben.</p> <p>Ist das NST "Pendeln durch die PLC" gesetzt, wird mit dem NST "Pendeldrehzahl" nur in Verbindung mit den Nahtstellensignalen: DB31, ... DBX18.6 - DBX18.7 (Solldrehrichtung links und rechts) eine Drehzahl ausgegeben.</p> <p>Das Pendeln, also das ständige Wechseln der Drehrichtung, wird durch den PLC-Anwender mit den Nahtstellensignalen "Solldrehrichtung links und rechts" durchgeführt (Pendeln durch die PLC).</p>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Spindel pendelt nicht.
Signal irrelevant bei ...	... allen Spindelbetriebsarten außer dem Pendelbetrieb.
Anwendungs- beispiel(e)	Die Pendeldrehzahl wird verwendet, um das Einrücken einer neuen Getriebestufe zu erleichtern.
korrespondierend mit ...	DB31, ... DBX18.4 (Pendeln durch die PLC) DB31, ... DBX18.7 (Solldrehrichtung links) DB31, ... DBX18.6 (Solldrehrichtung rechts)

<b>DB31, ... DBX18.6 - DBX18.7</b>	<b>Pendeldrehrichtung links / Pendeldrehrichtung rechts</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Wird das Nahtstellensignal: DB31, ... DBX18.4 (Pendeln durch die PLC) gesetzt, kann mit den beiden Nahtstellensignalen: DB31, ... DBX18.6 - DBX18.7 (Solldrehrichtung links und rechts) die Drehrichtung für die Pendeldrehzahl vorgegeben werden. Dabei werden die Zeiten für die Pendelbewegung des Spindelmotors dadurch festgelegt, dass die Nahtstellensignale "Solldrehrichtung links und rechts" entsprechend lang gesetzt werden.
Signal irrelevant bei ...	... anderen Spindelbetriebsarten als Pendeln.
Anwendungs- beispiel(e)	Siehe DB31, ... DBX18.4 (Pendeln durch die PLC).
Sonderfälle, Fehler, .....	Sind beide Nahtstellensignale gleichzeitig gesetzt, wird keine Pendeldrehzahl ausgegeben. Ist kein Nahtstellensignal gesetzt, wird keine Pendeldrehzahl ausgegeben.
korrespondierend mit ...	DB31, ... DBX18.4 (Pendeln durch die PLC) DB31, ... DBX18.5 (Pendeldrehzahl)

### 18.11.2 Signale von Achse/Spindel (DB31, ...)

<b>DB31, ... DBX60.0</b>	<b>Spindel/keine Achse</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenachse wird in einer der folgenden Spindelbetriebsarten betrieben: <ul style="list-style-type: none"> <li>• Steuerbetrieb</li> <li>• Pendelbetrieb</li> <li>• Positionierbetrieb</li> <li>• Gewindebohren ohne Ausgleichsfutter</li> <li>• Synchronbetrieb</li> </ul> Die Nahtstellensignale an Achse (DB31, ... DBB12 - DBB15) und von Achse (DB31, ... DBB74 - DBB81) sind <b>ungültig</b> . Die Nahtstellensignale an Spindel (DB31, ... DBB16 - DBB19) und von Spindel (DB31, ... DBB82 - DBB91) sind <b>gültig</b> .

<b>DB31, ... DBX60.0</b>	<b>Spindel/keine Achse</b>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse wird als Achse betrieben. Die Nahtstellensignale an Achse (DB31, ... DBB12 - DBB15) und von Achse (DB31, ... DBB74 - DBB81) sind <b>gültig</b> . Die Nahtstellensignale an Spindel (DB31, ... DBB16 - DBB19) und von Spindel (DB31, ... DBB82 - DBB91) sind <b>ungültig</b> .
Anwendungs- beispiel(e)	Wird eine Maschinenachse wechselweise als Spindel oder Rundachse betrieben: <ul style="list-style-type: none"> <li>• Drehmaschine: Spindel / C-Achse</li> <li>• Fräsmaschine: Spindel / Rundachse für Gewindebohren ohne Ausgleichsfutter</li> </ul> kann aus dem Nahtstellensignal: DB31, ... DBX60.0 (Spindel/keine Achse) erkannt werden, welcher Betriebsmodus momentan aktiv ist.

<b>DB31, ... DBX82.0 - 82.2</b>	<b>Sollgetriebestufe A bis C</b>																								
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch																								
	Siehe DB31, ... DB82.3 (Getriebe umschalten)																								
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Sollgetriebestufe wird codiert ausgegeben: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>1. Getriebestufe</td> <td>0 0 0</td> <td>(C B A)</td> </tr> <tr> <td>1. Getriebestufe</td> <td>0 0 1</td> <td></td> </tr> <tr> <td>2. Getriebestufe</td> <td>0 1 0</td> <td></td> </tr> <tr> <td>3. Getriebestufe</td> <td>0 1 1</td> <td></td> </tr> <tr> <td>4. Getriebestufe</td> <td>1 0 0</td> <td></td> </tr> <tr> <td>5. Getriebestufe</td> <td>1 0 1</td> <td></td> </tr> <tr> <td>ungültiger Wert</td> <td>1 1 0</td> <td></td> </tr> <tr> <td>ungültiger Wert</td> <td>1 1 1</td> <td></td> </tr> </table>	1. Getriebestufe	0 0 0	(C B A)	1. Getriebestufe	0 0 1		2. Getriebestufe	0 1 0		3. Getriebestufe	0 1 1		4. Getriebestufe	1 0 0		5. Getriebestufe	1 0 1		ungültiger Wert	1 1 0		ungültiger Wert	1 1 1	
1. Getriebestufe	0 0 0	(C B A)																							
1. Getriebestufe	0 0 1																								
2. Getriebestufe	0 1 0																								
3. Getriebestufe	0 1 1																								
4. Getriebestufe	1 0 0																								
5. Getriebestufe	1 0 1																								
ungültiger Wert	1 1 0																								
ungültiger Wert	1 1 1																								
Signal irrelevant bei ...	... anderen Spindelbetriebsarten außer Pendelbetrieb.																								
korrespondierend mit ...	DB31, ... DBX82.3 (Getriebe umschalten) DB31, ... DBX16.0 - DBX16.2 (Istgetriebestufe A bis C) DB31, ... DBX16.3 (Getriebe ist umgeschaltet)																								

<b>DB31, ... DBX82.3</b>	<b>Getriebe umschalten</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
	<p>Vorgabe der Getriebestufe:</p> <ul style="list-style-type: none"> <li>Manuelle Vorgabe durch M-Funktion M41 - M45 entsprechend Getriebestufe 1- 5 Bei Sollgetriebestufe &lt;&gt; Istgetriebestufe =&gt; DB31, ... DBX82.3 (Getriebe umschalten) = 1 DB31, ... DBX82.0 - DBX82.2 (Sollgetriebestufe) = Sollgetriebestufe</li> <li>Automatischen Getriebestufenauswahl abhängig von der progr. Spindeldrehzahl durch M-Funktion M40 Bei vorgegebene Solldrehzahl erfordert Getriebestufenwechsel =&gt; DB31, ... DBX82.3 (Getriebe umschalten) = 1 DB31, ... DBX82.0 - DBX82.2 (Sollgetriebestufe) = Sollgetriebestufe</li> </ul>
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Neue Sollgetriebestufe wurde vorgegeben UND Sollgetriebestufe <> Istgetriebestufe
Sonderfälle, Fehler, .....	Das Signal wird nicht ausgegeben bei: Sollgetriebestufe == Istgetriebestufe
korrespondierend mit ...	DB31, ... DBX82.0 - DBX82.2 (Sollgetriebestufe A bis C) DB31, ... DBX16.0 - DBX16.2 (Istgetriebestufe A bis C) DB31, ... DBX16.3 (Getriebe ist umgeschaltet)

<b>DB31, ... DBX83.0</b>	<b>Drehzahlgrenze überschritten</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Istdrehzahl überschreitet die max. Spindeldrehzahl: MD35100 \$MA_SPIND_VELO_LIMIT um mehr als die Spindeldrehzahltoleranz: MD35150 \$MA_SPIND_DES_VELO_TOL
korrespondierend mit ...	MD35150 \$MA_SPIND_DES_VELO_TOL (Spindeldrehzahltoleranz) MD35100 \$MA_SPIND_VELO_LIMIT (max. Spindeldrehzahl)

<b>DB31, ... DBX83.1</b>	<b>Solldrehzahl begrenzt (programmierte Drehzahl zu hoch)</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Die wirksame Solldrehzahl überschreitet den aktuellen max. Grenzwert. Die Solldrehzahl wird diesen Grenzwert begrenzt.</p> <p>Grenzwerte:</p> <ul style="list-style-type: none"> <li>• MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT (Maximaldrehzahl der Getriebestufe)</li> <li>• MD35100 \$MA_SPIND_VELO_LIMIT (max. Spindeldrehzahl)</li> <li>• DB31, ... DBX3.6 (Spindeldrehzahlbegrenzung auf MD35160 \$MA_SPIND_EXTERN_VELO_LIMIT)</li> <li>• G26 (obere Spindeldrehzahlbegrenzung)</li> <li>• LIMS (Drehzahlbegrenzung für die Masterspindel bei aktivem G96/G961/G97)</li> <li>• VELOLIM: programmierte Spindeldrehzahlbegrenzung im Drehzahlsteuerbetrieb</li> <li>• Safety Integrated</li> </ul> <p>MD36931 \$MA_SAFE_VELO_LIMIT (Grenzwert für sichere Geschwindigkeit)</p>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die wirksame Solldrehzahl der Spindel liegt unterhalb des max. Grenzwertes.
Anwendungs- beispiel(e)	<p>Über das Nahtstellensignal kann im PLC-Anwenderprogramm erkannt werden, dass die Solldrehzahl der Spindel nicht erreicht wird. Mögliche Reaktionen:</p> <ul style="list-style-type: none"> <li>• Zustand als zulässig anerkennen und Bahnvorschub freigeben: DB21, ... DBX6.0 = 0 (Vorschubsperr)</li> <li>• Bahnvorschub bzw. gesamten Kanal sperren: DB21, ... DBX6.0 = 1 (Vorschubsperr)</li> </ul> <p>DB31, ... DBX83.5 (Spindel im Sollbereich) wird bearbeitet</p> <p>Safety Integrated</p> <p>Zusätzlich zum Grenzwert MD36931 \$MA_SAFE_VELO_LIMIT sind abhängig von der aktiven sicheren Geschwindigkeitsstufe SG1 ... SGn folgende Maschinendaten zu berücksichtigen:</p> <ul style="list-style-type: none"> <li>• MD36932 \$MA_SAFE_VELO_OVR_FACTOR</li> <li>• MD36933 \$MA_SAFE_DES_VELO_LIMIT</li> </ul> <p>Beispiel: Alle Standard-Grenzwerte sind größer 1500 U/min.</p> <ul style="list-style-type: none"> <li>• SG1 ist aktiv</li> <li>• MD36932 \$MA_SAFE_VELO_OVR_FACTOR[&lt;SG1&gt;] = 1111,1111 [U/min]</li> <li>• MD36933 \$MA_SAFE_DES_VELO_LIMIT[&lt;SG1&gt;] = 90%</li> </ul> <p>Programmierung: M3 S1500 Der Drehzahlsollwert wird auf 1000 U/min (MD36932 * MD36933) begrenzt. DB31, ... DBX83.1 = 1</p>
korrespondierend mit ...	<p>DB21, ... DBX6.0 (Vorschubsperr)</p> <p>DB31, ... DBX4.3 (Vorschub /Spindel-Halt)</p> <p>DB31, ... DBX83.5 (Spindel im Sollbereich)</p>

<b>DB31, ... DBX83.2</b>	<b>Solldrehzahl erhöht (programmierte Drehzahl zu niedrig)</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die wirksame Solldrehzahl unterschreitet den aktuellen min. Grenzwert. Die Solldrehzahl wird diesen Grenzwert begrenzt. Grenzwerte: <ul style="list-style-type: none"> <li>• MD35120 \$MA_GEAR_STEP_MIN_VELO (min. Drehzahl für automatische Getriebestufenauswahl M40)</li> <li>• MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT (Minimaldrehzahl der Getriebestufe)</li> <li>• G25 (untere Spindeldrehzahlbegrenzung)</li> </ul>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Solldrehzahl der Spindel liegt oberhalb des min. Grenzwertes.
Anwendungs- beispiel(e)	Über das Nahtstellensignal kann im PLC-Anwenderprogramm erkannt werden, dass die Solldrehzahl der Spindel nicht erreicht wird. Mögliche Reaktionen: <ul style="list-style-type: none"> <li>• Zustand als zulässig anerkennen und Bahnvorschub freigeben: DB21, ... DBX6.0 = 0 (Vorschubsperrung)</li> <li>• Bahnvorschub bzw. gesamten Kanal sperren: DB21, ... DBX6.0 = 1 (Vorschubsperrung)</li> </ul> DB31, ... DBX83.5 (Spindel im Sollbereich) wird bearbeitet  Aus dem Nahtstellensignal lässt sich erkennen, dass die programmierte Solldrehzahl nicht erreicht werden kann. Über das PLC-Anwenderprogramm kann der Vorschub dennoch freigegeben werden.  Der PLC-Anwender kann diesen Zustand als zulässig anerkennen und den Bahnvorschub freigeben, oder er kann den Bahnvorschub bzw. den gesamten Kanal sperren, NST:
korrespondierend mit ...	DB21, ... DBX6.0 (Vorschubsperrung) DB31, ... DBX4.3 (Vorschub /Spindel-Halt) DB31, ... DBX83.5 (Spindel im Sollbereich)

<b>DB31, ... DBX83.5</b>	<b>Spindel im Sollbereich</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Istdrehzahl der Spindel weicht um weniger als die Spindeldrehzahltoleranz: MD35150 \$MA_SPIND_DES_VELO_TOL von der Solldrehzahl ab.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Istdrehzahl der Spindel weicht um mehr als die Spindeldrehzahltoleranz: MD35150 \$MA_SPIND_DES_VELO_TOL von der Solldrehzahl ab. Normaler Zustand während der Beschleunigungs-/Bremsphase der Spindel.
Signal irrelevant bei ...	... alle Betriebsarten der Spindel außer Steuerbetrieb (Drehzahlbetrieb).



<b>DB31, ... DBX83.5</b>	<b>Spindel im Sollbereich</b>
Anwendungs- beispiel(e)	Vorschubfreigabe im Kanal erst mit dem Ende der Beschleunigungsphase der Spindel: IF ( DB31, ... DBX83.5 (Spindel im Sollbereich) == 1 ) THEN ( DB21, ... DBX6.0 (Vorschubsperr) = 0 ) ELSE ( DB21, ... DBX6.0 (Vorschubsperr) = 1 ) Hinweis: Mit der Vorschubsperr werden auch Positionierachsen angehalten.
korrespondierend mit ...	MD35500 \$MA_SPIND_ON_SPEED_AT_IPO_START MD35500 \$MA_SPIND_DES_VELO_TOL (Spindeldrehzahltoleranz)

<b>DB31, ... DBX83.7</b>	<b>Istdrehrichtung rechts</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
	Das Nahtstellensignal ist nur gültig, wenn sich die Spindel dreht: DB31, ... DBX61.4 (Achse/Spindel steht) == 0 Die Istdrehrichtung wird vom Lagemessgeber der Spindel abgeleitet.
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Istdrehrichtung: rechts
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Istdrehrichtung: links
Signal irrelevant bei ...	<ul style="list-style-type: none"> <li>• Spindel steht: DB31, ... DBX61.4 (Achse/Spindel steht) == 1</li> <li>• Spindeln ohne Lagemessgeber</li> </ul>
korrespondierend mit ...	DB31, ... DBX61.4 (Achse/Spindel steht)

<b>DB31, ... DBX84.3</b>	<b>Gewindebohren ohne Ausgleichsfutter aktiv</b>
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
	<p>Die Spindel wird durch die Funktion "Gewindebohren ohne Ausgleichsfutter" (G331/G332) intern in den Achsbetrieb umgeschaltet. Dadurch erfolgt Reaktion auf bzw. Aktualisierung der spindelspezifischen Nahtstellensignale:</p> <ul style="list-style-type: none"> <li>• DB31, ... DBX2.2 (Spindel-Reset)</li> <li>• DB31, ... DBX16.4 - DBX16.5 (Spindel synchronisieren)</li> <li>• DB31, ... DBX17.6 (M3/M4 invertieren)</li> <li>• DB31, ... DBX83.5 (Spindel im Sollbereich)</li> <li>• DB31, ... DBX83.1 (Programmierbare Drehzahl zu hoch)</li> </ul>
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<ul style="list-style-type: none"> <li>• Gewindebohren ohne Ausgleichsfutter ist aktiv.</li> </ul>
Anwendungs- beispiel(e)	<p>Achtung! Das Setzen folgender Signale während des Gewindebohrens ohne Ausgleichsfutter führt zur Zerstörung des Gewindes:</p> <ul style="list-style-type: none"> <li>• DB11, ... DBX0.7 (BAG-Reset) = 1</li> <li>• DB21, ... DBX7.7 (Kanal-Reset) = 1</li> <li>• DB31, ... DBX2.1 (Reglerfreigabe) = 0</li> <li>• DB31, ... DBX4.3 (Vorschub Halt) = 1</li> </ul>

<b>DB31, ... DBX84.5</b>	<b>aktive Spindelbetriebsart: Positionierbetrieb</b>
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Positionierbetrieb (SPOS bzw. SPOSA) ist aktiv.
korrespondierend mit ...	<p>DB31, ... DBX84.7 (Spindelbetriebsart Steuerbetrieb) DB31, ... DBX84.6 (Spindelbetriebsart Pendelbetrieb)</p>

<b>DB31, ... DBX84.6</b>	<b>aktive Spindelbetriebsart: Pendelbetrieb</b>
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Pendelbetrieb ist aktiv. Hinweis: Die Spindel wechselt bei einer Getriebeumschaltung automatisch in den Pendelbetrieb.</p>
korrespondierend mit ...	<p>DB31, ... DBX84.7 (Spindelbetriebsart Steuerbetrieb) DB31, ... DBX84.5 (Spindelbetriebsart Positionierbetrieb) DB31, ... DBX82.3 (Getriebe umschalten)</p>

<b>DB31, ... DBX84.7</b>	<b>aktive Spindelbetriebsart: Steuerbetrieb</b>
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bei folgenden Funktionen befindet sich die Spindel im Steuerbetrieb: <ul style="list-style-type: none"> <li>• Spindeldrehrichtungsvorgabe M3/M4 oder Spindelstopp M5</li> <li>• M41...M45, bzw. automatischer Getriebestufenwechsel M40</li> </ul>
korrespondierend mit ...	DB31, ... DBX84.6 (Spindelbetriebsart Pendelbetrieb) DB31, ... DBX84.5 (Spindelbetriebsart Positionierbetrieb)

<b>DB31, ... DBX85.5</b>	<b>Spindel in Position</b>
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
	Das Nahtstellensignal wird ausschließlich bei der Funktion Spindelpositionieren bearbeitet. Hierzu gehören: <ul style="list-style-type: none"> <li>• SPOS, SPOSA und M19 im Teileprogramm</li> <li>• SPOS und M19 in Synchronaktionen</li> <li>• Spindelpositionieren unter Verwendung des FC18</li> <li>• Spindelpositionieren über PLC-Schnittstelle (DB31, ... DBX30.4)</li> </ul>
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Voraussetzung für die Ausgabe des Signals DB31, ... DBX85.5 (Spindel in Position) ist das Erreichen von "Genauhalt fein": DB31, ... DBX60.7 (Genauhalt fein) = 1 Zusätzlich muss die zuletzt programmierte Spindelposition sollwertseitig erreicht worden sein. Steht die Spindel nach einer Positionierung bereits auf der programmierten Position, so bleibt das Signal DB31, ... DBX85.5 (Spindel in Position) gesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Mit der Wegnahme des Signals DB31, ... DBX60.7 (Genauhalt fein) wird immer auch das Signal DB31, ... DBX85.5 (Spindel in Position) zurückgesetzt.
Anwendungs- beispiel(e)	Spindel in Position für den Werkzeugwechsel Wird der Werkzeugwechselzyklus durch den Maschinenbediener unterbrochen (z. B. mit NC-Stop, NC-Stop Achsen plus Spindeln, BAG-Stop etc.), dann kann das Erreichen der Position, mit der die Spindel in den Werkzeugwechsler hineinfahren soll, über das NC/PLC-Nahtstellensignal DB31, ... DBX85.5 abgefragt werden.
korrespondierend mit ...	DB31, ... DBX60.7 (Genauhalt fein)

18.11 Spindeln (S1)

<b>DB31, ... DBB86</b>	<b>M-Funktion für Spindel</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Vom NCK wurde eine der folgenden M-Funktionen: M3, M4, M5, M19, M70 an die PLC ausgegeben. Die Ausgabe erfolgt über: siehe unten " korrespondierend mit ..."
korrespondierend mit ...	DB31, ... DBB86 - DBB87 (M-Funktion für Spindel), achsspezifisch DB21, ... DBB58, DBB68 - DBB97 (M-Funktion für Spindel), kanalspezifisch

<b>DB31, ... DBB88</b>	<b>S-Funktion für Spindel</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Vom NCK wurde eine S-Funktionen an die PLC ausgegeben. Die Ausgabe erfolgt über: siehe unten " korrespondierend mit ..." Folgende S-Funktionen werden hier ausgegeben: <ul style="list-style-type: none"> <li>• S.... als Spindeldrehzahl in 1/min (programmierter Wert)</li> <li>• S.... als konstante Schnittgeschwindigkeit in m/min bzw. ft/min</li> </ul> Folgende S-Funktionen werden hier <b>nicht</b> ausgegeben: <ul style="list-style-type: none"> <li>• S.... als progr. Spindeldrehzahlbegrenzung G25</li> <li>• S.... als progr. Spindeldrehzahlbegrenzung G26</li> <li>• S.... als Spindeldrehzahl in 1/min, wenn in der Steuerung keine Spindel definiert wurde</li> <li>• S.... als Verweilzeit in Spindelumdrehungen</li> </ul>
korrespondierend mit ...	DB31, ... DBB88 - DBB91 (S-Funktion für Spindel), achsspezifisch DB21, ... DBB60, DBB98 - DBB115 (S-Funktion für Spindel), kanalspezifisch

**Spindel mit SMI 24 (Weiss-Spindel)**

<b>DB31, ... DBX132.0</b>	<b>Sensorik vorhanden</b>
Flankenbewertung: nein	Signal(e) aktualisiert: Hochlauf
Signalzustand 1	Die für die Spindel mit SMI 24 benötigte Sensorik ist vorhanden.
Signalzustand 0	Die für die Spindel mit SMI 24 benötigte Sensorik ist nicht vorhanden.
korrespondierend mit ...	DB31, ... DBX132.1: Sensor S1 (Spannzustand) ist vorhanden DB31, ... DBX132.4: Sensor S4 ist vorhanden DB31, ... DBX132.5: Sensor S5 ist vorhanden

<b>DB31, ... DBX132.1</b>	<b>Sensor S1 vorhanden (Spannzustand)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: Hochlauf	
Signalzustand 1	Sensor S1 ist vorhanden.	
Signalzustand 0	Sensor S1 ist nicht vorhanden.	
korrespondierend mit ...	DB31, ... DBW134 (Zustand des Spannsystems) DB31, ... DBW136 (Analogwert: Spannzustand)	

<b>DB31, ... DBX132.4</b>	<b>Sensor S4 vorhanden (Kolbenendlage)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: Hochlauf	
Signalzustand 1	Sensor S4 ist vorhanden.	
Signalzustand 0	Sensor S4 ist nicht vorhanden.	
korrespondierend mit ...	DB31, ... DBX138.4 (Sensor S4: Kolbenendlage)	

<b>DB31, ... DBX132.5</b>	<b>Sensor S5 vorhanden (Winkellage der Motorwelle)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: Hochlauf	
Signalzustand 1	Sensor S5 ist vorhanden.	
Signalzustand 0	Sensor S5 ist nicht vorhanden.	
korrespondierend mit ...	DB31, ... DBX138.5 (Sensor S5: Winkellage der Motorwelle)	

<b>DB31, ... DBX133.2</b>	<b>Zustandswert wird gebildet, Drehzahlbegrenzung p5043 aktiv</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: Hochlauf	
Signalzustand 1	Der Zustandswert wird gebildet und die Drehzahlbegrenzungen aus Antriebsparameter p5043 sind aktiv.	
Signalzustand 0	Der Zustandswert wird nicht gebildet und die Drehzahlbegrenzungen aus Antriebsparameter p5043 sind nicht aktiv.	
Hinweis	Bei der Bildung des Zustandswertes werden die analogen Spannungswerte des Sensors S1 in die diskreten Zustandswerte des Antriebsparameters r5001 transformiert.	
korrespondierend mit ...	DB31, ... DBX134 (Spannzustand) Antriebsparameter: r5001 Systemvariable: \$VA_MOT_CLAMPING_STATE[<Achse>] BTSS-Variable: vaMotClampingState	

<b>DB31, ... DBW134</b>	<b>Zustand des Spannsystems (Sensor S1)</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
	<p>Der Sensor S1 liefert abhängig von der Lage der Spannvorrichtung einen analogen Spannungswert. Zur Vereinfachung der Auswertung des Spannzustandes wird der analoge Spannungswert von Sensormodul SMI 24 in einen Zustandwert gewandelt.</p> <p>Die Zustandswerte entsprechen bestimmten Spannungsbereichen. Die Spannungsbereiche können eingestellt werden über: Antriebsparameter p5041[0...5].</p>	
	<b>Zustandswert</b>	<b>Spannzustand</b>
	0	Sensor S1 nicht vorhanden oder Zustandswerte inaktiv
	1	Zustandsinitialisierung läuft
	2	Gelöst mit Meldung (Fehlerzustand)
	3	Gelöst
	4	Spannend mit Werkzeug
	5	Lösend mit Werkzeug
	6	Lösend ohne Werkzeug
	7	Gespannt mit Werkzeug UND S4 == 0
	8	Gespannt mit Werkzeug UND S4 == 1
	9	Spannend ohne Werkzeug
	10	Gespannt ohne Werkzeug
	11	Gespannt mit Meldung (Fehlerzustand)
korrespondierend mit ...	DB31, ... DBW136 (Analogwert: Spannzustand) Antriebsparameter: p5041[0...5], p5043[0...6]	

<b>DB31, ... DBW136</b>	<b>Analogmesswert: des Spannsystems</b>	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
	<p>Der Sensor S1 liefert einen analogen Spannungswert: 0 - 10 V. Der Analogwert des Spannzustandes wird abgebildet auf: 0 - 10000 Inkremente, Auflösung 1 mV</p> <p>Hinweis SIMATIC S7-Eingangsbaugruppe: 0 - 27648 Inkremente, Auflösung 0,36 mV Anpassfaktor bei Umstellung auf eine Spindel mit SMI 24: 2,7648</p>	
korrespondierend mit ...	DB31, ... DBW134 (Spannzustand) Antriebsparameter: p5041[0...5], p5043[0...6]	

<b>DB31, ... DBX138.4</b>	<b>Sensor S4 Kolbenendlage</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Der ist Kolben in Position, d. h. der Kolben ist frei	
Signalzustand 0	Der Kolben ist nicht in Position	
korrespondierend mit ...	DB31, ... DBX132.4 (Sensor S4 vorhanden)	

<b>DB31, ... DBX138.5</b>	<b>Sensor S5 Winkellage der Motorwelle</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1	Die Motorwelle steht auf Position (Voraussetzung: Stillstand der Spindel)	
Signalzustand 0	Die Motorwelle ist nicht gerichtet	
korrespondierend mit ...	DB31, ... DBX132.5 (Sensor S5 vorhanden)	

## 18.12 Vorschübe (V1)

### 18.12.1 Signale an Kanal (DB21, ...)

<b>DB21, ... DBX0.6</b>	<b>Probelaufvorschub aktivieren</b>
Flankenbewertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Es wird statt mit dem programmierten Vorschub (bei G01, G02, G03) mit dem über das Settingdatum: SD42100 \$SC_DRY_RUN_FEED vorgegebenen Probelaufvorschub gefahren, wenn der Probelaufvorschub größer als der programmierte ist.</p> <p>Der Probelaufvorschub ist wirksam nach Reset-Zustand.</p> <p>Das Nahtstellensignal wird bei NC-Start ausgewertet, wenn der Kanal sich im Zustand "Reset" befand.</p> <p>Der Probelaufvorschub kann über PLC oder Bedientafelfront angewählt werden.</p> <p>Bei Anwahl über die Bedientafelfront wird das PLC-Nahtstellensignal: DB21, ... DBX24.6 (Probelaufvorschub angewählt) gesetzt und vom PLC-Grundprogramm auf das Nahtstellensignal: DB21, ... DBX0.6 (Probelaufvorschub aktivieren) übertragen.</p> <p>Bei Anwahl über PLC ist das NST "Probelaufvorschub aktivieren" vom PLC-Anwenderprogramm zu setzen.</p>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>Es wird mit dem programmierten Vorschub gefahren.</p> <p>Wirksam nach Reset-Zustand.</p>
Anwendungs- beispiel(e)	Test eines Werkstückprogramms mit erhöhtem Vorschub.
Sonderfälle, Fehler, .....	Wechselt innerhalb eines G33-Satzes das Signal nach "0", so wird der programmierte Vorschub erst am Ende des Satzes wirksam, da kein NC-Stop wirkt.
korrespondierend mit ...	DB21, ... DBX24.6 (Probelaufvorschub angewählt) SD42100 \$SC_DRY_RUN_FEED (Probelaufvorschub)



