

SIMATIC

Programmieren mit STEP 7

Handbuch

Vorwort	
Inhaltsverzeichnis	
Produkteinführung und Installation	1
Installation	2
Konzeption der Automatisierungslösung	3
Grundlagen zum Entwerfen einer Programmstruktur	4
Starten und Bedienen	5
Einrichten und Bearbeiten des Projekts	6
Bearbeiten von Projekten mit unterschiedlichen STEP 7- Versionen	7
Festlegen von Symbolen	8
Anlegen von Bausteinen und Bibliotheken	9
Erstellen von Codebausteinen	10
Erstellen von Datenbausteinen	11
Parametrieren von Datenbausteinen	12
Erstellen von AWL-Quellen	13
Anzeigen von Referenzdaten	14
Bausteinkonsistenz prüfen und Zeitstempel als Bausteineigenschaft	15
Projektieren von Meldungen	16
Bedienen und Beobachten von Variablen	17
Aufbau der Online-Verbindung und CPU-Einstellung	18
Laden	19
Testen	20
Testen mit Programmstatus	21
Testen mit Simulationsprogramm (Optionspaket)	22
Diagnose	23
Drucken und Archivieren	24
Arbeiten mit M7- Automatisierungssystemen	25
Tipps und Tricks	26
Anhang	
Index	

Diese Dokumentation ist Bestandteil des
Dokumentationspaketes mit der Bestellnummer:
6ES7810-4CA08-8AW0

Sicherheitshinweise

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.



Gefahr

bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Warnung

bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Vorsicht

mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Achtung

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 zugehörige Gerät/System darf nur in Verbindung mit dieser Dokumentation eingerichtet und betrieben werden. Inbetriebsetzung und Betrieb eines Gerätes/Systems dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieser Dokumentation sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie Folgendes:



Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden. Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

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

Zweck des Handbuchs

Dieses Handbuch gibt Ihnen einen vollständigen Überblick über das Programmieren mit **STEP 7**. Es unterstützt Sie bei der Installation und Inbetriebnahme der Software. Die Vorgehensweise für die Programmerstellung und die Bestandteile von Anwenderprogrammen werden erläutert.

Das Handbuch richtet sich an Personen, die für die Realisierung von Steuerungsaufgaben mit STEP 7 auf Basis der SIMATIC S7-Automatisierungssysteme tätig sind.

Wir empfehlen Ihnen, sich mit den Beispielen aus dem Handbuch Getting Started "Erste Schritte und Übungen mit STEP 7" vertraut zu machen. Sie bieten einen leichten Einstieg in die Thematik "Programmieren mit STEP

Erforderliche Grundkenntnisse

Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik erforderlich.

Außerdem werden Kenntnisse über die Verwendung von Computern oder Pc-ähnlichen Arbeitsmitteln (z. B. Programmiergeräten) unter den Betriebssystemen MS Windows 2000 Professional, MS Windows XP Professional oder MS Windows Server 2003 vorausgesetzt.

Gültigkeitsbereich des Handbuchs

Das Handbuch ist gültig für das Softwarepaket STEP 7 V5.4.

Informationen zu Servicepacks, die nach Drucklegung dieses Handbuchs erscheinen, finden Sie

- in der Datei "Liesmich.wri"
- in der aktualisierten Online-Hilfe zu STEP 7

Das Thema "Was ist neu?" der Online-Hilfe bietet einen guten Einstieg und ersten Überblick zu den Innovationen von STEP 7.

Dokumentationspakete zu STEP 7

Dieses Handbuch ist Bestandteil des Dokumentationspaketes "STEP 7 Grundwissen".

Die folgende Tabelle zeigt die Dokumentation zu STEP 7 im Überblick:

Handbücher	Zweck	Bestellnummer
STEP 7-Grundwissen mit <ul style="list-style-type: none"> • Erste Schritte und Übungen mit STEP 7 • Programmieren mit STEP 7 • Hardware konfigurieren und Verbindungen projektieren mit STEP 7 • Von S5 nach S7, Umsteigerhandbuch 	Das Grundwissen für technisches Personal, das das Vorgehen zur Realisierung von Steuerungsaufgaben mit STEP 7 und S7-300/400 beschreibt.	6ES7810-4CA08-8AW0
STEP 7-Referenzwissen mit <ul style="list-style-type: none"> • Handbücher KOP/FUP/AWL für S7-300/400 • Standard- und Systemfunktionen für S7-300/400 Band 1 und Band 2 	Das Referenzwissen zum Nachschlagen, das die Programmiersprachen KOP, FUP und AWL sowie Standard- und Systemfunktionen ergänzend zum STEP 7-Grundwissen beschreibt.	6ES7810-4CA08-8AW1

Online-Hilfen	Zweck	Bestellnummer
Hilfe zu STEP 7	Das Grundwissen zum Programmieren und Hardware konfigurieren mit STEP 7 als Online-Hilfe	Bestandteil des Softwarepaketes STEP 7
Referenzhilfen zu AWL/KOP/FUP Referenzhilfe zu SFBs/SFCs Referenzhilfe zu Organisationsbausteinen	Kontextsensitives Referenzwissen	Bestandteil des Softwarepaketes STEP 7

Online-Hilfe

Ergänzend zum Handbuch erhalten Sie bei der Nutzung der Software detaillierte Unterstützung durch die in die Software integrierte Online-Hilfe.

Das Hilfesystem ist über mehrere Schnittstellen in die Software integriert:

- Im Menü **Hilfe** stehen mehrere Menübefehle zur Verfügung: **Hilfethemen** öffnet das Inhaltsverzeichnis der Hilfe zu STEP 7.
- **Hilfe benutzen** gibt detaillierte Anweisungen zum Umgang mit der Online-Hilfe.
- Die kontext-sensitive Hilfe bietet Informationen zum aktuellen Kontext, z. B. zu einem geöffneten Dialogfeld oder zu einem aktiven Fenster. Sie lässt sich über die Schaltfläche "Hilfe" oder über die Taste F1 aufrufen.
- Eine weitere Form kontext-sensitiver Hilfe bietet die Statuszeile. Zu jedem Menübefehl wird hier eine kurze Erklärung angezeigt, sobald sich der Mauszeiger auf dem Menübefehl befindet.
- Auch zu den Symbolen in der Funktionsleiste wird eine kurze Erläuterung eingeblendet, wenn der Mauszeiger kurze Zeit über den Symbolen verweilt.

Wenn Sie Informationen der Online-Hilfe lieber in gedruckter Form lesen möchten, können Sie einzelne Hilfethemen, Bücher oder die gesamte Hilfe auch ausdrucken.

Dieses Handbuch ist ebenso wie die Handbücher "Hardware konfigurieren und Verbindungen projektieren mit STEP 7", "Anlagenänderungen im laufenden Betrieb mittels CiR" und "S7-400H - Hochverfügbare Systeme" ein Auszug der Hilfe zu STEP 7. Detaillierte Handlungsanweisungen finden Sie in der Hilfe zu STEP 7. Da Handbücher und Online-Hilfe nahezu identisch gegliedert sind, können Sie bequem zwischen Handbüchern und Online-Hilfe wechseln.

Die elektronischen Handbücher finden Sie nach der Installation von STEP 7 über die Startleiste unter **Start > SIMATIC > Dokumentation**.

Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

Ihren Ansprechpartner finden Sie unter:

<http://www.siemens.com/automation/partner>

Den Wegweiser zum Angebot an technischen Dokumentationen für die einzelnen SIMATIC Produkte und Systeme finden Sie unter:

<http://www.siemens.de/simatic-tech-doku-portal>

Den Online-Katalog und das Online-Bestellsystem finden Sie unter:

<http://mall.automation.siemens.com/>

Trainingscenter

Um Ihnen den Einstieg in das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90327 Nürnberg.

Telefon: +49 (911) 895-3200.

Internet: <http://www.sitrain.com>

Technical Support

Sie erreichen den Technical Support für alle A&D-Produkte

- Über das Web-Formular für den Support Request
<http://www.siemens.de/automation/support-request>
- Telefon: + 49 180 5050 222
- Fax: + 49 180 5050 223

Weitere Informationen zu unserem Technical Support finden Sie im Internet unter <http://www.siemens.de/automation/service>

Service & Support im Internet

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser komplettes Wissen online an.

<http://www.siemens.com/automation/service&support>

Dort finden Sie:

- den Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche in Service & Support.
- ein Forum, in welchem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Automation & Drives vor Ort.
- Informationen über Vor-Ort Service, Reparaturen, Ersatzteile. Vieles mehr steht für Sie unter dem Begriff "Leistungen" bereit

Inhaltsverzeichnis

1	Produkteinführung und Installation	1-1
1.1	Leitfaden durch STEP 7.....	1-1
1.2	Das STEP 7 - Basispaket	1-6
1.3	Was ist neu in STEP 7 Version 5.4	1-11
1.4	Erweiterungsmöglichkeiten des STEP 7 - Basispakets.....	1-14
1.4.1	Engineering Tools.....	1-15
1.4.2	Runtime Software	1-17
1.4.3	Human Machine Interface.....	1-19
2	Installation	2-1
2.1	Automation License Manager.....	2-1
2.1.1	Nutzungsberechtigung durch den Automation License Manager	2-1
2.1.2	Installieren des Automation License Managers.....	2-4
2.1.3	Regeln für den Umgang mit License Keys	2-5
2.2	Installieren von STEP 7	2-6
2.2.1	Vorgehen beim Installieren	2-8
2.2.2	Einstellen der PG/PC-Schnittstelle	2-11
2.3	Deinstallieren von STEP 7.....	2-13
3	Konzeption der Automatisierungslösung	3-1
3.1	Prinzipielle Vorgehensweise zur Konzeption einer Automatisierungslösung... 3-1	
3.2	Zerlegen des Prozesses in Aufgaben und Bereiche	3-2
3.3	Beschreiben der einzelnen Funktionsbereiche	3-5
3.4	Auflisten von Ein-, Aus- und Durchgängen.....	3-7
3.5	Erstellen eines Eingangs-, Ausgangsdiagramms für die Motoren	3-8
3.6	Erstellen eines Eingangs-/ Ausgangsdiagramms für die Ventile.....	3-9
3.7	Festlegen der Sicherheitsanforderungen	3-10
3.8	Beschreiben der notwendigen Anzeige- und Bedienelemente.....	3-11
3.9	Erstellen des Konfigurationsplans	3-12
4	Grundlagen zum Entwerfen einer Programmstruktur.....	4-1
4.1	Programme in einer CPU.....	4-1
4.2	Bausteine im Anwenderprogramm	4-2
4.2.1	Organisationsbausteine und Programmstruktur	4-3
4.2.2	Aufrufhierarchie im Anwenderprogramm.....	4-9
4.2.3	Bausteinarten.....	4-11
4.2.3.1	Organisationsbaustein für zyklische Programmbearbeitung (OB 1)	4-11
4.2.3.2	Funktionen (FC).....	4-16
4.2.3.3	Funktionsbausteine (FB).....	4-18
4.2.3.4	Instanz-Datenbausteine.....	4-21
4.2.3.5	Globale Datenbausteine (DB).....	4-24
4.2.3.6	Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC).....	4-25

4.2.4	Organisationsbausteine für alarmgesteuerte Programmbearbeitung	4-27
4.2.4.1	Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17)	4-28
4.2.4.2	Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)	4-30
4.2.4.3	Organisationsbausteine für den Weckalarm (OB 30 bis OB 38).....	4-31
4.2.4.4	Organisationsbausteine für den Prozessalarm(OB 40 bis OB 47).....	4-33
4.2.4.5	Organisationsbausteine für den Anlauf (OB 100 / OB 101 / OB 102).....	4-34
4.2.4.6	Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90)	4-36
4.2.4.7	Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)	4-38
5	Starten und Bedienen	5-1
5.1	Starten von STEP 7	5-1
5.2	Starten von STEP 7 mit vorgegebenen Startparametern.....	5-3
5.3	Aufrufen der Hilfefunktionen	5-5
5.4	Objekte und Objekthierarchie	5-6
5.4.1	Objekt Projekt	5-8
5.4.2	Objekt Bibliothek.....	5-10
5.4.3	Objekt Station	5-11
5.4.4	Objekt Programmierbare Baugruppe.....	5-13
5.4.5	Objekt S7/M7-Programm	5-15
5.4.6	Objekt Baustein-Ordner	5-17
5.4.7	Objekt Quellen-Ordner.....	5-21
5.4.8	S7/M7-Programm ohne Station und CPU	5-22
5.5	Benutzungsoberfläche und Bedienung.....	5-23
5.5.1	Bedienphilosophie	5-23
5.5.2	Aufbau des Fensters.....	5-24
5.5.3	Elemente in Dialogfeldern.....	5-25
5.5.4	Erzeugen und Manipulieren von Objekten	5-27
5.5.5	Auswahl von Objekten im Dialog.....	5-32
5.5.6	Sitzungsgedächtnis.....	5-34
5.5.7	Ändern der Fensteranordnung	5-34
5.5.8	Speichern und Wiederherstellen der Fensteranordnung.....	5-35
5.6	Bedienung über Tastatureingabe	5-36
5.6.1	Tastenkombinationen für Menübefehle	5-36
5.6.2	Tastenkombinationen für das Bewegen des Cursors.....	5-38
5.6.3	Tastenkombinationen für das Markieren von Texten	5-40
5.6.4	Tastenkombinationen für den Zugriff auf Online-Hilfe	5-40
5.6.5	Tastenkombinationen für das Umschalten zwischen Fenstertypen	5-41
6	Einrichten und Bearbeiten des Projekts.....	6-1
6.1	Projektstruktur.....	6-1
6.2	Wissenswertes zum Zugriffsschutz	6-3
6.3	Wissenswertes zum Änderungsprotokoll.....	6-4
6.4	Fremdsprachige Zeichensätze verwenden.....	6-5
6.5	Einstellen der Windows-Sprache.....	6-8
6.6	Einrichten eines Projekts	6-9
6.6.1	Anlegen eines Projekts	6-9
6.6.2	Einfügen von Stationen.....	6-11
6.6.3	Einfügen eines S7- / M7-Programms	6-13
6.7	Bearbeiten eines Projekts.....	6-15
6.7.1	Projekt auf verwendete Softwarepakete prüfen	6-16

6.8	Texte mehrsprachig verwalten	6-16
6.8.1	Texttypen mehrsprachig verwalteter Texte	6-18
6.8.2	Aufbau der Export-Datei	6-19
6.8.3	Verwalten von Anwendertexten, deren Sprach-Font nicht installiert ist	6-21
6.8.4	Informationen zur Protokoll-Datei	6-21
6.8.5	Optimieren der Vorlage für die Übersetzung	6-22
6.8.6	Optimierung des Übersetzungsvorgangs	6-24
6.9	Micro Memory Card (MMC) als Datenträger	6-25
6.9.1	Wissenswertes zu Micro Memory Cards (MMC)	6-25
6.9.2	Micro Memory Card als Datenträger verwenden.....	6-26
6.9.3	Memory Card-Datei	6-26
6.9.4	Projektdateien auf Micro Memory Card (MMC) ablegen.....	6-27
7	Bearbeiten von Projekten mit unterschiedlichen STEP 7-Versionen	7-1
7.1	Version 2-Projekte und Bibliotheken bearbeiten	7-1
7.2	Erweitern von DP-Slaves, die mit Vorgänger-Versionen von STEP 7 erstellt wurden	7-1
7.3	Aktuelle Konfigurationen mit Vorgänger-Versionen von STEP 7 bearbeiten ...	7-3
7.4	SIMATIC PC - Konfigurationen aus Vorgänger-Versionen nutzen	7-4
7.5	Darstellung von Baugruppen, die mit neueren STEP 7-Versionen oder einem Optionspaket projektiert sind	7-6
8	Festlegen von Symbolen.....	8-1
8.1	Absolute und symbolische Adressierung.....	8-1
8.2	Globale und lokale Symbole	8-3
8.3	Darstellung von globalen oder lokalen Symbolen	8-4
8.4	Einstellen des Operandenvorrangs (symbolisch/absolut)	8-5
8.5	Symboltabelle für globale Symbole	8-8
8.5.1	Struktur und Bestandteile der Symboltabelle	8-8
8.5.2	Zulässige Adressen und Datentypen in der Symboltabelle	8-10
8.5.3	Unvollständige und mehrdeutige Symbole in der Symboltabelle	8-11
8.6	Eingabemöglichkeiten von globalen Symbolen	8-12
8.6.1	Allgemeine Hinweise zur Eingabe von Symbolen	8-12
8.6.2	Eingeben einzelner globaler Symbole im Dialog.....	8-13
8.6.3	Eingeben mehrerer globaler Symbole in der Symboltabelle	8-14
8.6.4	Groß-/Kleinschreibung bei Symbolen.....	8-15
8.6.5	Exportieren und Importieren von Symboltabellen	8-17
8.6.6	Dateiformate für den Import/Export einer Symboltabelle	8-17
8.6.7	Bearbeiten von Bereichen in Symboltabellen.....	8-20
9	Anlegen von Bausteinen und Bibliotheken.....	9-1
9.1	Auswahl der Erstellmethode	9-1
9.2	Auswahl der Programmiersprache	9-3
9.2.1	Programmiersprache KOP (Kontaktplan)	9-5
9.2.2	Programmiersprache FUP (Funktionsplan)	9-6
9.2.3	Programmiersprache AWL (Anweisungsliste)	9-7
9.2.4	Programmiersprache S7-SCL	9-8
9.2.5	Programmiersprache S7-GRAPH (Ablaufsteuerung).....	9-9
9.2.6	Programmiersprache S7-HiGraph (Zustandsgraph)	9-10
9.2.7	Programmiersprache S7-CFC	9-12

9.3	Anlegen von Bausteinen.....	9-13
9.3.1	Bausteinordner	9-13
9.3.2	Anwenderdefinierte Datentypen (UDT)	9-14
9.3.3	Bausteineigenschaften	9-15
9.3.4	Anzeige von Baustein-Längen.....	9-18
9.3.5	Bausteine vergleichen	9-19
9.3.6	Umverdrahten	9-22
9.3.7	Attribute für Bausteine und Parameter	9-22
9.4	Arbeiten mit Bibliotheken.....	9-23
9.4.1	Hierarchischer Aufbau von Bibliotheken.....	9-25
9.4.2	Übersicht der Standardbibliotheken	9-25
10	Erstellen von Codebausteinen	10-1
10.1	Grundlagen zum Erstellen von Codebausteinen.....	10-1
10.1.1	Aufbau des Programmeditor-Fensters	10-1
10.1.2	Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen	10-3
10.1.3	Voreinstellungen für Programmeditor KOP/FUP/AWL	10-4
10.1.4	Zugriffsrechte auf Bausteine bzw. Quellen.....	10-4
10.1.5	Anweisungen aus der Programmelemente-Übersicht.....	10-5
10.2	Editieren der Variablendeklaration	10-6
10.2.1	Verwendung der Variablendeklaration in Codebausteinen	10-6
10.2.2	Zusammenspiel zwischen Variablendeklaration und Anweisungsteil	10-7
10.2.3	Aufbau des Variablendeklarationsfensters	10-8
10.3	Multiinstanzen in der Variablendeklaration.....	10-9
10.3.1	Verwendung von Multiinstanzen.....	10-9
10.3.2	Regeln für die Bildung von Multiinstanzen	10-10
10.3.3	Eingeben der Multiinstanz im Variablendeklarationsfenster.....	10-10
10.4	Allgemeine Hinweise zum Editieren von Anweisungen und Kommentaren	10-11
10.4.1	Aufbau des Anweisungsteils.....	10-11
10.4.2	Vorgehensweise beim Eingeben von Anweisungen	10-12
10.4.3	Eingeben von globalen Symbolen in ein Programm	10-13
10.4.4	Titel und Kommentare zu Bausteinen und Netzwerken	10-14
10.4.5	Eingeben von Baustein-/Netzwerkcommentaren	10-15
10.4.6	Arbeiten mit Netzwerkvorlagen.....	10-16
10.4.7	Suchfunktion für Fehler im Anweisungsteil.....	10-17
10.5	Editieren von KOP-Anweisungen im Anweisungsteil	10-18
10.5.1	Einstellungen für Programmiersprache KOP.....	10-18
10.5.2	Regeln für die Eingabe von KOP-Anweisungen.....	10-18
10.5.3	Unzulässige Verschaltungen in KOP.....	10-21
10.6	Editieren von FUP-Anweisungen im Anweisungsteil.....	10-22
10.6.1	Einstellungen für die Programmiersprache FUP	10-22
10.6.2	Regeln für die Eingabe von FUP-Anweisungen	10-23
10.7	Editieren von AWL-Anweisungen im Anweisungsteil	10-25
10.7.1	Einstellungen für die Programmiersprache AWL	10-25
10.7.2	Regeln für die Eingabe von AWL-Anweisungen	10-25
10.8	Aktualisieren von Bausteinaufrufen	10-26
10.8.1	Ändern von Schnittstellen	10-27
10.9	Speichern von Codebausteinen	10-28