<b>DB21, ... DBB4</b>	<b>Vorschubkorrektur</b>														
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch														
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Die Vorschubkorrektur kann über die PLC binär- oder graycodiert vorgegeben werden. Bei Binärcodierung wird der Vorschubwert in % interpretiert. 0% bis 200% Vorschubänderung sind möglich, entsprechend dem dualen Wert im Byte. Dabei gilt folgende feste Zuordnung:</p> <table border="1"> <thead> <tr> <th><u>Codes</u></th> <th><u>Vorschubkorrekturfaktor</u></th> </tr> </thead> <tbody> <tr> <td>00000000</td> <td>0.00 ± 0%</td> </tr> <tr> <td>00000001</td> <td>0.01 ± 1%</td> </tr> <tr> <td>00000010</td> <td>0.02 ± 2%</td> </tr> <tr> <td>00000011</td> <td>0.03 ± 3%</td> </tr> <tr> <td>.</td> <td>.</td> </tr> <tr> <td>11001000</td> <td>2.00 ± 200%</td> </tr> </tbody> </table> <p>Duale Werte &gt; 200 werden auf 200% begrenzt. Mit dem Maschinendatum: MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrektorschalter) kann die maximale Vorschubkorrektur zusätzlich begrenzt werden.</p>	<u>Codes</u>	<u>Vorschubkorrekturfaktor</u>	00000000	0.00 ± 0%	00000001	0.01 ± 1%	00000010	0.02 ± 2%	00000011	0.03 ± 3%	.	.	11001000	2.00 ± 200%
<u>Codes</u>	<u>Vorschubkorrekturfaktor</u>														
00000000	0.00 ± 0%														
00000001	0.01 ± 1%														
00000010	0.02 ± 2%														
00000011	0.03 ± 3%														
.	.														
11001000	2.00 ± 200%														

<b>DB21, ... DBB4</b>	<b>Vorschubkorrektur</b>																																																																																																
	<p>Bei Graycodierung sind den einzelnen Schalterstellungen folgende Codes zugeordnet:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p style="text-align: center;">Tabelle: Graycodierung für Vorschubkorrektur</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Schalter- stellung</th> <th style="text-align: center;">Code</th> <th style="text-align: center;">Vorschub- korrekturfaktor (Standardwerte)</th> </tr> </thead> <tbody> <tr><td style="text-align: center;">1</td><td style="text-align: center;">00001</td><td style="text-align: center;">0.0</td></tr> <tr><td style="text-align: center;">2</td><td style="text-align: center;">00011</td><td style="text-align: center;">0.01</td></tr> <tr><td style="text-align: center;">3</td><td style="text-align: center;">00010</td><td style="text-align: center;">0.02</td></tr> <tr><td style="text-align: center;">4</td><td style="text-align: center;">00110</td><td style="text-align: center;">0.04</td></tr> <tr><td style="text-align: center;">5</td><td style="text-align: center;">00111</td><td style="text-align: center;">0.06</td></tr> <tr><td style="text-align: center;">6</td><td style="text-align: center;">00101</td><td style="text-align: center;">0.08</td></tr> <tr><td style="text-align: center;">7</td><td style="text-align: center;">00100</td><td style="text-align: center;">0.10</td></tr> <tr><td style="text-align: center;">8</td><td style="text-align: center;">01100</td><td style="text-align: center;">0.20</td></tr> <tr><td style="text-align: center;">9</td><td style="text-align: center;">01101</td><td style="text-align: center;">0.30</td></tr> <tr><td style="text-align: center;">10</td><td style="text-align: center;">01111</td><td style="text-align: center;">0.40</td></tr> <tr><td style="text-align: center;">11</td><td style="text-align: center;">01110</td><td style="text-align: center;">0.50</td></tr> <tr><td style="text-align: center;">12</td><td style="text-align: center;">01010</td><td style="text-align: center;">0.60</td></tr> <tr><td style="text-align: center;">13</td><td style="text-align: center;">01011</td><td style="text-align: center;">0.70</td></tr> <tr><td style="text-align: center;">14</td><td style="text-align: center;">01001</td><td style="text-align: center;">0.75</td></tr> <tr><td style="text-align: center;">15</td><td style="text-align: center;">01000</td><td style="text-align: center;">0.80</td></tr> <tr><td style="text-align: center;">16</td><td style="text-align: center;">11000</td><td style="text-align: center;">0.85</td></tr> <tr><td style="text-align: center;">17</td><td style="text-align: center;">11001</td><td style="text-align: center;">0.90</td></tr> <tr><td style="text-align: center;">18</td><td style="text-align: center;">11011</td><td style="text-align: center;">0.95</td></tr> <tr><td style="text-align: center;">19</td><td style="text-align: center;">11010</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">20</td><td style="text-align: center;">11110</td><td style="text-align: center;">1.05</td></tr> <tr><td style="text-align: center;">21</td><td style="text-align: center;">11111</td><td style="text-align: center;">1.10</td></tr> <tr><td style="text-align: center;">22</td><td style="text-align: center;">11101</td><td style="text-align: center;">1.15</td></tr> <tr><td style="text-align: center;">23</td><td style="text-align: center;">11100</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">24</td><td style="text-align: center;">10100</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">25</td><td style="text-align: center;">10101</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">26</td><td style="text-align: center;">10111</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">27</td><td style="text-align: center;">10110</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">28</td><td style="text-align: center;">10010</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">29</td><td style="text-align: center;">10011</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">30</td><td style="text-align: center;">10001</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">31</td><td style="text-align: center;">10000</td><td style="text-align: center;">1.20</td></tr> </tbody> </table> </div> <p>Die in der Tabelle angegebenen Faktoren für die Vorschubkorrektur sind im Maschinendatum: MD12030 \$MN_OVR_FACTOR_FEEDRATE [n] abgelegt.</p> <p>In die Tabelle ist die Standardvorbelegung eingetragen.</p> <p>Die Anzahl der möglichen Schalterstellungen bei den Standardmaschinensteuertafeln sind in der Projektierungsanleitung für SINUMERIK 840D beschrieben.</p>	Schalter- stellung	Code	Vorschub- korrekturfaktor (Standardwerte)	1	00001	0.0	2	00011	0.01	3	00010	0.02	4	00110	0.04	5	00111	0.06	6	00101	0.08	7	00100	0.10	8	01100	0.20	9	01101	0.30	10	01111	0.40	11	01110	0.50	12	01010	0.60	13	01011	0.70	14	01001	0.75	15	01000	0.80	16	11000	0.85	17	11001	0.90	18	11011	0.95	19	11010	1.00	20	11110	1.05	21	11111	1.10	22	11101	1.15	23	11100	1.20	24	10100	1.20	25	10101	1.20	26	10111	1.20	27	10110	1.20	28	10010	1.20	29	10011	1.20	30	10001	1.20	31	10000	1.20
Schalter- stellung	Code	Vorschub- korrekturfaktor (Standardwerte)																																																																																															
1	00001	0.0																																																																																															
2	00011	0.01																																																																																															
3	00010	0.02																																																																																															
4	00110	0.04																																																																																															
5	00111	0.06																																																																																															
6	00101	0.08																																																																																															
7	00100	0.10																																																																																															
8	01100	0.20																																																																																															
9	01101	0.30																																																																																															
10	01111	0.40																																																																																															
11	01110	0.50																																																																																															
12	01010	0.60																																																																																															
13	01011	0.70																																																																																															
14	01001	0.75																																																																																															
15	01000	0.80																																																																																															
16	11000	0.85																																																																																															
17	11001	0.90																																																																																															
18	11011	0.95																																																																																															
19	11010	1.00																																																																																															
20	11110	1.05																																																																																															
21	11111	1.10																																																																																															
22	11101	1.15																																																																																															
23	11100	1.20																																																																																															
24	10100	1.20																																																																																															
25	10101	1.20																																																																																															
26	10111	1.20																																																																																															
27	10110	1.20																																																																																															
28	10010	1.20																																																																																															
29	10011	1.20																																																																																															
30	10001	1.20																																																																																															
31	10000	1.20																																																																																															
korrespondierend mit ...	DB21, ... DBX6.7 (Vorschubkorrektur wirksam) MD12030 \$MN_OVR_FACTOR_FEEDRATE [n] (Bewertung des Bahnvorschub-Korrekturschalters) MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrektorschalter)																																																																																																

<b>DB21, ... DBB5</b>	<b>Eilgangkorrektur</b>															
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch															
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Die Eilgangkorrektur kann über die PLC binär- oder graycodiert vorgegeben werden. Bei Binärcodierung wird die Eilgangkorrektur in % interpretiert. 0% bis 100% Vorschubänderung sind möglich, entsprechend dem dualen Wert im Byte. Dabei gilt folgende feste Zuordnung:</p> <table border="1" data-bbox="411 600 1463 898"> <thead> <tr> <th data-bbox="703 629 794 651">Codes</th> <th data-bbox="922 629 1134 651">Eilgangkorrekturfaktor</th> </tr> </thead> <tbody> <tr> <td data-bbox="703 685 799 707">00000000</td> <td data-bbox="986 685 1082 707">0.00 ± 0%</td> </tr> <tr> <td data-bbox="703 712 799 734">00000001</td> <td data-bbox="986 712 1082 734">0.01 ± 1%</td> </tr> <tr> <td data-bbox="703 739 799 761">00000010</td> <td data-bbox="986 739 1082 761">0.02 ± 2%</td> </tr> <tr> <td data-bbox="703 766 799 788">00000011</td> <td data-bbox="986 766 1082 788">0.03 ± 3%</td> </tr> <tr> <td data-bbox="703 792 719 815">.</td> <td data-bbox="986 792 1002 815">.</td> </tr> <tr> <td data-bbox="703 831 799 853">11001000</td> <td data-bbox="986 831 1102 853">1.00 ± 100%</td> </tr> </tbody> </table> <p>Duale Werte &gt; 100 werden auf 100% begrenzt. Mit dem Maschinendatum: MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrektorschalter) kann die maximale Eilgangkorrektur zusätzlich begrenzt werden.</p>		Codes	Eilgangkorrekturfaktor	00000000	0.00 ± 0%	00000001	0.01 ± 1%	00000010	0.02 ± 2%	00000011	0.03 ± 3%	.	.	11001000	1.00 ± 100%
Codes	Eilgangkorrekturfaktor															
00000000	0.00 ± 0%															
00000001	0.01 ± 1%															
00000010	0.02 ± 2%															
00000011	0.03 ± 3%															
.	.															
11001000	1.00 ± 100%															

<b>DB21, ... DBB5</b>	<b>Eilgangkorrektur</b>																																																																																																
	<p>Bei Graycodierung sind den einzelnen Schalterstellungen folgende Codes zugeordnet:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p style="text-align: center;">Tabelle: Graycodierung für Eilgangkorrektur</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Schalter- stellung</th> <th style="text-align: center;">Code</th> <th style="text-align: center;">Eilgang- korrekturfaktor (Standardwerte)</th> </tr> </thead> <tbody> <tr><td style="text-align: center;">1</td><td style="text-align: center;">00001</td><td style="text-align: center;">0.0</td></tr> <tr><td style="text-align: center;">2</td><td style="text-align: center;">00011</td><td style="text-align: center;">0.01</td></tr> <tr><td style="text-align: center;">3</td><td style="text-align: center;">00010</td><td style="text-align: center;">0.02</td></tr> <tr><td style="text-align: center;">4</td><td style="text-align: center;">00110</td><td style="text-align: center;">0.04</td></tr> <tr><td style="text-align: center;">5</td><td style="text-align: center;">00111</td><td style="text-align: center;">0.06</td></tr> <tr><td style="text-align: center;">6</td><td style="text-align: center;">00101</td><td style="text-align: center;">0.08</td></tr> <tr><td style="text-align: center;">7</td><td style="text-align: center;">00100</td><td style="text-align: center;">0.10</td></tr> <tr><td style="text-align: center;">8</td><td style="text-align: center;">01100</td><td style="text-align: center;">0.20</td></tr> <tr><td style="text-align: center;">9</td><td style="text-align: center;">01101</td><td style="text-align: center;">0.30</td></tr> <tr><td style="text-align: center;">10</td><td style="text-align: center;">01111</td><td style="text-align: center;">0.40</td></tr> <tr><td style="text-align: center;">11</td><td style="text-align: center;">01110</td><td style="text-align: center;">0.50</td></tr> <tr><td style="text-align: center;">12</td><td style="text-align: center;">01010</td><td style="text-align: center;">0.60</td></tr> <tr><td style="text-align: center;">13</td><td style="text-align: center;">01011</td><td style="text-align: center;">0.70</td></tr> <tr><td style="text-align: center;">14</td><td style="text-align: center;">01001</td><td style="text-align: center;">0.75</td></tr> <tr><td style="text-align: center;">15</td><td style="text-align: center;">01000</td><td style="text-align: center;">0.80</td></tr> <tr><td style="text-align: center;">16</td><td style="text-align: center;">11000</td><td style="text-align: center;">0.85</td></tr> <tr><td style="text-align: center;">17</td><td style="text-align: center;">11001</td><td style="text-align: center;">0.90</td></tr> <tr><td style="text-align: center;">18</td><td style="text-align: center;">11011</td><td style="text-align: center;">0.95</td></tr> <tr><td style="text-align: center;">19</td><td style="text-align: center;">11010</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">20</td><td style="text-align: center;">11110</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">21</td><td style="text-align: center;">11111</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">22</td><td style="text-align: center;">11101</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">23</td><td style="text-align: center;">11100</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">24</td><td style="text-align: center;">10100</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">25</td><td style="text-align: center;">10101</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">26</td><td style="text-align: center;">10111</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">27</td><td style="text-align: center;">10110</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">28</td><td style="text-align: center;">10010</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">29</td><td style="text-align: center;">10011</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">30</td><td style="text-align: center;">10001</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">31</td><td style="text-align: center;">10000</td><td style="text-align: center;">1.00</td></tr> </tbody> </table> </div> <p>Die in der Tabelle angegebenen Faktoren für die Eilgangkorrektur sind im Maschinendatum: MD12050 \$MN_OVR_FACTOR_RAPID_TRA[n] abgelegt.</p> <p>In der Tabelle ist die Standardvorbelegung eingetragen.</p> <p>Die Anzahl der möglichen Schalterstellungen bei den Standardmaschinensteuertafeln sind in der Projektierungsanleitung für SINUMERIK 840D beschrieben.</p>	Schalter- stellung	Code	Eilgang- korrekturfaktor (Standardwerte)	1	00001	0.0	2	00011	0.01	3	00010	0.02	4	00110	0.04	5	00111	0.06	6	00101	0.08	7	00100	0.10	8	01100	0.20	9	01101	0.30	10	01111	0.40	11	01110	0.50	12	01010	0.60	13	01011	0.70	14	01001	0.75	15	01000	0.80	16	11000	0.85	17	11001	0.90	18	11011	0.95	19	11010	1.00	20	11110	1.00	21	11111	1.00	22	11101	1.00	23	11100	1.00	24	10100	1.00	25	10101	1.00	26	10111	1.00	27	10110	1.00	28	10010	1.00	29	10011	1.00	30	10001	1.00	31	10000	1.00
Schalter- stellung	Code	Eilgang- korrekturfaktor (Standardwerte)																																																																																															
1	00001	0.0																																																																																															
2	00011	0.01																																																																																															
3	00010	0.02																																																																																															
4	00110	0.04																																																																																															
5	00111	0.06																																																																																															
6	00101	0.08																																																																																															
7	00100	0.10																																																																																															
8	01100	0.20																																																																																															
9	01101	0.30																																																																																															
10	01111	0.40																																																																																															
11	01110	0.50																																																																																															
12	01010	0.60																																																																																															
13	01011	0.70																																																																																															
14	01001	0.75																																																																																															
15	01000	0.80																																																																																															
16	11000	0.85																																																																																															
17	11001	0.90																																																																																															
18	11011	0.95																																																																																															
19	11010	1.00																																																																																															
20	11110	1.00																																																																																															
21	11111	1.00																																																																																															
22	11101	1.00																																																																																															
23	11100	1.00																																																																																															
24	10100	1.00																																																																																															
25	10101	1.00																																																																																															
26	10111	1.00																																																																																															
27	10110	1.00																																																																																															
28	10010	1.00																																																																																															
29	10011	1.00																																																																																															
30	10001	1.00																																																																																															
31	10000	1.00																																																																																															
korrespondierend mit ...	DB21, ... DBX6.6 (Eilgangkorrektur wirksam) MD12050 \$MN_OVR_FACTOR_RAPID_TRA[n] (Bewertung des Eilgang-Korrekturschalters) MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrekturschalter)																																																																																																

<b>DB21, ... DBX6.0</b>	<b>Vorschubsperr</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Das Signal ist in einem Kanal in allen Betriebsarten wirksam.</p> <p>Signal bewirkt Vorschubsperr aller im interpolatorischen Zusammenhang fahrenden Achsen (Geometrie- und Synchronachsen), sofern kein G33 (Gewinde) ansteht.</p> <p>Alle Achsen werden unter Einhaltung der Bahnkontur zum Stillstand gebracht. Nach Wegnahme der Vorschubsperr (0-Signal) wird das unterbrochene Teileprogramm wieder fortgesetzt.</p> <p>Signal bewirkt Vorschubsperr aller Positionierachsen. Bei einer fahrenden Achse bewirkt dieses Signal ein geführtes Bremsen zum Stillstand (Rampenstopp). Es erfolgt dabei keine Alarmmeldung.</p> <p>Die Lageregelung bleibt erhalten; d.h. der Schleppabstand wird abgebaut.</p> <p>Wird bei einer Achse, bei der "Vorschubsperr" ansteht, eine Fahranforderung gegeben, so bleibt diese erhalten. Diese anstehende Fahranforderung wird direkt mit der Wegnahme von "Vorschubsperr" ausgeführt.</p> <p>Steht die Achse im interpolatorischen Zusammenhang mit anderen, so gilt dies auch für diese Achsen.</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>Für alle Achsen des Kanals ist der Vorschub freigegeben.</p> <p>Steht für eine Achse oder einen Achsverbund bei Wegnahme von "Vorschubsperr" eine Fahranforderung ("Fahrbefehl") an, so wird diese direkt ausgeführt.</p>	
Anwendungs- beispiel(e)	Anhalten der Bearbeitung durch Vorgabe von VORSCHUB-AUS über die Maschinensteuertafel.	
Sonderfälle, Fehler, .....	Vorschubsperr ist bei aktivem G33 unwirksam.	

<b>DB21, ... DBX6.6</b>	<b>Eilgangkorrektur wirksam</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Die in die PLC-Nahtstelle eingetragene Eilgangkorrektur 0 bis maximal 100% ist kanalspezifisch wirksam. Der Korrekturfaktor wird vorgegeben über die Maschinendaten:</p> <p>MD12040 \$MN_OVR_RAPID_IS_GRAY_CODE (Eilgang-Korrekturschalter graycodiert) und</p> <p>MD12050 \$MN_OVR_FACTOR_RAPID_TRA [n] (Bewertung des Eilgang-Korrekturschalters).</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>Die in die PLC-Nahtstelle eingetragene Eilgangkorrektur wird nicht berücksichtigt.</p> <p>Bei unwirksamer Eilgangkorrektur wird als Korrekturfaktor NC intern 100% verwendet.</p> <p>Ausnahme bilden die Nullstellung bei binärer Schnittstelle und die 1.Schalterstellung für eine graycodierte Schnittstelle. Hier werden die Korrekturfaktoren, die in die PLC-Nahtstelle eingetragen sind, verwendet. Bei binärer Schnittstelle ist der Korrekturfaktor = 0. Bei graycodierter Schnittstelle wird der in den Maschinendaten für die 1.Schalterstellung eingetragene Wert als Korrekturwert ausgegeben.</p>	
Anwendungs- beispiel(e)	<p>Im Allgemeinen wird der Korrekturwert mit dem Eilgang-Korrekturschalter an der Maschinensteuertafel vorgegeben.</p> <p>Mit dem Nahtstellensignal: DB21, ... DBX6.6 (Eilgangkorrektur wirksam) kann der Eilgang-Korrekturschalter per PLC-Anwenderprogramm während der Inbetriebnahme eines neuen NC-Programms z.B. mit dem Schlüsselschalter freigegeben werden.</p>	

<b>DB21, ... DBX6.6</b>	<b>Eilgangkorrektur wirksam</b>
Sonderfälle, Fehler, .....	Die Eilgangkorrektur ist bei aktivem G33, G63, G331, G332 unwirksam.
korrespondierend mit ...	DB21, ... DBB5 (Eilgangkorrektur)

<b>DB21, ... DBX6.7</b>	<b>Vorschubkorrektur wirksam</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die in die PLC-Nahtstelle eingetragene Vorschubkorrektur 0 bis maximal 200% ist für Bahnvorschub und damit automatisch für die zugehörigen Achsen wirksam. In der Betriebsart JOG wirkt die Vorschubkorrektur direkt auf die Achsen. Der Korrekturfaktor wird vorgegeben über die Maschinendaten: MD12020 \$MN_OVR_FEED_IS_GRAY_CODE (Bahnvorschub-Korrekturschalter graycodiert) und MD12030 \$MN_OVR_FACTOR_FEEDRATE [n] (Bewertung des Bahnvorschub-Korrekturschalters)
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die in die PLC-Nahtstelle eingetragene Vorschubkorrektur wird nicht berücksichtigt. Bei unwirksamer Vorschubkorrektur wird als Korrekturfaktor intern 100% verwendet. Ausnahme bilden die Nullstellung bei binärer Schnittstelle und die 1.Schalterstellung für eine graycodierte Schnittstelle. Hier werden die Korrekturfaktoren, die in die PLC Nahtstelle eingetragen sind, verwendet. Bei binärer Schnittstelle ist der Korrekturfaktor = 0. Bei graycodierter Schnittstelle wird der in den Maschinendaten für die 1.Schalterstellung eingetragene Wert als Korrekturwert ausgegeben.
Anwendungs- beispiel(e)	Im Allgemeinen wird der Korrekturwert mit dem Vorschub-Korrekturschalter an der Maschinensteuertafel vorgegeben. Mit dem Nahtstellensignal: DB21, ... DBX6.7 (Vorschubkorrektur wirksam) kann der Vorschub-Korrekturschalter per PLC-Anwenderprogramm während der Inbetriebnahme eines neuen NC-Programms z.B. mit dem Schlüsselschalter freigegeben werden.
Sonderfälle, Fehler, .....	Die Vorschubkorrektur ist bei aktivem G33, G63, G331, G332 unwirksam.
korrespondierend mit ...	DB21, ... DBB4 (Vorschubkorrektur)

<b>DB21, ... DBX12.3, DBX16.3, DBX20.3</b>	<b>Vorschub Halt (Geometrieachse 1 bis 3)</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Das Signal ist nur im JOG-Betrieb wirksam.</p> <p>Signal bewirkt Vorschub Halt der jeweiligen Geometrieachse. Bei einer fahrenden Achse bewirkt dieses Signal ein geführtes Bremsen zum Stillstand (Rampenstopp). Es erfolgt dabei keine Alarmmeldung.</p> <p>Die Lageregelung bleibt erhalten; d.h. der Schleppabstand wird abgebaut.</p> <p>Wird bei einer Geometrieachse, bei der "Vorschub Halt" ansteht, eine Fahranforderung gegeben, so bleibt diese erhalten. Diese anstehende Fahranforderung wird direkt mit der Wegnahme von "Vorschub Halt" ausgeführt.</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>Für die Geometrieachse ist der Vorschub freigegeben.</p> <p>Steht für die Geometrieachse bei Wegnahme von "Vorschub Halt" eine Fahranforderung ("Fahrbefehl") an, so wird diese direkt ausgeführt.</p>	

<b>DB21, ... DBX24.6</b>	<b>Probelaufvorschub angewählt</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Probelaufvorschub ist angewählt.</p> <p>Anstelle des programmierten Vorschubs ist der in dem Settingdatum: SD42100 \$SC_DRY_RUN_FEED eingetragene Probelaufvorschub wirksam.</p> <p>Das Signal wird bei Aktivierung des Probelaufvorschubs über die Bedientafelfront automatisch in die PLC-Nahtstelle eingetragen und vom PLC-Grundprogramm auf das PLC-Nahtstellensignal: DB21, ... DBX0.6 (Probelaufvorschub aktivieren) übertragen.</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>Probelaufvorschub ist nicht angewählt.</p> <p>Der programmierte Vorschub ist wirksam.</p>	
korrespondierend mit ...	<p>DB21, ... DBX0.6 (Probelaufvorschub aktivieren)</p> <p>SD42100 \$SC_DRY_RUN_FEED (Probelaufvorschub)</p>	

<b>DB21, ... DBX25.3</b>	<b>Vorschubkorrektur für Eilgang angewählt</b>	
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Der Vorschub-Korrekturschalter soll auch als Eilgang-Korrekturschalter wirken. Korrekturen von über 100% werden auf den Maximalwert von 100%-Eilgangkorrektur begrenzt.</p> <p>Das Nahtstellensignal: DB21, ... DBX25.3 (Vorschubkorrektur für Eilgang angewählt) wird automatisch von der Bedientafelfront in die PLC-Nahtstelle eingetragen und vom PLC-Grundprogramm auf das PLC-Nahtstellensignal: DB21, ... DBX6.6 (Eilgangkorrektur wirksam) übertragen.</p> <p>Weiterhin wird das Nahtstellensignal: DB21, ... DBB4 (Vorschubkorrektur) vom PLC-Grundprogramm in das Nahtstellensignal: DB21, ... DBB5 (Eilgangkorrektur) kopiert.</p>	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der Vorschub-Korrekturschalter soll nicht als Eilgang-Korrekturschalter wirken.	
Anwendungs- beispiel(e)	Das Signal wird angewendet, wenn kein separater Eilgang-Korrekturschalter vorhanden ist.	