11	Erstellen von Datenbausteinen	11-1
11.1	Grundlagen zum Erstellen von Datenbausteinen	11-1
11.2	Deklarationssicht von Datenbausteinen	11-2
11.3	Datensicht von Datenbausteinen.....	11-3
11.4	Eingeben in Datenbausteine und Speichern	11-5
11.4.1	Eingeben der Datenstruktur von globalen Datenbausteinen.....	11-5
11.4.2	Eingeben / Anzeige der Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs)	11-6
11.4.3	Eingeben der Struktur von anwenderdefinierten Datentypen (UDT).....	11-8
11.4.4	Eingeben / Anzeige der Struktur von Datenbausteinen mit zugeordnetem UDT.....	11-9
11.4.5	Ändern von Datenwerten in der Datensicht.....	11-10
11.4.6	Rücksetzen von Datenwerten auf die Anfangswerte.....	11-10
11.4.7	Speichern von Datenbausteinen	11-11
12	Parametrieren von Datenbausteinen	12-1
12.1	Parametrieren von Technologischen Funktionen.....	12-2
13	Erstellen von AWL-Quellen.....	13-1
13.1	Grundlagen zum Programmieren in AWL-Quellen.....	13-1
13.2	Regeln zum Programmieren in AWL-Quellen	13-2
13.2.1	Regeln zur Eingabe von Anweisungen in AWL-Quellen	13-2
13.2.2	Regeln für Variablendeklaration in AWL-Quellen.....	13-3
13.2.3	Regeln zur Reihenfolge der Bausteine in AWL-Quellen	13-4
13.2.4	Regeln zum Festlegen von Systemattributen in AWL-Quellen	13-4
13.2.5	Regeln zum Festlegen von Bausteineigenschaften in AWL-Quellen.....	13-5
13.2.6	Zulässige Bausteineigenschaften je Bausteinart.....	13-7
13.3	Struktur von Bausteinen in AWL-Quellen.....	13-8
13.3.1	Struktur von Codebausteinen in AWL-Quellen.....	13-8
13.3.2	Struktur von Datenbausteinen in AWL-Quellen.....	13-9
13.3.3	Struktur von anwenderdefinierten Datentypen in AWL-Quellen.....	13-9
13.4	Syntax und Formate für Bausteine in AWL-Quellen.....	13-10
13.4.1	Formattabelle von OBs	13-10
13.4.2	Formattabelle von FBs.....	13-11
13.4.3	Formattabelle von FCs	13-12
13.4.4	Formattabelle von DBs	13-13
13.5	Erstellen von AWL-Quellen	13-14
13.5.1	Anlegen von AWL-Quellen	13-14
13.5.2	Bearbeiten von S7-Quellen	13-14
13.5.3	Layout des Quelltextes festlegen	13-15
13.5.4	Einfügen von Bausteinvorlagen in AWL-Quellen	13-15
13.5.5	Einfügen des Inhalts anderer AWL-Quellen	13-15
13.5.6	Einfügen des Quellcodes vorhandener Bausteine in AWL-Quellen.....	13-16
13.5.7	Einfügen von externen Quellen	13-16
13.5.8	Generieren von AWL-Quellen aus Bausteinen	13-17
13.5.9	Importieren von Quellen	13-17
13.5.10	Exportieren von Quellen	13-18
13.6	Speichern, Übersetzen von AWL-Quellen und Konsistenzprüfung.....	13-19
13.6.1	Speichern von AWL-Quellen	13-19
13.6.2	Prüfen der Konsistenz in AWL-Quellen.....	13-19
13.6.3	Fehlersuche in AWL-Quellen.....	13-20
13.6.4	Übersetzen von AWL-Quellen	13-20

13.7	Beispiele zu AWL-Quellen	13-22
13.7.1	Beispiele für Variablendeklarationen in AWL-Quellen.....	13-22
13.7.2	Beispiel für OBs in AWL-Quellen.....	13-23
13.7.3	Beispiel für FCs in AWL-Quellen	13-24
13.7.4	Beispiel für FBs in AWL-Quellen	13-26
13.7.5	Beispiele für DBs in AWL-Quellen.....	13-28
13.7.6	Beispiel für UDTs in AWL-Quellen	13-29
14	Anzeigen von Referenzdaten.....	14-1
14.1	Übersicht der möglichen Referenzdaten	14-1
14.1.1	Querverweisliste	14-2
14.1.2	Programmstruktur	14-4
14.1.3	Belegungsplan	14-6
14.1.4	Nicht verwendete Symbole	14-8
14.1.5	Operanden ohne Symbol.....	14-8
14.1.6	Anzeige sprachabhängiger Informationen zu KOP, FUP, AWL	14-9
14.2	Arbeiten mit Referenzdaten	14-10
14.2.1	Möglichkeiten zum Anzeigen von Referenzdaten	14-10
14.2.2	Anzeigen von Listen in zusätzlichen Arbeitsfenstern	14-10
14.2.3	Erzeugen und Anzeigen von Referenzdaten.....	14-11
14.2.4	Schnelles Positionieren auf Verwendungsstellen im Programm.....	14-12
14.2.5	Beispiel zum Arbeiten mit Verwendungsstellen	14-13
15	Bausteinkonsistenz prüfen und Zeitstempel als Bausteineigenschaft.....	15-1
15.1	Bausteinkonsistenz prüfen	15-1
15.2	Zeitstempel als Bausteineigenschaft und Zeitstempelkonflikte	15-3
15.3	Zeitstempel in Codebausteinen	15-4
15.4	Zeitstempel bei globalen Datenbausteinen	15-5
15.5	Zeitstempel bei Instanz-Datenbausteinen	15-5
15.6	Zeitstempel bei UDTs und von UDTs abgeleiteten DBs	15-6
15.7	Korrigieren der Schnittstellen in einem FC, FB oder UDT.....	15-6
15.8	Vermeiden von Fehlern beim Aufrufen von Bausteinen.....	15-7
16	Projektieren von Meldungen.....	16-1
16.1	Grundlagen des Meldekonzeptes.....	16-1
16.1.1	Welche Meldeverfahren gibt es?	16-1
16.1.2	Auswahl des Meldeverfahrens	16-3
16.1.3	SIMATIC-Komponenten.....	16-5
16.1.4	Bestandteile einer Meldung	16-5
16.1.5	Welche Meldebausteine gibt es?.....	16-6
16.1.6	Formalparameter, Systemattribute und Meldebausteine	16-8
16.1.7	Meldungstyp und Meldungen	16-9
16.1.8	Erzeugen einer AWL-Quelle von meldenden Bausteinen	16-10
16.1.9	Vergabe von Meldenummern	16-10
16.1.10	Unterschiede zwischen projektweiter und CPU-weiter Vergabe von Meldenummern	16-11
16.1.11	Möglichkeiten zum Abändern der Meldenummernvergabe eines Projekts..	16-11
16.2	Projektweites Projektieren von Meldungen	16-12
16.2.1	Projektweite Vergabe von Meldenummern.....	16-12
16.2.2	Bausteinbezogene Meldungen zuordnen und bearbeiten.....	16-12
16.2.2.1	Bausteinbezogene Meldungen anlegen (projektweit)	16-13
16.2.2.2	Bausteinbezogene Meldungen bearbeiten (projektweit)	16-16
16.2.2.3	PCS 7-Meldungsprojektierung (projektweit)	16-16
16.2.3	Symbolbezogene Meldungen zuordnen und bearbeiten.....	16-17
16.2.3.1	Symbolbezogene Meldungen zuordnen und bearbeiten (projektweit)	16-17

16.2.4	Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten	16-18
16.3	CPU-weites Projektieren von Meldungen.....	16-20
16.3.1	CPU-weite Vergabe von Meldenummern.....	16-20
16.3.2	Bausteinbezogene Meldungen zuordnen und bearbeiten.....	16-21
16.3.2.1	Bausteinbezogene Meldungen anlegen (CPU-weit)	16-21
16.3.2.2	Bausteinbezogene Meldungen bearbeiten (CPU-weit)	16-24
16.3.2.3	PCS 7-Meldungsprojektierung (CPU-weit).....	16-24
16.3.3	Symbolbezogene Meldungen zuordnen und bearbeiten.....	16-26
16.3.3.1	Symbolbezogene Meldungen zuordnen und bearbeiten (CPU-weit).....	16-26
16.3.4	Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten	16-27
16.4	Tipps zum Bearbeiten von Meldungen.....	16-28
16.4.1	Begleitwerte in Meldungen einfügen	16-28
16.4.2	Texte aus Textbibliotheken in Meldungen integrieren.....	16-30
16.4.3	Begleitwerte löschen.....	16-31
16.5	Bedienerrelevante Texte übersetzen und bearbeiten	16-32
16.5.1	Anwendertexte übersetzen und bearbeiten.....	16-32
16.6	Textbibliotheken bearbeiten und übersetzen	16-34
16.6.1	Anwender-Textbibliotheken	16-34
16.6.2	Erstellen von Anwender-Textbibliotheken	16-34
16.6.3	Bearbeiten von Anwender-Textbibliotheken.....	16-35
16.6.4	System-Textbibliotheken	16-36
16.6.5	Textbibliotheken übersetzen.....	16-36
16.7	Meldungsprojektierungsdaten zum Zielsystem transferieren	16-38
16.8	CPU-Meldungen und anwenderdefinierte Diagnosemeldungen anzeigen ..	16-39
16.8.1	CPU-Meldungen konfigurieren	16-42
16.8.2	Anzeigen von gespeicherten CPU-Meldungen	16-42
16.9	'Melden von Systemfehlern' projektieren.....	16-43
16.9.1	Unterstützte Komponenten und Funktionsumfang	16-45
16.9.2	Einstellungen für das Melden von Systemfehlern	16-48
16.9.3	Bausteine für die Meldung von Systemfehlern generieren.....	16-49
16.9.4	Erzeugter FB, DB.....	16-49
16.9.5	Anlegen von fremdsprachigen Meldetexten in 'Systemfehler melden'	16-51
17	Bedienen und Beobachten von Variablen	17-1
17.1	Projektieren von bedien- und beobachtbaren Variablen	17-1
17.2	Projektieren von BuB-Attributen mit AWL, KOP und FUP.....	17-3
17.3	Projektieren von BuB-Attributen über die Symboltabelle	17-4
17.4	Abändern von Attributen für Bedienen und Beobachten mit CFC.....	17-5
17.5	Transferieren der Projektierungsdaten zum BuB-Zielsystem.....	17-6
18	Aufbau der Online-Verbindung und CPU-Einstellung.....	18-1
18.1	Aufbau von Online-Verbindungen	18-1
18.1.1	Online-Verbindung aufbauen über das Fenster "Erreichbare Teilnehmer"	18-2
18.1.2	Online-Verbindung aufbauen über das Online-Fenster des Projekts.....	18-3
18.1.3	Online-Zugriff auf Zielsysteme im Multiprojekt	18-4
18.1.4	Passwortschutz für Zugriff auf Zielsysteme.....	18-6
18.1.5	Hinweis zur Aktualisierung des Fensterinhalts	18-7
18.2	Anzeigen und Ändern des Betriebszustands	18-8
18.3	Anzeigen und Einstellen von Uhrzeit und Datum	18-9
18.3.1	CPU-Uhren mit Zeitzone-Einstellung und Sommer-/Winterzeit.....	18-9
18.4	Aktualisieren der Firmware	18-10
18.4.1	Online-Aktualisierung der Firmware von Baugruppen und Modulen.....	18-10

19	Laden.....	19-1
19.1	Laden aus dem PG in das Zielsystem.....	19-1
19.1.1	Voraussetzungen für das Laden.....	19-1
19.1.2	Unterschied zwischen Speichern und Laden von Bausteinen	19-2
19.1.3	Lade- und Arbeitsspeicher in der CPU.....	19-3
19.1.4	Lademöglichkeiten abhängig vom Ladespeicher	19-4
19.1.5	Programm in S7-CPU laden	19-5
19.1.5.1	Laden mit Projektverwaltung	19-5
19.1.5.2	Laden ohne Projektverwaltung	19-6
19.1.5.3	Nachladen von Bausteinen in das Zielsystem.....	19-6
19.1.5.4	Speichern geladener Bausteine auf integriertem EPROM.....	19-7
19.1.5.5	Laden über EPROM-Memory Cards	19-7
19.2	Mehrere Objekte aus dem PG übersetzen und laden	19-9
19.2.1	Voraussetzungen und Hinweise zum Laden	19-9
19.2.2	Objekte übersetzen und laden.....	19-11
19.3	Laden aus dem Zielsystem in das PG.....	19-13
19.3.1	Station laden in PG.....	19-14
19.3.2	Zurückladen von Bausteinen aus S7-CPU.....	19-15
19.3.3	Bearbeiten von geladenen Bausteinen im PG/PC	19-16
19.3.3.1	Bearbeiten von geladenen Bausteinen, falls Anwenderprogramm im PG/PC vorliegt.....	19-17
19.3.3.2	Bearbeiten von geladenen Bausteinen, falls Anwenderprogramm nicht im PG/PC vorliegt.....	19-17
19.4	Löschen im Zielsystem	19-18
19.4.1	Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU	19-18
19.4.2	Löschen von S7-Bausteinen auf dem Zielsystem	19-19
19.5	Komprimieren des Anwenderspeichers (RAM)	19-20
19.5.1	Entstehung von Lücken im Anwenderspeicher (RAM).....	19-20
19.5.2	Komprimieren des Speicherinhalts einer S7-CPU	19-21
20	Testen mit der Variablen-tabelle.....	20-1
20.1	Einführung zum Testen mit Variablen-tabellen.....	20-1
20.2	Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablen-tabellen	20-2
20.3	Bearbeiten und Speichern von Variablen-tabellen	20-3
20.3.1	Erstellen und Öffnen einer Variablen-tabelle	20-3
20.3.2	Kopieren/Verschieben von Variablen-tabellen	20-4
20.3.3	Speichern einer Variablen-tabelle.....	20-4
20.4	Eingeben von Variablen in Variablen-tabellen.....	20-5
20.4.1	Einfügen von Operanden oder Symbolen in eine Variablen-tabelle.....	20-5
20.4.2	Einfügen eines Bereichs zusammenhängender Operanden in eine Variablen-tabelle	20-7
20.4.3	Einfügen von Steuerwerten	20-8
20.4.4	Obergrenzen für die Eingabe von Zeiten.....	20-8
20.4.5	Obergrenzen für die Eingabe von Zählern	20-9
20.4.6	Einfügen von Kommentarzeilen.....	20-10
20.4.7	Beispiele	20-10
20.4.7.1	Beispiel für die Eingabe von Operanden in Variablen-tabellen	20-10
20.4.7.2	Beispiel für die Eingabe eines zusammenhängenden Operandenbereichs.....	20-11
20.4.7.3	Beispiele für die Eingabe von Steuer-/Forcewerten	20-11
20.5	Herstellen einer Verbindung zur CPU	20-14
20.6	Beobachten von Variablen	20-15
20.6.1	Einführung zum Beobachten von Variablen	20-15
20.6.2	Festlegen des Triggers zum Beobachten von Variablen.....	20-15

20.7	Steuern von Variablen	20-17
20.7.1	Einführung zum Steuern von Variablen.....	20-17
20.7.2	Festlegen des Triggers zum Steuern von Variablen	20-18
20.8	Forcen von Variablen.....	20-20
20.8.1	Sicherheitsvorkehrungen zum Forcen von Variablen.....	20-20
20.8.2	Einführung zum Forcen von Variablen	20-21
20.8.3	Unterschiede zwischen Forcen und Steuern von Variablen.....	20-23
21	Testen mit Programmstatus	21-1
21.1	Anzeigen im Programmstatus	21-3
21.2	Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte	21-5
21.3	Wissenswertes zum Betriebszustand HALT.....	21-7
21.4	Programmstatus von Datenbausteinen	21-8
21.5	Festlegen der Anzeigen für den Programmstatus.....	21-9
21.6	Festlegen der Betriebsart für den Test.....	21-10
22	Testen mit Simulationsprogramm (Optionspaket)	22-1
22.1	Testen mit dem Simulationsprogramm S7-PLCSIM (Optionspaket).....	22-1
23	Diagnose	23-1
23.1	Diagnose der Hardware und Fehlersuche.....	23-1
23.2	Diagnosesymbole in der Online-Sicht	23-3
23.3	Hardware diagnostizieren: Schnellansicht.....	23-5
23.3.1	Aufruf der Schnellansicht.....	23-5
23.3.2	AuskunftsFunktionen der Schnellansicht.....	23-5
23.4	Hardware diagnostizieren: Diagnosesicht	23-6
23.4.1	Aufruf der Diagnosesicht von HW Konfig	23-6
23.4.2	AuskunftsFunktionen der Diagnosesicht	23-8
23.5	Baugruppenzustand.....	23-9
23.5.1	Aufrufmöglichkeiten des Baugruppenzustands	23-9
23.5.2	AuskunftsFunktionen des Baugruppenzustands	23-10
23.5.3	Baugruppentyp-abhängiger Umfang der AuskunftsFunktionen im Baugruppenzustand.....	23-12
23.5.4	Baugruppenzustand von PA-Feldgeräten und DP-Slaves hinter Y-Link anzeigen.....	23-13
23.6	Diagnose im Betriebszustand STOP	23-15
23.6.1	Prinzipielles Vorgehen zum Ermitteln einer STOP-Ursache	23-15
23.6.2	Stack-Inhalte im Betriebszustand STOP	23-15
23.7	Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern.....	23-17
23.8	Übermittlung von Diagnoseinformationen	23-18
23.8.1	Systemzustandsliste SZL	23-19
23.8.2	Eigene Diagnosemeldungen senden.....	23-22
23.8.3	Diagnosefunktionen	23-23
23.9	Maßnahmen im Programm zur Störungsbehandlung	23-24
23.9.1	Auswerten des Ausgangsparameters RET_VAL	23-25
23.9.2	Fehler-OBs als Reaktion auf die Erkennung eines Fehlers	23-26
23.9.3	Einfügen von Ersatzwerten bei Fehlererkennung	23-31
23.9.4	Peripherie-Redundanzfehler (OB 70).....	23-33
23.9.5	CPU-Redundanzfehler (OB 72).....	23-34
23.9.6	Zeitfehler (OB 80)	23-35
23.9.7	Stromversorgungsfehler (OB 81).....	23-36
23.9.8	Diagnosealarm (OB 82).....	23-37
23.9.9	Ziehen-/Stecken-Alarm (OB 83)	23-38
23.9.10	CPU-Hardwarefehler (OB 84).....	23-39
23.9.11	Programmablauffehler (OB 85)	23-40