<b>DB 21, ... DBX29.0 - DBX29.3</b>	<b>Festvorschub 1 - 4 aktivieren für Bahn-/Geometrieachsen</b>																															
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch																															
Beschreibung	<p>Mit diesen Signalen wird die Funktion "Festvorschub" an-/abgewählt und festgelegt, welcher Festvorschub für Bahn-/Geometrieachsen wirksam werden soll.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Bit 3</th> <th>Bit 2</th> <th>Bit 1</th> <th>Bit 0</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Festvorschub ist abgewählt</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>Festvorschub 1 ist angewählt</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>Festvorschub 2 ist angewählt</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>Festvorschub 3 ist angewählt</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>Festvorschub 4 ist angewählt</td> </tr> </tbody> </table>		Bit 3	Bit 2	Bit 1	Bit 0	Bedeutung	0	0	0	0	Festvorschub ist abgewählt	0	0	0	1	Festvorschub 1 ist angewählt	0	0	1	0	Festvorschub 2 ist angewählt	0	1	0	0	Festvorschub 3 ist angewählt	1	0	0	0	Festvorschub 4 ist angewählt
Bit 3	Bit 2	Bit 1	Bit 0	Bedeutung																												
0	0	0	0	Festvorschub ist abgewählt																												
0	0	0	1	Festvorschub 1 ist angewählt																												
0	0	1	0	Festvorschub 2 ist angewählt																												
0	1	0	0	Festvorschub 3 ist angewählt																												
1	0	0	0	Festvorschub 4 ist angewählt																												
korrespondierend mit ...	MD12202 \$MN_PERMANENT_FEED[n] MD12200 \$MN_RUN_OVERRIDE_0																															



## 18.12.2 Signale an Achse/Spindel (DB31, ...)

<b>DB31, ... DBB0</b>	<b>Vorschubkorrektur (achsspezifisch)</b>																							
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch																							
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Die achsspezifische Vorschubkorrektur kann über die PLC binär- oder graycodiert vorgegeben werden. Bei Binärcodierung wird der Vorschubwert in % interpretiert. 0% bis 200% Vorschubänderung sind möglich, entsprechend dem dualen Wert im Byte.</p> <p>Dabei gilt folgende feste Zuordnung:</p> <table border="1" data-bbox="408 654 1455 1111"> <thead> <tr> <th data-bbox="699 698 794 734">Codes</th> <th data-bbox="916 680 1145 734">Achsspezifischer Vorschubkorrekturfaktor</th> </tr> </thead> <tbody> <tr> <td data-bbox="699 757 794 784">00000000</td> <td data-bbox="976 757 1072 784">0.00 ± 0%</td> </tr> <tr> <td data-bbox="699 788 794 815">00000001</td> <td data-bbox="976 788 1072 815">0.01 ± 1%</td> </tr> <tr> <td data-bbox="699 819 794 846">00000010</td> <td data-bbox="976 819 1072 846">0.02 ± 2%</td> </tr> <tr> <td data-bbox="699 851 794 878">00000011</td> <td data-bbox="976 851 1072 878">0.03 ± 3%</td> </tr> <tr> <td data-bbox="743 882 750 909">⋮</td> <td data-bbox="1021 882 1027 909">⋮</td> </tr> <tr> <td data-bbox="743 913 750 940">⋮</td> <td data-bbox="1021 913 1027 940">⋮</td> </tr> <tr> <td data-bbox="699 945 794 972">01100100</td> <td data-bbox="976 945 1098 972">1.00 ± 100%</td> </tr> <tr> <td data-bbox="743 976 750 1003">⋮</td> <td data-bbox="1021 976 1027 1003">⋮</td> </tr> <tr> <td data-bbox="743 1008 750 1034">⋮</td> <td data-bbox="1021 1008 1027 1034">⋮</td> </tr> <tr> <td data-bbox="699 1039 794 1066">11001000</td> <td data-bbox="976 1039 1098 1066">2.00 ± 200%</td> </tr> </tbody> </table>		Codes	Achsspezifischer Vorschubkorrekturfaktor	00000000	0.00 ± 0%	00000001	0.01 ± 1%	00000010	0.02 ± 2%	00000011	0.03 ± 3%	⋮	⋮	⋮	⋮	01100100	1.00 ± 100%	⋮	⋮	⋮	⋮	11001000	2.00 ± 200%
Codes	Achsspezifischer Vorschubkorrekturfaktor																							
00000000	0.00 ± 0%																							
00000001	0.01 ± 1%																							
00000010	0.02 ± 2%																							
00000011	0.03 ± 3%																							
⋮	⋮																							
⋮	⋮																							
01100100	1.00 ± 100%																							
⋮	⋮																							
⋮	⋮																							
11001000	2.00 ± 200%																							

<b>DB31, ... DBB0</b>	<b>Vorschubkorrektur (achsspezifisch)</b>																																																																																																
	<p>Duale Werte &gt; 200 werden auf 200% begrenzt.</p> <p>Mit dem Maschinendatum: MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrektorschalter) kann die maximale achsspezifische Vorschubkorrektur zusätzlich begrenzt werden.</p> <p>Bei Graycodierung sind den einzelnen Schalterstellungen folgende Codes zugeordnet:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Tabelle: Graycodierung für achsspezifische Vorschubkorrektur</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Schalter- stellung</th> <th style="text-align: center;">Code</th> <th style="text-align: center;">Axialer Vorschub- korrekturfaktor (Standardwerte)</th> </tr> </thead> <tbody> <tr><td style="text-align: center;">1</td><td style="text-align: center;">00001</td><td style="text-align: center;">0.0</td></tr> <tr><td style="text-align: center;">2</td><td style="text-align: center;">00011</td><td style="text-align: center;">0.01</td></tr> <tr><td style="text-align: center;">3</td><td style="text-align: center;">00010</td><td style="text-align: center;">0.02</td></tr> <tr><td style="text-align: center;">4</td><td style="text-align: center;">00110</td><td style="text-align: center;">0.04</td></tr> <tr><td style="text-align: center;">5</td><td style="text-align: center;">00111</td><td style="text-align: center;">0.06</td></tr> <tr><td style="text-align: center;">6</td><td style="text-align: center;">00101</td><td style="text-align: center;">0.08</td></tr> <tr><td style="text-align: center;">7</td><td style="text-align: center;">00100</td><td style="text-align: center;">0.10</td></tr> <tr><td style="text-align: center;">8</td><td style="text-align: center;">01100</td><td style="text-align: center;">0.20</td></tr> <tr><td style="text-align: center;">9</td><td style="text-align: center;">01101</td><td style="text-align: center;">0.30</td></tr> <tr><td style="text-align: center;">10</td><td style="text-align: center;">01111</td><td style="text-align: center;">0.40</td></tr> <tr><td style="text-align: center;">11</td><td style="text-align: center;">01110</td><td style="text-align: center;">0.50</td></tr> <tr><td style="text-align: center;">12</td><td style="text-align: center;">01010</td><td style="text-align: center;">0.60</td></tr> <tr><td style="text-align: center;">13</td><td style="text-align: center;">01011</td><td style="text-align: center;">0.70</td></tr> <tr><td style="text-align: center;">14</td><td style="text-align: center;">01001</td><td style="text-align: center;">0.75</td></tr> <tr><td style="text-align: center;">15</td><td style="text-align: center;">01000</td><td style="text-align: center;">0.80</td></tr> <tr><td style="text-align: center;">16</td><td style="text-align: center;">11000</td><td style="text-align: center;">0.85</td></tr> <tr><td style="text-align: center;">17</td><td style="text-align: center;">11001</td><td style="text-align: center;">0.90</td></tr> <tr><td style="text-align: center;">18</td><td style="text-align: center;">11011</td><td style="text-align: center;">0.95</td></tr> <tr><td style="text-align: center;">19</td><td style="text-align: center;">11010</td><td style="text-align: center;">1.00</td></tr> <tr><td style="text-align: center;">20</td><td style="text-align: center;">11110</td><td style="text-align: center;">1.05</td></tr> <tr><td style="text-align: center;">21</td><td style="text-align: center;">11111</td><td style="text-align: center;">1.10</td></tr> <tr><td style="text-align: center;">22</td><td style="text-align: center;">11101</td><td style="text-align: center;">1.15</td></tr> <tr><td style="text-align: center;">23</td><td style="text-align: center;">11100</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">24</td><td style="text-align: center;">10100</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">25</td><td style="text-align: center;">10101</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">26</td><td style="text-align: center;">10111</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">27</td><td style="text-align: center;">10110</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">28</td><td style="text-align: center;">10010</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">29</td><td style="text-align: center;">10011</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">30</td><td style="text-align: center;">10001</td><td style="text-align: center;">1.20</td></tr> <tr><td style="text-align: center;">31</td><td style="text-align: center;">10000</td><td style="text-align: center;">1.20</td></tr> </tbody> </table> </div> <p>Die in der Tabelle angegebenen Faktoren für die axiale Vorschubkorrektur sind im NC-spezifischen Maschinendatum: MD12010 \$MN_OVR_FACTOR_AX_SPEED [n] abgelegt.</p> <p>In der Tabelle ist die Standardvorbelegung eingetragen.</p> <p>Die Anzahl der möglichen Schalterstellungen bei den Standardmaschinensteuertafeln sind in der Projektierungsanleitung für SINUMERIK 840D beschrieben.</p>	Schalter- stellung	Code	Axialer Vorschub- korrekturfaktor (Standardwerte)	1	00001	0.0	2	00011	0.01	3	00010	0.02	4	00110	0.04	5	00111	0.06	6	00101	0.08	7	00100	0.10	8	01100	0.20	9	01101	0.30	10	01111	0.40	11	01110	0.50	12	01010	0.60	13	01011	0.70	14	01001	0.75	15	01000	0.80	16	11000	0.85	17	11001	0.90	18	11011	0.95	19	11010	1.00	20	11110	1.05	21	11111	1.10	22	11101	1.15	23	11100	1.20	24	10100	1.20	25	10101	1.20	26	10111	1.20	27	10110	1.20	28	10010	1.20	29	10011	1.20	30	10001	1.20	31	10000	1.20
Schalter- stellung	Code	Axialer Vorschub- korrekturfaktor (Standardwerte)																																																																																															
1	00001	0.0																																																																																															
2	00011	0.01																																																																																															
3	00010	0.02																																																																																															
4	00110	0.04																																																																																															
5	00111	0.06																																																																																															
6	00101	0.08																																																																																															
7	00100	0.10																																																																																															
8	01100	0.20																																																																																															
9	01101	0.30																																																																																															
10	01111	0.40																																																																																															
11	01110	0.50																																																																																															
12	01010	0.60																																																																																															
13	01011	0.70																																																																																															
14	01001	0.75																																																																																															
15	01000	0.80																																																																																															
16	11000	0.85																																																																																															
17	11001	0.90																																																																																															
18	11011	0.95																																																																																															
19	11010	1.00																																																																																															
20	11110	1.05																																																																																															
21	11111	1.10																																																																																															
22	11101	1.15																																																																																															
23	11100	1.20																																																																																															
24	10100	1.20																																																																																															
25	10101	1.20																																																																																															
26	10111	1.20																																																																																															
27	10110	1.20																																																																																															
28	10010	1.20																																																																																															
29	10011	1.20																																																																																															
30	10001	1.20																																																																																															
31	10000	1.20																																																																																															
korrespondierend mit ...	DB31, ... DBX1.7 (Korrektur wirksam) MD12010 \$MN_OVR_FACTOR_AX_SPEED [n] (Bewertung des Achs-Vorschubkorrektorschalters) MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrektorschalter)																																																																																																

<b>DB31, ... DBX1.7</b>	<b>Korrektur wirksam</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p><b>Vorschubkorrektur wirksam:</b> Die in die PLC-Nahtstelle eingetragene achsspezifische Vorschubkorrektur 0 bis maximal 200% wird berücksichtigt. Der Korrekturfaktor wird vorgegeben über die Maschinendaten: MD12000 \$MN_OVR_AX_IS_GRAY_CODE (Achsvorschubkorrekturschalter Gray-codiert) und MD12010 \$MN_OVR_FACTOR_AX_SPEED [n] (Bewertung des Achsvorschubkorrekturschalters).</p> <p><b>Spindelkorrektur wirksam:</b> Die in die PLC-Nahtstelle eingetragene Spindelkorrektur 0 bis maximal 200% wird berücksichtigt. Der Korrekturfaktor wird vorgegeben über die Maschinendaten: MD12060 \$MN_OVR_SPIND_IS_GRAY_CODE (Spindelkorrekturschalter Gray-codiert) und MD12070 \$MN_OVR_FACTOR_SPIND_SPEED [n] (Bewertung des Spindelkorrekturschalters).</p>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p>Die anstehende achsspezifische Vorschubkorrektur bzw. Spindelkorrektur ist unwirksam. Bei unwirksamer Korrektur wird als Korrekturfaktor intern "100%" verwendet. Ausnahme bilden die Nullstellung bei binärer Schnittstelle und die 1. Schalterstellung für eine graycodierte Schnittstelle. Hier werden die Korrekturfaktoren, die in die PLC-Nahtstelle eingetragen sind, verwendet. Bei binärer Schnittstelle ist der Korrekturfaktor = 0. Bei graycodierter Schnittstelle wird der in den Maschinendaten für die 1. Schalterstellung eingetragene Wert als Korrekturwert ausgegeben.</p>
Anwendungs- beispiel(e)	<p>Im Allgemeinen wird der Korrekturwert mit dem achsspezifischen Vorschubkorrekturschalter bzw. mit dem Spindelkorrekturschalter an der Maschinensteuertafel vorgegeben. Mit dem Signal "Vorschubkorrektur wirksam" können die Korrekturschalter per PLC-Anwenderprogramm während der Inbetriebnahme eines neuen NC-Programms z.B. mit dem Schlüsselschalter freigegeben werden.</p>
Sonderfälle, Fehler, .....	<p>Die Spindelkorrektur wird in der Spindelbetriebsart "Pendelbetrieb" immer mit 100% angenommen. Die Spindelkorrektur wirkt auf die programmierten Werte, bevor die Begrenzungen (z.B. G26, LIMS...) eingreifen. Die Vorschubkorrektur ist unwirksam bei:</p> <ul style="list-style-type: none"> <li>• Aktivem G33</li> <li>• Aktivem G63 (die Korrektur wird in der NC auf 100% festgesetzt)</li> <li>• Aktivem G331, G332 (die Korrektur wird in der NC auf 100% festgesetzt)</li> </ul> <p>Die Spindelkorrektur ist unwirksam bei:</p> <ul style="list-style-type: none"> <li>• Aktivem G63 (die Korrektur wird in der NC auf 100% festgesetzt)</li> </ul>
korrespondierend mit ...	DB31, ... DBB0 (Vorschub-/Spindelkorrektur)

<b>DB 31, ... DBX3.2 - DBX3.5</b>	<b>Festvorschub 1 - 4 aktivieren für Maschinenachsen</b>																														
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch																														
Beschreibung	<p>Mit diesen Signalen wird die Funktion "Festvorschub" an-/abgewählt und festgelegt, welcher Festvorschub für Maschinenachsen wirksam werden soll.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Bit 5</th> <th>Bit 4</th> <th>Bit 3</th> <th>Bit 2</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Festvorschub ist abgewählt</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>Festvorschub 1 ist angewählt</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>Festvorschub 2 ist angewählt</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>Festvorschub 3 ist angewählt</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>Festvorschub 4 ist angewählt</td> </tr> </tbody> </table>	Bit 5	Bit 4	Bit 3	Bit 2	Bedeutung	0	0	0	0	Festvorschub ist abgewählt	0	0	0	1	Festvorschub 1 ist angewählt	0	0	1	0	Festvorschub 2 ist angewählt	0	1	0	0	Festvorschub 3 ist angewählt	1	0	0	0	Festvorschub 4 ist angewählt
Bit 5	Bit 4	Bit 3	Bit 2	Bedeutung																											
0	0	0	0	Festvorschub ist abgewählt																											
0	0	0	1	Festvorschub 1 ist angewählt																											
0	0	1	0	Festvorschub 2 ist angewählt																											
0	1	0	0	Festvorschub 3 ist angewählt																											
1	0	0	0	Festvorschub 4 ist angewählt																											
korrespondierend mit ...	MD12202 \$MN_PERMANENT_FEED[n] MD12200 \$MN_RUN_OVERRIDE_0																														

<b>DB31, ... DBX4.3</b>	<b>Vorschub-Halt/Spindel-Halt (achsspezifisch)</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Das Signal ist in allen Betriebsarten wirksam.</p> <p><b>Vorschub Halt:</b> Signal bewirkt "Vorschub Halt" der jeweiligen Achse. Bei einer fahrenden Achse bewirkt dieses Signal ein geführtes Bremsen zum Stillstand (Rampenstopp). Es erfolgt dabei keine Alarmmeldung. Signal bewirkt "Vorschub Halt" aller im interpolatorischen Zusammenhang fahrenden Bahnachsen, wenn "Vorschub Halt" für eine der Bahnachsen gegeben wird. In diesem Fall werden alle Achsen unter Einhaltung der Bahnkontur zum Stillstand gebracht. Nach Wegnahme des "Vorschub Halt"-Signals wird das unterbrochene Teileprogramm wieder fortgesetzt. Die Lageregelung bleibt erhalten; d.h. der Schleppabstand wird abgebaut. Wird bei einer Achse, bei der "Vorschub Halt" ansteht, eine Fahranforderung gegeben, so bleibt diese erhalten. Diese anstehende Fahranforderung wird direkt mit der Wegnahme von "Vorschub Halt" ausgeführt. Steht die Achse im interpolatorischen Zusammenhang mit anderen, so gilt dies auch für diese Achsen.</p> <p><b>Spindel Halt:</b> Die Spindel wird entlang der Beschleunigungskennlinie auf Stillstand abgebremst. Bei Positionierbetrieb wird durch Setzen des Signals "Spindel Halt" der Positioniervorgang unterbrochen. Es gilt obiges Verhalten bezüglich Einzelachsen.</p>
Signalzustand 0 bzw. Flankenwechsel 1 → 0	<p><b>Vorschub Halt:</b> Für die Achse ist der Vorschub freigegeben. Steht für die Achse bei Wegnahme von "Vorschub Halt" eine Fahranforderung ("Fahrbefehl") an, so wird diese direkt ausgeführt.</p> <p><b>Spindel Halt:</b> Für die Spindel ist die Drehzahl freigegeben. Mit Wegnahme von "Spindel Halt" wird die Spindel mit der Beschleunigungskennlinie auf den vorherigen Drehzahlsollwert beschleunigt bzw. bei Positionierbetrieb die Positionierung fortgesetzt.</p>

<b>DB31, ... DBX4.3</b>	<b>Vorschub-Halt/Spindel-Halt (achsspezifisch)</b>
Anwendungs- beispiel(e)	<p><b>Vorschub Halt:</b> Die Verfahrbewegungen der Maschinenachsen werden mit "Vorschub Halt" nicht gestartet, wenn beispielsweise an der Maschine gewisse Betriebszustände vorliegen, die eine Achsbewegung nicht erlauben (z.B. Tür nicht geschlossen).</p> <p><b>Spindel Halt:</b> Um einen Werkzeugwechsel durchzuführen. Zur Eingabe von Hilfsfunktionen (M, S, H, T, D und F-Funktionen) während des Einrichtens.</p>
Sonderfälle, Fehler, .....	Spindel Halt ist bei aktivem G331, G332 unwirksam.

<b>DB31, ... DBB19</b>	<b>Spindelkorrektur</b>																										
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch																										
Signalzustand 1 bzw. Flankenwechsel 0 → 1	<p>Die Spindelkorrektur kann über die PLC binär- oder graycodiert vorgegeben werden. Der Korrekturwert bestimmt den Prozentanteil des programmierten Drehzahlsollwertes, der an die Spindel ausgegeben wird. Bei Binärcodierung wird der Korrekturwert in % interpretiert. 0% bis 200% Drehzahländerung sind möglich, entsprechend dem dualen Wert im Byte. Dabei gilt folgende feste Zuordnung:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="text-align: center;">Codes</th> <th style="text-align: center;">Spindelkorrekturfaktor</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00000000</td> <td style="text-align: center;">0.00 ± 0%</td> </tr> <tr> <td style="text-align: center;">00000001</td> <td style="text-align: center;">0.01 ± 1%</td> </tr> <tr> <td style="text-align: center;">00000010</td> <td style="text-align: center;">0.02 ± 2%</td> </tr> <tr> <td style="text-align: center;">00000011</td> <td style="text-align: center;">0.03 ± 3%</td> </tr> <tr> <td style="text-align: center;">.</td> <td style="text-align: center;">.</td> </tr> <tr> <td style="text-align: center;">.</td> <td style="text-align: center;">.</td> </tr> <tr> <td style="text-align: center;">.</td> <td style="text-align: center;">.</td> </tr> <tr> <td style="text-align: center;">01100100</td> <td style="text-align: center;">1.00 ± 100%</td> </tr> <tr> <td style="text-align: center;">.</td> <td style="text-align: center;">.</td> </tr> <tr> <td style="text-align: center;">.</td> <td style="text-align: center;">.</td> </tr> <tr> <td style="text-align: center;">.</td> <td style="text-align: center;">.</td> </tr> <tr> <td style="text-align: center;">11001000</td> <td style="text-align: center;">2.00 ± 200%</td> </tr> </tbody> </table> <p>Duale Werte &gt; 200 werden auf 200% begrenzt. Mit dem Maschinendatum: MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrekturschalter) kann die maximale Spindelkorrektur zusätzlich begrenzt werden.</p>	Codes	Spindelkorrekturfaktor	00000000	0.00 ± 0%	00000001	0.01 ± 1%	00000010	0.02 ± 2%	00000011	0.03 ± 3%	.	.	.	.	.	.	01100100	1.00 ± 100%	.	.	.	.	.	.	11001000	2.00 ± 200%
Codes	Spindelkorrekturfaktor																										
00000000	0.00 ± 0%																										
00000001	0.01 ± 1%																										
00000010	0.02 ± 2%																										
00000011	0.03 ± 3%																										
.	.																										
.	.																										
.	.																										
01100100	1.00 ± 100%																										
.	.																										
.	.																										
.	.																										
11001000	2.00 ± 200%																										

<p><b>DB31, ... DBB19</b></p>	<p><b>Spindelkorrektur</b></p>																																																																																																
<p>Signalzustand 1 bzw. Flankenwechsel 0 → 1</p>	<p>Bei Graycodierung sind den einzelnen Schalterstellungen folgende Codes zugeordnet:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">Tabelle: Graycodierung für Spindelkorrektur</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 15%;">Schalter- stellung</th> <th style="width: 25%;">Code</th> <th style="width: 60%;">Spindel- korrekturfaktor (Standardwerte)</th> </tr> </thead> <tbody> <tr><td>1</td><td>00001</td><td>0.5</td></tr> <tr><td>2</td><td>00011</td><td>0.55</td></tr> <tr><td>3</td><td>00010</td><td>0.60</td></tr> <tr><td>4</td><td>00110</td><td>0.65</td></tr> <tr><td>5</td><td>00111</td><td>0.70</td></tr> <tr><td>6</td><td>00101</td><td>0.75</td></tr> <tr><td>7</td><td>00100</td><td>0.80</td></tr> <tr><td>8</td><td>01100</td><td>0.85</td></tr> <tr><td>9</td><td>01101</td><td>0.90</td></tr> <tr><td>10</td><td>01111</td><td>0.95</td></tr> <tr><td>11</td><td>01110</td><td>1.00</td></tr> <tr><td>12</td><td>01010</td><td>1.05</td></tr> <tr><td>13</td><td>01011</td><td>1.10</td></tr> <tr><td>14</td><td>01001</td><td>1.15</td></tr> <tr><td>15</td><td>01000</td><td>1.20</td></tr> <tr><td>16</td><td>11000</td><td>1.20</td></tr> <tr><td>17</td><td>11001</td><td>1.20</td></tr> <tr><td>18</td><td>11011</td><td>1.20</td></tr> <tr><td>19</td><td>11010</td><td>1.20</td></tr> <tr><td>20</td><td>11110</td><td>1.20</td></tr> <tr><td>21</td><td>11111</td><td>1.20</td></tr> <tr><td>22</td><td>11101</td><td>1.20</td></tr> <tr><td>23</td><td>11100</td><td>1.20</td></tr> <tr><td>24</td><td>10100</td><td>1.20</td></tr> <tr><td>25</td><td>10101</td><td>1.20</td></tr> <tr><td>26</td><td>10111</td><td>1.20</td></tr> <tr><td>27</td><td>10110</td><td>1.20</td></tr> <tr><td>28</td><td>10010</td><td>1.20</td></tr> <tr><td>29</td><td>10011</td><td>1.20</td></tr> <tr><td>30</td><td>10001</td><td>1.20</td></tr> <tr><td>31</td><td>10000</td><td>1.20</td></tr> </tbody> </table> </div> <p>Die in der Tabelle angegebenen Faktoren für die Spindelkorrektur sind in dem Maschinendatum: MD12070 \$MN_OVR_FACTOR_SPIND_SPEED [n] abgelegt.</p> <p>In die Tabelle ist die Standardvorbelegung eingetragen.</p> <p>Die Anzahl der möglichen Schalterstellungen bei den Standardmaschinensteuertafeln in der Projektierungsanleitung für SINUMERIK 840D beschrieben.</p>	Schalter- stellung	Code	Spindel- korrekturfaktor (Standardwerte)	1	00001	0.5	2	00011	0.55	3	00010	0.60	4	00110	0.65	5	00111	0.70	6	00101	0.75	7	00100	0.80	8	01100	0.85	9	01101	0.90	10	01111	0.95	11	01110	1.00	12	01010	1.05	13	01011	1.10	14	01001	1.15	15	01000	1.20	16	11000	1.20	17	11001	1.20	18	11011	1.20	19	11010	1.20	20	11110	1.20	21	11111	1.20	22	11101	1.20	23	11100	1.20	24	10100	1.20	25	10101	1.20	26	10111	1.20	27	10110	1.20	28	10010	1.20	29	10011	1.20	30	10001	1.20	31	10000	1.20
Schalter- stellung	Code	Spindel- korrekturfaktor (Standardwerte)																																																																																															
1	00001	0.5																																																																																															
2	00011	0.55																																																																																															
3	00010	0.60																																																																																															
4	00110	0.65																																																																																															
5	00111	0.70																																																																																															
6	00101	0.75																																																																																															
7	00100	0.80																																																																																															
8	01100	0.85																																																																																															
9	01101	0.90																																																																																															
10	01111	0.95																																																																																															
11	01110	1.00																																																																																															
12	01010	1.05																																																																																															
13	01011	1.10																																																																																															
14	01001	1.15																																																																																															
15	01000	1.20																																																																																															
16	11000	1.20																																																																																															
17	11001	1.20																																																																																															
18	11011	1.20																																																																																															
19	11010	1.20																																																																																															
20	11110	1.20																																																																																															
21	11111	1.20																																																																																															
22	11101	1.20																																																																																															
23	11100	1.20																																																																																															
24	10100	1.20																																																																																															
25	10101	1.20																																																																																															
26	10111	1.20																																																																																															
27	10110	1.20																																																																																															
28	10010	1.20																																																																																															
29	10011	1.20																																																																																															
30	10001	1.20																																																																																															
31	10000	1.20																																																																																															
<p>korrespondierend mit ...</p>	<p>DB 31, ... DBX1.7 (Korrektur wirksam) MD12070 \$MN_OVR_FACTOR_SPIND_SPEED [n] (Bewertung des Spindel-Korrekturschalters) MD12100 \$MN_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Korrekturschalter)</p>																																																																																																

## 18.12.3 Signale von Achse/Spindel (DB31, ...)

<b>DB31, ... DBX62.2</b>	<b>Umdrehungsvorschub aktiv</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bei Programmierung von G95 (Umdrehungsvorschub) im JOG- oder Automatikbetrieb.
korrespondierend mit ...	SD41100 \$SN_JOG_REV_IS_ACTIVE (Umdrehungsvorschub bei JOG aktiv) SD42600 \$SC_JOG_FEED_PER_REV_SOURCE (In der Betriebsart JOG Umdrehungsvorschub für Geometrieachsen auf die ein Frame mit Rotation wirkt) SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE (Umdrehungsvorschub für Positionachsen/Spindeln) MD32040 \$MA_JOG_REV_VELO_RAPID (Umdrehungsvorschub bei JOG mit Eilgangsüberlagerung) MD32050 \$MA_JOG_REV_VELO (Umdrehungsvorschub bei JOG)

<b>DB31, ... DBB78 - DBB81</b>	<b>F-Funktion für Positionierachse</b>
Flankenbewertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	In das achsspezifische PLC-Nahtstellensignal wird der F-Wert einer im aktuellen Satz programmierten Positionierachse eingetragen. Die Zuordnung zwischen DB-Nummer und Maschinenachsnnummer geschieht über den Achsnamen. Der Wert bleibt erhalten, bis er durch einen neuen überschrieben wird. Format: Dualzahl im Real-Format.
Anwendungs- beispiel(e)	Beeinflussen des programmierten F-Wertes durch die PLC, z.B. durch Überschreiben der eingestellten achsspezifischen Vorschubkorrektur.
korrespondierend mit ...	MD22240 \$MC_AUXFU_F_SYNC_TYPE (Ausgabezeitpunkt der F-Funktionen)





## Anhang

## A.1 Liste der Abkürzungen

A	
A	Ausgang
ADI4	Analog Drive Interface for 4 Axes
AC	Adaptive Control
ALM	Active Line Module
ARM	Asynchroner rotatorischer Motor
AS	Automatisierungssystem
ASCII	American Standard Code for Information Interchange: Amerikanische Code-Norm für den Informationsaustausch
ASIC	Application Specific Integrated Circuit: Anwender-Schaltkreis
ASUP	Asynchrones Unterprogramm
AUXFU	Auxiliary Function: Hilfsfunktion
AWL	Anweisungsliste
AWP	Anwenderprogramm

B	
BA	Betriebsart
BAG	Betriebsartengruppe
BCD	Binary Coded Decimals: Im Binärcode verschlüsselte Dezimalzahlen
BERO	Berührungsloser Endschalter mit rückgekoppeltem Oszillator
BI	Binector Input
BICO	Binector Connector
BIN	Binary Files: Binärdateien
BIOS	Basic Input Output System
BKS	Basiskoordinatensystem
BO	Binector Output
BTSS	Bedientafelschnittstelle

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Compile-Zyklen
CI	Connector Input

CF-Card	Compact Flash-Card
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
CO	Connector Output
CoL	Certificate of License
COM	Communication
CPA	Compiler Projecting Data: Projektierdaten des Compilers
CRT	Cathode Ray Tube: Bildröhre
CSB	Central Service Board: PLC-Baugruppe
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: Zentrale Rechereinheit
CR	Carriage Return
CTS	Clear To Send: Meldung der Sendebereitschaft bei seriellen Daten-Schnittstellen
CUTCOM	Cutter Radius Compensation: Werkzeugradiuskorrektur

<b>D</b>	
DAU	Digital-Analog-Umwandler
DB	Datenbaustein (PLC)
DBB	Datenbaustein-Byte (PLC)
DBD	Datenbaustein-Doppelwort (PLC)
DBW	Datenbaustein-Wort (PLC)
DBX	Datenbaustein-Bit (PLC)
DDE	Dynamic Data Exchange
DIN	Deutsche Industrie Norm
DIO	Data Input/Output: Datenübertragungs-Anzeige
DIR	Directory: Verzeichnis
DLL	Dynamic Link Library
DO	Drive Object
DPM	Dual Port Memory
DPR	Dual Port RAM
DRAM	Dynamischer Speicher (ungepuffert)
DRF	Differential Resolver Function: Differential-Drehmelder-Funktion (Handrad)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run: Probelaufvorschub
DSB	Decoding Single Block: Dekodierungseinzelsatz
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Datenwort
DWORD	Doppelwort (aktuell 32 Bit)

<b>E</b>	
E	Eingang
E/A	Ein-/Ausgabe
ENC	Encoder: Istwertgeber
EFP	Einfach Peripheriemodul (PLC–E/A–Baugruppe)
EGB	Elektronisch gefährdete Baugruppen/Bauelemente
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Norm
EnDat	Geberschnittstelle
EPROM	Erasable Programmable Read Only Memory: Löscharer, elektrisch programmierbarer nur Lesespeicher
ePS Network Services	Dienste zur internetgestützten Maschinen-Fernwartung
EQN	Typbezeichnung eines Absolutwertgebers mit 2048 Sinussignalen/Umdrehung
ES	Engineering System
ESR	Erweitertes Stillsetzen und Rückziehen
ETC	ETC–Taste ">"; Erweiterung der Softkeyleiste im gleichen Menü

<b>F</b>	
FB	Funktionsbaustein (PLC)
FC	Function Call: Funktionsbaustein (PLC)
FEPROM	Flash–EPROM: Les– und schreibbarer Speicher
FIFO	First In First Out: Speicher, der ohne Adressangabe arbeitet und dessen Daten in derselben Reihenfolge gelesen werden, in der sie gespeichert wurden
FIPO	Feininterpolator
FPU	Floating Point Unit: Gleitpunkteinheit
FRK	Fräsradiuskorrektur
FST	Feed Stop: Vorschub Halt
FUP	Funktionsplan (Programmiermethode für PLC)
FW	Firmware

<b>G</b>	
GC	Global Control (PROFIBUS: Broadcast-Telegramm)
GEO	Geometrie, z.B. Geometrieachse
GIA	Gear Interpolation Data: Getriebeinterpolationsdaten
GND	Signal Ground
GP	Grundprogramm (PLC)

## Anhang

### A.1 Liste der Abkürzungen

GS	Getriebestufe
GSD	Gerätstammdatei zur Beschreibung eines PROFIBUS Slaves
GSDML	Generic Station Description Markup Language: XML-basierte Beschreibungssprache zur Erstellung einer GSD-Datei
GUD	Global User Data: Globale Anwenderdaten

<b>H</b>	
HEX	Kurzbezeichnung für hexadezimale Zahl
HiFu	Hilfsfunktion
HLA	Hydraulischer Linearantrieb
HMI	Human Machine Interface: SINUMERIK-Bedienoberfläche
HSA	Hauptspindeltrieb
HW	Hardware

<b>I</b>	
IBN	Inbetriebnahme
IKA	Interpolatorische Kompensation
IM	Interface-Modul: Anschaltungsbaugruppe
IMR	Interface-Modul Receive: Anschaltungsbaugruppe für Empfangsbetrieb
IMS	Interface-Modul Send: Anschaltungsbaugruppe für Sendebetrieb
INC	Increment: Schrittmaß
INI	Initializing Data: Initialisierungsdaten
IPO	Interpolator
ISA	International Standard Architecture
ISO	International Standard Organization

<b>J</b>	
JOG	Jogging: Einrichtbetrieb

<b>K</b>	
K <sub>v</sub>	Verstärkungsfaktor des Regelkreises
K <sub>p</sub>	Proportionalverstärkung
K <sub>Ü</sub>	Übersetzungsverhältnis
KOP	Kontaktplan (Programmiermethode für PLC)

<b>L</b>	
LAI	Logic Machine Axis Image: Logisches Maschinenachsen-Abbild
LAN	Local Area Network
LCD	Liquid-Crystal Display: Flüssigkristallanzeige
LED	Light Emitting Diode: Leuchtdiode
LF	Line Feed
LMS	Lagemesssystem
LR	Lageregler
LSB	Least Significant Bit: Niederwertigstes Bit
LUD	Local User Data: Anwenderdaten (lokal)

<b>M</b>	
MAC	Media Access Control
MAIN	Main program: Hauptprogramm (OB1, PLC)
MB	Megabyte
MCI	Motion Control Interface
MCIS	Motion–Control–Information–System
MCP	Machine Control Panel: Maschinensteuertafel
MD	Maschinendatum bzw. Maschinendaten
MDA	Manual Data Automatic: Handeingabe
MELDW	Meldungswort
MKS	Maschinenkoordinatensystem
MLFB	Maschinenlesbare Fabrikatbezeichnung
MM	Motor Module
MPF	Main Program File: Hauptprogramm (NC)
MSTT	Maschinensteuertafel

<b>N</b>	
NC	Numerical Control: Numerische Steuerung
NCK	Numerical Control Kernel: Numerik-Kern mit Satzaufbereitung, Verfahrbereich usw.
NCU	Numerical Control Unit: Hardware-Einheit des NCK
NRK	Bezeichnung des Betriebssystems des NCK
NST	Nahtstellensignal
NURBS	Non-Uniform Rational B-Spline
NV	Nullpunktverschiebung
NX	Numerical Extension: Achserweiterungsbaugruppe

O	
OB	Organisationsbaustein in der PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Bedieneinrichtung
OPI	Operation Panel Interface: Bedientafel-Anschaltung
OPT	Options: Optionen
OLP	Optical Link Plug: Busstecker für Lichtleiter
OSI	Open Systems Interconnection: Normung für Rechnerkommunikation

P	
PAA	Prozessabbild der Ausgänge
PAE	Prozessabbild der Eingänge
PC	Personal Computer
PCIN	Name der SW für den Datenaustausch mit der Steuerung
PCMCIA	Personal Computer Memory Card International Association: Speichersteckkarten-Normierung
PCU	PC Unit: PC-Box (Rechereinheit)
PG	Programmiergerät
PKE	Parameterkennung: Teil eines PKW
PKW	Parameterkennung: Wert (Parametrierteil eines PPO)
PLC	Programmable Logic Control: Anpass-Steuerung
PN	PROFINET
PNO	PROFIBUS-Nutzerorganisation
PO	POWER ON
POE	Programmorganisationseinheit
POS	Position/Positionieren
POSMO A	Positioning Motor Actuator: Positioniermotor
POSMO CA	Positioning Motor Compact AC: Komplette Antriebseinheit mit integrierter Leistungs- und Reglungsbaugruppe sowie Positioniereinheit und Programmspeicher; Wechselstrom-Einspeisung
POSMO CD	Positioning Motor Compact DC: wie CA, jedoch Gleichstromspeisung
POSMO SI	Positioning Motor Servo Integrated: Positioniermotor; Gleichstromspeisung
PPO	Parameter Prozessdaten Objekt ; Zyklisches Datentelegramm bei der Übertragung mit PROFIBUS-DP und Profil "Drehzahlveränderbare Antriebe"
PPU	Panel Processing Unit (zentrale Hardware einer Panel-basierten CNC-Steuerung z.B. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Serieller Datenbus
PRT	Programmtest
PSW	Programmsteuerwort
PTP	Point to Point: Punkt zu Punkt
PUD	Program Global User Data: Programmglobale Anwendervariable
PZD	Prozessdaten: Prozessdatenteil eines PPO

<b>Q</b>	
QFK	Quadrantenfehler Kompensation

<b>R</b>	
RAM	Random Access Memory: Schreib-/Lese-Speicher
REF	Funktion Referenzpunkt anfahren
REPOS	Funktion Repositionieren
RISC	Reduced Instruction Set Computer: Prozessortyp mit kleinem Befehlssatz und schnellem Befehlsdurchsatz
ROV	Rapid Override: Eingangskorrektur
RP	R-Parameter, Rechenparameter, vordefinierte Anwendervariable
RPA	R-Parameter Active: Speicherbereich in NCK für R-Parameternummern
RPY	Roll Pitch Yaw: Drehungsart eines Koordinatensystems
RTL	Rapid Traverse Linear Interpolation: Lineare Interpolation bei Eilgangbewegung
RTS	Request To Send: Sendeteil einschalten, Steuersignal von seriellen Daten-Schnittstellen
RTCP	Real Time Control Protocol

<b>S</b>	
SA	Synchronaktion
SBC	Safe Break Control: Sichere Bremsenansteuerung
SBL	Single Block: Einzelsatz
SBR	Subroutine: Unterprogramm (PLC)
SD	Settingdatum bzw. Settingdaten
SDB	System Datenbaustein
SEA	Setting Data Active: Kennzeichnung (Dateityp) für Settingdaten
SERUPRO	Search-Run by Program Test: Suchlauf via Programmtest
SFB	System Funktionsbaustein
SFC	System Function Call
SGE	Sicherheitsgerichteter Eingang
SGA	Sicherheitsgerichteter Ausgang
SH	Sicherer Halt
SIM	Single in Line Module
SK	Softkey
SKP	Skip: Funktion zum Ausblenden eines Teileprogrammsatzes
SLM	Synchroner Linearmotor
SM	Schrittmotor
SMC	Sensor Module Cabinet Mounted

## Anhang

### A.1 Liste der Abkürzungen

SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Sub Program File: Unterprogramm (NC)
SPS	Speicherprogrammierbare Steuerung = PLC
SRAM	Statischer Speicher (gepuffert)
SRK	Schneidenradiuskorrektur
SRM	Synchron rotatorischer Motor
SSFK	Spindelsteigungsfehlerkompensation
SSI	Serial Synchron Interface: Serielle synchrone Schnittstelle
SSL	Satzsuchlauf
STW	Steuerwort
SUG	Scheibenumfangsgeschwindigkeit
SW	Software
SYF	System Files: Systemdateien
SYNACT	Synchronized Action: Synchronaktion

<b>T</b>	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point: Werkzeugspitze
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: Kennung für Maschinendaten
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Werkzeugkorrektur
TOA	Tool Offset Active: Kennzeichnung (Dateityp) für Werkzeugkorrekturen
TRANSMIT	Transform Milling Into Turning: Koordinatentransformation für Fräsbearbeitungen an einer Drehmaschine
TTL	Transistor–Transistor–Logik (Schnittstellen–Typ)
TZ	Technologiezyklus

<b>U</b>	
UFR	User Frame: Nullpunktverschiebung
UP	Unterprogramm
USB	Universal Serial Bus
USV	Unterbrechungsfreie Stromversorgung



<b>V</b>	
VDI	Interne Kommunikationsschnittstelle zwischen NCK und PLC
VDI	Verein Deutscher Ingenieure
VDE	Verband Deutscher Elektrotechniker
VI	Voltage Input
VO	Voltage Output
VSA	Vorschubantrieb

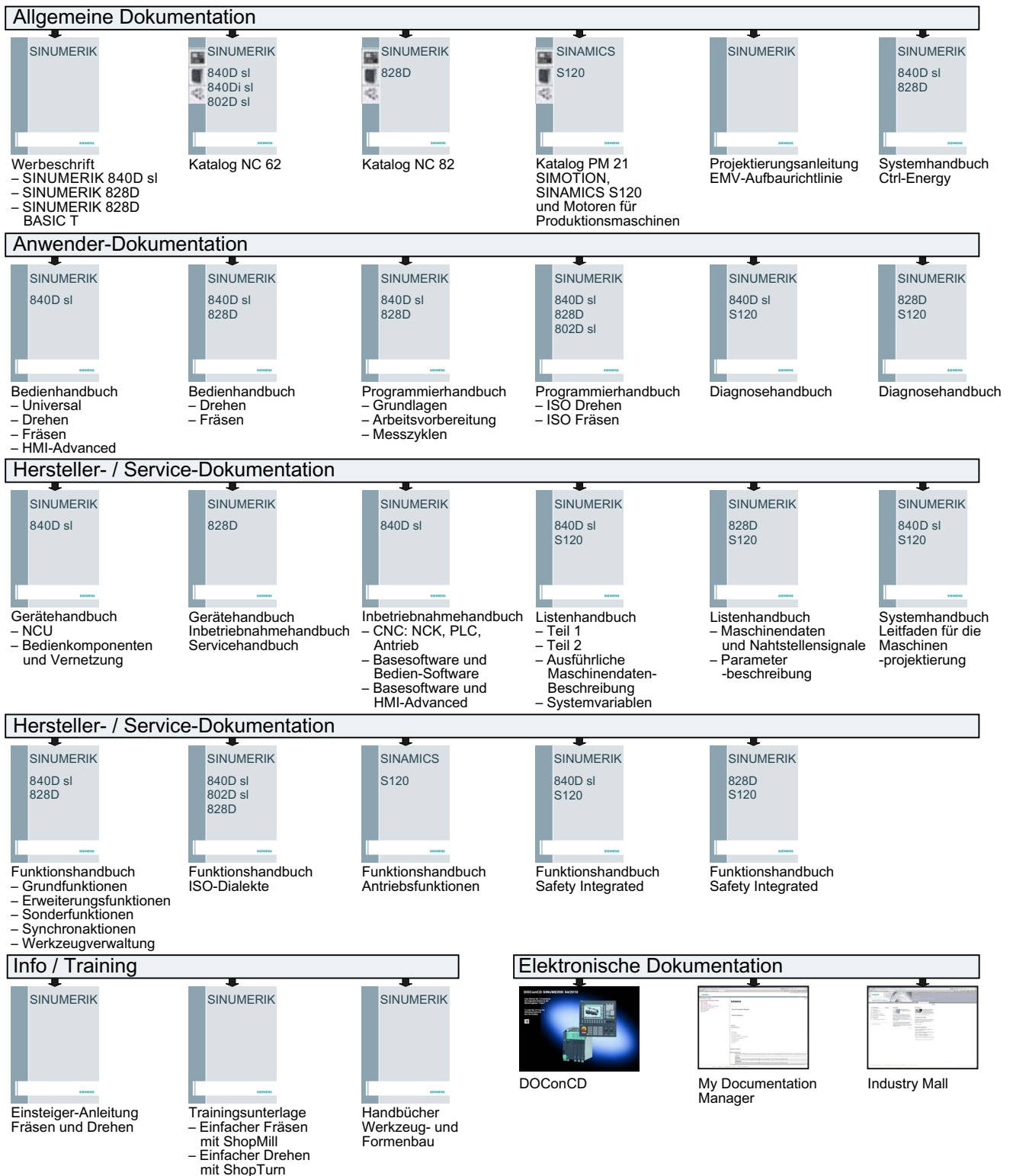
<b>W</b>	
WAB	Funktion Weiches An- und Abfahren
WKS	Werkstückkoordinatensystem
WKZ	Werkzeug
WLK	Werkzeuglängenkorrektur
WOP	Werkstatt-orientierte Programmierung
WPD	Work Piece Directory: Werkstückverzeichnis
WRK	Werkzeug-Radius-Korrektur
WZ	Werkzeug
WZK	Werkzeugkorrektur
WZV	Werkzeugverwaltung
WZW	Werkzeugwechsel

<b>X</b>	
XML	Extensible Markup Language

<b>Z</b>	
ZOA	Zero Offset Active: Kennung für Nullpunktverschiebungen
ZSW	Zustandswort (des Antriebs)



## A.2 Dokumentationsübersicht





# Glossar

## Absolutmaß

Angabe des Bewegungsziels einer Achsbewegung durch ein Maß, das sich auf den Nullpunkt des momentan gültigen Koordinatensystems bezieht. Siehe → Kettenmaß.

## Achsadresse

Siehe → Achsbezeichner

## Achsbezeichner

Achsen werden nach DIN 66217 für ein rechtsdrehendes, rechtwinkliges → Koordinatensystem bezeichnet mit X, Y, Z.

Um X, Y, Z drehende → Rundachsen erhalten die Bezeichner A, B, C. Zusätzliche Achsen, parallel zu den angegebenen, können mit weiteren Adressbuchstaben gekennzeichnet werden.

## Achsen

Die CNC-Achsen werden entsprechend ihres Funktionsumfangs abgestuft in:

- Achsen: interpolierende Bahnachsen
- Hilfsachsen: nicht interpolierende Zustell- und Positionierachsen mit achsspezifischem Vorschub. Hilfsachsen sind an der eigentlichen Bearbeitung nicht beteiligt, z. B. Werkzeugzubringer, Werkzeugmagazin.

## Achsname

Siehe → Achsbezeichner

## Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden oder Operandenbereich, z. B. Eingang, Ausgang usw.

## Alarmer

Alle → Meldungen und Alarmer werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

### 1. Alarmer und Meldungen im Teileprogramm

Alarmer und Meldungen können direkt aus dem Teileprogramm im Klartext zur Anzeige gebracht werden.

### 2. Alarmer und Meldungen von PLC

Alarmer- und Meldungen der Maschine können aus dem PLC-Programm im Klartext zur Anzeige gebracht werden. Dazu sind keine zusätzlichen Funktionsbaustein-Pakete notwendig.

## Antrieb

Der Antrieb ist diejenige Einheit der CNC, welche die Drehzahl- und Momentenregelung aufgrund der Vorgaben der NC ausführt.

## Anwenderdefinierte Variable

Anwender können für beliebige Nutzung im → Teileprogramm oder Datenbaustein (globale Anwenderdaten) anwenderdefinierte Variablen vereinbaren. Eine Definition enthält eine Datentypangabe und den Variablennamen. Siehe → Systemvariable.

## Anwenderprogramm

Anwenderprogramme für Automatisierungssysteme S7-300 werden mit der Programmiersprache STEP 7 erstellt. Das Anwenderprogramm ist modular aufgebaut und besteht aus einzelnen Bausteinen.

Die grundlegenden Bausteintypen sind:

- Code-Bausteine

Diese Bausteine enthalten die STEP 7-Befehle.

- Datenbausteine

Diese Bausteine enthalten Konstanten und Variablen für das STEP 7-Programm.

## Anwenderspeicher

Alle Programme und Daten wie Teileprogramme, Unterprogramme, Kommentare, Werkzeugkorrekturen, Nullpunktverschiebungen/Frames sowie Kanal- und Programm-anwenderdaten können in den gemeinsamen CNC-Anwenderspeicher abgelegt werden.

**Arbeitsfeldbegrenzung**

Mit der Arbeitsfeldbegrenzung kann der Verfahrbereich der Achsen zusätzlich zu den Endschaltern eingeschränkt werden. Je Achse ist ein Wertepaar zur Beschreibung des geschützten Arbeitsraumes möglich.

**Arbeitsraum**

Dreidimensionaler Raum, in den die Werkzeugspitze aufgrund der Konstruktion der Werkzeugmaschine hineinfahren kann. Siehe → Schutzraum.

**Arbeitsspeicher**

Der Arbeitsspeicher ist ein RAM-Speicher in der → CPU, auf den der Prozessor während der Programmbearbeitung auf das Anwenderprogramm zugreift.

**Archivieren**

Auslesen von Dateien und/oder Verzeichnissen auf ein **externes** Speichergerät.

**Asynchrones Unterprogramm**

Teileprogramm, das asynchron (unabhängig) zum aktuellen Programmzustand durch ein Interruptsignal (z. B. Signal "schneller NC-Eingang") gestartet werden kann.

**Automatik**

Betriebsart der Steuerung (Satzfolgebetrieb nach DIN): Betriebsart bei NC-Systemen, in der ein → Teileprogramm angewählt und kontinuierlich abgearbeitet wird.

**Bahnachse**

Bahnachsen sind alle Bearbeitungsachsen des → Kanals, die vom → Interpolator so geführt werden, dass sie gleichzeitig starten, beschleunigen, stoppen und den Endpunkt erreichen.

**Bahngeschwindigkeit**

Die maximal programmierbare Bahngeschwindigkeit ist abhängig von der Eingabefeinheit. Bei einer Auflösung von beispielsweise 0,1 mm beträgt die maximal programmierbare Bahngeschwindigkeit 1000 m/min.

**Bahnsteuerbetrieb**

Ziel des Bahnsteuerbetriebes ist es, ein größeres Abbremsen der → Bahnachsen an den Teileprogramm-Satzgrenzen zu vermeiden und mit möglichst gleicher Bahngeschwindigkeit in den nächsten Satz zu wechseln.

### **Bahnvorschub**

Bahnvorschub wirkt auf → Bahnachsen. Er stellt die geometrische Summe der Vorschübe der beteiligten → Geometrieachsen dar.

### **Basisachse**

Achse, deren Soll- oder Istwert für die Berechnung eines Kompensationswertes herangezogen wird.

### **Basiskoordinatensystem**

Kartesisches Koordinatensystem, wird durch Transformation auf das Maschinenkoordinatensystem abgebildet.

Im → Teileprogramm verwendet der Programmierer Achsnamen des Basiskoordinatensystems. Es besteht, wenn keine → Transformation aktiv ist, parallel zum → Maschinenkoordinatensystem. Der Unterschied zu diesem liegt in den → Achsbezeichnungen.

### **Baudrate**

Geschwindigkeit bei der Datenübertragung (Bit/s).

### **Baustein**

Als Bausteine werden alle Dateien bezeichnet, die für die Programmerstellung und Programmverarbeitung benötigt werden.

### **Bearbeitungskanal**

Über eine Kanalstruktur können durch parallele Bewegungsabläufe Nebenzeiten verkürzt werden, z. B. Verfahren eines Ladeportals simultan zur Bearbeitung. Ein CNC-Kanal ist dabei als eigene CNC-Steuerung mit Dekodierung, Satzaufbereitung und Interpolation anzusehen.

### **Bedienoberfläche**

Die Bedienoberfläche (BOF) ist das Anzeigemedium einer CNC-Steuerung in Gestalt eines Bildschirms. Sie ist mit horizontalen und vertikalen Softkeys gestaltet.

### **Beschleunigung mit Ruckbegrenzung**

Zur Erzielung eines optimalen Beschleunigungsverhaltens an der Maschine bei gleichzeitiger Schonung der Mechanik kann im Bearbeitungsprogramm zwischen sprunghafter Beschleunigung und stetiger (ruckfreier) Beschleunigung umgeschaltet werden.



**Betriebsart**

Ablaufkonzept für den Betrieb einer SINUMERIK-Steuerung. Es sind die Betriebsarten → Jog, → MDA, → Automatik definiert.

**Betriebsartengruppe**

Technologisch zusammengehörige Achsen und Spindeln können zu einer Betriebsartengruppe (BAG) zusammengefasst werden. Achsen/Spindeln einer BAG können von einem oder mehreren → Kanälen gesteuert werden. Den Kanälen der BAG ist immer die gleiche → Betriebsart zugeordnet.

**Bezeichner**

Die Wörter nach DIN 66025 werden durch Bezeichner (Namen) für Variable (Rechenvariable, Systemvariable, Anwendervariable), für Unterprogramme, für Schlüsselwörter und Wörter mit mehreren Adressbuchstaben ergänzt. Diese Ergänzungen kommen in der Bedeutung den Wörtern beim Satzaufbau gleich. Bezeichner müssen eindeutig sein. Derselbe Bezeichner darf nicht für verschiedene Objekte verwendet werden.

**Booten**

Laden des Systemprogramms nach Power On.

**C-Achse**

Achse, um die eine gesteuerte Drehbewegung und Positionierung mit der Werkstückspindel erfolgt.

**CNC**

Siehe → NC

**COM**

Komponente der NC-Steuerung zur Durchführung und Koordination von Kommunikation.

**CPU**

Central Processing Unit, siehe → Speicherprogrammierbare Steuerung

**C-Spline**

Der C-Spline ist der bekannteste und am meisten verwendete Spline. Die Übergänge an den Stützpunkten sind tangential- und krümmungsstetig. Es werden Polynome 3. Grades verwendet.

### Datenbaustein

1. Dateneinheit der → PLC, auf die → HIGHSTEP-Programme zugreifen können.
2. Dateneinheit der → NC: Datenbausteine enthalten Datendefinitionen für globale Anwenderdaten. Die Daten können bei der Definition direkt initialisiert werden.

### Datenübertragungsprogramm PCIN

PCIN ist ein Hilfsprogramm zum Senden und Empfangen von CNC-Anwenderdaten über die serielle Schnittstelle, wie z. B. Teileprogramme, Werkzeugkorrekturen etc. Das PCIN-Programm ist unter MS-DOS auf Standard-Industrie-PCs lauffähig.

### Datenwort

Zwei Byte große Dateneinheit innerhalb eines → Datenbausteins.

### Diagnose

1. Bedienbereich der Steuerung
2. Die Steuerung besitzt sowohl ein Selbstdiagnose-Programm als auch Testhilfen für den Service: Status-, Alarm- und Serviceanzeigen

### DRF

Differential Resolver Function: NC-Funktion, die in Verbindung mit einem elektronischen Handrad eine inkrementale Nullpunktverschiebung im Automatik-Betrieb erzeugt.

### Editor

Der Editor ermöglicht das Erstellen, Ändern, Ergänzen, Zusammenschieben und Einfügen von Programmen/Texten/Programmsätzen.

### Eilgang

Schnellste Verfahrgeschwindigkeit einer Achse. Sie wird z. B. verwendet, wenn das Werkzeug aus einer Ruhestellung an die → Werkstückkontur herangefahren oder von der Werkstückkontur zurückgezogen wird. Die Eilganggeschwindigkeit wird maschinenspezifisch über Maschinendatum eingestellt.

### Externe Nullpunktverschiebung

Von der → PLC vorgegebene Nullpunktverschiebung.

### Fertigteilkontur

Kontur des fertig bearbeiteten Werkstücks. Siehe → Rohteil.

## Festpunkt-Anfahren

Werkzeugmaschinen können feste Punkte wie Werkzeugwechsellpunkt, Beladepunkt, Palettenwechsellpunkt etc. definiert anfahren. Die Koordinaten dieser Punkte sind in der Steuerung hinterlegt. Die Steuerung verfährt die betroffenen Achsen, wenn möglich, im → Eilgang.

## Frame

Ein Frame stellt eine Rechenvorschrift dar, die ein kartesisches Koordinatensystem in ein anderes kartesisches Koordinatensystem überführt. Ein Frame enthält die Komponenten → Nullpunktverschiebung, → Rotation, → Skalierung, → Spiegelung.

## Führungssachse

Die Führungssachse ist die → Gantry-Achse, die aus Sicht des Bedieners und des Programmierers vorhanden und damit entsprechend wie eine normale NC-Achse beeinflussbar ist.

## Funktionalität

Die Begrenzung des Bahnrucks kann durch Programmierung des Settingdatums ein/ausgeschaltet werden.

Parameter: *Wert*

- Wertebereich: TRUE, FALSE

Anwendbarkeit:

- Teileprogramm
- Statische Synchronaktion

## Genauhalt

Bei programmierter Genauhalt-Anweisung wird die in einem Satz angegebene Position genau und ggf. sehr langsam angefahren. Zur Reduktion der Annäherungszeit werden für Eilgang und Vorschub → Genauhaltsgrenzen definiert.

## Genauhaltgrenze

Erreichen alle Bahnachsen ihre Genauhaltgrenze, so verhält sich die Steuerung als habe sie einen Zielpunkt exakt erreicht. Es erfolgt Satzweilerschaltung des → Teileprogramms.

## Geometrie

Beschreibung eines → Werkstücks im → Werkstückkoordinatensystem.

### **Geometrieachse**

Geometrieachsen dienen der Beschreibung eines 2- oder 3-dimensionalen Bereichs im Werkstückkoordinatensystem.

### **Geradeninterpolation**

Das Werkzeug wird auf einer Geraden zum Zielpunkt verfahren und dabei das Werkstück bearbeitet.

### **Geschwindigkeitsführung**

Um bei Verfahrbewegungen um sehr kleine Beträge je Satz eine akzeptable Verfahrgeschwindigkeit erreichen zu können, kann vorausschauende Auswertung über mehrere Sätze (→ Look Ahead) eingestellt werden.

### **Gewindebohren ohne Ausgleichsfutter**

Mit dieser Funktion können Gewinde ohne Ausgleichsfutter gebohrt werden. Durch das interpolierende Verfahren der Spindel als Rundachse und der Bohrachse werden Gewinde exakt auf Endbohrtiefe geschnitten, z. B. Sacklochgewinde (Voraussetzung: Achsbetrieb der Spindel).

### **Gleichlaufachse**

Die Gleichlaufachse ist die → Gantry-Achse, deren Sollposition stets von der Verfahrbewegung der → Führungsachse abgeleitet und damit synchron verfahren wird. Aus Sicht des Bedieners und des Programmierers ist die Gleichlaufachse "nicht vorhanden".

### **Grenzdrehzahl**

Maximale/minimale (Spindel-)Drehzahl: Durch Vorgaben von Maschinendaten, der → PLC oder → Settingdaten kann die maximale Drehzahl einer Spindel begrenzt sein.