23.9.12	Baugruppenträgerausfall (OB 86).....	23-41
23.9.13	Kommunikationsfehler (OB 87)	23-42
23.9.14	Programmierfehler (OB 121)	23-43
23.9.15	Peripheriezugriffsfehler (OB 122).....	23-44
24	Drucken und Archivieren	24-1
24.1	Drucken einer Projektdokumentation	24-1
24.1.1	Prinzipielle Vorgehensweise beim Drucken	24-2
24.1.2	Funktionen zum Drucken.....	24-2
24.1.3	Besonderheiten zum Drucken des Objektbaums	24-3
24.2	Archivieren von Projekten und Bibliotheken	24-4
24.2.1	Anwendungsfälle für Speichern / Archivieren.....	24-5
24.2.2	Voraussetzungen für das Archivieren.....	24-6
24.2.3	Vorgehensweise beim Archivieren/Dearchivieren	24-6
25	Arbeiten mit M7-Automatisierungssystemen	25-1
25.1	Vorgehensweise für M7-Systeme.....	25-1
25.2	Optionssoftware für M7-Programmierung	25-3
25.3	Betriebssysteme für M7-300/400.....	25-6
26	Tipps und Tricks	26-1
26.1	Baugruppen tauschen in der Konfigurationstabelle.....	26-1
26.2	Projekte mit einer großen Anzahl vernetzter Stationen	26-1
26.3	Reorganisation.....	26-2
26.4	Symbole über mehrere Netzwerke bearbeiten	26-3
26.5	Testen mit der Variablen-tabelle.....	26-3
26.6	Variablen vom Programmierer aus steuern	26-5
26.7	Virtueller Arbeitsspeicher.....	26-6
A	Anhang.....	A-1
A.1	Betriebszustände	A-1
A.1.1	Betriebszustände und Übergänge	A-1
A.1.2	Betriebszustand STOP	A-4
A.1.3	Betriebszustand ANLAUF.....	A-5
A.1.4	Betriebszustand RUN	A-13
A.1.5	Betriebszustand HALT.....	A-14
A.2	Speicherbereiche von S7-CPU's.....	A-15
A.2.1	Aufteilung der Speicherbereiche	A-15
A.2.2	Lade- und Arbeitsspeicher.....	A-16
A.2.3	Systemspeicher	A-19
A.2.3.1	Verwenden der Systemspeicherbereiche.....	A-19
A.2.3.2	Prozessabbild der Ein-/Ausgänge	A-21
A.2.3.3	Lokaldaten-Stack	A-25
A.2.3.4	Unterbrechungs-Stack.....	A-26
A.2.3.5	Baustein-Stack.....	A-27
A.2.3.6	Diagnosepuffer	A-28
A.2.3.7	Auswertung des Diagnosepuffers.....	A-28
A.2.3.8	Remanente Speicherbereiche in S7-300-CPU's.....	A-30
A.2.3.9	Remanente Speicherbereiche in S7-400-CPU's.....	A-31
A.2.3.10	Konfigurierbare Speicherobjekte im Arbeitsspeicher	A-32

A.3	Daten- und Parametertypen	A-33
A.3.1	Einführung zu Daten- und Parametertypen	A-33
A.3.2	Elementare Datentypen	A-34
A.3.2.1	Format des Datentyps INT (16-Bit-Ganzzahlen)	A-35
A.3.2.2	Format des Datentyps DINT (32-Bit-Ganzzahlen)	A-36
A.3.2.3	Format des Datentyps REAL (Gleitpunktzahlen)	A-37
A.3.2.4	Format der Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen	A-41
A.3.2.5	Format des Datentyps S5TIME (Zeitdauer)	A-42
A.3.3	Zusammengesetzte Datentypen	A-43
A.3.3.1	Format des Datentyps DATE_AND_TIME (Datum und Uhrzeit)	A-44
A.3.3.2	Verwenden von zusammengesetzten Datentypen	A-45
A.3.3.3	Verwenden von Feldern für den Datenzugriff	A-46
A.3.3.4	Verwenden von Strukturen für den Datenzugriff	A-49
A.3.3.5	Verwenden von anwenderdefinierten Datentypen für den Datenzugriff	A-51
A.3.4	Parametertypen	A-54
A.3.4.1	Format der Parametertypen BLOCK, COUNTER, TIMER	A-55
A.3.4.2	Format des Parametertyps POINTER	A-55
A.3.4.3	Verwenden des Parametertyps POINTER	A-56
A.3.4.4	Baustein zum Verändern des Pointers	A-58
A.3.4.5	Format des Parametertyps ANY	A-61
A.3.4.6	Verwenden des Parametertyps ANY	A-63
A.3.4.7	Zuordnen von Datentypen zu Lokaldaten von Codebausteinen	A-67
A.3.4.8	Zulässige Datentypen beim Übergeben von Parametern	A-69
A.3.4.9	Übergabe an IN_OUT-Parameter eines FB	A-74
A.4	Arbeiten mit älteren Projekten	A-75
A.4.1	Umsetzen von Version 1-Projekten	A-75
A.4.2	Umsetzen von Version 2-Projekten	A-76
A.4.3	Hinweise zu STEP 7 V2.1-Projekten mit GD-Kommunikation	A-77
A.4.4	DP-Slaves mit fehlenden oder fehlerhaften GSD-Dateien	A-77
A.5	Beispielprogramme	A-78
A.5.1	Beispielprojekte und Beispielprogramme	A-78
A.5.2	Beispielprogramm für einen industriellen Mischprozess	A-80
A.5.2.1	Definieren von Codebausteinen	A-83
A.5.2.2	Zuordnen von symbolischen Namen	A-84
A.5.2.3	Erstellen des FB für den Motor	A-86
A.5.2.4	Erstellen der FC für die Ventile	A-90
A.5.2.5	Erstellen des OB 1	A-92
A.5.3	Beispiel zur Hantierung mit Uhrzeitalarmen	A-99
A.5.3.1	Struktur des Anwenderprogramms Uhrzeitalarme	A-99
A.5.3.2	FC 12	A-101
A.5.3.3	OB 10	A-103
A.5.3.4	OB 1 und OB 80	A-105
A.5.4	Beispiel zur Hantierung mit Verzögerungsalarman	A-106
A.5.4.1	Struktur des Anwenderprogramms Verzögerungsalarman	A-106
A.5.4.2	OB 20	A-108
A.5.4.3	OB 1	A-110
A.5.4.4	Beispiel für das Maskieren und Demaskieren von Synchronfehlerereignissen	A-112
A.5.4.5	Beispiel zum Sperren und Freigeben von Alarm- und Asynchronfehlerereignissen (SFC 39 und 40)	A-116
A.5.4.6	Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronfehlerereignissen (SFC 41 und 42)	A-117

A.6	Zugriff auf Prozess- und Peripheriedatenbereiche	A-118
A.6.1	Zugriff auf den Prozessdatenbereich.....	A-118
A.6.2	Zugriff auf den Peripheriedatenbereich	A-119
A.7	Einstellen des Betriebsverhaltens	A-122
A.7.1	Ändern des Verhaltens und der Eigenschaften von Baugruppen	A-123
A.7.2	Offline-Aktualisierung der Firmware (des Betriebssystems) von Baugruppen und Modulen	A-125
A.7.3	Nutzen der Uhrzeitfunktionen	A-126
A.7.4	Verwenden von Taktmerkern und Zeiten	A-128
Index	Index1

1 Produkteinführung und Installation

1.1 Leitfaden durch STEP 7

Was ist STEP 7?

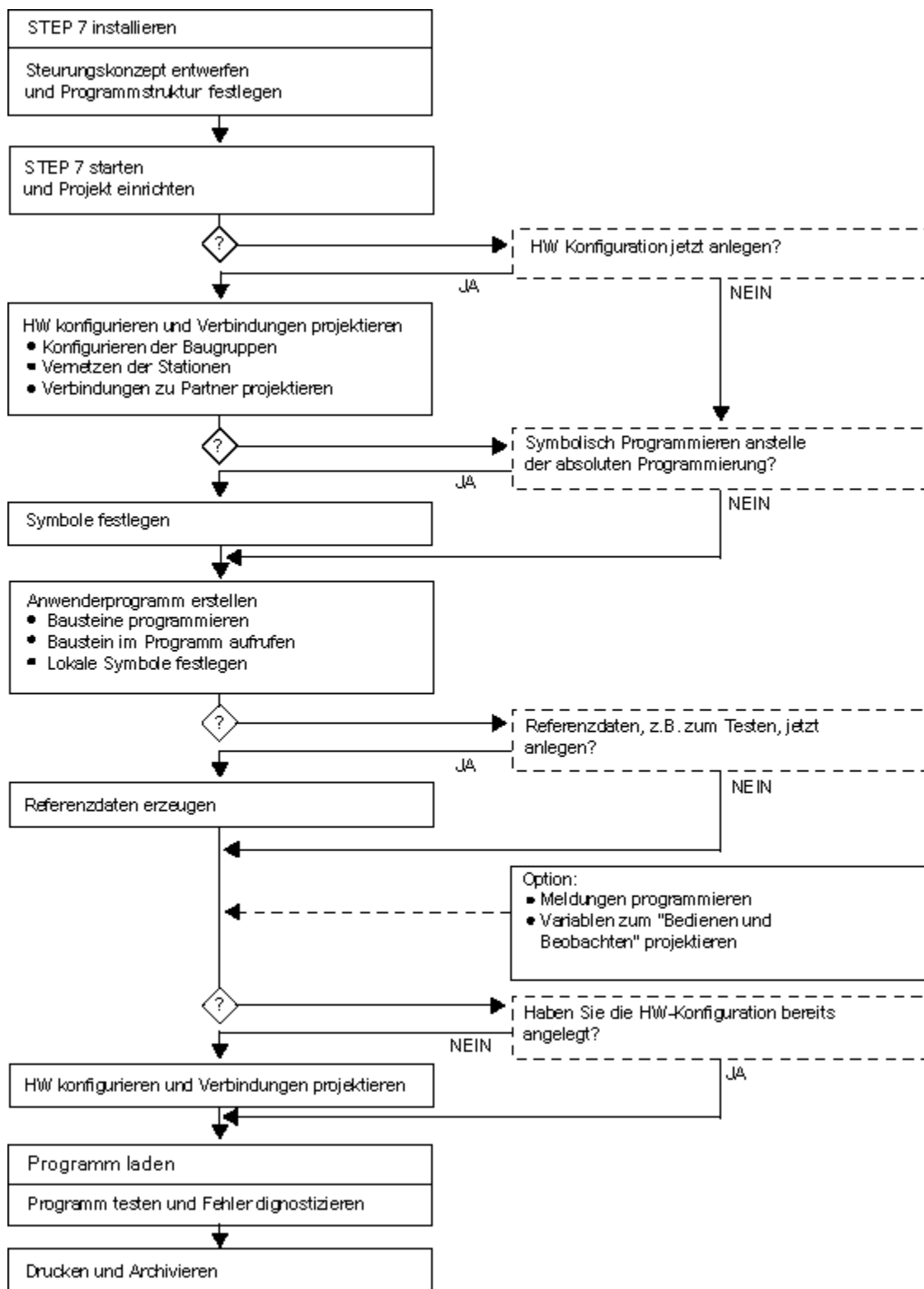
STEP 7 ist das Basispaket zur Konfigurierung und Programmierung von SIMATIC-Automatisierungssystemen. Es ist Teil der SIMATIC Industrie Software. Vom Basispaket STEP 7 gibt es folgende Ausführungen:

- STEP 7-Micro/DOS und STEP 7-Micro/Win für einfachere Stand-alone-Anwendungen auf SIMATIC S7-200.
- STEP 7 für Anwendungen auf SIMATIC S7-300/400, SIMATIC M7-300/400 und SIMATIC C7 mit zusätzlichem Funktionsumfang:
 - optional erweiterbar durch die in SIMATIC Industrie Software enthaltenen Softwareprodukte (siehe auch Erweiterungsmöglichkeiten des STEP 7 - Basispakets)
 - Möglichkeit, Funktionsbaugruppen und Kommunikationsbaugruppen zu parametrieren
 - Forcen und Multicomputing-Betrieb
 - Globaldaten-Kommunikation
 - ereignisgesteuerte Datenübertragung mit Kommunikations-/Funktionsbausteinen
 - Projektieren von Verbindungen

STEP 7 ist Gegenstand dieser Dokumentation, STEP 7-Micro und STEP 7 Lite werden in getrennten Dokumentationen beschrieben.

Grundlegende Arbeiten

Bei der Erstellung einer Automatisierungslösung mit STEP 7 fallen grundlegende Aufgaben an. Nachfolgendes Bild zeigt die Aufgaben, die in den meisten Projekten durchgeführt werden, und ordnet sie zu einer prinzipiellen Vorgehensweise in Form eines Leitfadens. Er verweist dabei auf die jeweiligen Kapitel und gibt Ihnen damit die Möglichkeit, sich aufgabenspezifisch durch das Handbuch zu bewegen.



Alternativen in der Vorgehensweise

Wie im vorherigen Bild dargestellt, haben Sie für die Vorgehensweise zwei Alternativen:

- Sie können zuerst die Hardware konfigurieren und danach die Bausteine programmieren.
- Sie können aber auch zuerst die Bausteine programmieren, ohne vorher die Hardware konfigurieren zu müssen. Das empfiehlt sich vor allem bei Service- und Wartungsarbeiten, um z. B. programmierte Bausteine in ein bereits bestehendes Projekt zu integrieren.

Kurzbeschreibung der einzelnen Arbeitsschritte:

- STEP 7 und License Keys installieren
Bei der Erstanwendung installieren Sie STEP 7 und übertragen die License Keys von Diskette auf die Festplatte (siehe auch Installieren von STEP 7 und Autorisierung).
- Steuerung entwerfen
Bevor Sie mit STEP 7 arbeiten, planen Sie Ihre Automatisierungslösung von der Untergliederung des Prozesses in einzelne Aufgaben bis zur Erstellung eines Konfigurationsplanes (siehe auch Prinzipielle Vorgehensweise zur Konzeption einer Automatisierungslösung).
- Programmstruktur entwerfen
Die beim Steuerungsentwurf beschriebenen Aufgaben setzen Sie mit den von STEP 7 zur Verfügung gestellten Bausteinen in eine Programmstruktur um (siehe auch Bausteine im Anwenderprogramm).
- STEP 7 starten
Sie starten STEP 7 von der Windows-Oberfläche aus (siehe auch Starten von STEP 7).
- Projektstruktur anlegen
Ein Projekt ist wie ein Ordner, in dem alle Daten hierarchisch gegliedert abgelegt werden und jederzeit zur Verfügung stehen. Nachdem Sie ein Projekt angelegt haben, werden alle weiteren Arbeiten in diesem Projekt ausgeführt (siehe auch Projektstruktur).
- Station einrichten
Mit dem Einrichten der Station legen Sie die Steuerung fest: z. B. SIMATIC 300, SIMATIC 400, SIMATIC S5 (siehe auch Einfügen von Stationen).
- Hardware konfigurieren
Beim Konfigurieren legen Sie in einer Konfigurationstabelle fest, welche Baugruppen Sie für Ihre Automatisierungslösung einsetzen und über welche Adressen die Baugruppen aus dem Anwenderprogramm heraus angesprochen werden sollen. Darüber hinaus lassen sich noch Eigenschaften der Baugruppen über Parameter einstellen (siehe auch Grundsätzliche Bedienung beim Hardware konfigurieren).

- Netze und Kommunikationsverbindungen projektieren
Grundlage für die Kommunikation ist ein zuvor konfiguriertes Netz. Dazu erzeugen Sie die für Ihre Automatisierungsnetze benötigten Subnetze, legen die Subnetz-Eigenschaften und für die vernetzten Stationen die Netzanschluss-Eigenschaften und ggf. die benötigten Kommunikationsverbindungen fest (siehe auch Vorgehensweise zum Projektieren eines Subnetzes).
- Symbole festlegen
Sie können in einer Symboltabelle anstelle von Adressen lokale oder globale Symbole mit einer aussagekräftigen Bezeichnung definieren, um sie dann in Ihrem Programm zu verwenden (siehe auch Anlegen einer Symboltabelle).
- Programm erstellen
Mit einer der zur Verfügung stehenden Programmiersprachen erstellen Sie ein baugruppenzugeordnetes oder baugruppenunabhängiges Programm und hinterlegen dieses als Bausteine, Quellen oder Pläne (siehe auch Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen und Grundlagen zum Programmieren in AWL-Quellen).
- Nur S7: Referenzdaten erzeugen und auswerten
Sie können sich dieser Referenzdaten bedienen, um sich das Testen und Ändern Ihres Anwenderprogramms zu erleichtern (siehe auch Übersicht der möglichen Referenzdaten).
- Meldungen projektieren
Sie legen z. B. bausteinbezogene Meldungen mit ihren Texten und Attributen an. Mit dem Transferprogramm transferieren Sie die erstellten Meldungsprojektierungsdaten in den Datenbestand des Bedien- und Beobachtungssystems (z. B. SIMATIC WinCC, SIMATIC ProTool, (siehe auch Projektieren von Meldungen)).
- B+B-Variable projektieren
Sie legen einmalig in STEP 7 B+B-Variablen an und statten diese mit den gewünschten Attributen aus. Mit dem Transferprogramm transferieren Sie die erstellten B+B-Variablen in den Datenbestand des Bedien- und Beobachtungssystems WinCC (siehe auch Projektieren von bedien- und beobachtbaren Variablen).
- Programme auf Zielsystem laden
Nur S7: Nach Abschluss von Konfiguration, Parametrierung und Programmerstellung können Sie Ihr komplettes Anwenderprogramm oder einzelne Bausteine davon auf das Zielsystem (programmierbare Baugruppe Ihrer Hardwarelösung) übertragen (siehe auch Voraussetzungen für das Laden). Die CPU enthält bereits das Betriebssystem.
Nur M7: Unter verschiedenen Betriebssystemen wählen Sie das für Ihre Automatisierungslösung passende aus und übertragen dieses einzeln oder zusammen mit dem Anwenderprogramm auf den gewünschten Datenträger des M7-Zielsystems.
- Programme testen
Nur S7: Zum Testen haben Sie die Möglichkeit, sich Werte von Variablen aus Ihrem Anwenderprogramm oder einer CPU anzeigen zu lassen, diesen Werte zuzuweisen und eine Variablen-tabelle für die Variablen zu erstellen, die Sie anzeigen oder steuern wollen (siehe auch Einführung zum Testen mit der Variablen-tabelle).
Nur M7: Testen des Anwenderprogramms mit einem Hochsprachen-Debugger.