### **Hauptprogramm**

Die Bezeichnung Hauptprogramm stammt noch aus der Zeit, als Teileprogramm fest in Haupt- und → Unterprogramme unterteilt waren. Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Prinzipiell kann jedes Teileprogramm im Kanal angewählt und gestartet werden. Es läuft dann in der → Programmebene 0 (Hauptprogramm-Ebene) ab. Im Hauptprogramm können weitere Teileprogramme oder → Zyklen als Unterprogramme aufgerufen werden

### **Hauptsatz**

Durch ":" eingeleiteter Satz, der alle Angaben enthält, um den Arbeitsablauf in einem → Teileprogramm starten zu können.

## HIGHSTEP

Zusammenfassung der Programmiermöglichkeiten für die → PLC des Systems AS300/AS400.

## Hilfsfunktionen

Mit Hilfsfunktionen können in → Teileprogrammen → Parameter an die → PLC übergeben werden, die dort vom Maschinenhersteller definierte Reaktionen auslösen.

## Hochsprache CNC

Die Hochsprache bietet: → Anwenderdefinierte Variable, → Systemvariable, → Makrotechnik.

## HW-Konfig

SIMATIC S7-Tool zum Konfigurieren und Parametrieren von Hardware-Komponenten innerhalb eines S7-Projekts.

## Interpolator

Logische Einheit des → NCK, die nach Angaben von Zielpositionen im Teileprogramm Zwischenwerte für die in den einzelnen Achsen zu fahrenden Bewegungen bestimmt.

## Interpolatorische Kompensation

Mit Hilfe der interpolatorischen Kompensation können fertigungsbedingte **Spindelsteigungsfehler** und **Messsystemfehler** kompensiert werden (SSFK, MSFK).

## Interruptroutine

Interruptroutinen sind spezielle → Unterprogramme, die durch Ereignisse (externe Signale) vom Bearbeitungsprozess gestartet werden können. Ein in Abarbeitung befindlicher Teileprogrammsatz wird abgebrochen, die Unterbrechungsposition der Achsen wird automatisch gespeichert.

## JOG

Betriebsart der Steuerung (Einrichtebetrieb): In der Betriebsart JOG kann die Maschine eingerichtet werden. Einzelne Achsen und Spindeln können über die Richtungstasten im Tippbetrieb verfahren werden. Weitere Funktionen in der Betriebsart JOG sind das → Referenzpunktfahren, → Repos sowie → Preset (Istwert setzen).

## Kanal

Ein Kanal ist dadurch gekennzeichnet, dass er unabhängig von anderen Kanälen ein → Teileprogramm abarbeiten kann. Ein Kanal steuert exklusiv die ihm zugeordneten Achsen und Spindeln. Teileprogrammabläufe verschiedener Kanäle können durch → Synchronisation koordiniert werden.

### **Kettenmaß**

Auch Inkrementmaß: Angabe eines Bewegungsziels einer Achse durch eine zu verfahrenende Wegstrecke und Richtung bezogen auf einen bereits erreichten Punkt. Siehe → Absolutmaß.

### **Kompensationsachse**

Achse, deren Soll- oder Istwert durch den Kompensationswert modifiziert wird.

### **Kompensationstabelle**

Tabelle von Stützpunkten. Sie liefert für ausgewählte Positionen der Basisachse die Kompensationswerte der Kompensationsachse.

### **Kompensationswert**

Differenz zwischen der durch den Messgeber gemessenen Achsposition und der gewünschten, programmierten Achsposition.

### **Kontur**

Umriss des → Werkstücks

### **Konturüberwachung**

Als Maß für die Konturtreue wird der Schleppfehler innerhalb eines definierbaren Toleranzbandes überwacht. Ein unzulässig hoher Schleppfehler kann sich z. B. durch Überlastung des Antriebs ergeben. In diesem Fall kommt es zu einem Alarm und die Achsen werden stillgesetzt.

### **Koordinatensystem**

Siehe → Maschinenkoordinatensystem, → Werkstückkoordinatensystem

### **Korrekturspeicher**

Datenbereich in der Steuerung, in dem Werkzeugkorrekturdaten hinterlegt sind.

### **Kreisinterpolation**

Das → Werkzeug soll zwischen festgelegten Punkten der Kontur mit einem gegebenen Vorschub auf einem Kreis fahren und dabei das Werkstück bearbeiten.

### **Krümmung**

Die Krümmung  $k$  einer Kontur ist das Inverse des Radius  $r$  des anschmiegenden Kreises in einem Konturpunkt ( $k = 1/r$ ).

**KÜ**

Übersetzungsverhältnis

**KV**

Kreisverstärkungsfaktor, regelungstechnische Größe eines Regelkreises

**Ladespeicher**

Der Ladespeicher ist bei der CPU 314 der → SPS gleich dem → Arbeitsspeicher.

**Linearachse**

Die Linearachse ist eine Achse, welche im Gegensatz zur Rundachse eine Gerade beschreibt.

**Look Ahead**

Mit der Funktion **Look Ahead** wird durch das "Vorausschauen" über eine parametrierbare Anzahl von Verfahrstrichen ein Optimum an Bearbeitungsgeschwindigkeit erzielt.

**Losekompensation**

Ausgleich einer mechanischen Maschinenlose, z. B. Umkehrlose bei Kugelrollspindeln. Für jede Achse kann die Losekompensation getrennt eingegeben werden.

**Makrotechnik**

Zusammenfassung einer Menge von Anweisungen unter einem Bezeichner. Der Bezeichner repräsentiert im Programm die Menge der zusammengefassten Anweisungen.

**Maschinenachsen**

In der Werkzeugmaschine physikalisch existierende Achsen.

**Maschinenfestpunkt**

Durch die Werkzeugmaschine eindeutig definierter Punkt, z. B. Maschinen-Referenzpunkt.

**Maschinenkoordinatensystem**

Koordinatensystem, das auf die Achsen der Werkzeugmaschine bezogen ist.

**Maschinennullpunkt**

Fester Punkt der Werkzeugmaschine, auf den sich alle (abgeleiteten) Messsysteme zurückführen lassen.

### **Maschinensteuertafel**

Bedientafel der Werkzeugmaschine mit den Bedienelementen Tasten, Drehschalter usw. und einfachen Anzeigeelementen wie LEDs. Sie dient der unmittelbaren Beeinflussung der Werkzeugmaschine über die PLC.

### **Maßangabe metrisch und inch**

Im Bearbeitungsprogramm können Positions- und Steigungswerte in inch programmiert werden. Unabhängig von der programmierbaren Maßangabe (*G70/G71*) wird die Steuerung auf ein Grundsystem eingestellt.

### **Masse**

Als Masse gilt die Gesamtheit aller untereinander verbundenen inaktiven Teile eines Betriebsmittels, die auch im Fehlerfall keine gefährliche Berührungsspannung annehmen können.

### **MDA**

Betriebsart der Steuerung: Manual Data Automatic. In der Betriebsart MDA können einzelne Programmsätze oder Satzfolgen ohne Bezug auf ein Haupt- oder Unterprogramm eingegeben und anschließend über die Taste NC-Start sofort ausgeführt werden.

### **Meldungen**

Alle im Teileprogramm programmierten Meldungen und vom System erkannte → Alarme werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

### **Metrisches Messsystem**

Genormtes System von Einheiten: für Längen z. B. mm (Millimeter), m (Meter).

### **NC**

Numerical Control: NC-Steuerung umfasst alle Komponenten der Werkzeugmaschinensteuerung: → NCK, → PLC, HMI, → COM.

---

#### **Hinweis**

Für die SINUMERIK-Steuerungen wäre CNC-Steuerung korrekter: Computerized Numerical Control.

---



**NCK**

Numerical Control Kernel: Komponente der NC-Steuerung, die → Teileprogramme abarbeitet und im Wesentlichen die Bewegungsvorgänge für die Werkzeugmaschine koordiniert.

**Nebensatz**

Durch "N" eingeleiteter Satz mit Informationen für einen Arbeitsschritt, z. B. eine Positionsangabe.

**Netz**

Ein Netz ist die Verbindung von mehreren S7-300 und weiteren Endgeräten, z. B. einem PG, über → Verbindungskabel. Über das Netz erfolgt ein Datenaustausch zwischen den angeschlossenen Geräten.

**NRK**

Numeric Robotic Kernel (Betriebssystem des → NCK)

**Nullpunktverschiebung**

Vorgabe eines neuen Bezugspunkts für ein Koordinatensystem durch Bezug auf einen bestehenden Nullpunkt und ein → Frame.

1. Einstellbar

Es steht eine projektierbare Anzahl von einstellbaren Nullpunktverschiebungen für jede CNC-Achse zur Verfügung. Die über G-Funktionen anwählbaren Verschiebungen sind alternativ wirksam.

2. Extern

Zusätzlich zu allen Verschiebungen, die die Lage des Werkstücknullpunkts festlegen, kann eine externe Nullpunktverschiebung durch Handrad (DRF-Verschiebung) oder von der PLC überlagert werden.

3. Programmierbar

Mit der Anweisung `TRANS` sind für alle Bahn- und Positionierachsen Nullpunktverschiebungen programmierbar.

**NURBS**

Die steuerungsinterne Bewegungsführung und Bahninterpolation wird auf Basis von NURBS (**N**on **U**niform **R**ational **B**-**S**plines) durchgeführt. Damit steht steuerungsintern für alle Interpolationen ein einheitliches Verfahren zur Verfügung.

**OEM**

Für Maschinenhersteller, die ihre eigene Bedienoberfläche erstellen oder technologiespezifische Funktionen in die Steuerung einbringen wollen, sind Freiräume für individuelle Lösungen (OEM-Applikationen) vorgesehen.

### Orientierter Spindelhalt

Halt der Werkstückspindel in vorgegebener Winkellage, z. B. um an bestimmter Stelle eine Zusatzbearbeitung vorzunehmen.

### Orientierter Werkzeugrückzug

**RETTTOOL:** Bei Bearbeitungsunterbrechungen (z. B. bei Werkzeugbruch) kann das Werkzeug per Programmbefehl mit vorgegebener Orientierung um einen definierten Weg zurückgezogen werden.

### Override

Manuelle bzw. programmierbare Eingriffsmöglichkeit, die es dem Bediener gestattet, programmierte Vorschübe oder Drehzahlen zu überlagern, um sie einem bestimmten Werkstück oder Werkstoff anzupassen.

### Peripheriebaugruppe

Peripheriebaugruppen stellen die Verbindung zwischen CPU und Prozess her.

Peripheriebaugruppen sind:

- → Digital-Ein-/Ausgabebaugruppen
- → Analog-Ein-/Ausgabebaugruppen
- → Simulatorbaugruppen

### PLC

**Programmable Logic Control:** → Speicherprogrammierbare Steuerung. Komponente der → NC: Anpass-Steuerung zur Bearbeitung der Kontroll-Logik der Werkzeugmaschine.

### PLC-Programmierung

Die PLC wird mit der Software **STEP 7** programmiert. Die Programmiersoftware STEP 7 basiert auf dem Standardbetriebssystem **WINDOWS** und enthält die Funktionen der STEP 5 -Programmierung mit innovativen Weiterentwicklungen.

### PLC-Programmspeicher

**SINUMERIK 840D sl:** Im PLC-Anwenderspeicher werden das PLC-Anwenderprogramm und die Anwenderdaten gemeinsam mit dem PLC-Grundprogramm abgelegt.

### Polarkoordinaten

Koordinatensystem, das die Lage eines Punktes in einer Ebene durch seinen Abstand vom Nullpunkt und den Winkel festlegt, den der Radiusvektor mit einer festgelegten Achse bildet.

### Polynom-Interpolation

Mit der Polynom-Interpolation können die unterschiedlichsten Kurvenverläufe erzeugt werden, wie **Gerade-, Parabel-, Potenzfunktionen** (SINUMERIK 840D sl).

### Positionierachse

Achse, die eine Hilfsbewegung an einer Werkzeugmaschine ausführt. (z. B. Werkzeugmagazin, Palettentransport). Positionierachsen sind Achsen, die nicht mit den → Bahnachsen interpolieren.

### Programmbaustein

Programmbausteine enthalten die Haupt- und Unterprogramme der → Teileprogramme.

### Programmebene

Ein im Kanal gestartetes Teileprogramm läuft als → Hauptprogramm auf Programmebene 0 (Hauptprogramm-Ebene). Jedes im Hauptprogramm aufgerufene Teileprogramm läuft als → Unterprogramm auf einer eigenen Programmebene 1 ... n.

### Programmierbare Arbeitsfeldbegrenzung

Begrenzung des Bewegungsraumes des Werkzeugs auf einen durch programmierte Begrenzungen definierten Raum.

### Programmierbare Frames

Mit programmierbaren → Frames können dynamisch im Zuge der Teileprogramm-Abarbeitung neue Koordinatensystem-Ausgangspunkte definiert werden. Es wird unterschieden nach absoluter Festlegung anhand eines neuen Frames und additiver Festlegung unter Bezug auf einen bestehenden Ausgangspunkt.

### Programmierschlüssel

Zeichen und Zeichenfolgen, die in der Programmiersprache für → Teileprogramme eine festgelegte Bedeutung haben.

### Pufferbatterie

Die Pufferbatterie gewährleistet, dass das → Anwenderprogramm in der → CPU netzausfallsicher hinterlegt ist und festgelegte Datenbereiche und Merker, Zeiten und Zähler remanent gehalten werden.

### **Quadrantenfehlerkompensation**

Konturfehler an Quadrantenübergängen, die durch wechselnde Reibverhältnisse an Führungsbahnen entstehen, sind mit der Quadrantenfehlerkompensation weitgehend eliminierbar. Die Parametrierung der Quadrantenfehlerkompensation erfolgt durch einen Kreisformtest.

### **Referenzpunkt**

Punkt der Werkzeugmaschine, auf den sich das Messsystem der → Maschinenachsen bezieht.

### **Rohteil**

Teil, mit dem die Bearbeitung eines Werkstücks begonnen wird.

### **Rotation**

Komponente eines → Frames, die eine Drehung des Koordinatensystems um einen bestimmten Winkel definiert.

### **R-Parameter**

Rechenparameter, kann vom Programmierer des → Teileprogramms für beliebige Zwecke im Programm gesetzt oder abgefragt werden.

### **Rundachse**

Rundachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine vorgegebene Winkellage.

### **Rundungsachse**

Rundungsachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine einem Teilungsraster entsprechende Winkellage. Beim Erreichen eines Rasters ist die Rundungsachse "in Position".

### **Satzsuchlauf**

Zum Austesten von Teileprogrammen oder nach einem Abbruch der Bearbeitung kann über die Funktion "Satzsuchlauf" eine beliebige Stelle im Teileprogramm angewählt werden, an der die Bearbeitung gestartet oder fortgesetzt werden soll.

### **Schlüsselschalter**

Der Schlüsselschalter auf der → Maschinensteuertafel besitzt 4 Stellungen, die vom Betriebssystem der Steuerung mit Funktionen belegt sind. Zum Schlüsselschalter gehören drei verschiedenfarbige Schlüssel, die in den angegebenen Stellungen abgezogen werden können.

## Schlüsselwörter

Wörter mit festgelegter Schreibweise, die in der Programmiersprache für → Teileprogramme eine definierte Bedeutung haben.

## Schneidenradiuskorrektur

Bei der Programmierung einer Kontur wird von einem spitzen Werkzeug ausgegangen. Da dies in der Praxis nicht realisierbar ist, wird der Krümmungsradius des eingesetzten Werkzeugs der Steuerung angegeben und von dieser berücksichtigt. Dabei wird der Krümmungsmittelpunkt um den Krümmungsradius verschoben äquidistant um die Kontur geführt.

## Schnellabheben von der Kontur

Beim Eintreffen eines Interrupts kann über das CNC-Bearbeitungsprogramm eine Bewegung eingeleitet werden, die ein schnelles Abheben des Werkzeugs von der gerade bearbeiteten Werkstückkontur ermöglicht. Zusätzlich kann der Rückzugwinkel und der Betrag des Weges parametrisiert werden. Nach dem Schnellabheben kann zusätzlich eine Interruptroutine ausgeführt werden.

## Schnelle digitale Ein-/Ausgänge

Über die digitalen Eingänge können z. B. schnelle CNC-Programmroutinen (Interruptroutinen) gestartet werden. Über die digitalen CNC-Ausgänge können schnelle, programmgesteuerte Schaltfunktionen ausgelöst werden.

## Schrägenbearbeitung

Bohr- und Fräsbearbeitungen an Werkstückflächen, die nicht in den Koordinatenebenen der Maschine liegen, können mit Unterstützung der Funktion "Schrägenbearbeitung" komfortabel ausgeführt werden.

## Schraubenlinien-Interpolation

Die Schraubenlinien-Interpolation eignet sich besonders zum einfachen Herstellen von Innen- oder Außengewinden mit Formfräsern und zum Fräsen von Schmiernuten.

Dabei setzt sich die Schraubenlinie aus zwei Bewegungen zusammen:

- Kreisbewegung in einer Ebene
- Linearbewegung senkrecht zu dieser Ebene

## Schrittmaß

Verfahrweglängenangabe über Inkrementanzahl (Schrittmaß). Inkrementanzahl kann als → Settingdatum hinterlegt sein bzw. durch entsprechend beschriftete Tasten 10, 100, 1000, 10000 gewählt werden.

### **Schutzraum**

Dreidimensionaler Raum innerhalb des → Arbeitsraumes, in den die Werkzeugspitze nicht hineinreichen darf.

### **Settingdaten**

Daten, die Eigenschaften der Werkzeugmaschine auf durch die Systemsoftware definierte Weise der NC-Steuerung mitteilen.

### **Sicherheitsfunktionen**

Die Steuerung enthält ständig aktive Überwachungen, die Störungen in der → CNC, der Anpass-Steuerung (→ PLC) und der Maschine so frühzeitig erkennen, dass Schäden an Werkstück, Werkzeug oder Maschine weitgehend ausgeschlossen werden. Im Störfall wird der Bearbeitungsablauf unterbrochen und die Antriebe werden stillgesetzt, die Störungsursache gespeichert und als Alarm angezeigt. Gleichzeitig wird der PLC mitgeteilt, dass ein CNC-Alarm ansteht.

### **Skalierung**

Komponente eines → Frames, die achsspezifische Maßstabsveränderungen bewirkt.

### **Softkey**

Taste, deren Beschriftung durch ein Feld im Bildschirm repräsentiert wird, das sich dynamisch der aktuellen Bediensituation anpasst. Die frei belegbaren Funktionstasten (Softkeys) werden softwaremäßig definierten Funktionen zugeordnet.

### **Software-Endschalter**

Software-Endschalter begrenzen den Verfahrbereich einer Achse und verhindern ein Auffahren des Schlittens auf die Hardware-Endschalter. Je Achse sind 2 Wertepaare vorgebar, die getrennt über die → PLC aktiviert werden können.

### **Speicherprogrammierbare Steuerung**

Speicherprogrammierbare Steuerungen (SPS) sind elektronische Steuerungen, deren Funktion als Programm im Steuerungsgerät gespeichert ist. Aufbau und Verdrahtung des Gerätes hängen also nicht von der Funktion der Steuerung ab. Die speicherprogrammierbare Steuerung hat die Struktur eines Rechners; sie besteht aus CPU (Zentralbaugruppe) mit Speicher, Ein-/Ausgabebaugruppen und internem Bus-System. Die Peripherie und die Programmiersprache sind auf die Belange der Steuerungstechnik ausgerichtet.

### **Spiegelung**

Bei Spiegelung werden die Vorzeichen der Koordinatenwerte einer Kontur bezüglich einer Achse vertauscht. Es kann bezüglich mehrerer Achsen zugleich gespiegelt werden.

## Spindelsteigungsfehler-Kompensation

Ausgleich mechanischer Ungenauigkeiten einer am Vorschub beteiligten Kugelrollspindel durch die Steuerung anhand von hinterlegten Messwerten der Abweichungen.

## Spline-Interpolation

Mit der Spline-Interpolation kann die Steuerung aus nur wenigen vorgegebenen Stützpunkten einer Sollkontur einen glatten Kurvenverlauf erzeugen.

## Standardzyklen

Für häufig wiederkehrende Bearbeitungsaufgaben stehen Standardzyklen zur Verfügung:

- Für die Technologie Bohren/Fräsen
- Für die Technologie Drehen

Im Bedienbereich "Programm" werden unter dem Menü "Zyklusunterstützung" die zur Verfügung stehenden Zyklen aufgelistet. Nach Anwahl des gewünschten Bearbeitungszyklus werden die notwendigen Parameter für die Wertzuweisung im Klartext angezeigt.

## Synchronachsen

Synchronachsen benötigen für ihren Weg die gleiche Zeit wie die Geometrieachsen für ihren Bahnweg.

## Synchronaktionen

### 1. Hilfsfunktionsausgabe

Während der Werkstückbearbeitung können aus dem CNC-Programm heraus technologische Funktionen (→ Hilfsfunktionen) an die PLC ausgegeben werden. Über diese Hilfsfunktionen werden beispielsweise Zusatzeinrichtungen der Werkzeugmaschine gesteuert, wie Pinole, Greifer, Spannfutter etc.

### 2. Schnelle Hilfsfunktionsausgabe

Für zeitkritische Schaltfunktionen können die Quittierungszeiten für die → Hilfsfunktionen minimiert und unnötige Haltepunkte im Bearbeitungsprozess vermieden werden.

## Synchronisation

Anweisungen in → Teileprogrammen zur Koordination der Abläufe in verschiedenen → Kanälen an bestimmten Bearbeitungsstellen.

## Syntax

`$SC_IS_SD_MAX_PATH_JERK = Wert`

## Systemspeicher

Der Systemspeicher ist ein Speicher in der CPU, in der folgende Daten abgelegt werden:

- Daten, die das Betriebssystem benötigt
- Die Operanden Zeiten, Zähler, Merker

## Systemvariable

Ohne Zutun des Programmierers eines → Teileprogramms existierende Variable. Sie ist definiert durch einen Datentyp und dem Variablennamen, der durch das Zeichen \$ eingeleitet wird. Siehe → Anwenderdefinierte Variable.

## Teileprogramm

Folge von Anweisungen an die NC-Steuerung, die insgesamt die Erzeugung eines bestimmten → Werkstücks bewirken. Ebenso Vornahme einer bestimmten Bearbeitung an einem gegebenen → Rohteil.

## Teileprogrammsatz

Teil eines → Teileprogramms, durch Line Feed abgegrenzt. Es werden → Hauptsätze und → Nebensätze unterschieden.

## Teileprogrammverwaltung

Die Teileprogrammverwaltung kann nach → Werkstücken organisiert werden. Die Größe des Anwenderspeichers bestimmt die Anzahl der zu verwaltenden Programme und Daten. Jede Datei (Programme und Daten) kann mit einem Namen von maximal 24 alphanumerischen Zeichen versehen werden.

## Text-Editor

Siehe → Editor

## TOA-Bereich

Der TOA-Bereich umfasst alle Werkzeug- und Magazindaten. Standardmäßig fällt der Bereich bzgl. der Reichweite der Daten mit dem Bereich → Kanal zusammen. Über Maschinendaten kann jedoch festgelegt werden, dass sich mehrere Kanäle eine → TOA-Einheit teilen, so dass diesen Kanälen dann gemeinsame WZV-Daten zur Verfügung stehen.

## TOA-Einheit

Jeder → TOA-Bereich kann mehrere TOA-Einheiten enthalten. Die Anzahl der möglichen TOA-Einheiten wird über die maximale Anzahl aktiver → Kanäle begrenzt. Eine TOA-Einheit umfasst genau einen WZ-Daten-Baustein und einen Magazindaten-Baustein. Zusätzlich kann noch ein WZ-Trägerdaten-Baustein enthalten sein (optional).



## Transformation

Additive oder absolute Nullpunktverschiebung einer Achse.

## Unterprogramm

Die Bezeichnung Unterprogramm stammt noch aus der Zeit, als Teileprogramm fest in → Haupt- und Unterprogramme unterteilt waren. Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Prinzipiell kann jedes Teileprogramm oder jeder → Zyklus innerhalb eines anderen Teileprogramms als Unterprogramm aufgerufen werden. Es läuft dann in der nächsten → Programmebene (x+1) (Unterprogrammebene (x+1)) ab.

## Urlöschen

Beim Urlöschen werden folgende Speicher der → CPU gelöscht:

- → Arbeitsspeicher
- Schreib-/Lesebereich des → Ladespeichers
- → Systemspeicher
- → Backup-Speicher

## V.24

Serielle Schnittstelle für die Dateneingabe/-ausgabe. Über diese Schnittstelle können Bearbeitungsprogramme sowie Hersteller- und Anwenderdaten geladen und gesichert werden.

## Variablendefinition

Eine Variablendefinition umfasst die Festlegung eines Datentyps und eines Variablennamens. Mit dem Variablennamen kann der Wert der Variablen angesprochen werden.

## Verfahrbereich

Der maximal zulässige Verfahrbereich bei Linearachsen beträgt  $\pm 9$  Dekaden. Der absolute Wert ist abhängig von der gewählten Eingabe- und Lageregelfeinheit und dem Einheitensystem (inch oder metrisch).

## Vorkoinzidenz

Satzwechsel bereits, wenn Bahnweg um ein vorgegebenes Delta der Endposition nahe gekommen ist.

### Vorschub-Override

Der programmierten Geschwindigkeit wird die aktuelle Geschwindigkeitseinstellung über → Maschinensteuertafel oder von der → PLC überlagert (0-200%). Die Vorschubgeschwindigkeit kann zusätzlich im Bearbeitungsprogramm durch einen programmierbaren Prozentfaktor (1-200%) korrigiert werden.

### Vorsteuerung, dynamisch

Ungenauigkeiten der → Kontur, bedingt durch Schleppfehler, lassen sich durch die dynamische, beschleunigungsabhängige Vorsteuerung nahezu eliminieren. Dadurch ergibt sich auch bei hohen → Bahngeschwindigkeiten eine hervorragende Bearbeitungsgenauigkeit. Die Vorsteuerung kann achsspezifisch über das → Teileprogramm an- und abgewählt werden.

### Werkstück

Von der Werkzeugmaschine zu erstellendes/zu bearbeitendes Teil.

### Werkstückkontur

Sollkontur des zu erstellenden/bearbeitenden → Werkstücks.

### Werkstückkoordinatensystem

Das Werkstückkoordinatensystem hat seinen Ausgangspunkt im → Werkstücknullpunkt. Bei Programmierung im Werkstückkoordinatensystem beziehen sich Maße und Richtungen auf dieses System.

### Werkstücknullpunkt

Der Werkstücknullpunkt bildet den Ausgangspunkt für das → Werkstückkoordinatensystem. Er ist durch Abstände zum → Maschinennullpunkt definiert.

### Werkzeug

An der Werkzeugmaschine wirksames Teil, das die Bearbeitung bewirkt (z. B. Drehmeißel, Fräser, Bohrer, LASER-Strahl ...).

### Werkzeugkorrektur

Berücksichtigung der Werkzeug-Abmessungen bei der Berechnung der Bahn.

### Werkzeugradiuskorrektur

Um eine gewünschte → Werkstückkontur direkt programmieren zu können, muss die Steuerung unter Berücksichtigung des Radius des eingesetzten Werkzeugs eine äquidistante Bahn zur programmierten Kontur verfahren ( $G41/G42$ ).

**WinSCP**

WinSCP ist ein frei verfügbares Open Source-Programm für Windows zum Transferieren von Dateien.

**Zeitreziproker Vorschub**

Anstelle der Vorschubgeschwindigkeit kann für die Achsbewegung auch die Zeit programmiert werden, die der Bahnweg eines Satzes benötigen soll (G93).

**Zoll-Maßsystem**

Maßsystem, das Entfernungen in "inch" und Bruchteilen davon definiert.

**Zwischensätze**

Verfahrbewegungen mit angewählter → Werkzeugkorrektur ( $G_{41}/G_{42}$ ) dürfen durch eine begrenzte Anzahl Zwischensätze (Sätze ohne Achsbewegungen in der Korrektorebene) unterbrochen werden, wobei die Werkzeugkorrektur noch korrekt verrechnet werden kann. Die zulässige Anzahl Zwischensätze, die die Steuerung vorausliest, ist über Systemparameter einstellbar.

**Zyklen**

Geschützte Unterprogramme zur Ausführung von wiederholt auftretenden Bearbeitungsvorgängen am → Werkstück.



# Index

## \$

\$A\_MONIFACT, 1771  
\$AA\_ACC, 1558  
\$AA\_ATOL, 234  
\$AA\_ETRANS, 840  
\$AA\_FGREF, 354  
\$AA\_FGROUP, 354  
\$AA\_MOTEND, 1559  
\$AA\_S, 1388  
\$AA\_SCPAR, 1560  
\$AA\_VLFCT, 1484  
\$AA\_VMAXB, 1484  
\$AA\_VMAXM, 1484  
\$SAC\_ACT\_PROG\_NET\_TIME, 698  
\$SAC\_ACTUAL\_PARTS, 706  
\$SAC\_ASUP, 632  
\$SAC\_AUXFU\_EXT, 473  
\$SAC\_AUXFU\_M\_EXT, 473  
\$SAC\_AUXFU\_M\_STATE, 474  
\$SAC\_AUXFU\_M\_TICK, 451  
\$SAC\_AUXFU\_M\_VALUE, 473  
\$SAC\_AUXFU\_PREDEF\_INDEX, 457, 473  
\$SAC\_AUXFU\_SPEC, 470, 473  
\$SAC\_AUXFU\_STATE, 474  
\$SAC\_AUXFU\_TICK, 458  
\$SAC\_AUXFU\_TYPE, 473  
\$SAC\_AUXFU\_VALUE, 473  
\$SAC\_CONSTCUT\_S, 1476  
\$SAC\_CTOL, 234  
\$SAC\_CUT\_INV, 1700  
\$SAC\_CUTMOD, 1700  
\$SAC\_CUTMOD\_ANG, 1693, 1700  
\$SAC\_CUTTING\_TIME, 701  
\$SAC\_CYCLE\_TIME, 701  
\$SAC\_FGROUP\_MASK, 354  
\$SAC\_OLD\_PROG\_NET\_TIME, 698  
\$SAC\_OLD\_PROG\_NET\_TIME\_COUNT, 699  
\$SAC\_OPERATING\_TIME, 701  
\$SAC\_OTOL, 234  
\$SAC\_PATHACC, 258, 271  
\$SAC\_PATHJERK, 270, 271  
\$SAC\_PROG\_NET\_TIME\_TRIGGER, 699  
\$SAC\_REQUIRED\_PARTS, 706  
\$SAC\_SGEAR, 1436, 1459  
\$SAC\_SMAXACC, 1483  
\$SAC\_SMAXACC\_INFO, 1483  
\$SAC\_SMAXVELO, 1483  
\$SAC\_SMAXVELO\_INFO, 1483  
\$SAC\_SMINVELO, 1483  
\$SAC\_SMINVELO\_INFO, 1483  
\$SAC\_SPECIAL\_PARTS, 706  
\$SAC\_SPIND\_STATE, 1483  
\$SAC\_STOLF, 238  
\$SAC\_TOTAL\_PARTS, 706  
\$SAN\_AUXFU\_LIST\_CHANNO, 459  
\$SAN\_AUXFU\_LIST\_ENDINDEX, 462  
\$SAN\_AUXFU\_LIST\_GROUPINDEX, 459  
\$SAN\_POWERON\_TIME, 697  
\$SAN\_SETUP\_TIME, 697  
\$C\_AUX\_EXT, 677  
\$C\_AUX\_IS\_QUICK, 677  
\$C\_AUX\_VALUE, 677  
\$C\_D, 677  
\$C\_D\_PROG, 677  
\$C\_DL, 677  
\$C\_DL\_PROG, 677  
\$C\_DUPLO, 677  
\$C\_DUPLO\_PROG, 677  
\$C\_M, 677  
\$C\_M\_PROG, 677  
\$C\_ME, 677  
\$C\_T, 677  
\$C\_T\_PROG, 677  
\$C\_TCA, 677  
\$C\_TE, 677  
\$C\_THNO, 677  
\$C\_THNO\_PROG, 677  
\$C\_TS, 677  
\$C\_TS\_PROG, 677  
\$P\_AD, 1701  
\$P\_CHANNO, 609  
\$P\_CONSTCUT\_S, 1476  
\$P\_CTOL, 235  
\$P\_CUT\_INV, 1700  
\$P\_CUTMOD, 1700  
\$P\_CUTMOD\_ANG, 1693, 1700  
\$P\_FGROUP\_MASK, 354  
\$P\_GEAR, 1459  
\$P\_ISTEST, 512  
\$P\_OTOL, 235  
\$P\_PROG\_EVENT, 609  
\$P\_REPINF, 627  
\$P\_SEARCH\_S, 453, 527

\$P\_SEARCH\_SDIR, 453, 527  
 \$P\_SEARCH\_SGEAR, 453, 527, 1436  
 \$P\_SEARCH\_SPOS, 453, 527  
 \$P\_SEARCH\_SPOSMODE, 453, 527  
 \$P\_SGEAR, 1436  
 \$P\_SIM, 520  
 \$P\_STOLF, 238  
 \$P\_SUB\_AUTOGEAR, 687  
 \$P\_SUB\_AXFCT, 687  
 \$P\_SUB\_CA, 687  
 \$P\_SUB\_GEAR, 687  
 \$P\_SUB\_LA, 687  
 \$P\_SUB\_M19, 687  
 \$P\_SUB\_SPOS, 687  
 \$P\_SUB\_SPOSA, 687  
 \$P\_SUB\_SPOSIT, 687  
 \$P\_SUB\_SPOSMODE, 687  
 \$P\_SUB\_STAT, 675  
 \$P\_TOOLENV, 1752  
 \$P\_TOOLENVN, 1752  
 \$P\_WORKAREA\_CS\_COORD\_SYSTEM, 121  
 \$PA\_ATOL, 235  
 \$PA\_FGREF, 354  
 \$PA\_FGROUP, 354  
 \$SC\_IS\_SD\_MAX\_PATH\_ACCEL, 256  
 \$SC\_IS\_SD\_MAX\_PATH\_JERK, 269  
 \$SC\_SD\_MAX\_PATH\_ACCEL, 255  
 \$SC\_SD\_MAX\_PATH\_JERK, 269  
 \$TC\_DP1, 1690  
 \$TC\_DP10, 1692  
 \$TC\_DP11, 1691  
 \$TC\_DP2, 1690, 1695  
 \$TC\_DP24, 1692  
 \$TC\_DPCE[t,d], 1578  
 \$TC\_DPNT, 1513  
 \$VA\_ABSOLUTE\_ENC\_DELTA\_INIT, 111  
 \$VA\_ENC\_ZERO\_MON\_ERR\_CNT, 108, 111  
 \$VA\_TORQUE\_AT\_LIMIT, 325  
 \$VC\_SGEAR, 1436

## A

Ablage von Winkeln in den Werkzeugträgerdaten, 1661  
 ACC, 1469, 1557  
 ACC[Achse], 253  
 Achse/Spindel steht, 50  
 Achsensperre, 40, 1535  
 Achsklemmung, 93  
 Achskonfiguration, 742  
 Achsüberwachungen, 85  
   Achse-/Spindelspezifische Settingdaten, 162  
   Drehzahlsollwert, 101

Istgeschwindigkeit, 103  
 Randbedingungen, 145  
 Schleppabstand, 87  
 Stillstand, 91  
 ACN, 1468  
 ACP, 1468  
 Adaptermaß, 1608  
 ADDFRAME, 839  
 Adress  
   -bereiche, 1202  
   -eingabe, absolute, 1217  
   -eingabe, symbolische, 1217  
 Adressierung  
   absolute, 1205  
   direkte, 1204, 1217  
   indirekte, 1204, 1217  
   symbolische, 1205  
 AG\_SEND, AG\_RECV, 951  
 Aktions-Einzelsatz, 514  
 Aktionssätze, 524  
 Aktivierung  
   von Maschinensteuertafel, Bedienhandgerät, 962  
 Aktualwert, 1211  
 aktuelle Bildnummer von JobShop-Oberfläche, 1804  
 Alarm bei Vorlaufstopp, 1651  
 ALF, 626, 1524  
 Anlauf und Synchronisation NCK-PLC, 942  
 Anpassfaktor  
   der Bahndynamik, 211  
 Anpassungen der Motor/Last-Verhältnisse, 361  
 Antrieb ready, 36  
 Antriebe im zyklischenBetrieb, 36  
 Antriebstest Fahr Anforderung, 50  
 Antriebstest Fahrfreigabe, 40  
 Anwahl der Schneide bei WZ-Wechsel, 1572  
 Anwender  
   -datenbaustein, 1204  
   -nahtstelle, 1204  
 ANY, 1174  
 ANY im FB, 1175  
 ANY im FC, 1174  
 Anzeigefeinheit, 56, 335  
 Anzeigesatz, Aufbau (DIN), 648  
 Arbeitsfeldbegrenzung, 115  
   im BKS, 117  
   im WKS/ENS, 120  
 Arbeitsfeldbegrenzungsgruppe, 120  
 ASUP  
   Aktivierung, 618  
   interne, 630  
   nach Satzsuchlauf mit Berechnung, 525  
   Priorität, 626

- Reorganisation, 619
  - SERUPRO-Ende, 462
  - ATOL, 232
  - ATRANS, 725
  - Ausblendeebenen, 641
  - Ausgabe
    - sequenz, 458
    - verhalten einer Hilfsfunktion, 423
    - zähler, 458
  - Autarke Einzelachsvorgänge, 572
  - AUXFUDEL, 463
  - AUXFUDELG, 464
  - AUXFUMSEQ, 451
  - AUXFUSYNC, 462
- B**
- BAG, 943
  - Bahn
    - geschwindigkeit, Maximale, 1510
    - vorschub F, 1509
  - Bahnachsen, 737
  - Bahninterpolator, 506
  - Bahnsteuerbetrieb, 176
    - Impliziter, 180
  - Bahnvorschub, 333
  - Basis-Anzeige
    - Größe des Anzeigebuffers, 646
  - Basiskoordinatensystem (BKS), 723, 754
  - Basismaß, 1608
  - Basis-Satzanzeige
    - aktivieren, 646
    - konfigurieren, 646
  - Bearbeitung in Richtung der  
Werkzeugorientierung, 1704
  - Bearbeitungszeit, 701
  - Benutzerdefinierten ASUP
    - nach SERUPRO-Vorgang, 539
  - Beschreibung einer Drehung, 1671
  - Besonderheiten im Zielsatz
    - STOPPRE-Satz, 565
  - Betriebsart
    - AUTOMATIC, 498
    - JOG, 498
    - JOG in AUTOMATIC, 498
    - MDA, 498
  - Betriebsarten
    - der Betriebsartengruppe, 498
    - Prioritäten, 499
    - übergreifende Synchronaktionen, 499
    - Überwachungen, 503
    - Verriegelungen, 503
    - wechsel, 498, 504
  - Betriebsartengruppe (BAG), 493
    - Anwendernahtstelle, 495
    - Kanalspezifische Zuordnungen, 494
    - Konfigurationsänderung der  
Betriebsartengruppe, 495
    - Nummer, 495
  - Betriebsartenwechsel
    - von/nach Betriebsarten AUTOMATIK, JOG,  
MDA, 504
  - Betriebszustände, 500
  - Bewegungsendekriterium
    - bei Einzelachsen, 1559
  - Bewertung einzelner Verschleißkomponenten, 1746
  - Bezugsachse
    - für G96 / G961 / G962, 893
  - Bildschirm dunkel steuern, 37
  - Bildschirm hell, 37
  - BLSYNC, 626
  - BRISK, 250
  - BRISKA, 250
- C**
- Cancelalarme, 38
  - CFC, 1510
  - CFCIN, 1510
  - CFINE, 725
  - CFTCP, 1510
  - CLRINT, 628
  - COMPCAD, 226
  - COMPCURV, 226
  - COMPON, 226
  - CTOL, 232
  - CTRANS, 725
- D**
- D/DL-Funktionsersetzung, 674
  - Darstellungsweise der Programmanzeige, 645
  - Daten
    - baustein, 1194, 1211
    - format, 1256
    - klassen, 1193, 1226
    - typ, 1207
  - Datenaustausch
    - mit Bedientafel, 922
  - Datenkanal, schneller, 61
  - Datenschnittstelle, 926
  - Datentyp
    - prüfung, 1207

DB10

- DBB4-7, 888
- DBX103.0, 36, 1791
- DBX103.5, 1792
- DBX103.6, 1792
- DBX103.7, 1792
- DBX104.3, 965
- DBX104.4, 965
- DBX104.7, 36, 947, 1792
- DBX106.1, 887, 889, 890, 1886
- DBX107.7, 349
- DBX108.3, 36, 1793
- DBX108.5, 36, 1793
- DBX108.6, 36, 1793
- DBX108.7, 36, 1794
- DBX109.0, 37, 1794
- DBX109.5, 1795
- DBX109.6, 36, 1795
- DBX109.7, 36, 1796
- DBX180.2, 1793
- DBX56.1, 46, 581, 886, 888, 890, 1885
- DBX56.2, 886, 889, 1885
- DBX56.4, 68, 1791
- DBX56.5, 68, 1791
- DBX56.6, 68, 1791
- DBX56.7, 68, 1791
- DBX92.0, 946
- DBX92.1, 946

DB11

- DBX0.0, 499, 501, 1851
- DBX0.1, 499, 1321, 1324, 1851
- DBX0.2, 499, 1321, 1324, 1852
- DBX0.4, 505, 1852
- DBX0.5, 496, 1853
- DBX0.6, 496, 1853
- DBX0.7, 497, 584, 889, 1323, 1325, 1326, 1854
- DBX07.7, 581
- DBX1.0, 1854
- DBX1.0 - DBX1.2, 500
- DBX1.1, 1855
- DBX1.2, 1321, 1324, 1855
- DBX1.6, 639, 1856
- DBX1.7, 639, 1856
- DBX26.4, 502
- DBX26.5, 502
- DBX4.0, 1857
- DBX4.0 - DBX4.2, 499
- DBX4.1, 1857
- DBX4.2, 1857
- DBX4.4, 581
- DBX46.4, 502
- DBX46.5, 502

- DBX5.0, 1858
- DBX5.0 - DBX5.2, 500
- DBX5.1, 1858
- DBX5.2, 1858
- DBX6.0, 499, 502, 1859
- DBX6.1, 499, 1859
- DBX6.2, 499, 1859
- DBX6.3, 497, 887, 890, 1860
- DBX6.4, 502
- DBX6.5, 502
- DBX6.7, 497, 1860
- DBX7.0, 502, 1860
- DBX7.0 - DBX7.2, 500
- DBX7.1, 1861
- DBX7.2, 1861

DB1600, 1188

DB19

- DBB10, 1799
- DBB13, 38, 39
- DBB15, 1801
- DBB16, 38, 39
- DBB17, 38, 39, 1801
- DBB22, 1804
- DBB24, 1804
- DBB26, 39
- DBB27, 39
- DBB6, 1799
- DBB7, 1799
- DBB8, 1799
- DBX 0.3, 38
- DBX 0.4, 38
- DBX0.0, 37, 1796
- DBX0.1, 37, 38, 1797
- DBX0.2, 37, 38, 1797
- DBX0.3, 1798
- DBX0.4, 1798
- DBX0.7, 38, 1798
- DBX13.5, 1799
- DBX13.6, 1800
- DBX13.7, 1800
- DBX14.0 - DBX14.6, 1800
- DBX14.7, 1800
- DBX16.6, 1801
- DBX16.7, 1801
- DBX20.1, 1803
- DBX20.3, 38, 1798, 1803
- DBX20.4, 38, 1798, 1803
- DBX20.6, 1804
- DBX20.7, 1799, 1804
- DBX26.1, 1805
- DBX26.2, 1805
- DBX26.3, 1805



- DBX26.5, 1806  
 DBX26.6, 1806  
 DBX26.7, 1806, 1807  
 DBX44.0, 1802  
 DBX45.0, 1802  
 DBX45.1, 1802  
 DBX45.2, 1802  
 DBX45.3, 1802  
 DB21  
   DBX378.1, 625  
 DB21, ...  
   D35.5, 582, 583  
   DBB116 - DBB136, 443  
   DBB118, 1847  
   DBB129, 1847  
   DBB140 - DBB157, 1848  
   DBB140 - DBB190, 443  
   DBB158 - DBB193, 1848  
   DBB194, 1470  
   DBB194 - DBB206, 443, 1849  
   DBB2, 1863  
   DBB202, 1470  
   DBB208 - DBB271, 1876  
   DBB317.1, 1876  
   DBB376, 603, 1880  
   DBB4, 1536, 1913  
   DBB5, 1536, 1915  
   DBB58, 1844  
   DBB58 - DBB67, 443  
   DBB60 - DBB64, 1845  
   DBB60 - DBB65, 1844  
   DBB66 - DBB67, 1845  
   DBB68 - DBB112, 443  
   DBB68 - DBB97, 1846  
   DBB68ff., 1470  
   DBB98 - DBB115, 1846  
   DBX0.4, 514, 1862  
   DBX0.5, 1862  
   DBX0.6, 517, 1541, 1912  
   DBX1.0, 1324, 1888  
   DBX1.1, 1833  
   DBX1.6, 523, 526, 1862  
   DBX1.7, 512, 581, 1863  
   DBX10.0 - DBX11.1, 135, 1834  
   DBX12.3, 1534, 1919  
   DBX16.3, 1534, 1919  
   DBX2.0, 518, 582  
   DBX2.0 - 7, 642  
   DBX20.3, 1534, 1919  
   DBX24.6, 516, 1541, 1919  
   DBX25.3, 1920  
   DBX25.7, 511  
   DBX26.0, 518  
   DBX272.0 - DBX273.1, 1834  
   DBX274.0 - DBX275.1, 1835  
   DBX276.0 - DBX277.1, 138, 1835  
   DBX278.0 - DBX279.1, 1836  
   DBX29.0, 1550  
   DBX29.0 - DBX29.3, 1920  
   DBX29.1, 1550  
   DBX29.2, 1550  
   DBX29.3, 1550  
   DBX30.5, 434, 1844  
   DBX31.0 - DBX31.2, 545, 546, 547, 1866  
   DBX31.0-31.2, 553  
   DBX31.4, 545, 546, 547, 548, 1866  
   DBX31.6 - 7, 642  
   DBX317.1,  
   DBX318.0, 620, 1877  
   DBX318.1, 1878  
   DBX318.5, 434, 1849  
   DBX318.6, 1542  
   DBX319.0, 546, 547, 548, 1878  
   DBX319.1 - DBX319.3, 546, 547, 548, 549, 1878  
   DBX319.5, 547, 551, 1880  
   DBX32.3, 523, 1867  
   DBX32.4, 523, 524, 1867  
   DBX32.5, 1868  
   DBX32.6, 454, 523, 1869  
   DBX33.0, 1324, 1889  
   DBX33.4, 523, 1869  
   DBX33.5, 1870  
   DBX33.6, 1871  
   DBX33.7, 512, 1871  
   DBX35.0, 582, 585, 1871  
   DBX35.0 - DBX35.4, 625  
   DBX35.1, 585, 1872  
   DBX35.2, 585, 1872  
   DBX35.3, 515, 585, 1873  
   DBX35.4, 585, 603, 1873  
   DBX35.5, 582, 586, 1873  
   DBX35.5 - DBX35.7, 625  
   DBX35.6, 581, 586, 1874  
   DBX35.7, 581, 584, 586, 603, 1874  
   DBX36.2, 1322, 1325, 1889  
   DBX36.3, 1839  
   DBX36.4, 1874  
   DBX36.5, 1875  
   DBX36.6, 37, 526, 1794, 1809  
   DBX36.7, 37, 526, 1794, 1809  
   DBX37.6, 1875  
   DBX37.7, 1875  
   DBX378.0, 1880  
   DBX378.1, 1881

- DBX384.0, 591, 1881
- DBX4.3, 1535
- DBX59.0 - DBX59.4, 1845
- DBX6.0, 1534, 1917
- DBX6.1, 622, 1393, 1429, 1863
- DBX6.2, 39, 1808
- DBX6.4, 1864
- DBX6.6, 1536, 1917
- DBX6.7, 1536, 1918
- DBX7.0, 581, 1864
- DBX7.1, 512, 524, 584, 1864
- DBX7.2, 581, 582, 1865
- DBX7.3, 581, 582, 1865
- DBX7.4, 581, 582, 1431, 1482, 1865
- DBX7.5, 510
- DBX7.7, 581, 584, 889, 1323, 1325, 1326, 1866
- DBX8.0 - DBX9.1, 135, 1833
- DB21, ...
  - DBX378.0, 618
- DB31, ...
  - DBB0, 1538, 1921
  - DBB1.5, 1370
  - DBB1.6, 1370
  - DBB19, 1538, 1925
  - DBB2.1, 1370
  - DBB60.4, 1351, 1362, 1365
  - DBB60.5, 1351, 1362, 1365
  - DBB68ff., 1470
  - DBB78 - DBB81, 1927
  - DBB86, 1908
  - DBB88, 1908
  - DBD78, 1850
  - DBD86, 1850
  - DBD88, 1850
  - DBW134, 1910
  - DBW136, 1910
  - DBX1.0, 40, 50, 1810
  - DBX1.1, 309, 312, 1841
  - DBX1.2, 308, 1842
  - DBX1.3, 40, 309, 1482, 1535, 1810
  - DBX1.4, 41, 42, 44, 47, 93, 1356, 1357, 1813
  - DBX1.5, 45, 106, 123, 356, 1326, 1419
  - DBX1.5 - DBX1.6, 1814
  - DBX1.6, 45, 106, 123, 356, 1326, 1419
  - DBX1.7, 1538, 1923
  - DBX10.0, 542, 544, 545, 546, 547, 548, 553, 1881
  - DBX12.0, 112
  - DBX12.0 - DBX12.1, 1837
  - DBX12.1, 112
  - DBX12.2, 113
  - DBX12.2 - DBX12.3, 1837
  - DBX12.3, 113
  - DBX12.7, 1332, 1335, 1890
  - DBX132.0, 1908
  - DBX132.1, 1909
  - DBX132.4, 1909
  - DBX132.5, 1909
  - DBX138.4, 1911
  - DBX138.5, 1911
  - DBX16.0 - DBX16.2, 1389, 1425, 1430, 1431, 1432, 1438, 1447, 1451, 1894
  - DBX16.3, 1385, 1389, 1425, 1431, 1432, 1445, 1447, 1451, 1455, 1895
  - DBX16.4, 1419
  - DBX16.4 - DBX16.5, 1896
  - DBX16.5, 1419
  - DBX16.7, 1482, 1896
  - DBX17.0, 1454
  - DBX17.4 - DBX17.5, 1450, 1897
  - DBX17.6, 1462, 1897
  - DBX18.4, 1443, 1445, 1451, 1898
  - DBX18.5, 1389, 1425, 1426, 1443, 1444, 1445, 1447, 1451, 1482, 1899
  - DBX18.6, 1445
  - DBX18.6 - DBX18.7, 1900
  - DBX18.7, 1445
  - DBX2.1, 41, 42, 44, 46, 123, 309, 1326, 1356, 1357, 1450, 1816
  - DBX2.2, 48, 94, 524, 543, 1388, 1397, 1431, 1482, 1817, 1893
  - DBX2.3, 93, 100, 1836
  - DBX2.4 - DBX2.7, 1340, 1890
  - DBX20.1, 1432, 1819
  - DBX21.0, 51
  - DBX21.0 - DBX21.2, 1819
  - DBX21.1, 51
  - DBX21.2, 51
  - DBX21.3, 52
  - DBX21.3 - DBX21.4, 1820
  - DBX21.4, 52
  - DBX21.5, 52, 1820
  - DBX21.6, 52, 1821
  - DBX21.7, 46, 53, 312, 1821
  - DBX28.7, 509
  - DBX3.0, 840, 880, 1884
  - DBX3.1, 308, 311, 312, 1842
  - DBX3.2, 1550
  - DBX3.2 - DBX3.5, 1924
  - DBX3.3, 1550
  - DBX3.4, 1550
  - DBX3.5, 1550
  - DBX3.6, 1836
  - DBX3.7, 572
  - DBX30.0, 1472

- DBX30.1, 1472  
 DBX30.2, 1472  
 DBX30.3, 1472  
 DBX30.4, 1472  
 DBX39.0, 141  
 DBX4.3, 1445, 1482, 1924  
 DBX4.6, 1321  
 DBX4.7, 1321  
 DBX60.0, 1900  
 DBX60.2, 106, 1487  
 DBX60.2 - DBX60.3, 1838  
 DBX60.3, 106, 1487  
 DBX60.4, 44, 123, 1322, 1323, 1325, 1326, 1340, 1348, 1353, 1356, 1357, 1487, 1891  
 DBX60.4 - DBX60.5, 1439, 1446, 1447, 1450, 1451  
 DBX60.5, 44, 123, 890, 1322, 1323, 1325, 1326, 1340, 1348, 1353, 1356, 1357, 1487, 1891  
 DBX60.6, 45, 91, 100, 1393, 1401, 1405, 1489, 1839  
 DBX60.7, 45, 91, 100, 1401, 1405, 1455, 1489, 1840  
 DBX61.0, 40, 50, 1822  
 DBX61.3, 41, 42, 50, 1355, 1356, 1357, 1823  
 DBX61.4, 49, 50, 1356, 1388, 1389, 1395, 1425, 1480, 1823  
 DBX61.5, 46, 47, 50, 1387, 1446, 1450, 1824  
 DBX61.6, 47, 50, 1825  
 DBX61.7, 50, 1825  
 DBX62.2, 1514, 1927  
 DBX62.4, 308, 311, 312, 1843  
 DBX62.5, 309, 312, 315, 1843  
 DBX64.6, 91, 448  
 DBX64.7, 91, 448  
 DBX68.0 - DBX68.3, 1473  
 DBX69.0, 51, 77  
 DBX69.0 - DBX69.2, 1438, 1825  
 DBX69.1, 51, 77  
 DBX69.2, 51, 77  
 DBX7.7, 1482  
 DBX70.0, 547, 550, 1882  
 DBX70.1, 547, 549, 550, 1882  
 DBX70.2, 547, 548, 549, 1883  
 DBX71.4, 1420, 1892  
 DBX71.5, 1420, 1892  
 DBX72.0, 547, 1883  
 DBX76.0, 51, 1826  
 DBX76.4, 547, 551, 1883  
 DBX82.0 - DBX82.2, 1425, 1427, 1429, 1431, 1442, 1446, 1450, 1453, 1901  
 DBX82.3, 1425, 1427, 1429, 1443, 1446, 1447, 1450, 1453, 1902  
 DBX83.0, 1902  
 DBX83.1, 1389, 1400, 1428, 1429, 1481, 1482, 1903  
 DBX83.2, 1428, 1429, 1481, 1482, 1904  
 DBX83.5, 1389, 1393, 1395, 1396, 1481, 1904  
 DBX83.7, 1905  
 DBX84.0, 1476  
 DBX84.3, 1906  
 DBX84.5, 1450, 1455, 1906  
 DBX84.6, 1385, 1443, 1446, 1450, 1453, 1455, 1906  
 DBX84.7, 1387, 1426, 1451, 1907  
 DBX85.5, 1406, 1907  
 DBX9.0, 49, 51, 76  
 DBX9.0 - DBX9.2, 1818  
 DBX9.1, 49, 51, 76  
 DBX9.2, 49, 51, 76  
 DBX9.3, 50, 1818  
 DBX92.1, 53, 1826  
 DBX93.0, 51, 53  
 DBX93.0 - DBX93.2, 1827  
 DBX93.1, 51, 53  
 DBX93.2, 51, 53  
 DBX93.3, 53  
 DBX93.3 - DBX93.4, 1827  
 DBX93.4, 53  
 DBX93.5, 36, 46, 53, 1793, 1828  
 DBX93.6, 52, 54, 1828  
 DBX93.7, 53, 54, 1829  
 DBX94.0, 54, 1829  
 DBX94.1, 54, 1830  
 DBX94.2, 55, 1830  
 DBX94.3, 55, 1831  
 DBX94.4, 55, 1831  
 DBX94.5, 55, 1831, 1832  
 DBX94.6, 55, 1832  
 DB31, ...  
     DBX9.3, 1560  
 DB4500, 1189  
 DB9000 - DB9063, 1189  
 DC, 1468  
 Default-Kennwort, 67  
 Dekodier-Einzelsatz, 514  
 DELTOOLENV, 1751  
 D-Funktionen, 400, 1572  
 DIACYCOFA, 899  
 DIAM90, 898  
 DIAM90/DIAM90A[AX], 901  
 DIAM90A, 899  
 DIAMCHAN, 899  
 DIAMCHANA, 899  
 DIAMCYCOF, 898  
 DIAMCYCOF/DIACYCOFA[AX], 901

DIAMOF, 898  
DIAMOF A, 899  
DIAMON, 898  
DIAMON/DIAMONA[AX], 900  
DIAMONA, 899  
DILF, 1524  
Direkttasten  
    Adresse, 965  
    Alarmer, 965  
    OPs am Ethernet-Bus, 964  
DISABLE, 628  
DISC, 1638  
DITE, 1520  
DITS, 1520  
DL-Funktionen, 401  
D-Nummern  
    Vergabe von freien ..., 1577  
D-Nummern-Struktur  
    - flache (ohne WZV), 1589  
Drehzahl  
    -vorsteuerung, 383  
Drehzahlbegrenzung bei G96, G961, 1517  
Drehzahlregelkreis, 377  
Drehzahlregler aktiv, 50  
Drehzahlsollwertanpassung, 365  
Drehzahlsollwertausgabe, 365  
Drehzahlvorgabe, 1475  
DRIVE, 286  
DRIVEA, 286  
Dynamik  
    -anpassung, 210  
Dynamikanpassung, 379  
DYNFINISH, 219  
DYNNORM, 219  
DYNPOS, 219  
DYNROUGH, 219  
DYNSEMIFIN, 219