- **Betrieb überwachen, Hardware diagnostizieren**
Sie ermitteln die Ursache einer Baugruppenstörung, indem Sie sich Online-Informationen zu einer Baugruppe anzeigen lassen. Die Ursache für die Störung im Ablauf eines Anwenderprogramms ermitteln Sie mit Hilfe des Diagnosepuffers und der Stack-Inhalte. Darüber hinaus können Sie prüfen, ob ein Anwenderprogramm auf einer bestimmten CPU ablauffähig ist (siehe auch Diagnose der Hardware und Anzeigen des Baugruppenzustands).
- **Anlage dokumentieren**
Nach dem Erstellen eines Projekts/einer Anlage ist es sinnvoll, die Projektdaten übersichtlich zu dokumentieren, um die weitere Bearbeitung des Projekts als auch Servicearbeiten zu erleichtern (siehe auch Drucken einer Projektdokumentation). DOCPRO, das optionale Werkzeug zum Erstellen und Verwalten von Anlagendokumentation, ermöglicht die Strukturierung der Projektdaten, die Aufbereitung in Form von Schaltbüchern und den Ausdruck in einem einheitlichen Druckbild.

Erweiterung des Leitfadens um spezielle Themen

Bei der Erstellung einer Automatisierungslösung gibt es eine Reihe von speziellen Themen, die für Sie von Interesse sein können:

- **Multicomputing - Synchroner Betrieb mehrerer CPUs** (siehe auch Multicomputing - Synchroner Betrieb mehrerer CPUs).
- **Arbeiten von mehreren Personen in einem Projekt** (siehe auch Bearbeiten von Projekten durch mehrere Personen).
- **Arbeiten mit M7-Systemen** (siehe auch Vorgehensweise für M7-Systeme).

1.2 Das STEP 7 - Basispaket

Verwendete Standards

Die in STEP 7 integrierten SIMATIC-Programmiersprachen erfüllen die Norm DIN EN 6.1131-3. Das Basispaket läuft unter den Betriebssystemen MS Windows 2000 Professional (nachfolgend als Windows 2000 bezeichnet) sowie MS Windows XP Professional (nachfolgend als Windows XP bezeichnet) sowie MS Windows Server 2003 und ist an deren grafische und objektorientierte Arbeitsweise angepasst.

Funktionen des Basispakets

Das Basispaket unterstützt Sie in allen Phasen des Erstellungsprozesses von Automatisierungslösungen, wie z.B.:

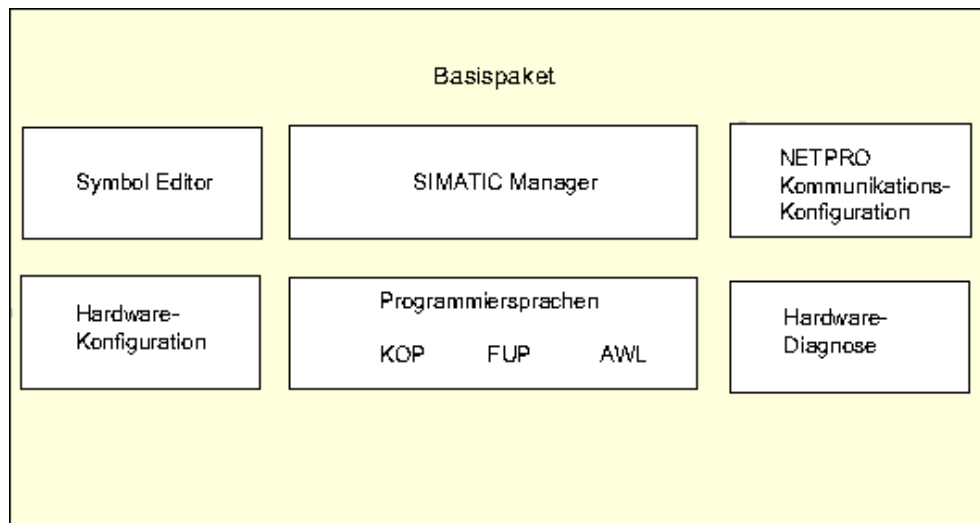
- Einrichten und Verwalten von Projekten,
- Konfigurieren und Parametrieren der Hardware und der Kommunikation,
- Verwaltung von Symbolen,
- Programmerstellung z. B. für S7-Zielsysteme,
- Laden von Programmen auf Zielsysteme,
- Test der Automatisierungsanlage,
- Diagnose bei Anlagenstörungen.

Die Benutzungsoberfläche der STEP 7-Software ist nach modernen ergonomischen Erkenntnissen gestaltet und ermöglicht einen leichten Einstieg.

Die Dokumentation zum Softwareprodukt STEP 7 bietet Ihnen die gesamte Information online in der Online-Hilfe und in elektronischen Handbüchern im PDF-Format.

Unterstützende Tools

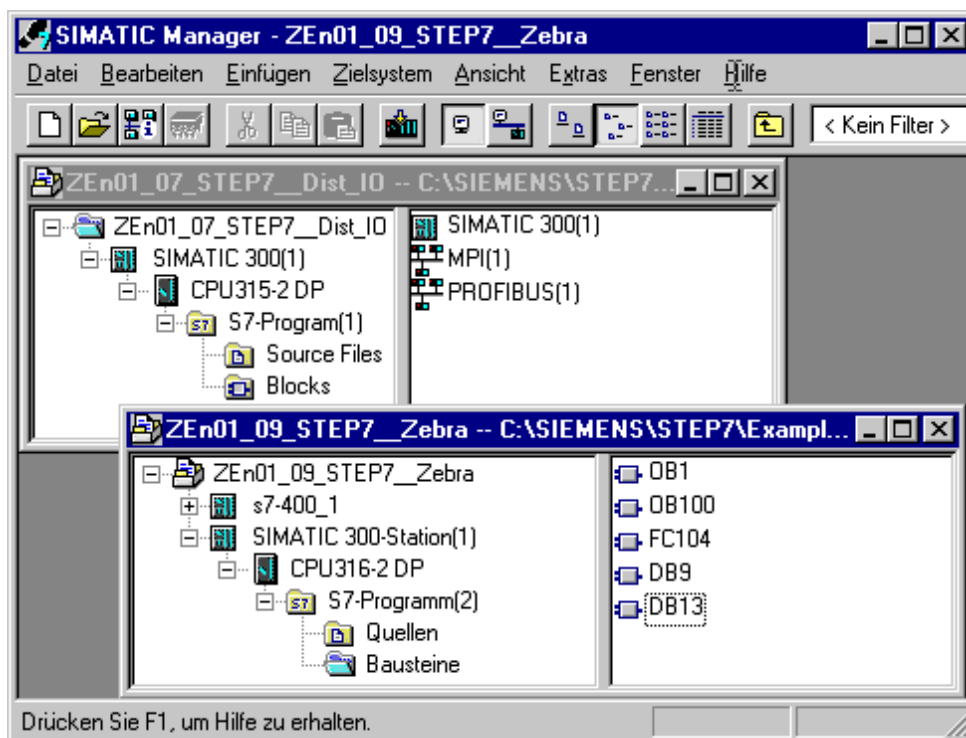
Das STEP 7-Basispaket stellt Ihnen eine Reihe von Tools zur Verfügung:



Die Tools müssen Sie nicht separat aufrufen, sie werden automatisch gestartet, wenn Sie eine zugehörige Funktion anwählen oder ein Objekt öffnen.

SIMATIC Manager

Der SIMATIC Manager verwaltet alle Daten, die zu einem Automatisierungsprojekt gehören – unabhängig davon, auf welchem Zielsystem (S7/M7/C7) sie realisiert sind. Die Tools, die zur Bearbeitung der angewählten Daten erforderlich sind, werden automatisch vom SIMATIC Manager gestartet.



Symbol Editor

Mit dem Symbol Editor verwalten Sie alle globalen Variablen. Es stehen folgende Funktionen zur Verfügung:

- Festlegen von symbolischen Bezeichnungen und Kommentaren zu den Prozesssignalen (Ein-/Ausgänge), Merkern und Bausteinen,
- Sortierfunktionen,
- Import/Export mit anderen Windows-Programmen.

Die dabei entstehende Symboltabelle steht allen Werkzeugen zur Verfügung. Die Änderung eines Symbolparameters wird deshalb automatisch von allen Tools erkannt.

Hardware-Diagnose

Die Hardware-Diagnose bietet Ihnen einen Überblick über den Zustand des Automatisierungssystems. In einer Übersichtsdarstellung kann für jede Baugruppe anhand eines Symbols angezeigt werden, ob sie gestört ist oder nicht. Durch Doppelklick auf die gestörte Baugruppe werden detaillierte Informationen zur Störung angezeigt. Der Umfang dieser Informationen ist abhängig von der einzelnen Baugruppe:

- Anzeige allgemeiner Informationen zur Baugruppe (z.B. Bestellnummer, Version, Bezeichnung) und des Zustands der Baugruppe (z. B. gestört),
- Anzeige der Baugruppenfehler (z. B. Kanalfehler) von zentraler Peripherie und DP-Slaves,
- Anzeige der Meldungen aus dem Diagnosepuffer.

Für CPUs wird zusätzlich Information angezeigt:

- Ursachen für Störung im Ablauf eines Anwenderprogramms,
- Anzeige der Zyklusdauer (längster, kürzester und letzter Zyklus),
- Möglichkeiten und Auslastung der MPI-Kommunikation,
- Anzeige der Leistungsdaten (Anzahl möglicher Ein-/Ausgänge, Merker, Zähler, Zeiten und Bausteine).

Programmiersprachen

Die Programmiersprachen KOP, AWL und FUP für S7-300/400 sind integraler Bestandteil des Basispakets.

- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Syntax der Anweisungen ähnelt einem Stromlaufplan. KOP ermöglicht Ihnen eine einfache Verfolgung des Signalfusses zwischen Stromschienen über Kontakte, komplexe Elemente und Spulen.
- AWL (Anweisungsliste) ist eine textuelle, maschinennahe Programmiersprache. Wird ein Programm in AWL programmiert, so entsprechen die einzelnen Anweisungen weitgehend den Arbeitsschritten, mit denen die CPU das Programm bearbeitet. Zur Erleichterung der Programmierung wurde AWL um einige Hochsprachenkonstrukte (wie z.B. strukturierte Datenzugriffe und Bausteinparameter) erweitert.
- FUP (Funktionsplan) ist eine grafische Programmiersprache und benutzt zur Darstellung der Logik die von der Booleschen Algebra bekannten logischen Boxen. Außerdem können komplexe Funktionen (z.B. mathematische Funktionen) direkt in Verbindung mit den logischen Boxen dargestellt werden.

Weitere Programmiersprachen sind als Optionssoftware erhältlich.

Hardware-Konfiguration

Dieses Tool setzen Sie zur Konfiguration und Parametrierung der Hardware eines Automatisierungsprojektes ein. Es stehen folgende Funktionen zur Verfügung:

- Um das Automatisierungssystem zu konfigurieren, wählen Sie Baugruppenträger (Racks) aus einem elektronischen Katalog aus und ordnen die ausgewählten Baugruppen den gewünschten Steckplätzen in den Racks zu.
- Die Konfiguration der Dezentralen Peripherie erfolgt identisch zur Konfiguration der zentralen Peripherie. Dabei wird auch kanalgranulare Peripherie unterstützt.
- Bei der Parametrierung der CPU können Sie Eigenschaften wie Anlaufverhalten und Zykluszeitüberwachung menügeführt einstellen. Multicomputing wird unterstützt. Die eingegebenen Daten werden in Systemdatenbausteinen abgelegt.
- Bei der Parametrierung der Baugruppen lassen sich alle einstellbaren Parameter über Dialogfelder festlegen. Einstellungen über DIP-Schalter entfallen. Die Parametrierung der Baugruppen erfolgt automatisch im Hochlauf der CPU. So ist z. B. ein Baugruppentausch ohne erneute Parametrierung möglich.
- Die Parametrierung von Funktionsmodulen (FMs) und Kommunikationsprozessoren (CPs) erfolgt ebenso innerhalb der Hardwarekonfiguration identisch zur Parametrierung der restlichen Baugruppen. Zu jedem FM und CP werden hierzu baugruppenspezifische Dialogfelder und Regeln zur Verfügung gestellt (im Lieferumfang FM-/CP-Funktionspaket). Das System verhindert fehlerhafte Eingaben, indem in den Dialogfeldern nur zulässige Eingabemöglichkeiten angeboten werden.

NetPro

Durch NetPro ist eine zeitgesteuerte zyklische Datenübertragung über MPI möglich mit:

- Auswahl der Kommunikationsteilnehmer,
- Eintragen von Datenquelle und Datenziel in eine Tabelle; das Generieren aller zu ladenden Bausteine (SDB) und ihr vollständiges Übertragen auf alle CPUs erfolgt automatisch.

Darüber hinaus ist eine ereignisgesteuerte Datenübertragung möglich mit:

- Festlegen der Kommunikationsverbindungen,
- Auswahl der Kommunikations-/Funktionsbausteine aus der integrierten Bausteinbibliothek,
- Parametrierung der ausgewählten Kommunikations-/Funktionsbausteine in der gewohnten Programmiersprache.

1.3 Was ist neu in STEP 7 Version 5.4

Die folgenden Themenbereiche wurden aktualisiert:

- SIMATIC Manager
- Hardware konfigurieren und diagnostizieren
- Netze und Verbindungen projektieren
- Standardbibliotheken
- Systemfehler_melden

SIMATIC Manager

- Ab STEP 7 V5.4 stehen Ihnen zwei Formate für die Anzeige von Datum und Uhrzeit zur Verfügung. Sie haben die Auswahl zwischen der Anzeige in der jeweiligen STEP 7-Landessprache und dem ISO 8601-Standard. Die Einstellung nehmen Sie im SIMATIC Manager vor, im Register "Datum und Uhrzeit" des Dialogfeldes "Einstellungen".
- Ab STEP 7 V5.4 können Sie sich die Baugruppenzeiten in der Lokalzeit Ihres PG/PC anzeigen lassen. Die Einstellung nehmen Sie im SIMATIC Manager vor, im Register "Datum und Uhrzeit" des Dialogfeldes "Einstellungen".
- Ab STEP 7 V5.4 gibt es die Möglichkeit, durch die Vergabe eines Projektpasswortes einen Zugriffsschutz für Projekte und Bibliotheken einzurichten. Dies setzt jedoch die Installation von SIMATIC Logon V1.3 SP1, nachfolgend SIMATIC Logon genannt, voraus (siehe Wissenswertes zum Zugriffsschutz).
- Ab STEP 7 V5.4 gibt es die Möglichkeit, nach dem Einrichten eines Zugriffsschutzes für Projekte und Bibliotheken, ein Änderungsprotokoll zu führen, das Online-Aktionen, wie z.B. „Laden“, „Betriebszustandsänderungen“ oder „Löschen“ mitprotokolliert. Dies setzt jedoch die Installation von SIMATIC Logon V1.3 SP1, nachfolgend SIMATIC Logon genannt, voraus (siehe Wissenswertes zum Änderungsprotokoll).

Hardware konfigurieren und diagnostizieren

- Das Verfahren "Information and Maintenance" wird unterstützt, um Identifikationsdaten von Baugruppen lesen bzw. auf Baugruppen schreiben zu können. Diese Funktion steht auch im SIMATIC Manager zur Verfügung (siehe Identification und Maintenance (I&M)).
- Das Schreiben von Identifikationsdaten ist auch für PROFIBUS DP-Anschaltungen im redundanten Betrieb möglich (über "Erreichbare Teilnehmer"). Die Anschaltung (IM) muss diese Funktion unterstützen.
- CAx-Daten können importiert und exportiert werden. So können gemeinsame Daten zwischen STEP 7 und z. B. CAD- oder CAE-Engineering-Systemen ausgetauscht werden (siehe CAx-Daten exportieren und importieren)
- Die Firmware von PROFIBUS DP-Anschaltungen (IMs) können Sie auch im redundanten Betrieb aktualisieren, sofern die IMs dafür geeignet sind. Jede der redundant eingesetzten IMs ist in der Lage, die aktualisierte Firmware über den aktiven Rückwandbus zur redundant eingesetzten IM automatisch weiter zu geben.
- Für die Funktion "Software Redundanz" können auch PA-Links mit unterlagerten PA-Slaves kopiert und redundant eingefügt werden (siehe Konfigurieren von SW-Redundanz)
- Applikationen zum Bearbeiten von Objekten in HW Konfig können mit dem Menübefehl Bearbeiten > Objekt öffnen gestartet werden (siehe Objekte öffnen in HW Konfig).
- Bei PROFINET IO-Devices können Sie eine Ansprechüberwachungszeit projektieren (siehe Projektieren der Ansprechüberwachungszeit)
- Ab STEP 7 V5.4 können Sie sich die Baugruppenzeiten in der Lokalzeit Ihres PG/PC anzeigen lassen.

Netze und Verbindungen projektieren

- PROFINET IO mit IRT-Kommunikation (Isochronous Realtime) wird unterstützt. Damit können auch für PROFINET IO kurze und gleichlange Buszykluszeiten projektiert werden (siehe Einführung: Isochronous Realtime Ethernet).
- Verbessertes Hantieren beim Einfügen kopierter IO-Devices in eine andere Station. Wenn die IP-Adressen bereits belegt sind, können Sie das Verhalten beim Einfügen bestimmen (Adressen beibehalten oder neu vergeben lassen).
- Ähnlich wie bei PROFIBUS DP-Slaves können Sie die Ansprechüberwachung nun bei PROFINET IO-Devices einstellen: als Objekteigenschaft des IO-Devices im Register "IO-Zyklus".
- Wenn Sie optische Komponenten für PROFIBUS DP verwenden: Sie können bei Konfigurationen mit optischem Ring die verwendeten OLMs spezifizieren. Die Berechnung der Busparameter wird dadurch genauer und bei Verwendung leistungsstärkerer Komponenten auch die Buszykluszeit verkürzt.

Standardbibliotheken

- Die Standardbibliothek "Communication Blocks" wird erweitert um die Bausteine FB 67 und FB 68 für die offene TCP/IP-Kommunikation.
- Die Standardbibliothek "Communication Blocks" wird erweitert um die Bausteine FB 20, FB 21, FB 22 und FB 23 zum zyklischen Zugriff auf Nutzdaten gemäß PNO.
- Ab STEP 7 V5.4 gibt es zusätzlich zur bisherigen auf Baugruppen ausgerichteten Redundanzbibliothek Redundant IO (V1) die neue Bausteinbibliothek Redundant IO CGP (channel granular peripheral devices). Diese unterstützt Redundanz für einzelne Baugruppenkanäle. Weitere Hinweise entnehmen Sie bitte der kontext-sensitiven Baustein-Hilfe bzw. der Readme zu STEP 7. Eine aktuelle Liste der unterstützten Baugruppen finden Sie als FAQ unter <http://support.automation.siemens.com/>.

Systemfehler melden

- Ab V5.4 wird ein Datenbaustein (DB 125) für PROFIBUS unterstützt, mit dem Diagnoseereignisse auf einem HMI-Gerät grafisch ausgegeben werden können.

1.4 Erweiterungsmöglichkeiten des STEP 7 - Basispakets

Das Basispaket kann durch optionale Software, die in den drei folgenden Softwareklassen zusammengefasst sind, erweitert werden.

- Engineering Tools;
diese umfassen höhere Programmiersprachen und technologieorientierte Software.
- Runtime Software;
enthält fertige Ablaufsoftware für den Produktionsprozess.
- Human Machine Interfaces (HMI);
steht für Software speziell zum Bedienen und Beobachten.