## E

Eilgang  
    -korrektur, 1536  
Eingabefeinheit, 335  
Eingeschränkte Orientierung des  
Werkzeuträgers, 1660  
Einzelachsdynamik, 1556  
Einzelatz  
    -betrieb, 513  
    Kanal-Klassifizierung, 639  
    SBL1, 633  
    SBL2, 633  
    SBL2 mit impliziten Vorlaufstop, 634

SBL3, 633  
    situationsabhängig nicht anhalten, 638  
    Stopp-Unterdrückung, 634  
    -Unterdrückung beim gestarteten ASUP, 635  
    -Unterdrückung im ASUP wieder aktivieren, 638  
    -Unterdrückung im Programm SBLOF, 635  
ENABLE, 628  
Endschalter-Überwachung, 112  
Ereignisgesteuerte Programmabläufe, 599  
Ersatznullmarke, 1367  
Ethernet  
    -Ankopplung, 963  
Externe Nullpunktverschiebung, 903  
Externe Programmspeicher, 652  
Externes Unterprogramm Abarbeiten, 652

## F

FA, 1469, 1532  
FA-Funktionen, 402  
Fahren auf Festanschlag, 303  
    Abwahl, 312  
    Anwahl, 308  
    Festanschlag wird erreicht, 308  
    Festanschlag wird nicht erreicht, 310  
    Funktionsabbruch, 317  
    Konturüberwachung, 318  
    Positionierachsen, 318  
    Satzsuchlauf, 313  
    Überwachungsfenster, 309  
FB, 1555  
FB1 RUN\_UP (Grundprogramm, Anlaufteil), 999  
FB10 Sicherheits-Relais, 1072  
FB11 Bremsentest, 1075  
FB2 GET (NC-Variable lesen), 1008  
FB29 Diagnose (Signalrekorder und  
Datentrigger), 1081  
FB3 PUT (NC-Variable schreiben, 1016  
FB4 PI\_SERV (PI-Dienste), 1025  
    Verfügbare PI-Dienste, 1028, 1029  
FB5 GETGUD (GUD-Variable lesen), 1057  
FB7 PI\_SERV2 (PI Dienste), 1065  
FB9 MzuN (Bedieneinheitenumschaltung), 1066  
FC10 AL\_MSG, 1107  
FC1005 AG\_SEND, 1166  
FC1006 AG\_RECV, 1168  
FC12 AUXFU, 1109  
FC13 BHGDisp, 1111  
FC17, 1116  
FC18 SpinCtrl, 1119  
FC19 MCP\_IFM, 1131  
FC2 GP\_HP (Grundprogramm, zyklischer Teil), 1085

- FC21 Transfer, 1138  
 FC22 TM\_DIR, 1146  
 FC24 MCP\_IFM2, 1149  
 FC25 MCP\_IFT, 1153  
 FC3 GP\_PRAL (Grundprogramm, alarmgesteuerter Teil), 1087  
 FC3 GP\_PRAL (Grundprogramm, Diagnose), 1090  
 FC6 TM\_TRANS2, 1092  
 FC7 TM\_REV, 1093  
 FC8 TM\_TRANS, 1097  
 FC9 ASUP, 1104  
 FCUB, 1551  
 Feininterpolation, 378  
 Feinverschiebung, 725  
 Ferndiagnose, 36  
 Festvorschübe, 1549  
 F-Funktionen, 401  
 FGROU, 737, 740  
 FIFO-Puffer, 653  
 Firmware, 1185  
 FL, 1510  
 Flache D-Nummern-Struktur, 1589  
 FLIN, 1551  
 FNORM, 1551  
 FOC, 304  
 FOCOF, 304  
 FOCON, 304  
 Format  
   -kennzeichen, 1210  
 FPO, 1551  
 FPRAOF, 1468  
 FPRAON, 1468  
 FRAME, 731  
 Framedrehungen, 726  
   in Werkzeugrichtung, 848, 850  
   mit Raumwinkeln, 846  
 FRC, 1552  
 FRCM, 1552  
 Freiformflächen, 223  
   -modus, 168, 222  
 Freischneidwinkel, 1609  
 Freiwinkel, 1691  
 Funktionenwahl über Bedientafelfront oder über PLC, 640  
 Funktionsschnittstelle, 927  
 Funktionsübersicht Inch/Metrisch Umschaltung  
   Datensicherung, 351  
   Runden von Maschinendaten, 351  
   Synchronaktionen, 343  
 FXS, 304  
 FXS-REPOS, 316  
 FXST, 304  
 FXSW, 304  
 FZ, 1513
- ## G
- G0-Toleranzfaktor, 236  
 G25, 118, 1469  
 G26, 118, 1469  
 G33, 1518  
 G331, 1529  
 G332, 1529  
 G40, 1613  
 G41, 1613  
 G42, 1613  
 G450/G451, 1637  
 G451, 1640  
 G460, 1653  
 G461, 1569, 1654  
 G462, 1569, 1655  
 G58, 725  
 G59, 725  
 G60, 171  
 G601, 173  
 G602, 173  
 G603, 173  
 G63, 1531  
 G64, 179  
 G642, 185  
 G643, 185  
 G644, 189  
 G645, 193  
 G9, 171  
 G91 Erweiterung, 1703  
   Nullpunktverschiebung, 1704  
 G93, 1511  
 G94, 1512  
 G95, 1512  
 G96, 1514  
 G961, 1514  
 G97, 1516  
 G971, 1516  
 Geber Codierung, 368  
 Geber direkt am Werkzeug, 363  
 Geberüberwachungen, 106  
   Geberfrequenz, 106  
 Genauhalt, 171  
   Impliziter, 177  
 Genauhaltkriterien, 172, 173  
 Geometrieachsen, 731, 737, 754  
 Geometrieachsen beim Handradfahren, 903  
 Geschwindigkeiten, 332  
 Getriebestufe

- bei M70, 1456
- Manuelle Vorgabe, 1430
- Vorgabe durch PLC, 1429
- Getriebestufe vorgeben, 1423
- Getriebestufen, 1421
- Getriebestufenauswahl
  - Automatische, 1489
- GETTCOR, 1754
  - Kompatibilität, 1760
  - orientierbarer Werkzeugträger, 1758
- GETTENV, 1753
- Gewinde
  - schneiden, 1524
- Gewindeschneiden G33, 1518
- G-Gruppen, 578
- Glättung
  - der Bahngeschwindigkeit, 206
- GOTOS, 593
- Grenz
  - geschwindigkeit, für Bahnachsen, 1510
- Grobverschiebung, 725
- Grundorientierung der Werkzeuge, 1708

## H

- Halterwinkel, 1691
- Hängende Achsen, 318
- Hardware-Endschalter, 112
- Hauptlauf
  - Achsen, 1558
- Hauptlaufachsen, 739
- Helixinterpolation, 741
- H-Funktionen, 399
- Hilfsfunktion
  - Adresserweiterung, 422
  - anwenderdefiniert, 395
  - Assoziierte, 433
  - Ausgabeverhalten, 423
  - Definition, 396
  - Typ, 422
  - vordefiniert, 395
  - Wert, 423
- Hilfsfunktionen
  - Anwenderspezifische, 429
  - Vordefinierte, 404
- Hilfsfunktions
  - zähler, 458
- Hilfsfunktionsausgabe, 577
- HMI - CPU ab BTSS eady, 36
- Hochlaufvorgang beendet, 55

## I

- IC, 1468
- Identitätsvergleich, 708
- Impliziter Bahnsteuerbetrieb, 180
- Impliziter Genauhalt, 177
- impliziter Vorlaufstopp, 566
- Inkrementell programmierte Korrekturwerte, 1703
- interpolarischer Achsverbund, 47
- Interpolator-Ende, 172
- Interrupt
  - routine, 616
  - signal, 617
  - sperre, 628
- Interruptroutine
  - Ende, 619
- Istwert synchronisieren, 48
- Istwertauflösung, 367
- Istwerte in Werkstückkoordinatensystem, 38
- Istwerterfassung, 355
- Istwertkorrektur, 357
- Istwertsystem
  - werkstücknahes, 863
- Istwertverarbeitung, 365

## J

- Joggen
  - in Betriebsart AUTOMATIK, 501
  - während Unterbrechung eines JOG-ASUP, 623
- JOG-Retract, 665

## K

- Kanal
  - Aktueller, 609
  - Bahninterpolator, 506
  - Eigenschaften, 506
  - Grundeinstellungen, 577
  - Konfiguration, 507
  - Zustand anzeigen, 586
  - Zustände, 586
- Kanalachsen, 731
- Kanalspezifische Durchmesserprogrammierung, 898
- Kanalspezifische Grundstellung nach Hochlauf, RESET, 897
- Kanalspezifischer NCK-Alarm steht an, 37
- Kanalzustand
  - Kanal aktiv, 500
  - Kanal unterbrochen, 500
  - Kanal-Reset, 500
- Kaskadierter Satzsuchlauf, 530

Kennwort, 66  
     setzen, 66  
     zurücksetzen, 66  
 Kinematik der Maschine, 1661  
 Kinematiktyp, 1661  
 kinematische Transformation, 754  
 Klemmungsüberwachung, 93  
 Kollisionsüberwachung, 1644  
 Konkurrierende Achsen, 949  
 Konstante, 1210  
 Konstante Drehzahl, 1516  
 Konstante Schnittgeschwindigkeit, 1514  
 KONT, 1614  
 KONTC, 1614  
 Kontrast, 56  
 KONTT, 1614  
 Kontur  
     -abtastfaktor, 224  
     -abtastzeit, 224  
     -toleranz, 232  
 Konturfehler, 86  
 Korrekturnummer, 1578  
 Krümmung, 222  
 krümmungsstetig, 1616  
 Kv-Faktor, 378

## L

Laderachsen, 737  
 Lage der Koordinatensysteme und Referenzpunkte, 748  
 Lagemesssystem, 45  
 Lageregelkreis, 377  
 Lageregler aktiv, 50  
 Laufzeiten  
     Programm-, 697  
 LENTOAX, 1767  
 Lesen  
     Einfaches, 1251  
 LFOF, 1524  
 LFON, 1524  
 LFPOS, 1524  
 LFTXT, 1524  
 LFWP, 1524  
 LIFTFAST, 626  
 LIMS, 1469  
 Lineare Signalverzerrungen, 86  
 Linearvorschub (G94), 1512  
 LookAhead, 194  
     An- und Abwahl, 196  
 Lücke, 496  
 Luft-Temperatur-Alarm, 36

## M

M1, 477  
 M17, 476  
 M19, 1391, 1468  
 M2, 476  
 M3, 1467  
 M30, 476  
 M4, 1467  
 M40, 1470, 1489  
 M41, 1470  
 M42, 1470  
 M43, 1470  
 M44, 1470  
 M45, 1470  
 M5, 1467  
 M70, 1468  
 MAIN, 1197  
 Manuelle Umschaltung Grundsystem  
     Allgemeines, 346  
     Eingabe- und Rechenfeinheit, 350  
     JOG und Handratbewertung, 350  
     Referenzpunkt, 349  
     Systemdaten, 348  
     Werkzeugdaten, 349  
 Maschinen mit aufgelöster Kinematik, 1669  
 Maschinen mit drehbarem Werkstück, 1668  
 Maschinen mit drehbarem Werkzeug, 1667  
 Maschinenachsen, 730  
 Maschinenkoordinatensystem (MKS), 38, 723, 752  
 Maschinennullpunkt, 1319  
 Maschinennullpunkt M, 746  
 Master-Slave Umschaltung bei G96, G961, 1518  
 MCP-Identifizierung, 966  
 Md < Mdx, 55  
 MD10000, 722  
 MD10010, 493, 495, 496  
 MD10050, 1371  
 MD10070, 1371  
 MD10131, 57  
 MD10192, 1441  
 MD10200, 333, 334, 336, 350, 370  
 MD10210, 333, 334, 336, 370, 372, 375  
 MD10220, 338  
 MD10230, 338  
 MD10240, 342, 347, 348, 350, 351, 1476, 1562  
 MD10260, 346, 347, 351  
 MD10270, 349  
 MD10290, 349  
 MD10292, 349  
 MD10366, 1199  
 MD10368, 1199  
 MD10600, 768, 848

MD10602, 803, 804, 807, 816, 822  
MD10610, 727, 796  
MD10612, 797  
MD10613, 860  
MD10615, 858  
MD10618, 137  
MD10680, 224  
MD10682, 224  
MD10700, 647  
MD10702, 515, 524, 625, 633, 637  
MD10704, 1542  
MD10707, 538  
MD10708, 538  
MD10710, 119, 1517  
MD10712, 221  
MD10713, 445  
MD10714, 417, 673  
MD10715, 671  
MD10716, 671  
MD10719, 674  
MD10735, 501, 668  
MD10804, 673  
MD10806, 673  
MD10814, 673  
MD1103, 319  
MD1104, 319  
MD1105, 319  
MD11100, 430  
MD11110, 437  
MD11220, 351  
MD11270, 60  
MD11300, 1322  
MD11346, 1588  
MD11410, 1426, 1519, 1645  
MD11411, 510  
MD11450, 453, 526, 527, 528, 531, 540, 571  
MD11470, 541, 542, 543, 544, 572  
MD11550, 1530  
MD11600, 622  
MD11602, 571, 621, 623, 624  
MD11604, 571, 621, 624  
MD11610, 630  
MD11620, 604  
MD12000, 1538  
MD12010, 1538  
MD12020, 1536  
MD12030, 197, 199, 1537  
MD12040, 1536  
MD12050, 1537  
MD12060, 1538  
MD12070, 1539  
MD12080, 1539  
MD12082, 1537  
MD12090, 1529  
MD12100, 197, 199, 1539  
MD12200, 1550  
MD12202, 1549  
MD12204, 1549  
MD1230/1231, 319  
MD14504, 957  
MD14506, 957  
MD14508, 957  
MD15700, 684  
MD15702, 684  
MD17200, 646  
MD18080, 1725  
MD18088, 1662, 1680, 1685  
MD18094, 349  
MD18096, 349  
MD18100, 1573, 1590  
MD18102, 1581, 1589, 1590, 1591  
MD18104, 1749  
MD18105, 1578, 1580, 1581, 1582  
MD18106, 1578, 1580, 1582  
MD18108, 1729  
MD18112, 1731, 1748  
MD18114, 1707  
MD18116, 1750, 1751, 1752, 1753, 1754, 1761  
MD18150, 60  
MD18190, 126, 138  
MD18360, 653  
MD18362, 653  
MD18600, 766  
MD18601, 783, 791, 856  
MD18602, 783, 793, 856  
MD18960, 249, 267  
MD20000, 507  
MD20050, 661, 722, 829  
MD20060, 722  
MD20070, 722  
MD20080, 722, 740  
MD20090, 475, 1467, 1594  
MD20092, 1464  
MD20094, 417, 673  
MD20095, 417, 673  
MD20096, 1575, 1576, 1577  
MD20100, 896, 903, 1515, 1518, 1758  
MD20105, 571, 623  
MD20106, 529, 605  
MD20107, 529, 606  
MD20108, 603  
MD20109, 605  
MD20110, 346, 621, 734, 858, 859, 867, 897, 900,  
1571, 1593, 1594, 1595, 1687, 1704, 1736



MD20112, 539, 621, 661, 734, 862, 867, 897, 1593, 1687  
MD20115, 571, 623  
MD20116, 624  
MD20117, 624, 636, 638  
MD20118, 661, 734  
MD20120, 661, 1593, 1687  
MD20121, 661, 1571, 1593  
MD20124, 475  
MD20125, 1697  
MD20126, 1687, 1698  
MD20127, 1698  
MD20130, 661, 1593  
MD20140, 661  
MD20144, 663  
MD20150, 119, 239, 250, 266, 286, 287, 341, 347, 507, 578, 661, 782, 859, 861, 897, 900, 1562  
MD20152, 661, 859, 897, 900  
MD20172, 1551  
MD20180, 1676  
MD20184, 852, 853, 1678  
MD20188, 1670  
MD20190, 1670  
MD20191, 625  
MD20192, 607  
MD20193, 608  
MD20200, 1553  
MD20201, 1553  
MD20202, 1632  
MD20204, 1624  
MD20210, 1640  
MD20220, 1639  
MD20230, 1641  
MD20240, 1642, 1644  
MD20250, 1613  
MD20252, 1650  
MD20256, 1652  
MD20270, 400, 1572, 1585, 1591, 1593, 1737, 1738  
MD20272, 401, 1729, 1737, 1738  
MD20360, 1588, 1748, 1758, 1765, 1785  
MD20390, 1720, 1723, 1724  
MD20392, 1721  
MD20400, 198  
MD20430, 199  
MD20440, 199  
MD20443, 203  
MD20450, 200  
MD20460, 207, 208  
MD20462, 208  
MD20465, 211, 212  
MD20480, 187, 190, 232  
MD20482, 232, 237  
MD20488, 230  
MD20490, 179  
MD20500, 251, 252  
MD20550, 174  
MD20552, 175  
MD20560, 236  
MD20600, 265  
MD20602, 262, 263  
MD20606, 223  
MD20610, 254  
MD20700, 1364  
MD20750, 1516  
MD20800, 397, 476, 477  
MD20850, 1392  
MD21015, 58  
MD21016, 59  
MD21020, 116  
MD21110, 850  
MD21158, 290  
MD21159, 290  
MD21166, 289  
MD21168, 289  
MD21202, 625  
MD21220, 1545  
MD21230, 1544  
MD21330, 661  
MD22000, 430, 1391  
MD22010, 431, 1391  
MD22020, 431, 1391  
MD22030, 432, 1391  
MD22035, 432  
MD22040, 421, 430  
MD22050, 422  
MD22060, 422  
MD22070, 423  
MD22080, 423, 685  
MD22100, 444  
MD22110, 399, 403  
MD22200, 435  
MD22210, 398, 435  
MD22220, 400, 435  
MD22230, 399, 435  
MD22240, 401, 402, 435, 1533  
MD22250, 400, 435  
MD22252, 401, 435  
MD22254, 417, 433, 673  
MD22256, 417, 433, 673  
MD22410, 1533, 1541, 1562  
MD22510, 578, 1171  
MD22530, 469, 1687  
MD22532, 469, 735  
MD22534, 469

MD22550, 675, 1571, 1585, 1586, 1591, 1593, 1594  
MD22560, 417, 673, 1571, 1593, 1594  
MD22560, 417, 673, 1571, 1593, 1594  
MD22562, 1586, 1587  
MD22600, 554  
MD22601, 554, 571  
MD22620, 575  
MD24000, 797  
MD24002, 860  
MD24004, 858  
MD24006, 758, 782, 859  
MD24007, 861  
MD24008, 758, 858  
MD24010, 796  
MD24020, 790  
MD24030, 788  
MD24040, 829  
MD24100, 1723  
MD24110, 1723  
MD24120, 1723  
MD24550, 1759  
MD24558, 1759  
MD24560, 1759  
MD24570, 1723  
MD24572, 1723  
MD24650, 1759  
MD24658, 1759  
MD24660, 1759  
MD24805, 815, 816  
MD24855, 815, 816  
MD24905, 808  
MD24955, 808  
MD25574, 1723  
MD26008, 417, 673  
MD26012, 673  
MD27100, 647, 902  
MD27800, 508  
MD27850, 702  
MD27860, 592, 702, 703  
MD27880, 592, 707  
MD27882, 707  
MD28060, 643, 646  
MD28070, 230  
MD28080, 783, 791  
MD28081, 783, 792  
MD28082, 757, 782, 799, 841, 842, 853, 856, 1678  
MD28085, 1573, 1680, 1685  
MD28150, 63  
MD28200, 126, 138  
MD28210, 138  
MD28400, 646  
MD28402, 646  
MD28530, 60, 188  
MD28533, 202  
MD28560, 862  
MD28600, 120  
MD28610, 223  
MD30130, 357  
MD30200, 355, 369  
MD30240, 356, 357  
MD30242, 357, 1352  
MD30250, 1359  
MD30300, 367, 369, 370, 372, 375, 1562  
MD30310, 114  
MD30330, 1350  
MD30340, 1350  
MD30350, 357  
MD30455, 1350, 1469  
MD30460, 896, 899, 900, 903  
MD30550, 1323  
MD31000, 367, 368, 370, 372, 375  
MD31010, 367, 368, 370  
MD31020, 367, 368, 370, 372, 375, 1371  
MD31025, 370, 372  
MD31030, 145, 367, 369, 370, 372  
MD31040, 367, 368, 370, 372, 375, 1402, 1450  
MD31044, 363, 367, 368, 370, 1440  
MD31050, 76, 145, 362, 367, 368, 370, 372, 375,  
1439, 1446  
MD31060, 77, 145, 362, 367, 368, 370, 372, 375, 1439  
MD31064, 363, 369, 370, 1440  
MD31066, 363, 369, 370, 1440  
MD31070, 145, 368, 370, 372, 375  
MD31080, 145, 368, 370, 372, 375  
MD31090, 350  
MD31122, 1418  
MD31123, 1418  
MD32000, 87, 285, 332, 366, 369, 1487, 1510  
MD32040, 1513  
MD32050, 1513  
MD32060, 572, 1556  
MD32074, 858  
MD32100, 365  
MD32200, 49, 76, 87, 378, 390  
MD32200, 49, 76, 87, 378, 390  
MD32210, 389, 390  
MD32220, 389, 390  
MD32250, 145, 365  
MD32260, 145  
MD32300, 87, 249, 259, 260, 261, 262, 285, 1556,  
1557  
MD32301, 288  
MD32310, 179, 261  
MD32320, 1558

- MD32400, 278, 387  
 MD32402, 278, 387  
 MD32410, 278, 387, 388  
 MD32420, 251, 266, 268  
 MD32430, 267  
 MD32431, 265, 272  
 MD32432, 273  
 MD32433, 260  
 MD32434, 259, 260, 272  
 MD32435, 272  
 MD32436, 288  
 MD32439, 274  
 MD32440, 208, 211  
 MD32610, 87  
 MD32620, 383  
 MD32630, 383  
 MD32640, 386  
 MD32711, 349  
 MD32750, 1720, 1723, 1724  
 MD32800, 77, 87, 386  
 MD32810, 77, 87, 145, 384  
 MD32900, 379  
 MD32910, 77, 379, 386  
 MD33050, 51  
 MD33100, 186, 224, 232  
 MD33120, 194, 232  
 MD34000, 1337, 1349  
 MD34010, 1321, 1332, 1334, 1336, 1346, 1361  
 MD34020, 1332, 1336  
 MD34040, 1334, 1336, 1337, 1346, 1347, 1418  
 MD34050, 1334  
 MD34060, 1339, 1347, 1349, 1403  
 MD34070, 1340, 1348  
 MD34080, 364, 368, 1340  
 MD34090, 364, 368, 1340, 1343, 1344, 1360, 1362, 1363, 1364  
 MD34092, 1337  
 MD34093, 1338  
 MD34100, 1340, 1348, 1361, 1364  
 MD34102, 1351, 1352  
 MD34104, 1355  
 MD34110, 1324  
 MD34200, 1344, 1352, 1355, 1361, 1365, 1418  
 MD34210, 1358, 1359, 1360, 1361, 1362, 1363, 1369  
 MD34230, 1369  
 MD34232, 1369, 1370  
 MD34300, 1347  
 MD34320, 367, 368, 1343  
 MD34330, 1366  
 MD34990, 1511  
 MD35000, 722, 777, 1463, 1470  
 MD35010, 1422, 1426, 1443, 1446, 1450, 1456  
 MD35012, 1421, 1450, 1453  
 MD35014, 1457  
 MD35020, 1415  
 MD35030, 1415  
 MD35035, 1392, 1458, 1461  
 MD35040, 1388  
 MD35090, 1421  
 MD35092, 1426  
 MD3510, 1455  
 MD35100, 332, 1485  
 MD35110, 1421, 1490  
 MD35112, 1427  
 MD35120, 1421, 1490  
 MD35122, 1427  
 MD35130, 77, 1421, 1427, 1429, 1482  
 MD35135, 1421  
 MD35140, 1421, 1427, 1429, 1482, 1517, 1519  
 MD35150, 1335, 1394, 1396, 1481, 1485  
 MD35200, 1387, 1397, 1399, 1403, 1421, 1455  
 MD35210, 1387, 1397, 1399, 1400, 1403, 1404, 1421, 1455  
 MD35212, 1427  
 MD35220, 282, 285  
 MD35230, 282, 285  
 MD35240, 188, 285  
 MD35242, 279  
 MD35300, 1371, 1396, 1398, 1399, 1403, 1404, 1421, 1454, 1469  
 MD35310, 1421, 1450, 1453  
 MD35350, 1403, 1454  
 MD35400, 1122, 1444, 1445  
 MD35410, 1443, 1446  
 MD35430, 1444  
 MD35440, 1444  
 MD35450, 1444  
 MD35500, 1389, 1394  
 MD35510, 1394, 1480  
 MD35550, 1421, 1427  
 MD35590, 49, 76, 1422, 1437  
 MD36000, 172, 1401, 1405, 1488, 1489  
 MD36010, 90, 172, 1401, 1405, 1488, 1489  
 MD36012, 92, 173  
 MD36020, 90, 389, 390  
 MD36030, 91, 389, 390  
 MD36040, 91, 325, 389, 390  
 MD36042, 325  
 MD36050, 93, 100  
 MD36052, 94, 95, 97, 99  
 MD36060, 49, 50, 1388, 1395, 1480  
 MD36100, 113  
 MD36110, 113  
 MD36120, 113

MD36130, 113  
 MD36200, 103  
 MD36210, 101, 366  
 MD36220, 102  
 MD36300, 106, 1370, 1371  
 MD36302, 1398  
 MD36310, 104, 107  
 MD36312, 108  
 MD36400, 87, 389, 390  
 MD36500, 356  
 MD36510, 356  
 MD36600, 112  
 MD36610, 47, 88, 91, 92, 93, 102, 103, 106, 109, 887  
 MD36620, 47, 887  
 MD37002, 312  
 MD37010, 308, 324  
 MD37012, 312  
 MD37020, 309, 319  
 MD37030, 308  
 MD37040, 308  
 MD37050, 310  
 MD37060, 308, 309, 311, 312  
 MD37080, 323  
 MD51029, 641  
 MD9000, 56  
 MD9001, 56  
 MD9003, 56  
 MD9004, 56, 335, 648  
 MD9006, 38, 1797  
 MD9010, 648  
 MD9011, 335, 648  
 MD9424, 649, 762  
 MD9440, 866  
 M-Dekodierung nach Liste, 953  
 Mehrere Planachsen  
   Achsspezifische Durchmesserprogrammierung, 899  
   Achstausch in Synchronaktionen, 899  
   Achstausch über Achscontainer-Drehung, 900  
   Übernahme der zusätzlichen Planachse, 899  
 Meldesignale in DB2, 1172  
 Messsysteme, 356, 1319  
 M-Funktionsersetzung, 671  
 MKS, 38  
 Modale Aktivierung (FOCON/FOCOF), 323  
 Monitortyp, 56  
 Motor-/Last-Getriebe, 362  
 Multi-Instanz-DB, 1179  
 Multitool, 1052

## N

Nachführen aktiv, 50

Nahtstelle  
   MCP/PLC und BHG/PLC, 924  
   PLC/HMI, 924, 933  
   PLC/HMI-Meldungen, 934  
   PLC/MCP, 938  
   PLC/NCK, 926  
 Nahtstellensignale  
   Achse/Spindel steht, 1823  
   Achsen-/Spindelsperre, 1810  
   Aktionssatz aktiv, 1867  
   Aktiv, 1805  
   aktive Betriebsart AUTOMATIK, 1859  
   aktive Betriebsart JOG, 1859  
   aktive Betriebsart MDA, 1859  
   Aktive G-Funktion der Gruppe 1 bis 60, 1876  
   aktive Maschinenfunktion REF, 1861  
   aktive Maschinenfunktion REPOS, 1861  
   aktive Maschinenfunktion TEACH IN, 1860  
   aktive Spindelbetriebsart Pendelbetrieb, 1906  
   aktive Spindelbetriebsart Positionierbetrieb, 1906  
   aktive Spindelbetriebsart Steuerbetrieb, 1907  
   Aktiver Antriebsparametersatz A, B, C, 1827  
   Aktiver Motor A, B, 1827  
   Aktives oder passives Filesystem, 1800, 1801  
   Alle Achsen stehen, 1839  
   alle Kanäle im Reset-Zustand, 1860  
   alle referenzpunktpflichtigen Achsen sind referenziert, 1889  
   Analog Spindel 1, Auslastung in Prozent, 1799  
   Analog Spindel 2, Auslastung in Prozent, 1799  
   Analogmesswert des Spannsystems, 1910  
   Anfahrersatz aktiv, 1867  
   angewählt Maschinenfunktion REF, 1858  
   angewählte Betriebsart AUTOMATIK, 1857  
   angewählte Betriebsart JOG, 1857  
   angewählte Betriebsart MDA, 1857  
   angewählte Betriebsart REPOS, 1858  
   angewählte Betriebsart TEACH IN, 1858  
   angezeigte Kanalnummer von HMI, 1804  
   Antriebe im zyklischen Betrieb, 1793  
   Antriebe ready, 1793  
   Antriebsparametersatz-Anwahl A, B, C, 1819  
   Antriebstest Fahrenforderung, 1822  
   Antriebstest Fahrenfreigabe, 1810  
   Anwahl, 1806, 1807  
   Assoziiertes M01/M00 aktiv, 1849  
   Assoziiertes M1 aktivieren, 1844  
   ASUP aktiv, 1880  
   ASUP ist angehalten, 1877  
   AT-Box ready, 1792  
   BAG Betriebsbereit, 1860  
   BAG-Reset, 1854

- BAG-Stop, 1853  
 BAG-Stopp Achsen plus Spindeln, 1853  
 Bahnachse, 1883  
 BA-Wechselsperre, 1802  
 Betriebsart AUTOMATIK, 1851  
 Betriebsart JOG, 1852  
 Betriebsart MDA, 1851  
 Betriebsarten Wechselsperre, 1852  
 Bildschirm dunkelsteuern, 1797  
 Bildschirm hell, 1796  
 Bildschirm ist dunkel, 1803  
 Cancelalarm gelöscht, 1803  
 Cancelalarme löschen, 1798  
 D-Funktion 1, 1847  
 Drehzahlgrenze überschritten, 1902  
 Drehzahlregler aktiv, 1825  
 Drive Ready, 1828  
 Eilgangkorrektur, 1915  
 Eilgangkorrektur wirksam, 1917  
 Einlesefreigabe wird ignoriert, 1875  
 Einlesesperre, 1863  
 Einzelsatz aktivieren, 1862  
 Einzelsatz Typ A, 1856  
 Einzelsatz Typ B, 1856  
 Entladen, 1806  
 Error, 1805  
 Erweiterte Adresse F-Funktion 1 bis 6, 1848  
 Erweiterte Adresse H-Funktion 1 bis 3, 1848  
 Erweiterte Adresse M-Funktion 1 bis 5, 1846  
 Erweiterte Adresse S-Funktion 1 bis 3, 1846  
 Externe Nullpunktverschiebung, 1884  
 Fahren auf Festanschlag aktivieren, 1843  
 Fahren auf Festanschlag freigeben, 1842  
 FC9 Out\ Active, 1802  
 FC9 Out\ Done, 1802  
 FC9 Out\ Error, 1802  
 FC9 Out\ StartError, 1802  
 Ferndiagnose aktiv, 1791  
 Festanschlag erreicht, 1843  
 Festanschlag erreicht quittieren, 1841  
 Festvorschub 1 - 4 aktivieren für Bahn-  
 /Geometrieachsen, 1920  
 Festvorschub 1 - 4 aktivieren für  
 Maschinenachsen, 1924  
 F-Funktion für Positionierachse, 1927  
 F-Hilfsfunktion für Positionierachse, 1850  
 Gebergrenzfrequenz überschritten, 1838  
 Geschwindigkeits-  
 /Spindeldrehzahlbegrenzung, 1836  
 Getriebe ist umgeschaltet, 1895  
 Getriebe umschalten, 1902  
 Gewindebohren ohne Ausgleichsfutter aktiv, 1906  
 Hardwareendschalter plus und minus, 1837  
 HMI Batteriealarm, 1792  
 HMI-CPU1-Ready, 1793  
 HMI-Temperaturgrenze, 1792  
 Hochlaufgebersperre, 1819  
 Hochlaufgebersperre aktiv, 1826  
 Hochlaufvorgang beendet, 1830  
 Impulse freigegeben, 1829  
 Impulsfreigabe, 1821  
 Integrator n-Regler gesperrt, 1828  
 Integratorsperre n-Regler, 1821  
 Interruptbehandlung aktiv, 1874  
 Istdrehrichtung rechts, 1905  
 Istgetriebestufe A bis C, 1894  
 Istwerte in WKS, 1798  
 Kanal betriebsbereit, 1875  
 Kanalnummer der Maschinensteuertafel an  
 HMI, 1799  
 kanalspezifischen Schutzbereich aktivieren, 1834  
 Kanalspezifischer NCK-Alarm steht an, 1809  
 kanalspezifischer Schutzbereich verletzt, 1836  
 kanalspezifischer Schutzbereich voraktiviert, 1835  
 Kanalzustand aktiv, 1873  
 Kanalzustand Reset, 1874  
 Kanalzustand unterbrochen, 1874  
 Klemmvorgang läuft, 1836  
 Korrektur wirksam, 1923  
 Kühlk.--Temperatur.Alarm NCU, 1795  
 Laden, 1806  
 Lagemesssystem 1 und 2, 1814  
 Lageregler aktiv, 1824  
 letzter Aktionssatz aktiv, 1869  
 Lufttemperatur-Alarm, 1795  
 M(d) kleiner M(dx), 1831  
 M-, S-, T-, D-, H-, F-Fkt. Änderung, 1844  
 M-, S-, T-, D-, H-, F-Fkt. Zusatzinformation Quick  
 (schnelle Quittung), 1845  
 M00/M01 aktiv, 1868  
 M01 aktivieren, 1862  
 M02/M30 aktiv, 1870  
 M3/M4 invertieren, 1897  
 maschinenbezogenen Schutzbereich  
 aktivieren, 1833  
 maschinenbezogener Schutzbereich verletzt, 1835  
 maschinenbezogener Schutzbereich  
 voraktiviert, 1834  
 Maschinenfunktion REF, 1855  
 Maschinenfunktion REPOS, 1855  
 Maschinenfunktion TEACH IN, 1854  
 M-Fkt. 1-5 nicht decodiert, 1845  
 M-Funktion für Spindel, 1908  
 M-Hilfsfunktion für Spindel, 1850

- MKS-WKS umschalten, 1804
- Motor-Anwahl A, B, 1820
- Motoranwahl erfolgt, 1820
- n(ist) gleich n(soll), 1832
- n(ist) kleiner n(min), 1831
- n(ist) kleiner n(x), 1831, 1832
- Nachführbetrieb, 1813
- Nachführen aktiv, 1823
- NCK-Alarm mit Bearbeitungsstillstand steht an, 1809
- NCK-Alarm steht an, 1794
- NCK-Batteriealarm, 1796
- NCK-CPU-Ready, 1792
- NC-Ready, 1794
- NC-Start, 1864
- NC-Start-Sperre, 1864
- NC-Stop, 1865
- NC-Stop Achsen plus Spindeln, 1865
- NC-Stop an Satzgrenze, 1865
- Not-Halt, 1885
- Not-Halt aktiv, 1886
- Not-Halt quittieren, 1885
- O.K., 1805
- Parametersatzvorgaben von NC sperren, 1818
- Pendelfreigabe, 1899
- Pendeln durch die PLC, 1898
- PLC-Aktion beendet, 1862
- PLC-Hardkeys, 1799
- PLC-Index, 1800, 1801
- PLC-Zeilenoffset, 1801
- POS\_RESTORED 1, 1892
- POS\_RESTORED2, 1892
- Position erreicht mit Genauhalt fein, 1840
- Position erreicht mit Genauhalt grob, 1839
- Probelaufvorschub aktivieren, 1912
- Probelaufvorschub angewählt, 1919
- ProgEventDisplay, 1880
- Programmebenenabbruch, 1864
- Programmtest aktiv, 1871
- Programmtest aktivieren, 1863
- Programmverzweigung steuern, 1881
- Programmzustand abgebrochen, 1873
- Programmzustand angehalten, 1872
- Programmzustand läuft, 1871
- Programmzustand unterbrochen, 1873
- Programmzustand warten, 1872
- Recallalarm gelöscht, 1803
- Recallalarme löschen, 1798
- Referenziert/Synchronisiert 1, 1891
- Referenziert/Synchronisiert 2, 1891
- Referenzpunktwert 1 bis 4, 1890
- Referieren aktiv, 1889
- Referieren aktivieren, 1888
- Reglerfreigabe, 1816
- Regler-Parametersatz-Umschaltung (Anforderung), 1818
- Repos DEFERAL Chan, 1880
- REPOS Delay Quitt, 1883
- Repos Path Mode Quitt0-2, 1878
- REPOS Verschiebung, 1882
- REPOS Verschiebung gültig, 1882
- REPOSDELAY, 1881, 1883
- REPOSMODEEDGEACKN, 1878
- Reset, 1866
- Restweg löschen (achsspezifisch)/Spindel-Reset, 1817
- Restweg löschen (kanalspezifisch), 1808
- Satz ausblenden aktivieren, 1863
- Satzsuchlauf aktiv, 1869
- Satzsuchlauf via Programmtest ist aktiv, 1878
- Schlüsselschalter-Stellung, 1791
- Schmierimpuls, 1826
- Schutzbereiche freigeben, 1833
- Sensor Festanschlag, 1842
- Sensor S1 vorhanden (Spannzustand), 1909
- Sensor S4 Kolbenendlage, 1911
- Sensor S4 vorhanden (Kolbenlage), 1909
- Sensor S5 vorhanden (Winkellage der Motorwelle), 1909
- Sensor S5 Winkellage der Motorwelle, 1911
- Sensorik vorhanden, 1908
- S-Funktion für Spindel, 1908
- S-Hilfsfunktion für Spindel, 1850
- Simulation aktiv, 1804
- Solldrehrichtung links/Solldrehrichtung rechts, 1900
- Soll-Drehzahl begrenzt, 1903
- Soll-Drehzahl erhöht, 1904
- Sollgetriebestufe A bis C, 1901
- Spindel im Sollbereich, 1904
- Spindel in Position, 1907
- Spindel neu synchronisieren 1 und 2, 1896
- Spindel neu synchronisieren beim Positionieren 1 und 2, 1897
- Spindel/keine Achse, 1900
- Spindelkorrektur, 1925
- Spindel-Reset/Restweg löschen, 1893
- Stilles ASUP aktiv, 1881
- Stoppen am Blockende bei SBL wird unterdrückt, 1875
- Stromregler aktiv, 1825
- S-Wert löschen, 1896
- Synchronaktionen (S5, H2)|Signale Dynamische M-Funktionen,
- Tastensperre, 1797

- Teileprogramm Anwahl, 1800  
 Teileprogramm Entladen, 1799  
 Teileprogramm Laden, 1800  
 T-Funktion 1, 1847  
 Transformation aktiv, 1871  
 Umdrehungsvorschub aktiv, 1927  
 Verzögerung Referenzpunktfahren, 1890  
 Vorschub Halt (Geometrieachse 1 bis 3), 1919  
 Vorschub-Halt/Spindel-Halt (achsspezifisch), 1924  
 Vorschubkorrektur, 1913  
 Vorschubkorrektur (achsspezifisch), 1921  
 Vorschubkorrektur für Eilgang angewählt, 1920  
 Vorschubkorrektur wirksam, 1918  
 Vorschubsperrung, 1917  
 Werkstück-Soll erreicht, 1876  
 Zustand des Spannsystems, 1910  
 Zustandswert wird gebildet, Drehzahlbegrenzung  
 p5043 aktiv, 1909  
 Zweiter Softwareendschalter plus bzw. minus, 1837
- NC**
- Ausfall, 947
  - Satz-Kompressor, 226
  - Sprachumfang, 578
  - Start, 581
  - Variable lesen/schreiben, 951
- NCK-Alarm mit Bearbeitungsstillstand, 37  
 NCK-Alarm steht an, 37  
 NCK-Batterie-Alarm, 36  
 NCK-CPU ready, 36  
 NC-VAR-Selector, 986
  - Bedienoptionen, 990
  - Inbetriebnahme, Installation, 998
- negative Adresserweiterung, 477  
 Nicht quittierter Getriebestufenwechsel, 1441  
 Nichtlineare Signalverzerrungen, 86  
 nist, 55  
 NORM, 1614  
 Normen, 884  
 Not-Halt
  - Ablauf, 887
  - Nahtstelle, 886
  - Quittierung, 889
- Not-Halt-Stellteile, 886  
 Nullmarken-Diagnose, 108  
 Nullpunkte, 746  
 Nullpunktverschiebung \$P\_EXTFRAME, 903  
 Nullvektoren, 1665  
 Nutsäge, 1600
- O**
- Offset, 1203
- Operationen, 1226  
 Orientierbare Werkzeugträger, 1568, 1657
  - Beispiele, 1772
  - Berechnung der aktiven Werkzeuglänge, 1675
  - Kinematische Kette, 1665
  - Neuanlegen, 1680
- Orientierung, 125  
 Orientierungs
  - toleranz, 232
- OTOL, 232  
 Override
  - bei G331/332, 1529
- OVR, 1540  
 OVRA, 1469, 1540
- P**
- Paket
  - zähler, 458
- Parametersatz
  - bei Achsen, 381
  - orientierbare Werkzeugträger, 1661
- Parametrierung der Drehachsen, 1662  
 Parken, 123  
 PCOF, 1468  
 Physikalische Größen, 337  
 Planachse, 893
  - Löschstellung, 900
  - Maßangaben, 898
- PLC
  - Achsen, 738
  - Grundprogramm-Funktionen, 909
  - Variable lesen und schreiben, 61
  - Versionen, 909
- POINTER, 1174  
 POINTER im FB, 1175  
 POINTER im FC, 1174  
 POLF, 1524  
 POLFMASK, 1524  
 POLFMLIN, 1524  
 Polynome, Schnittpunktverfahren, 1652  
 POS, 737, 738  
 POSA, 737, 738  
 Positionierachsen, 333, 738  
 Positioniergenauigkeit, 334  
 Positionswerte im Durchmesser anzeigen, 902  
 Probelaufvorschub, 515, 1541  
 PROFIBUS
  - Ankopplung, 967
- PROFIBUS-Ankopplung, 968  
 PROFIBUS-Diagnose, 946  
 PROFINET

- Ankopplung, 970
- Programm
  - aktion, 588
  - Interrupt-, 1198
  - laufzeiten, 697
  - organisation, 1197
  - organisationseinheit, 1194
  - test, 511
  - zustand anzeigen, 585
  - zustände, 585
- Programmbearbeitung ohne Sollwertausgaben, 511
- Programmbeeinflussung, Nahtstellen-Signale, 641
- Programmbetrieb, 577
  - Grundeinstellungen, 577
- Programmiergerät
  - Hardware-Voraussetzungen, 983
- Programmteilwiederholung, 594
- Projekt, 1193
- Prozessalarm, 946

## Q

- Querverweis, 1196

## R

- Radius bezogene Daten, 901
- Recallalarme, 38
- Rechenfeinheit, 335
- Referenzierarten, 1319
- Referenzieren
  - bei inkrementellen Messsystem, 1330
  - bei rotatorischen Absolutwertgebern, 1366
- Referenzpunkt R, 746
- Referenzpunkte, 746
- Regelsinn, 365
- Regelung, 377
- Reglerfreigabe, 46
- Regler-Parametersatz-Umschaltung, 76
- Repositionieren neutraler Achsen nach SERUPRO, 544
- REPOS-Verschiebung, 315
- Reset
  - verhalten, 658
- RESET
  - Kommando, 584
- Restweg löschen, 39
- Restzeit
  - für ein Werkstück, 700
- Richtungsvektor, 1708
- Rotationsanteil, 844

- Ruck
  - überhöhung, geschwindigkeitsabhängige, 273
- Ruckbegrenzung, 191
- Rückzug
  - richtung beim Gewindeschneiden, 1525
- Rundachsen, 737

## S

- S..., 1467
- Satz
  - ausblenden, 641
- Satzbezogene Begrenzung (FOC), 324
- Satzsuchlauf
  - Kaskadierter, 522
  - mit Berechnung an der Kontur (Typ 2), 521
  - mit Berechnung an Satzendpunkt (Typ 4), 522
  - mit Berechnung im Modus Programmtest SERUPRO (Typ 5), 522
  - ohne Berechnung (Typ 1), 521
  - Zeitlicher Ablauf der Typen 1, 2 und 4, 523
- Satzsuchlauf mit Berechnung
  - aktuelle Istposition, 525
  - aufgesammelte Istposition, 525
  - aufgesammelte Spindelfunktionen, 527
- Satzsuchlauf SERUPRO
  - Achse mit REPOS-Verschiebung verzögert anfahren, 545
  - Bahnachsen, 545
  - Bedingungen für Achsfunktionen, 572
  - Getriebestufenwechsel, 573
  - Grundeinstellung, 575
  - Positionierachsen Wiederanfahren, 544
  - REPOS mit NC/PLC-Nahtstellensignale steuern, 547
  - REPOS vorziehen oder ignorieren, 544
  - REPOS-Quittungen, 547
  - REPOS-Verhalten einstellen, 542
  - REPOS-Verschiebung bei Synchronspindelkopplung, 551
  - REPOS-Verschiebung im Gültigkeitsbereich, 550
  - REPOS-Verschiebung nach einem Achstausch, 551
  - REPOS-Vorgang, 541
  - Soll- und Istwertkopplungen, 568
  - Überlagerte Bewegungen, 574
  - Wiederanfahren mit gesteuertem REPOS, 542
  - Zeitlicher Ablauf, 538
- SAVE, 628
- SBLOF, 635
- SBLON, 638
- Schlüsselschalter, 67
- Schmierimpuls, 51



- Schneiden  
  -bezugspunkt, 1694  
  -form, 1692  
  -lage, 1690, 1695  
  -mittelpunkt, 1694  
Schneidenlage, 1602  
  relevante, 1611  
Schneidenummer, 1578  
Schneller Datenkanal, 61  
Schnittrichtung, 1690  
Schraubenlinieninterpolation, 741  
Schutzbereiche, 85, 124  
  Beispiel, Aktivierung, 158  
  Beispiel, Definition, 149  
  Datenablage, 133  
  Definition mit Systemvariablen, 132  
  Definition per Teileprogramm, 128  
  Einschränkungen, 143  
Schutzstufe  
  bei Anwender-ASUP, 631  
Schutzstufen, 65  
  parametrierbare, 68  
SCPARA, 1560  
SD41100, 1513  
SD41200, 1461  
SD42010, 1521  
SD42100, 515, 1542  
SD42101, 515, 1543  
SD42200, 634  
SD42440, 727, 1704  
SD42442, 1703  
SD42444, 524  
SD42465, 186, 232  
SD42466, 186, 232  
SD42480, 1651  
SD42496, 1647  
SD42500, 255  
SD42502, 255  
SD42510, 268  
SD42512, 268  
SD42600, 1478, 1514  
SD42676, 232  
SD42678, 232  
SD42700, 654  
SD42750, 645  
SD42800, 1463, 1465  
SD42900, 1569, 1713, 1740, 1748, 1768  
SD42910, 1709, 1713, 1714, 1740, 1746, 1747, 1748  
SD42920, 1713, 1714, 1740, 1746, 1747, 1749  
SD42930, 1713, 1715, 1740, 1746, 1749, 1783  
SD42935, 1713, 1741, 1747, 1749, 1771, 1778  
SD42940, 1699, 1709, 1713, 1717, 1740, 1746, 1749  
SD42950, 1690, 1708, 1709, 1713, 1740, 1746, 1749  
SD42960, 1569, 1713, 1721, 1722  
SD42974, 1670  
SD42980, 851, 852  
SD42984, 1699  
SD42990, 643  
SD43200, 1474, 1475  
SD43202, 1474, 1476  
SD43206, 1474, 1477  
SD43210, 1517, 1519  
SD43220, 1517  
SD43235, 1486  
SD43240, 1391, 1468, 1477  
SD43250, 1391, 1477  
SD43300, 1478, 1514  
SD43400, 118  
SD43410, 118  
SD43420, 117  
SD43430, 117  
SD43500, 319  
SD43510, 319  
SD43520, 319  
Sekundenfehler, 224  
Self-Acting SERUPRO, 559  
SERUPRO  
  automatischer Unterbrechungszeiger, 563  
  -Ende-ASUP, 462  
  programmierbarer Unterbrechungszeiger, 561  
  SPEED-Faktor bei Kanalachsen im Hochlauf, 554  
SERUPRO-Anfahren, 541  
  Bahnachsen einzeln beeinflussen, 551  
  vom PLC beeinflussen, 547  
  Wiederanfahren an den nächstliegenden Punkt, 552  
SERUPRO-ASUP, 315  
  Besonderheiten, 555  
seruproMasterChan, 559  
SETINT, 617, 626  
SETMS, 1467  
SETTCOR, 1760  
S-Funktionen, 398  
Signalaustausch  
  Ereignisgesteuerter, 907  
  Zyklischer, 907  
Signale  
  Achsen-/Spindelspezifische (DB31, ...), 35  
  Alarm-Signale, 36  
  Compilezyklen, 928  
  Kanalspezifische (DB21, ...), 35  
  NCK / PLC, 929  
  PLC / Achsen, Spindeln, 932  
  PLC / BAG, 929, 930  
  PLC / NCK-Kanäle, 930

PLC/NCK, 928  
Ready-Signale, 36  
Signalverzerrungen, 86  
Signatur, 1207  
Simulation, 519  
Simulationsachsen, 357  
Simulierten Zielpunkt für LEAD mit JOG erreichen, 569  
SOFT, 266  
SOFTA, 267  
Software-Endschalter, 113  
Sollwertausgabe, 355  
Sollwertsystem, 355  
Sondermerker, 1225  
SPCOF, 1468  
SPCON, 1468  
Speicher  
-art, 1201  
Speicherbedarf  
des PLC-Grundprogramms, 979  
SPI, 1470  
Spiegelung  
Frames, 777  
Rückzugsrichtung (Schnellabheben), 625  
Spindel  
-auftrag, 1472  
-betriebsarten, 1384  
-drehzahl, Maximale, 1482  
-drehzahl, Minimale, 1482  
-drehzahlbegrenzungen, 1483  
-getriebestufe 0, 1433  
-korrektur, 1536, 1538  
-schnittstelle, 1472  
-solldrehzahl, 1481  
-vorgaben, 1474  
Spindeldrehzahl, 333  
Spindelfunktionen über PLC, 508  
Spindelsperre, 40, 1535  
Spline, 169  
SPOS, 1390, 1467  
SPOSA, 1468  
Status, 1240  
-tabelle, 1195, 1250  
Stern-/Dreieck Umschaltung mit FC17, 1432  
Stern-Dreieck-Umschaltung, 1116  
Steuerungsverhalten  
bei Hochlauf, 658  
bei Reset, 658  
bei Teileprogrammende, 658  
bei Teileprogrammstart, 658  
Stirnfräsen 3D, 1606  
STOLF, 237  
Stopp-Delay-Bereich, 612

Stopp-Ereignis, 612  
Strings, 1180  
Stromregler aktiv, 50  
SVC, 1467  
Symbol  
-tabelle, 1195  
Symbolische Programmierung, 952  
Synchronachsen, 182, 740

## T

Tabelle  
Status-, 1195  
-Symbol, 1195  
Tabellenstatus, 1251  
tangentenstetig, 1616  
Tastensperre, 38  
TCARR, 1676  
TCOABS, 1677  
TCOFR, 1676, 1677  
TCP-Tool Center Position, 746  
TEACH IN, 498  
Teileprogramm  
Anwahl, 581  
Kanalfreigabe, 581  
-sätze ausblenden, 517  
starten, 589  
T-Funktion, 399, 1571  
T-Funktionsersetzung, 674  
Tick, 458  
TOA  
-Daten, 1600  
-Einheit, 1573  
Toleranz  
bei G0, 236  
-faktor für Spindeldrehzahl, 1481  
TOOLENV, 1748  
Torsion, 222  
TOWBCS, 1745  
TOWKCS, 1745  
TOWMCS, 1744  
TOWSTD, 1744  
TOWTCS, 1745  
TOWWCS, 1744  
TRANS, 725  
TRANSMIT, 754

## U

Überlastfaktor, 179  
Überschleifen, 180

UDT-Zuordnungen, 952  
 Umdrehungsvorschub (G95), 1512  
 Umrechnung Grundsystem, 346  
 Umschaltbare Geometrieachsen, 732  
 Umverdrahten, 1236  
 Unterprogramme, 670  
 UTD-Bausteine, 952

## V

Variablen  
   lokale, 1206  
   -tabelle, 1206  
 VELOLIM, 1469  
 Verfahrbereiche, 334  
 Verfahrbewegungsfreigabe, 1366  
 Verriegelbare Datenbereiche, 68  
 Vordergrundsprache, 56  
 Vorgabe der konstanten Schnittgeschwindigkeit, 1476  
 Vorsatz-Getriebe, 363, 1440  
 Vorschub  
   -arten, 1507  
   -arten (G93, G94, G95), 1511  
   Bahnvorschub F, 1509  
   für Fase/Rundung, 1552  
   -korrektur, 1536  
   Linear- (G94), 1512  
   Umdrehungs- (G95), 1512  
   Zahn-, 1513  
   Zeitreziproker (G93), 1511  
 Vorschübe  
   Probelaufvorschub, 1541  
   Vorschub/Spindel Halt, 1534  
   Vorschubsperr, 1534  
 Vorschubkorrektur  
   Programmierbare, 1540  
 Vorschubsperr  
   kanalspezifisch, 1534

## W

WAITENC, 663  
 WAITS, 1469  
 WALCS0, 121  
 WALIMOF, 119  
 WALIMON, 119  
 Wechsel des Frames, 1659  
 Wegkriterium, 183  
 Weiches An- und Abfahren  
   Bedeutung, 1620  
   Teilbewegungen, 1621

Werkstück  
   -simulation, 519  
   -zähler, 705  
 Werkstückkoordinatensystem (WKS), 38, 723, 724, 763  
 Werkstücknullpunkt W, 746  
 Werkzeug, 1571  
   anwählen, 1571  
   Basismaß/Adaptermaß, 1608  
   DISC, 1638  
   -form, 1605  
   -form, aktive, 1607  
   -größe, aktive, 1607  
   -korrekturdaten, 1600  
   -länge, 1604  
   -parameter, 1596  
   -rückzug, 663  
   -schneide, 1596  
   Schneide, 1596  
   T-Funktion, 1571  
   -typ, 1598  
   -verschleiß, 1607  
   -verwaltung, 492  
   Wechsel, 1571  
   Wechsel mit M06, 1571  
   Werkzeugradiuskorrektur-2D (WRK), 1612  
   Werkzeugschneide, 1572  
 Werkzeuggrundorientierung, 1707  
 Werkzeuggrundorientierung, Beispiel, 1776  
 Werkzeugkorrektur  
   Typen, 1660  
   Verrechnung in der NC, 1575  
 Werkzeuglängenkorrektur  
   Geometrie-, 1604  
   Verschleiß-, 1607  
   werkstückspezifisch einrechnen, 1739  
   werkstückspezifisch einrechnen, Beispiel, 1776  
 Werkzeugmagazinachsen, 737  
 Werkzeugradiuskorrektur  
   konstant halten, 1648  
   Verschleiß-, 1607  
 Werkzeugradiuskorrektur-2D, 1612  
   Abwahl, 1637  
   An - und Abfahrverhalten, 1614  
   Anwahl, 1613  
   geändertes Alarmverhalten, 1651  
   Geometrie-, 1605  
   Kollisionsüberwachung, 1644  
   Korrektur an Außenecken, 1637  
   Korrektur an Innenecken, 1642  
   Schnittpunkt G451, 1640  
   Übergangskreis, 1638

- veränderlicher Korrekturwert, 1646
- Weiches An- und Abfahren, 1620
- Werkzeugrevolverachsen, 737
- Werkzeugträger, orientierbare, 1657
  - drehbarer Werk Tisch, 1675
  - Neuanlegen, 1680
  - Programmierung, 1685
  - Randbedingungen, 1686
  - Schrägbearbeitung, 1674
  - Steuerungsverhalten bei Reset, Programmstart, Repos, 1686
- Werkzeugträgeranwahl, 1657
- Werkzeugträgerbezugspunkt T, 746
- Werkzeugumgebungen, 1748
- Werkzeugwechsel
  - D-Funktion, 1572
  - Korrekturspeicher, 1573
- Wiederanfahren
  - an den Satzanfang, 552
  - an den Unterbrechungspunkt, 552
- Wiederanfahren an Kontur
  - Wiederanfahrpunkt, 553
- Winkel
  - Frei-, 1691
  - Halter-, 1691
- WKS, 38

## X

- XE \\* MERGEFORMAT, 397

## Y

- YDelta, 1116

## Z

- Zählimpuls, 708
- Zahnvorschub, 1513
- Zeitreziproker Vorschub (G93), 1511
- Zugriffsart, 1203
- Zugriffsmerkmale, 66
- Zugriffsrechte, 65
- Zugriffsschutz, 65
- Zuordnung Werkzeug / Werkzeugträger, 1658
- Zusammenfassen von Hilfsfunktionen, 422
- Zusatzachsen, 737
- Zyklischer Betrieb, 943
- Zyklischer Signalaustausch, 34