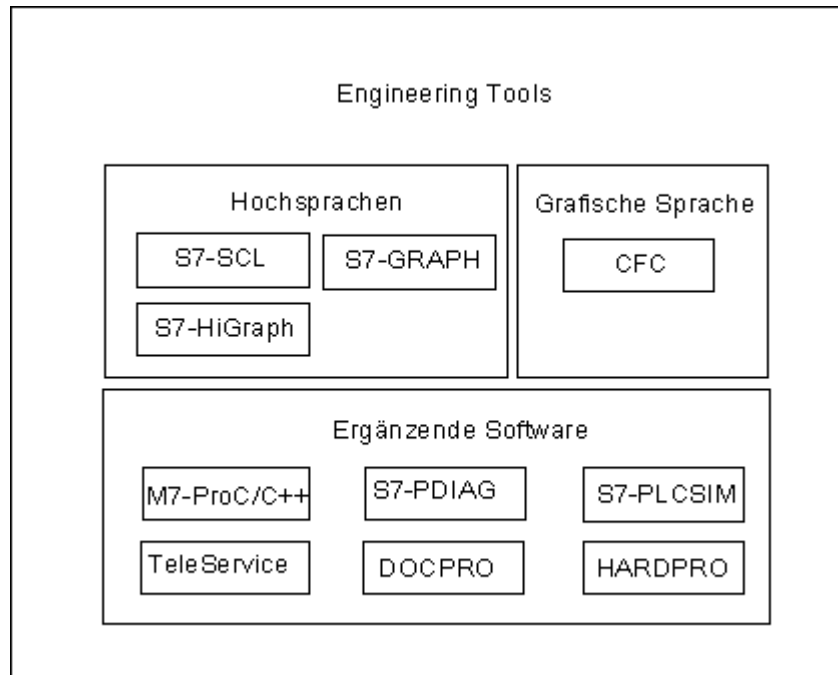
Nachfolgende Tabelle zeigt die einsetzbare Optionssoftware in Abhängigkeit von Automatisierungssystem:

	STEP 7		
	S7-300 S7-400	M7-300 M7-400	C7-620
Engineering Tools			
• Borland C/C++		o	
• CFC	+ ¹⁾	+	+ ²⁾
• DOCPRO	+	+ ³⁾	+
• HARDPRO	+		
• M7-ProC/C++		o	
• S7-GRAPH	+ ¹⁾		+ ²⁾
• S7-HiGraph	+		+
• S7-PDIAG	+		
• S7-PLCSIM	+		+
• S7-SCL	+		+
• TeleService	+	+	+
Runtime Software			
• Fuzzy Control	+		+
• M7-DDE-Server		+	
• M7-SYS RT		o	
• Modular PID Control	+		+
• PC-DDE-Server	+		
• PRODAVE MPI	+		
• Standard PID Control	+		+
Human Machine Interface			
• ProAgent			
• SIMATIC ProTool			
• SIMATIC ProTool/Lite			o
• SIMATIC WinCC			
o = unbedingt erforderlich + = optional ¹⁾ = empfohlen ab S7-400 ²⁾ = nicht empfohlen für C7-620 ³⁾ = nicht für C-Programme			

1.4.1 Engineering Tools

Engineering Tools sind aufgabenorientierte Tools, die als Erweiterung des Basispakets eingesetzt werden können. Engineering Tools umfassen:

- Hochsprachen für den Programmierer
- Grafische Sprache für den Technologen
- Ergänzende Software für Diagnose, Simulation, Fernwartung, Anlagendokumentation, etc.



Hochsprachen

Für die Programmierung der Automatisierungssysteme SIMATIC S7-300/400 stehen folgende Sprachen als Optionsoftware zur Verfügung:

- S7-GRAPH ist eine Programmiersprache zur komfortablen Beschreibung von Ablaufsteuerungen (Schrittkettenprogrammierung). Der Prozessablauf wird dabei in Schritte zerlegt. Die Schritte enthalten insbesondere Aktionen zum Steuern der Ausgänge. Der Übergang von einem Schritt zum nächsten wird durch Weiterschaltbedingungen gesteuert.
- S7-HiGraph ist eine Programmiersprache zur komfortablen Beschreibung asynchroner, nicht sequenzieller Vorgänge in Form von Zustandsgraphen. Dazu wird die Anlage in Funktionseinheiten gegliedert, die verschiedene Zustände einnehmen können. Die Funktionseinheiten können sich durch Austausch von Nachrichten synchronisieren.
- S7-SCL ist eine textuelle Hochsprache nach DIN EN 61131-3. Sie enthält Sprachkonstrukte, wie man sie in ähnlicher Form auch in den Programmiersprachen Pascal und C findet. S7-SCL eignet sich daher

besonders für Benutzer, die schon gewöhnt sind, mit einer höheren Programmiersprache zu arbeiten. S7-SCL ist beispielsweise einsetzbar zur Programmierung häufig wiederkehrender oder sehr komplexer Funktionen.

Grafische Sprache

CFC für S7 und M7 ist eine Programmiersprache zur grafischen Verschaltung vorhandener Funktionen. Die Funktionen decken einen weiten Bereich von einfachen Logikverknüpfungen bis hin zu komplexen Regelungen und Steuerungen ab. Eine große Anzahl solcher Funktionen steht in Form von Bausteinen in einer Bibliothek zur Verfügung. Sie programmieren, indem Sie Bausteine auf einen Plan kopieren und Bausteinanschlüsse durch Linien verbinden.

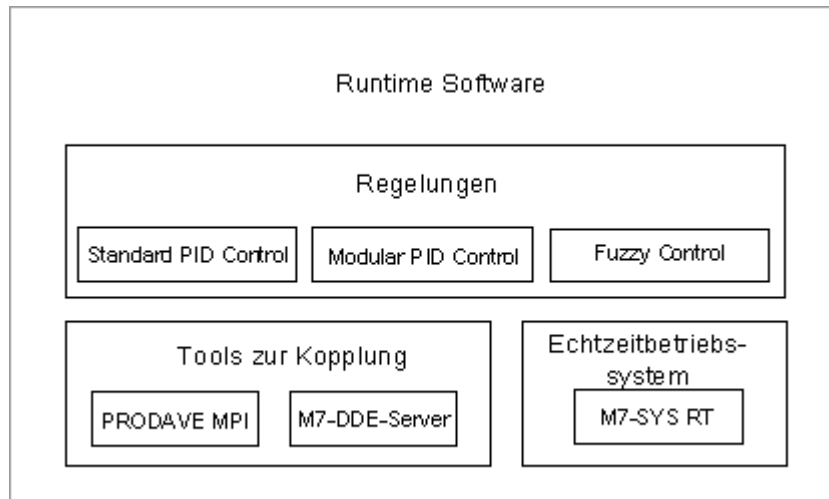
Ergänzende Software

- Borland C++ (nur M7) enthält die Borland Entwicklungsumgebung.
- Mit DOCPRO können Sie alle Projektierungsdaten, die mit STEP 7 erstellt wurden, in Schaltbüchern organisieren. Schaltbücher ermöglichen das bequeme Verwalten der Projektierungsdaten und eine normgerechte Druckaufbereitung.
- HARDPRO ist das Hardwareprojektierungssystem für S7-300 zur Anwenderunterstützung bei umfangreichen Projektierungen für komplexe Automatisierungsaufgaben.
- M7-ProC/C++ (nur M7) ermöglicht die Einbindung der Borland-Entwicklungsumgebung für die Programmiersprachen C und C++ in die STEP 7-Entwicklungsumgebung.
- Mit S7-PLCSIM (nur S7) können Sie zu Testzwecken an das Erstellsystem (PC/PG) angeschlossene S7-Automatisierungssysteme simulieren.
- S7-PDIAG (nur S7) ermöglicht die einheitliche Projektierung der Prozessdiagnose für SIMATIC S7-300/400. Mit Hilfe der Prozessdiagnose lassen sich fehlerhafte Zustände außerhalb des Automatisierungssystems erkennen (z.B. Endschalter nicht erreicht).
- TeleService bietet die Möglichkeit S7- und M7-Automatisierungssysteme mit dem PG/PC über das Telefonnetz zu programmieren und zu warten.

1.4.2 Runtime Software

Runtime Software umfasst fertig programmierte Lösungen, die durch das Anwenderprogramm aufgerufen werden. Runtime Software wird direkt in die Automatisierungslösung eingebaut. Sie umfasst:

- Regelungen für SIMATIC S7, z. B. Standard-, Modular- und Fuzzyregelungen
- Tools zur Kopplung der Automatisierungssysteme mit Windows-Applikationen
- ein Echtzeitbetriebssystem für SIMATIC M7



Regelungen für SIMATIC S7

- Standard PID Control ermöglicht die Integration von kontinuierlichen Reglern, Impulsreglern und Schrittreglern in das Anwenderprogramm. Das Parametriertool mit integrierter Reglereinstellung erlaubt es, den Regler in kürzester Zeit zu parametrieren und optimal einzustellen.
- Modular PID Control kommt dann zum Einsatz, wenn ein einfacher PID-Regler zur Lösung der Automatisierungsaufgabe nicht ausreicht. Durch Verschaltung der mitgelieferten Standard-Funktionsbausteine lässt sich nahezu jede regelungstechnische Struktur realisieren.
- Mit Fuzzy Control können Fuzzy-Systeme erstellt werden. Diese Systeme kommen dann zum Einsatz, wenn Prozesse mathematisch nur schwer oder gar nicht zu beschreiben sind, Abläufe und Prozesse unvorhersehbar verlaufen, Nichtlinearitäten auftreten, andererseits aber Erfahrungswissen im Umgang mit dem Prozess vorliegt.

Tools zur Kopplung

- PRODAVE MPI ist eine Toolbox für den Prozessdatenverkehr zwischen SIMATIC S7, SIMATIC M7 und SIMATIC C7. Sie wickelt den Datenverkehr selbstständig über die MPI-Schnittstelle ab.
- Mit dem M7-DDE-Server (**D**ynamic **D**ata **E**xchange) lassen sich Windows-Applikationen ohne zusätzlichen Programmieraufwand an Prozessvariablen auf SIMATIC M7 anbinden.

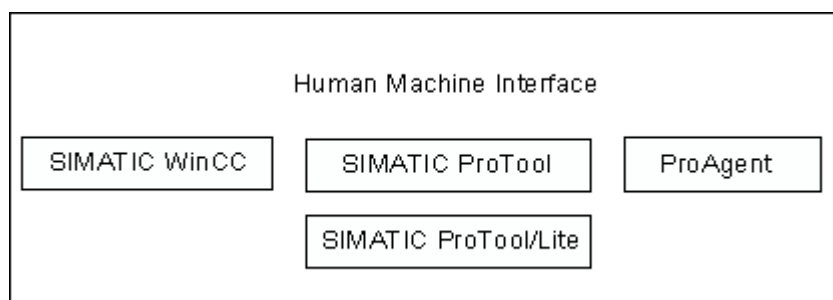
Echtzeitbetriebssystem

- M7-SYS RT enthält das Betriebssystem M7 RMOS 32 und Systemprogramme. Es ist Voraussetzung für den Einsatz der Pakete M7-ProC/C++ und CFC für SIMATIC M7.

1.4.3 Human Machine Interface

Human Machine Interface ist Software speziell zum Bedienen und Beobachten für SIMATIC.

- Das offene Prozessvisualisierungssystem SIMATIC WinCC ist ein branchen- und technologieunabhängiges Basissystem mit allen wichtigen Funktionen zum Bedienen und Beobachten.
- SIMATIC ProTool und SIMATIC ProTool/Lite sind moderne Tools zur Projektierung der SIMATIC Operator Panels und der SIMATIC C7-Kompaktgeräte.
- ProAgent ermöglicht die gezielte und schnelle Prozessdiagnose in Anlagen und Maschinen durch die Ermittlung von Informationen zu Fehlerort und Fehlerursache.



2 Installation

2.1 Automation License Manager

2.1.1 Nutzungsberechtigung durch den Automation License Manager

Automation License Manager

Für die Nutzung der STEP 7-Programmiersoftware wird ein produktspezifischer License Key (Nutzungsberechtigung) benötigt, dessen Installation ab der V5.3 von STEP 7 mit dem Automation License Manager durchgeführt wird.

Der Automation License Manager ist ein Software-Produkt der Siemens AG. Er wird systemübergreifend zur Handhabung von License Keys (technische Repräsentanten von Lizenzen) eingesetzt.

Den Automation License Manager finden Sie:

- auf dem Installationsmedium des jeweiligen Software-Produktes, für den ein License Key benötigt wird, oder
- auf einem separaten Installationsmedium sowie
- auf den Internetseiten des A&D Customer Support der Siemens AG als WebDownload.

Im Automation License Manager ist eine Online-Hilfe integriert, die Sie nach der Installation kontextsensitiv über die F1-Taste oder über den Menübefehl **Hilfe > Hilfe zum License Manager** aufrufen können. In dieser Hilfe erhalten Sie detaillierte Informationen zur Funktionalität und Handhabung des Automation License Managers.

Lizenzen

Für die Nutzung von lizenzrechtlich geschützten Programmpaketten von STEP 7 werden Lizenzen benötigt. Eine Lizenz wird als Recht zur Nutzung von Produkten vergeben. Die Repräsentanten dieses Rechtes sind:

- das CoL (**C**ertificate **o**f **L**icense) und
- der License Key.

Certificate of License (CoL)

Das im Lieferumfang der jeweiligen Produkte enthaltene "Certificate of License" ist der juristische Nachweis des Nutzungsrechtes. Das jeweilige Produkt darf nur durch den Besitzer des CoL oder beauftragte Personen genutzt werden.

License Keys

Der License Key ist der technische Repräsentant einer Lizenz (elektronischer Lizenzstempel).

Für jede Software, die lizenzrechtlich geschützt ist, wird von der SIEMENS AG ein License Key vergeben. Erst wenn nach dem Starten der Software auf einem Rechner das Vorhandensein eines gültigen License Keys festgestellt wurde, kann die jeweilige Software entsprechend der mit diesem License Key verbundenen Lizenz- und Nutzungsbedingungen genutzt werden.

Hinweise

- Sie können die Basissoftware von STEP 7 zum kurzen kennen lernen von Bedienoberfläche und Funktionsumfang auch ohne License Key verwenden.
 - Die uneingeschränkte Nutzung unter Berücksichtigung der lizenzrechtlichen Vereinbarungen ist jedoch nur mit installiertem License Key zulässig und möglich.
 - Wenn Sie den License Key **nicht** installiert haben, werden Sie in regelmäßigen Abständen aufgefordert, die Installation vorzunehmen.
-

License Keys können wie folgt abgelegt sein und zwischen den einzelnen Speichermedien transferiert werden:

- auf License Key-Disketten,
- auf lokalen Festplattenspeichern und
- auf Festplattenspeichern von Rechnern im Netzwerk.

Wenn Software-Produkte installiert sind, für die kein License Key verfügbar ist, so können Sie den License Key-Bedarf ermitteln und anschließend die notwendigen Lizenzen bestellen.

Weiterführende Informationen über die Handhabung von License Keys entnehmen Sie bitte der Online-Hilfe zum Automation License Manager.

Lizenz-Typen

Für Software-Produkte der Siemens AG wird zwischen folgenden anwendungsorientierten Lizenz-Typen unterschieden. Das Verhalten der Software wird durch die für diese Lizenz-Typen unterschiedlichen License Keys gesteuert. Die Art der Nutzung ergibt sich aus dem jeweiligen Certificate of License.

Lizenz-Typ	Beschreibung
Single License	Die Nutzung der Software ist zeitlich unbegrenzt auf einem beliebigen Rechner zulässig.
Floating License	Zeitlich unbegrenzte, auf Nutzung über ein Netzwerk bezogenes Nutzungsrecht ("remote"Nutzung) einer Software.
Trial License	Die Nutzung der Software ist beschränkt auf: <ul style="list-style-type: none"> • eine Gültigkeit von maximal 14 Tagen, • eine bestimmte Anzahl von Tagen ab der Erstnutzung, • die Nutzung für Tests und zur Validierung (Haftungsausschluss).
Rental License	Die Nutzung der Software ist beschränkt auf: <ul style="list-style-type: none"> • eine Gültigkeit von maximal 50 Tagen • eine bestimmte Anzahl von Stunden bei Nutzung
Upgrade License	Für ein Upgrade können spezifische Anforderungen an den Systemzustand gefordert sein: <ul style="list-style-type: none"> • Mit einer Upgrade License kann eine Lizenz einer "Alt-"Version x auf eine Version >x+... umgestellt werden. • Ein Upgrade kann z. B. durch eine Erweiterung des Mengengerüsts notwendig sein.

2.1.2 Installieren des Automation License Managers

Der Automation License Manager wird über ein MSI-Setup installiert. Die Installations-Software für den Automation License Manager finden Sie auf der STEP 7-Produkt-CD.

Sie können den Automation License Manager im Zusammenhang mit STEP 7 oder erst zu einem späteren Zeitpunkt installieren.

Hinweise

- Detaillierte Informationen zur Vorgehensweise beim Installieren des Automation License Managers entnehmen Sie bitte der aktuellen Liesmich.wri.
 - In der Online-Hilfe zum Automation License Manager erhalten Sie alle benötigten Informationen zur Funktionalität und Handhabung von License Keys.
-

License Keys später installieren

Wenn Sie die STEP 7-Software starten und keine License Keys vorhanden sind, so erhalten Sie eine entsprechende Meldung.

Hinweise

- Sie können die Basissoftware von STEP 7 zum kurzen kennen lernen von Bedienoberfläche und Funktionsumfang auch ohne License Key verwenden.
 - Die uneingeschränkte Nutzung unter Berücksichtigung der lizenzrechtlichen Vereinbarungen ist jedoch nur mit installiertem License Key zulässig und möglich.
 - Wenn Sie den License Key **nicht** installiert haben, werden Sie in regelmäßigen Abständen aufgefordert, die Installation vorzunehmen.
-

Zum nachträglichen Installieren von License Keys haben Sie folgende Möglichkeiten:

- Installieren der License Keys von Disketten
- Installieren der License Keys über WebDownLoad (vorherige Bestellung erforderlich)
- Nutzung von im Netzwerk vorhandenen Floating License Keys.

Detaillierte Informationen zur Vorgehensweise entnehmen Sie bitte der Online-Hilfe zum Automation License Manager, die Sie nach der Installation kontextsensitiv über die F1-Taste oder über den Menübefehl **Hilfe > Hilfe zum License Manager** aufrufen können.

Hinweise

- License Keys sind unter Windows 2000/XP/Server 2003 nur dann funktionsfähig, wenn sie auf einem Festplattenlaufwerk liegen, auf dem schreibende Zugriffe zugelassen sind.
- Floating Licenses können auch über ein Netzwerk, also "remote" genutzt werden.

2.1.3 Regeln für den Umgang mit License Keys



Vorsicht

Beachten Sie die Hinweise zum Umgang mit License Keys, die in der Online-Hilfe zum Automation License Manager beschrieben sind und in der STEP 7-Liesmich.wri auf der CD-ROM. Bei Nichtbeachtung besteht die Gefahr, dass License Keys unwiderruflich verloren gehen.

Die Online-Hilfe zum Automation License Manager können Sie kontextsensitiv über die F1-Taste oder über den Menübefehl **Hilfe > Hilfe zum Automation License Manager** aufrufen.

In dieser Hilfe erhalten Sie alle benötigten Informationen zur Funktionalität und Handhabung von License Keys.

2.2 Installieren von STEP 7

STEP 7 enthält ein Setup-Programm, das die Installation automatisch durchführt. Eingabeaufforderungen auf dem Bildschirm führen Sie Schritt für Schritt durch den gesamten Installationsvorgang. Es wird mit der unter Windows 2000/XP/Server 2003 üblichen Standardprozedur zur Installation von Software aufgerufen.

Die wesentlichen Phasen der Installation sind:

- das Kopieren der Daten auf Ihr Erstellsystem,
- das Einrichten der Treiber für EPROM und Kommunikation,
- das Installieren der License Keys (falls gewünscht).

Hinweis

Siemens Programmiergeräte werden mit installierbarer STEP 7-Software auf der Festplatte ausgeliefert.

Installationsvoraussetzungen

- Betriebssystem
Microsoft Windows 2000, Windows XP, Windows Server 2003.
- Basishardware
PC oder Programmiergerät mit
 - Pentium-Prozessor (600 MHz)
 - RAM-Speicherausbau: mindestens 256 MB.
 - Farbmonitor, Tastatur und Maus, die von Microsoft Windows unterstützt werden.

Ein Programmiergerät (PG) ist ein Personal Computer in spezieller industrietauglicher und kompakter Ausführung. Es ist komplett ausgestattet für die Programmierung der SIMATIC-Automatisierungssysteme.

- Speicherkapazität
Erforderlicher Speicherplatz auf der Festplatte siehe LIESMICH.WRI.
- MPI-Schnittstelle (optional)
Die MPI-Schnittstelle zwischen Erstellsystem (Programmiergerät oder PC) und Zielsystem ist nur erforderlich, wenn Sie unter STEP 7 über MPI mit dem Zielsystem kommunizieren wollen.
Dazu verwenden Sie entweder
 - einen PC-Adapter USB, der an die Kommunikationsschnittstelle Ihres Geräts angeschlossen wird oder
 - eine MPI-Baugruppe (z. B. CP 5611), die in Ihrem Gerät installiert wird.

Bei Programmiergeräten ist die MPI-Schnittstelle bereits eingebaut.

- Externer Prommer (optional)
Ein externer Prommer ist bei Einsatz eines PCs nur dann erforderlich, wenn Sie EPROMs schießen wollen.

Hinweise

Beachten Sie auch die Hinweise zum Installieren von STEP 7 in der Datei LIESMICH.WRI und die "Liste der Verträglichkeit von SIMATIC-Softwarepaketen mit den Versionen des STEP 7-Basispaketes".

Die Liesmich-Datei finden Sie in der Startleiste unter **Start > Simatic > Produkt-Hinweise**.

Die Verträglichkeitsliste finden Sie in der Startleiste unter **Start > Simatic > Dokumentation**.

2.2.1 Vorgehen beim Installieren

Vorbereitungen

Bevor Sie mit der Installation beginnen können, muss das Betriebssystem (Windows 2000, XP oder Server 2003) gestartet sein.

- Sie benötigen keine externen Datenträger, wenn sich die installierbare STEP 7-Software bereits auf der Festplatte des PG befindet.
- Um von CD-ROM zu installieren, legen Sie die CD-ROM in das CD-ROM-Laufwerk ihres PC.

Starten des Installationsprogramms

Gehen Sie zur Installation wie folgt vor:

1. Legen Sie die CD-ROM ein und starten Sie das Setup durch Doppelklick auf die Datei "setup.exe".
2. Befolgen Sie Schritt für Schritt die Anweisungen, die Ihnen das Installationsprogramm anzeigt.

Das Programm führt Sie schrittweise durch den Installationsprozess. Sie können jeweils zum nachfolgenden oder vorhergehenden Schritt weiterschalten.

Während des Installationsvorgangs werden Ihnen in Dialogfeldern Fragen angezeigt oder Optionen zur Auswahl angeboten. Lesen Sie bitte die folgenden Hinweise, um die Dialoge schneller und leichter beantworten zu können.

Falls schon eine STEP 7-Version installiert ist ...

Wenn das Installationsprogramm feststellt, dass sich bereits eine STEP 7-Installation auf dem Erstellsystem befindet, wird eine entsprechende Meldung angezeigt, und Sie haben folgende Wahlmöglichkeiten:

- Installation abbrechen (um danach alte STEP 7-Version unter Windows zu deinstallieren und anschließend die Installation erneut zu starten) oder
- Installation fortsetzen und damit alte Version durch neue Version überschreiben.

Hinsichtlich einer sauberen Softwarepflege sollten Sie vor einer Installation eine eventuell vorhandene ältere Version deinstallieren. Das einfache Überschreiben einer älteren Version hat außerdem den Nachteil, dass bei einem anschließenden Deinstallieren die eventuell noch vorhandenen Teile aus einer älteren Installation nicht entfernt werden.

Zum Installationsumfang

Zur Festlegung des Installationsumfangs haben Sie drei Auswahlmöglichkeiten:

- Maximalkonfiguration: alle Sprachen der Benutzungsoberfläche, alle Applikationen und alle Beispiele. Den dazu benötigten Speicherplatz entnehmen Sie bitte der aktuellen Produktinformation.
- Minimalkonfiguration: nur eine Sprache, keine Beispiele. Den dazu benötigten Speicherplatz entnehmen Sie bitte der aktuellen Produktinformation.
- Benutzerdefiniert: Sie können den Installationsumfang wählen hinsichtlich Programme, Datenbasis, Beispiele und Kommunikation.

Zur Ident-Nummer

Sie werden bei der Installation nach einer Ident-Nummer gefragt. Tragen Sie die Ident-Nummer ein. Sie finden diese auf dem Software-Produktschein oder auf der zugehörigen License Key-Diskette.

Zum Installieren von License Keys

Bei der Installation wird überprüft, ob ein entsprechender License Key auf der Festplatte vorhanden ist. Wird kein gültiger License Key erkannt, so erscheint ein Hinweis, dass die Software nur mit vorhandenem License Key benutzt werden kann. Wenn Sie es wünschen, können Sie gleich die License Keys installieren oder aber die Installation von STEP 7 fortsetzen und die License Keys zu einem späteren Zeitpunkt nachinstallieren. Im erstgenannten Fall legen Sie die mitgelieferte License Key-Diskette ein, wenn Sie dazu aufgefordert werden.

Zur PG/PC-Schnittstelleneinstellung

Während des Installationsvorgangs wird ein Dialog zur Einstellung der PG/PC-Schnittstelle angezeigt. Lesen Sie dazu "Einstellen der PG/PC-Schnittstelle".

Zur Memory Card-Parametrierung

Während des Installationsvorgangs wird ein Dialog zur Memory Card-Parametrierung angezeigt.

- Wenn Sie keine Memory Cards einsetzen, benötigen Sie keinen EPROM-Treiber. Wählen Sie die Option "kein EPROM-Treiber".
- Ansonsten wählen Sie den Eintrag, der zu Ihrem PG gehört.
- Wenn Sie einen PC einsetzen, so können Sie einen Treiber für einen externen Prommer auswählen. Dabei müssen Sie zusätzlich die Schnittstelle angeben, an der der Prommer angeschlossen ist (z. B. LPT1).

Sie können die eingestellten Parameter nach der Installation ändern, wenn Sie das Programm "Memory Card parametrieren" in der STEP 7-Programmgruppe oder der Systemsteuerung aufrufen.

Zum Flash-File-System

Im Dialog zur Memory Card-Parametrierung können Sie angeben, ob ein Flash-File-System installiert werden soll.

Das Flash-File-System wird z. B. benötigt, wenn Sie bei SIMATIC M7 einzelne Dateien auf eine EPROM-Memory Card schreiben oder dort löschen wollen, ohne den restlichen Inhalt der Memory Card zu verändern.

Wenn Sie ein geeignetes Programmiergerät (PG 720/740/760, Field PG und Power PG) oder einen geeigneten externen Prommer einsetzen und diese Funktionalität nutzen wollen, so wählen Sie die Installation des Flash-File-Systems.

Fehler während der Installation

Folgende Fehler führen zum Abbruch der Installation:

- Wenn sofort nach dem Start von Setup ein Initialisierungsfehler auftritt, so wurde höchst wahrscheinlich *setup* nicht unter Windows gestartet.
- Speicherplatz reicht nicht aus: Sie benötigen abhängig vom Installationsumfang ca. 100 MB freien Speicherplatz auf Ihrer Festplatte für die Basissoftware.
- Defekte CD: Wenn Sie feststellen, dass die CD defekt ist, wenden Sie sich bitte an Ihre SIEMENS-Vertretung.
- Bedienungsfehler: Beginnen Sie die Installation erneut und beachten Sie die Anweisungen sorgfältig.

Zum Abschluss der Installation ...

Wenn die Installation erfolgreich war, wird dies durch eine entsprechende Meldung auf dem Bildschirm angezeigt.

Falls bei der Installation Systemdateien aktualisiert wurden, werden Sie aufgefordert, Windows neu zu starten. Nach dem Neustart (Warmstart) können Sie die Oberfläche von STEP 7, den SIMATIC Manager starten.

Nach einer erfolgreichen Installation ist eine Programmgruppe für STEP 7 eingerichtet.

2.2.2 Einstellen der PG/PC-Schnittstelle

Mit den hier gemachten Einstellungen legen Sie die Kommunikation zwischen PG/PC und dem Automatisierungssystem fest. Während des Installationsvorgangs wird ein Dialog zur Einstellung der PG/PC-Schnittstelle angezeigt. Sie können sich den Dialog auch nach der Installation anzeigen lassen, indem Sie das Programm "PG/PC-Schnittstelle einstellen" in der STEP 7-Programmgruppe oder der Systemsteuerung aufrufen. Dadurch ist es möglich, die Schnittstellenparameter auch später unabhängig von einer Installation zu ändern.

Prinzipielles Vorgehen

Zum Betrieb einer Schnittstelle sind erforderlich:

- Einstellungen im Betriebssystem
- eine geeignete Schnittstellenparametrierung

Wenn Sie einen PC mit MPI-Karte oder Kommunikationsprozessoren (CP) einsetzen, so sollten Sie über die "Systemsteuerung" von Windows die Interrupt- und Adressbelegung prüfen, um sicherzustellen, dass es zu keinen Interrupt-Konflikten oder Adressbereichsüberschneidungen kommt.

Unter Windows 2000, Windows XP und Windows Server 2003 wird die ISA-Komponente MPI-ISA-Card nicht mehr unterstützt und nicht mehr zur Installation angeboten.

Um die Parametrierung der PG/PC-Schnittstelle zu vereinfachen, werden Ihnen in Dialogfeldern vordefinierte Sätze von Grundparametern (Schnittstellenparametrierungen) zur Auswahl angeboten.

Parametrieren der PG/PC-Schnittstelle

Gehen Sie folgendermaßen vor (ausführliche Beschreibung in der Online-Hilfe):

1. Doppelklicken Sie in der "Systemsteuerung" von Windows auf "PG/PC-Schnittstelle einstellen".
2. Stellen Sie den "Zugangspunkt der Applikation" auf "S7ONLINE".
3. Wählen Sie in der Liste "Benutzte Schnittstellenparametrierung" die gewünschte Schnittstellenparametrierung aus. Wird die von Ihnen gewünschte Schnittstellenparametrierung nicht angezeigt, müssen Sie zunächst über die Schaltfläche "Auswählen" eine Baugruppe bzw. ein Protokoll installieren. Die Schnittstellenparametrierung wird dann automatisch erstellt. Bei Plug&Play-Systemen können die Plug&Play-fähigen CPs (CP 5611 und CP 5511) nicht manuell installiert werden. Sie werden automatisch in "PG/PC-Schnittstelle einstellen" integriert, wenn Sie die Hardware in Ihrem PG/PC eingebaut haben.
 - Wenn Sie eine Schnittstelle **mit automatischer Erkennung der Busparameter** wählen (z. B. CP 5611 (Auto)), dann können Sie das PG bzw. den PC an MPI bzw. PROFIBUS anschließen, ohne Busparameter einstellen zu müssen. Bei Übertragungsgeschwindigkeiten kleiner 187,5 kBit/s können allerdings Wartezeiten bis zu einer Minute entstehen.
Voraussetzung für die automatische Erkennung: Am Bus sind Master angeschlossen, die zyklisch Busparameter verteilen. Alle neuen MPI-

Komponenten tun das; bei PROFIBUS-Subnetzen darf das zyklische Verteilen der Busparameter nicht ausgeschaltet sein (voreingestellte PROFIBUS-Netzeinstellung).

- Wenn Sie eine Schnittstelle **ohne automatische Erkennung der Busparameter** wählen, dann lassen Sie sich die Eigenschaften anzeigen und passen sie an das Subnetz an.

Änderungen sind auch dann erforderlich, wenn sich Konflikte mit anderen Einstellungen ergeben (z. B. Interrupt- oder Adressbelegungen). Nehmen Sie in diesem Fall die entsprechenden Änderungen mit der Hardwareerkennung und der Systemsteuerung von Windows vor (siehe unten).



Vorsicht

Eine eventuell angezeigte Schnittstellenparametrierung "TCP/IP" **nicht** entfernen!
Dies könnte den Ablauf von anderen Anwendungen beeinträchtigen.

Prüfen der Interrupt- und Adressbelegung

Wenn Sie einen PC mit MPI-Karte einsetzen, so sollten Sie in jedem Fall prüfen, ob der voreingestellte Interrupt und der voreingestellte Adressbereich frei sind.

2.3 Deinstallieren von STEP 7

Benutzen Sie das unter Windows übliche Verfahren zur Deinstallation:

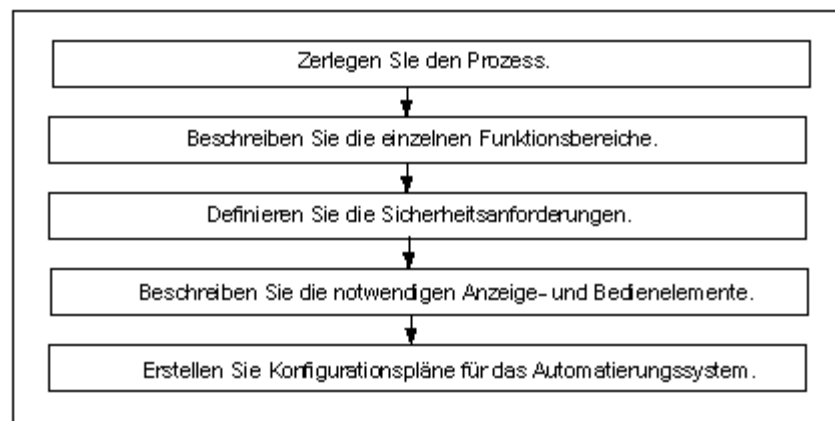
1. Starten Sie unter Windows den Dialog zu Installation von Software durch Doppelklick auf das Symbol "Software" in "Systemsteuerung".
2. Markieren Sie den STEP 7-Eintrag in der angezeigten Liste der installierten Software. Betätigen Sie dann die Schaltfläche zum "Entfernen" der Software.
3. Falls Dialogfelder "Freigegebene Datei entfernen" erscheinen, so klicken Sie im Zweifelsfall auf die Schaltfläche "Nein":

3 Konzeption der Automatisierungslösung

3.1 Prinzipielle Vorgehensweise zur Konzeption einer Automatisierungslösung

Dieses Kapitel enthält Informationen zu den grundlegenden Aufgaben beim Planen einer Automatisierungslösung für ein Automatisierungssystem (AS). Anhand eines Beispiels zur Automatisierung eines industriellen Mischprozesses wird beschrieben, wie Sie Schritt für Schritt dabei vorgehen können.

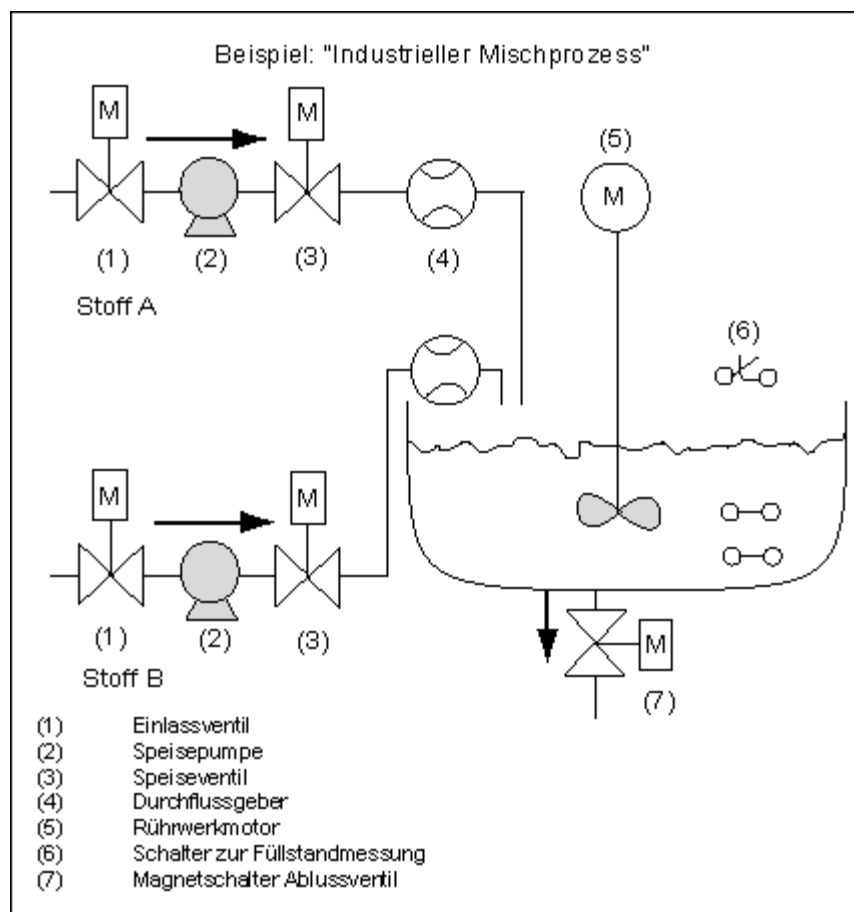
Es gibt viele Methoden, eine Automatisierungslösung vorzubereiten. Die grundsätzliche Vorgehensweise, die Sie für jedes beliebige Projekt verwenden können, ist im nachfolgenden Bild dargestellt.



3.2 Zerlegen des Prozesses in Aufgaben und Bereiche

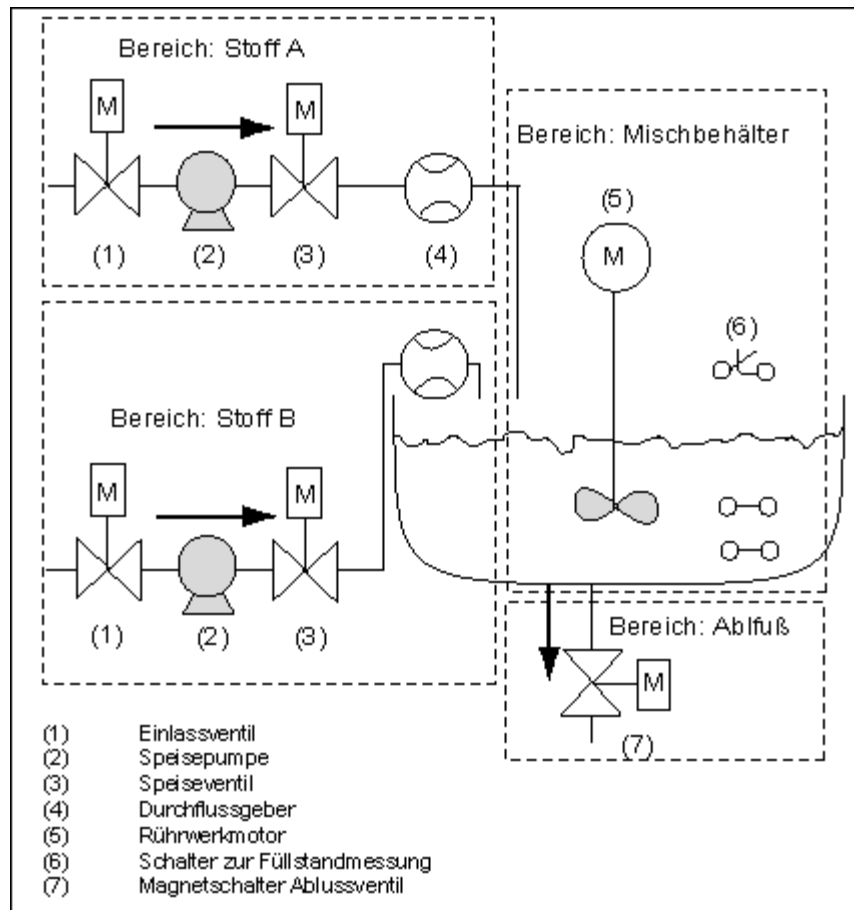
Ein Automatisierungsprozess besteht aus einzelnen Aufgaben. Selbst der komplizierteste Prozess kann definiert werden, wenn Sie zusammenhängende Aufgabenbereiche innerhalb des Prozesses bestimmen und diese in kleinere Teilaufgaben zerlegen.

Im folgenden Beispiel wird anhand eines industriellen Mischprozesses gezeigt, wie Sie einen Prozess in Funktionsbereiche und einzelne Aufgaben strukturieren können:



Bestimmen der Bereiche eines Prozesses

Nachdem Sie den Prozess, der gesteuert werden soll, definiert haben, unterteilen Sie das Projekt in zusammengehörige Gruppen bzw. Bereiche:



Da jeder Bereich in kleinere Aufgaben unterteilt wird, sind die Aufgaben zur Steuerung dieses Prozessabschnitts nicht sehr kompliziert.

Im Beispiel des industriellen Mischprozesses können Sie vier Bereiche bestimmen (siehe nachfolgende Tabelle). In diesem Beispiel enthält der Bereich für Stoff A die gleichen Geräte wie der Bereich für Stoff B.

Funktionsbereich	Zugehörige Geräte
Stoff A	Speisepumpe für Stoff A Einlassventil für Stoff A Speiseventil für Stoff A Durchflussgeber für Stoff A
Stoff B	Speisepumpe für Stoff B Einlassventil für Stoff B Speiseventil für Stoff B Durchflussgeber für Stoff B
Mischbehälter	Rührwerkmotor Schalter zur Füllstandsmessung
Abfluss	Abflussventil

3.3 Beschreiben der einzelnen Funktionsbereiche

Wenn Sie jeden Bereich und jede Aufgabe in Ihrem Prozess beschreiben, dann definieren Sie nicht nur die Funktionsweise jedes Bereichs, sondern auch die verschiedenen Elemente, die diesen Bereich steuern. Diese umfassen:

- Elektrische, mechanische und logische Eingänge und Ausgänge für jede Aufgabe
- Verriegelungen und Abhängigkeiten zwischen den einzelnen Aufgaben

In dem Beispiel des industriellen Mischprozesses werden Pumpen, Motoren und Ventile eingesetzt. Diese müssen genau beschrieben werden, um die Betriebsmerkmale und die Art der Verriegelungen festzulegen, die während des Betriebs erforderlich sind. In den nachfolgenden Tabellen finden Sie Beispiele für die Beschreibung der Geräte, die in dem industriellen Mischprozess eingesetzt werden. Diese Beschreibung können Sie auch für die Beschaffung der benötigten Geräte verwenden.

Stoff A/B: Speisepumpenmotoren
Die Speisepumpen fördern die Stoffe A und B in den Mischbehälter. <ul style="list-style-type: none"> • Durchflussleistung: 400 l pro Minute • Leistung: 100 KW bei 1200 UPM
Die Pumpen werden von einem Bedienpult gesteuert (Starten/Stoppen), das sich in der Nähe des Mischbehälters befindet. Die Anzahl der Starts wird zu Wartungszwecken gezählt. Der Zähler und die Anzeige können mit einem gemeinsamen Taster zurückgesetzt werden.
Es gelten die folgenden Freigabebedingungen: <ul style="list-style-type: none"> • Der Mischbehälter ist nicht voll. • Das Abflussventil des Mischbehälters ist geschlossen. • NOT-AUS ist nicht aktiviert.
Es gilt die folgende Abschaltbedingung: <ul style="list-style-type: none"> • Durchflussgeber meldet 7 s nach Start des Pumpenmotors keinen Durchfluss. • Durchflussgeber meldet während des Laufs keinen Durchfluss mehr.

Stoff A/B: Einlass- und Speiseventile
Die Einlass- und Speiseventile für die Stoffe A und B ermöglichen/verhindern den Zufluss der Stoffe in den Mischbehälter. Die Ventile haben einen Magnetschalter mit Federrückholeinrichtung. <ul style="list-style-type: none"> • Wenn der Magnetschalter aktiviert ist, ist das Ventil geöffnet. • Wenn der Magnetschalter deaktiviert ist, ist das Ventil geschlossen.
Die Einlass- und Speiseventile werden vom Anwenderprogramm gesteuert.
Es gilt die folgende Freigabebedingung: <ul style="list-style-type: none"> • Der Speisepumpenmotor ist mindestens 1 Sekunde angesteuert.
Es gilt die folgende Abschaltbedingung: <ul style="list-style-type: none"> • Der Durchflussgeber meldet keinen Durchfluss.

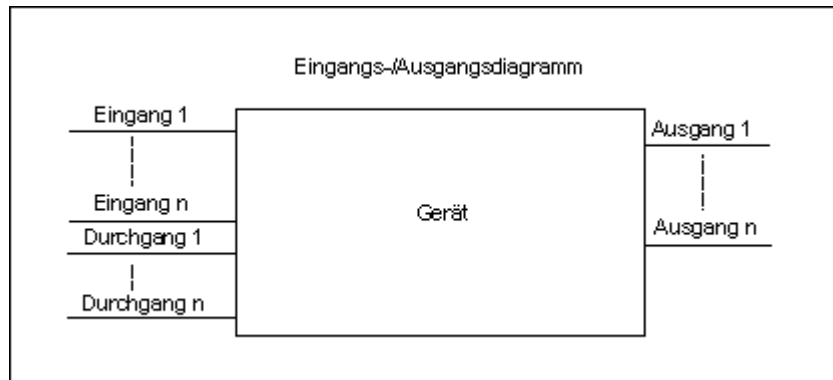
Rührwerkmotor
Mit dem Rührwerkmotor werden die Stoffe A und B im Mischbehälter gemischt. <ul style="list-style-type: none">• Leistung: 100 KW bei 1200 UPM
Der Rührwerkmotor wird von einem Bedienpult gesteuert (Starten/Stoppen), das sich in der Nähe des Mischbehälters befindet. Die Anzahl der Starts wird zu Wartungszwecken gezählt. Der Zähler und die Anzeige können mit einem gemeinsamen Taster zurückgesetzt werden.
Es gelten die folgenden Freigabebedingungen: <ul style="list-style-type: none">• Der Füllstandsmesser zeigt nicht "Behälter unter Minimum".• Das Abflussventil des Mischbehälters ist geschlossen.• NOT-AUS ist nicht aktiviert.
Es gilt die folgende Abschaltbedingung: <ul style="list-style-type: none">• Drehzahlgeber meldet nicht spätestens 10 s nach Start des Motors, dass die Nennzahl erreicht ist.

Abflussventil
Durch Fallspeisung gelangt das Gemisch durch das Abflussventil zum nächsten Prozessschritt. Das Ventil hat einen Magnetschalter mit Federrückholeinrichtung. <ul style="list-style-type: none">• Wenn der Magnetschalter aktiviert ist, ist das Abflussventil geöffnet.• Wenn der Magnetschalter deaktiviert ist, ist das Abflussventil geschlossen.
Das Abflussventil wird vom Bedienpult gesteuert (Öffnen/Schließen).
Das Abflussventil kann unter folgenden Bedingungen geöffnet werden: <ul style="list-style-type: none">• Der Rührwerkmotor ist abgeschaltet.• Der Füllstandsmesser zeigt nicht "Behälter ist leer".• NOT-AUS ist nicht aktiviert.
Es gilt folgende Abschaltbedingung: <ul style="list-style-type: none">• Die Füllstandsmessung ergibt "Behälter leer".

Schalter zur Füllstandsmessung
Die Schalter im Mischbehälter informieren über den Füllstand im Behälter und werden für die Verriegelung der Speisepumpen und des Rührwerkmotors verwendet.

3.4 Auflisten von Ein-, Aus- und Durchgängen

Nachdem Sie jedes Gerät, das gesteuert werden soll, physikalisch beschrieben haben, zeichnen Sie für jedes Gerät oder jeden Aufgabenbereich ein Eingangs-/Ausgangsdiagramm.



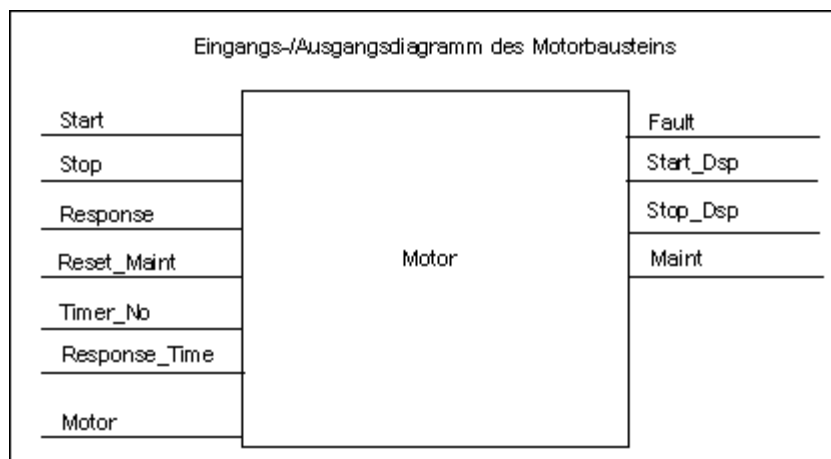
Diese Diagramme entsprechen den Codebausteinen, die zu programmieren sind.

3.5 Erstellen eines Eingangs-, Ausgangsdiagramms für die Motoren

In dem Beispiel für den industriellen Mischprozess werden zwei Speisepumpen und ein Rührwerk eingesetzt. Die Ansteuerung der jeweiligen Motoren erfolgt durch je einen "Motorbaustein", der für alle drei Geräte gleich ist. Dieser Baustein benötigt sechs Eingänge: Zwei Eingänge zum Starten bzw. Stoppen des Motors, einen Eingang zum Rücksetzen der Wartungsanzeige, einen Eingang für die Motorrückmeldung (Motor läuft / läuft nicht), einen Eingang für die Zeitdauer, in der die Rückmeldung erfolgen muss, und einen Eingang für die Nummer des Timers, der zur Zeitmessung verwendet werden soll.

Außerdem benötigt der Codebaustein vier Ausgänge: Zwei Ausgänge zur Anzeige des Betriebszustandes des Motors, einen Ausgang zur Anzeige von Fehlern und einen Ausgang, der anzeigt, dass die Wartung des Motors fällig ist.

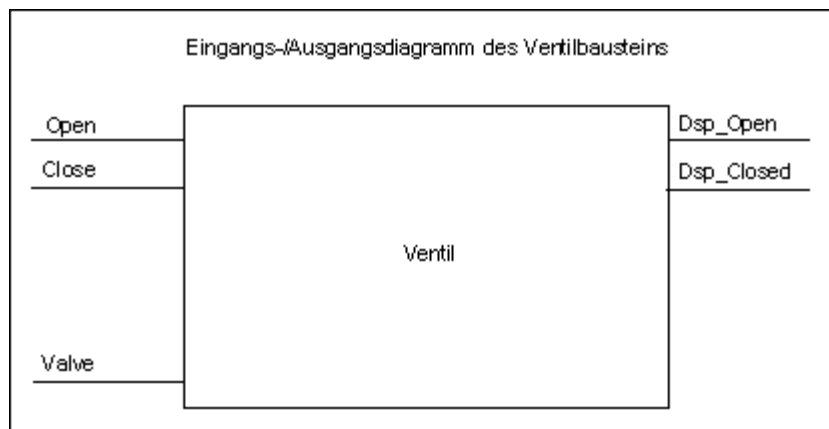
Weiterhin wird ein Durchgang benötigt: Er dient zur Ansteuerung des Motors, wird aber gleichzeitig im Programm des "Motorbausteins" bearbeitet bzw. verändert.



3.6 Erstellen eines Eingangs-/ Ausgangsdiagramms für die Ventile

Die Ansteuerung des jeweiligen Ventils erfolgt über einen "Ventilbaustein", der für alle verwendeten Ventile gleich ist. Der Codebaustein hat zwei Eingänge: Einen Eingang zum Öffnen und einen zum Schließen des Ventils. Außerdem sind zwei Ausgänge vorhanden: Ein Ausgang zeigt an, dass das Ventil geöffnet ist, der andere, dass es geschlossen ist.

Der Baustein hat einen Durchgang: Er dient zur Ansteuerung des Ventils, wird aber gleichzeitig im Programm des "Ventilbausteins" bearbeitet bzw. verändert.



3.7 Festlegen der Sicherheitsanforderungen

Entscheiden Sie - im Rahmen der gesetzlichen Vorschriften und der Verfahrensanweisungen Ihres Unternehmens - welche Elemente notwendig sind, um die Sicherheit des Prozesses zu gewährleisten. Beschreiben Sie auch, wie diese Sicherheitselemente die Bereiche Ihres Prozesses beeinflussen.

Definieren der Sicherheitsanforderungen

Bestimmen Sie die Geräte, die aus Sicherheitsgründen fest verdrahtete Schaltungen benötigen. Per Definition arbeiten diese Sicherheitsstromkreise unabhängig vom Automatisierungssystem (obwohl der Sicherheitsstromkreis in der Regel eine E/A-Schnittstelle zur Koordination mit dem Anwenderprogramm zur Verfügung stellt). Üblicherweise konfigurieren Sie eine Matrix, um jeden Aktor mit einem eigenen NOT-AUS-Bereich zu verbinden. Diese Matrix ist die Grundlage für die Schaltpläne der Sicherheitsstromkreise.

Zum Entwerfen von Schutzvorrichtungen gehen Sie wie folgt vor:

- Legen Sie die logischen und mechanischen/elektrischen Verriegelungen zwischen den einzelnen Automatisierungsaufgaben fest.
- Entwerfen Sie Schaltungen, um im Notfall die am Prozess beteiligten Geräte manuell bedienen zu können.
- Bestimmen Sie weitere Sicherheitsanforderungen für einen sicheren Ablauf des Prozesses.

Erstellen eines Sicherheitsstromkreises

In dem Beispiel für den industriellen Mischprozess verwenden wir für den Sicherheitsstromkreis folgende Schaltung:

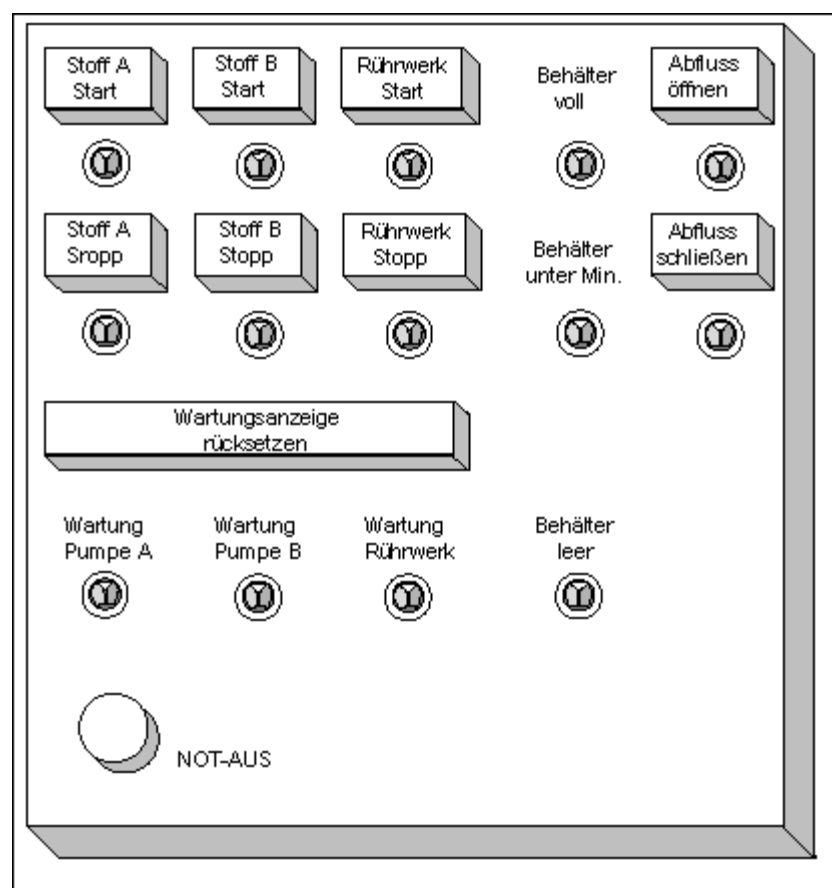
- Einen NOT-AUS-Schalter, der unabhängig vom Automatisierungssystem (AS) die folgenden Geräte ausschaltet:
 - Speisepumpe für Stoff A
 - Speisepumpe für Stoff B
 - Rührwerkmotor
 - Ventile
- Der NOT-AUS-Schalter befindet sich am Bedienpult.
- Ein Eingang der Steuerung erfasst den Zustand des NOT-AUS-Schalters.

3.8 Beschreiben der notwendigen Anzeige- und Bedienelemente

Jeder Prozess benötigt ein Bedien- und Beobachtungssystem, das menschliches Eingreifen in den Prozess ermöglicht. Als Teil der Entwurfsbeschreibung wird auch die Ausführung eines Bedienpultes beschrieben.

Definieren eines Bedienpultes

Beim industriellen Mischprozess in unserem Beispiel wird jedes Gerät über einen Taster, der sich auf dem Bedienpult befindet, gestartet bzw. gestoppt. Dieses Bedienpult ist mit Anzeigen versehen, die über den Betriebszustand informieren (siehe nachfolgendes Bild).



Hier befinden sich auch die Anzeigelampen für die Geräte, die nach einer bestimmten Anzahl von Starts gewartet werden müssen, und der NOT-AUS-Schalter, mit dem der Prozess sofort angehalten werden kann. Auf dem Bedienpult befindet sich außerdem ein Rücksetztaster für die Wartungsanzeige der drei Motoren. Mit ihm schalten Sie die Wartungsanzeigelampen für diejenigen Motoren aus, bei denen eine Wartung fällig ist, und setzen die zugehörigen Zählerstände für das Wartungsintervall auf 0.

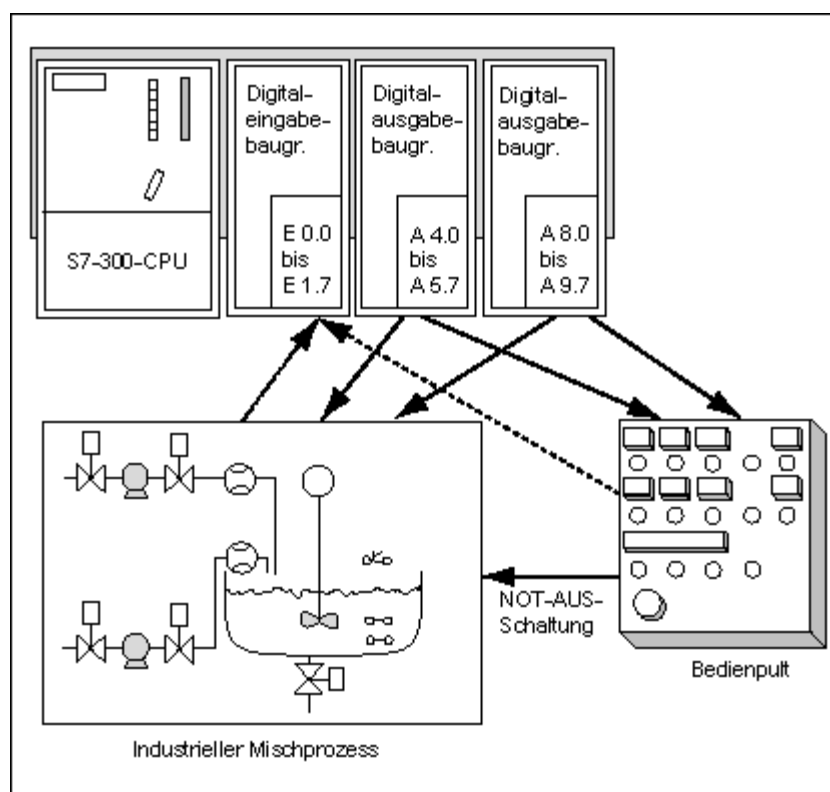
3.9 Erstellen des Konfigurationsplans

Nachdem Sie die Entwurfsanforderungen dokumentiert haben, bestimmen Sie die Steuergeräte, die für dieses Projekt erforderlich sind.

Mit der Entscheidung, welche Baugruppen Sie einsetzen wollen, legen Sie den Aufbau des Automatisierungssystems fest. Erstellen Sie einen Konfigurationsplan, in dem Sie folgende Punkte festlegen:

- Art der CPU
- Anzahl und Art der Signalbaugruppen
- Konfiguration der physikalischen Ein- und Ausgänge

Nachfolgendes Bild zeigt das Beispiel einer S7-Konfiguration für den industriellen Mischprozess.



4 Grundlagen zum Entwerfen einer Programmstruktur

4.1 Programme in einer CPU

In einer CPU laufen grundsätzlich zwei verschiedene Programme ab:

- das Betriebssystem und
- das Anwenderprogramm.

Betriebssystem

Das Betriebssystem ist bereits in jeder CPU enthalten und organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer spezifischen Steuerungsaufgabe verbunden sind. Zu seinen Aufgaben gehören:

- das Abwickeln von Neustart (Warmstart) und Wiederanlauf
- das Aktualisieren des Prozessabbildes der Eingänge und die Ausgabe des Prozessabbildes der Ausgänge
- das Aufrufen des Anwenderprogramms
- das Erfassen von Alarmen und das Aufrufen der Alarm-OBs
- das Erkennen und Behandeln von Fehlern
- das Verwalten von Speicherbereichen
- das Kommunizieren mit Programmiergeräten und anderen Kommunikationspartnern.

Durch Ändern der Betriebssystemparameter (Voreinstellungen des Betriebssystems) können Sie das Verhalten der CPU in bestimmten Bereichen beeinflussen.

Anwenderprogramm

Das Anwenderprogramm müssen Sie erstellen und in die CPU laden. Es enthält alle Funktionen, die zur Bearbeitung Ihrer spezifischen Automatisierungsaufgabe erforderlich sind. Zu den Aufgaben des Anwenderprogramms gehören:

- das Festlegen der Voraussetzungen für den Neustart (Warmstart) und den Wiederanlauf der CPU (z. B. Signale mit einem bestimmten Wert vorbesetzen)
- das Bearbeiten von Prozessdaten (z. B. Binärsignale verknüpfen, Analogwerte einlesen und auswerten, Binärsignale für die Ausgabe festlegen, Analogwerte ausgeben)
- das Reagieren auf Alarme
- das Bearbeiten von Störungen im normalen Programmablauf.

4.2 Bausteine im Anwenderprogramm

Die Programmiersoftware STEP 7 bietet Ihnen die Möglichkeit, das Anwenderprogramm zu strukturieren, d. h. in einzelne, in sich geschlossene Programmabschnitte aufzuteilen. Daraus ergeben sich die folgenden Vorteile:

- umfangreiche Programme lassen sich übersichtlich programmieren
- einzelne Programmteile können standardisiert werden
- die Programmorganisation wird vereinfacht
- Änderungen des Programms lassen sich leichter durchführen
- der Programmtest wird vereinfacht, weil er abschnittsweise erfolgen kann
- die Inbetriebnahme wird erleichtert

Am Beispiel des industriellen Mischprozesses haben Sie gesehen, wie ein Automatisierungsprozess sinnvoll in einzelne Aufgaben zerlegt werden kann. Die Programmabschnitte eines strukturierten Anwenderprogramms entsprechen diesen einzelnen Aufgaben, sie werden als Bausteine eines Programms bezeichnet.

Bausteinarten

Es gibt verschiedene Arten von Bausteinen, die Sie innerhalb eines S7-Anwenderprogramms verwenden können:

Baustein	Kurzbeschreibung der Funktion	Näheres siehe
Organisationsbausteine (OB)	OBs legen die Struktur des Anwenderprogramms fest.	"Organisationsbausteine und Programmstruktur"
Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)	SFBs und SFCs sind in die S7-CPU integriert und machen Ihnen einige wichtige Systemfunktionen zugänglich.	"Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)"
Funktionsbausteine (FB)	FBs sind Bausteine mit "Gedächtnis", die Sie selbst programmieren.	"Funktionsbausteine (FB)"
Funktionen (FC)	FCs enthalten Programmroutinen für häufig verwendete Funktionen.	"Funktionen (FC)"
Instanzen-Datenbausteine (Instanz-DB)	Instanzen-DBs werden bei Aufruf eines FB/SFB dem Baustein zugeordnet. Beim Übersetzen werden sie automatisch generiert.	"Instanz-Datenbausteine"
Datenbausteine (DB)	DBs sind Datenbereiche zur Speicherung von Anwenderdaten. Zusätzlich zu den Daten, die jeweils einem Funktionsbaustein zugeordnet sind, können globale Daten definiert und von beliebigen Bausteinen genutzt werden.	"Globale Datenbausteine (DB)"

OBs, FBs, SFBs, FCs und SFCs enthalten Programmteile und werden deshalb auch als Codebausteine bezeichnet. Die erlaubte Anzahl der Bausteine pro Bausteintyp und die erlaubte Länge der Bausteine ist CPU-abhängig.

4.2.1 Organisationsbausteine und Programmstruktur

Organisationsbausteine (OBs) bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. Sie werden vom Betriebssystem aufgerufen und steuern die zyklische und alarmgesteuerte Programmbearbeitung, das Anlaufverhalten des Automatisierungssystems und die Behandlung von Fehlern. Sie können die Organisationsbausteine programmieren und so das Verhalten der CPU bestimmen.

Priorität von Organisationsbausteinen

Organisationsbausteine bestimmen die Reihenfolge (Startereignisse), in der die einzelnen Programmteile bearbeitet werden. Die Bearbeitung eines OB kann durch den Aufruf eines anderen OB unterbrochen werden. Welcher OB einen anderen OB unterbrechen darf, hängt von seiner Priorität ab. Höherprioritäre OBs unterbrechen niederprioritäre OBs. Die niedrigste Priorität hat der Hintergrund-OB.

Alarmarten und Prioritätsklassen

Die Startereignisse, die den Aufruf eines bestimmten OB veranlassen, werden auch Alarme genannt. Nachfolgende Tabelle zeigt die Alarmarten bei STEP 7 und die Priorität der zugeordneten Organisationsbausteine. Nicht alle angegebenen Organisationsbausteine und deren Prioritätsklassen sind in allen S7-CPUs vorhanden (siehe Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Baugruppendaten").

Alarmart	Organisationsbaustein	Prioritätsklasse (voreingestellt)	Näheres siehe
Freier Zyklus	OB 1	1	"Organisationsbaustein für zyklische Programmbearbeitung (OB 1)"
Uhrzeitalarme	OB 10 bis OB 17	2	"Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17)"
Verzögerungsalarme	OB 20	3	"Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)"
	OB 21	4	
	OB 22	5	
	OB 23	6	
Weckalarme	OB 30	7	"Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)"
	OB 31	8	
	OB 32	9	
	OB 33	10	
	OB 34	11	
	OB 35	12	
	OB 36	13	
	OB 37	14	
	OB 38	15	

Alarmart	Organisationsbaustein	Prioritätsklasse (voreingestellt)	Näheres siehe
Prozessalarme	OB 40 OB 41 OB 42 OB 43 OB 44 OB 45 OB 46 OB 47	16 17 18 19 20 21 22 23	"Organisationsbausteine für den Prozessalarm (OB 40 bis OB 47)"
DPV1Alarmer	OB 55 OB 56 OB 57	2 2 2	Programmieren von DPV1-Geräten
Multicomputingalarm	OB 60 Multicomputing	25	Multicomputing - Synchroner Betrieb mehrerer CPUs
Taktsynchronalarmer	OB 61 OB 62 OB 63 OB 64	25	Kurze und gleichlange Prozessreaktionszeiten am PROFIBUS-DP projektieren
Redundanzfehler	OB 70 Peripherie-Redundanzfehler (nur in H-Systemen) OB 72 CPU-Redundanzfehler (nur in H-Systemen)	25 28	"Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)"
Asynchrone Fehler	OB 80 Zeitfehler OB 81 Stromversorgungsfehler OB 82 Diagnosealarm OB 83 Ziehen/Stecken-Alarm OB 84 CPU-Hardwarefehler OB 85 Programmablauffehler OB 86 Baugruppenträgerausfall OB 87 Kommunikationsfehler	25 (bzw. 28, wenn der asynchrone Fehler-OB im Anlaufprogramm vorkommt)	"Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)"
Hintergrundzyklus	OB 90	29 ¹⁾	"Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90)"
Anlauf	OB 100 Neustart (Warmstart) OB 101 Wiederanlauf OB 102 Kaltstart	27 27 27	"Organisationsbausteine für den Anlauf (OB 100/OB 101/OB 102)"
Synchrone Fehler	OB 121 Programmierfehler OB 122 Zugriffsfehler	Prioritätsklasse des fehlerverursachenden OB	"Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)"

1) Der Prioritätsklasse 29 entspricht die Priorität 0.29. Der Hintergrundzyklus hat also eine niedrigere Priorität als der Freie Zyklus.

Verändern der Priorität

Alarmer sind mit STEP 7 parametrierbar. Mit der Parametrierung können Sie z. B. Alarm-OBs abwählen oder Prioritätsklassen in den folgenden Parameterblöcken ändern: Uhrzeitalarmer, Verzögerungsalarmer, Weckalarmer und Prozessalarmer.

Bei S7-300-CPUs ist die Priorität der Organisationsbausteine fest zugeordnet.

Bei S7-400-CPU (und der CPU 318) können Sie mit STEP 7 die Priorität folgender Organisationsbausteine ändern:

- OB 10 bis OB 47
- OB 70 bis OB 72 (nur H-CPU) und OB 81 bis OB 87 im Betriebszustand RUN.

Zulässig sind:

- für OB 10 bis OB 47 die Prioritätsklassen 2 bis 23,
- für OB 70 bis OB 72 die Prioritätsklassen 2 bis 28 sowie
- für OB 81 bis OB 87 die Prioritätsklassen 24 bis 26; für CPUs ab ca. Mitte 2001 (Firmware Version 3.0) wurden die Bereiche erweitert: Für OB 81 bis OB 84 sowie OB 86 und OB 87 können die Prioritätsklassen 2 bis 26 eingestellt werden.

Sie können mehreren OBs die gleiche Priorität zuteilen. OBs mit gleicher Priorität werden in der Reihenfolge des Auftretens ihrer Startereignisse bearbeitet.

Fehler-OBs, die bei synchronen Fehlern gestartet werden, werden in derselben Prioritätsklasse bearbeitet wie der Baustein, der beim Erkennen des Fehlers bearbeitet wurde.

Lokaldaten

Beim Erstellen von Codebausteinen (OBs, FCs, FBs) können Sie temporäre Lokaldaten vereinbaren. Der in der CPU zur Verfügung stehende Lokaldatenbereich wird unter den Prioritätsklassen aufgeteilt.

Bei S7-400-CPU können Sie mit STEP 7 im Parameterblock "Prioritätsklassen" die Anzahl der Lokaldaten pro Prioritätsklasse abändern.

Startinformation eines OB

Jeder Organisationsbaustein verfügt über eine Startinformation von 20 Byte Lokaldaten, die das Betriebssystem beim Starten eines OB übergibt. Die Startinformation gibt Auskunft über das Startereignis des OB, Datum und Uhrzeit des OB-Starts, aufgetretene Fehler und Diagnoseereignisse.

Ein Prozessalarm-OB 40 enthält z. B. in der Startinformation die Adresse der alarmerzeugenden Baugruppe.

Abgewählte Alarm-OBs

Wählen Sie die Prioritätsklassen 0 oder ordnen Sie einer Prioritätsklasse weniger als 20 Byte Lokaldaten zu, wird der entsprechende Alarm-OB abgewählt.

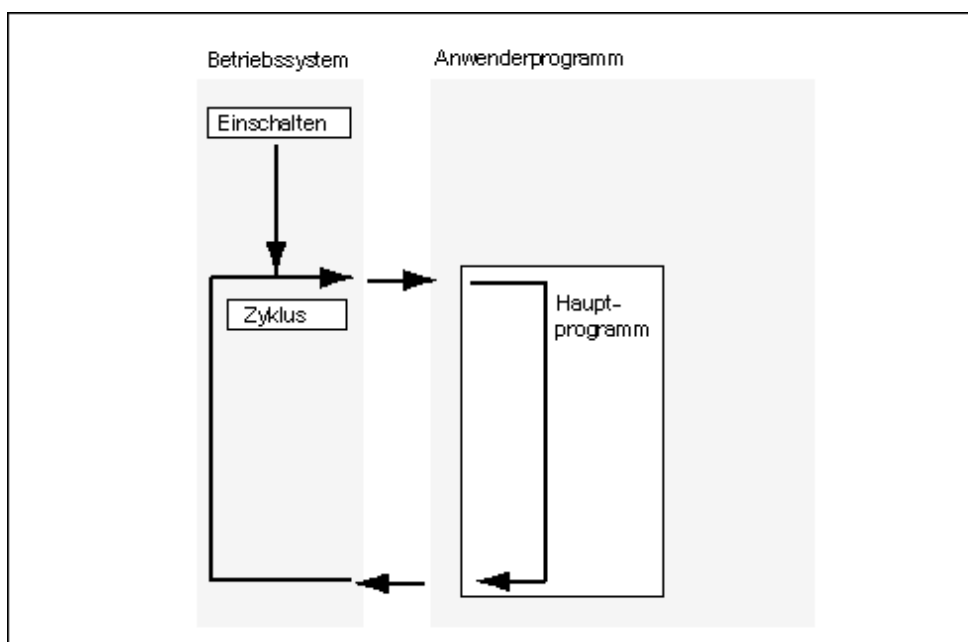
Abgewählte Alarm-OBs können:

- im Betriebszustand "RUN" nicht kopiert bzw. in Ihr Anwenderprogramm eingebunden werden.
- im Betriebszustand "STOP" zwar kopiert bzw. in Ihr Anwenderprogramm eingebunden werden, führen jedoch beim Neustart (Warmstart) der CPU zum Abbruch des Anlaufs und erzeugen einen Eintrag in den Diagnosepuffer.

Durch das Abwählen nicht benötigter Alarm-OBs wird der zur Verfügung stehende freie Lokaldatenbereich größer und kann für die Speicherung temporärer Daten in anderen Prioritätsklassen genutzt werden.

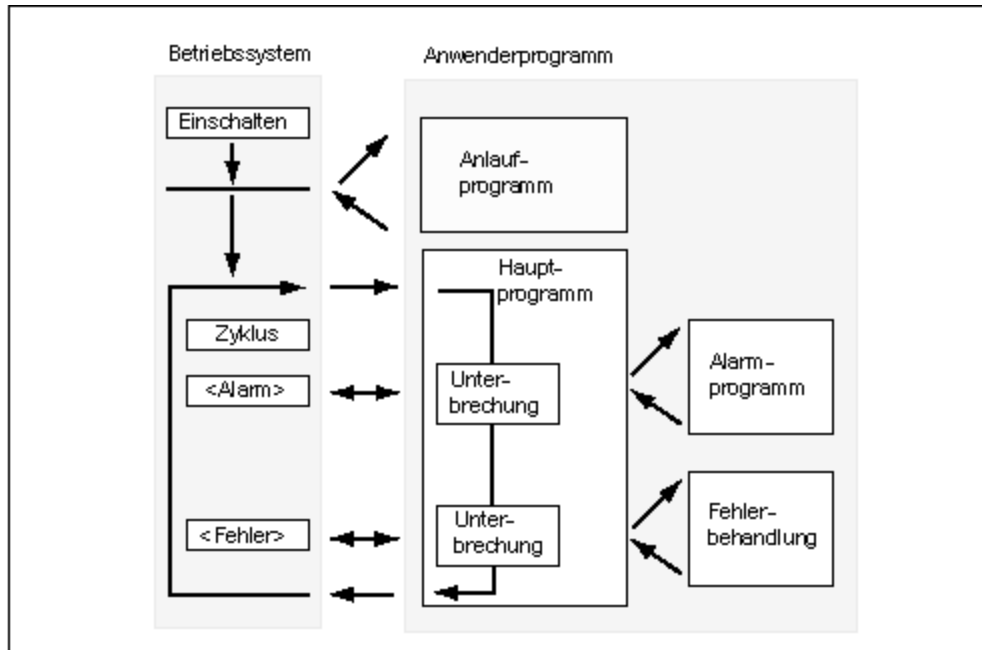
Zyklische Programmbearbeitung

Bei speicherprogrammierbaren Steuerungen herrscht typischerweise die zyklische Programmbearbeitung vor, d. h. das Betriebssystem läuft in einer Programmschleife (dem Zyklus) und ruft dabei in jeder Schleife einmal den Organisationsbaustein OB 1 im Hauptprogramm auf. Das Anwenderprogramm im OB 1 wird also zyklisch bearbeitet.



Ereignisgesteuerte Programmbearbeitung

Die zyklische Programmbearbeitung kann durch bestimmte Startereignisse (Alarme) unterbrochen werden. Tritt ein solches Startereignis ein, wird der gerade bearbeitete Baustein an einer Befehlsgränze unterbrochen und ein anderer Organisationsbaustein abgearbeitet, der dem Startereignis zugeordnet ist. Danach wird die Bearbeitung des zyklischen Programms an der Unterbrechungsstelle wieder fortgesetzt.

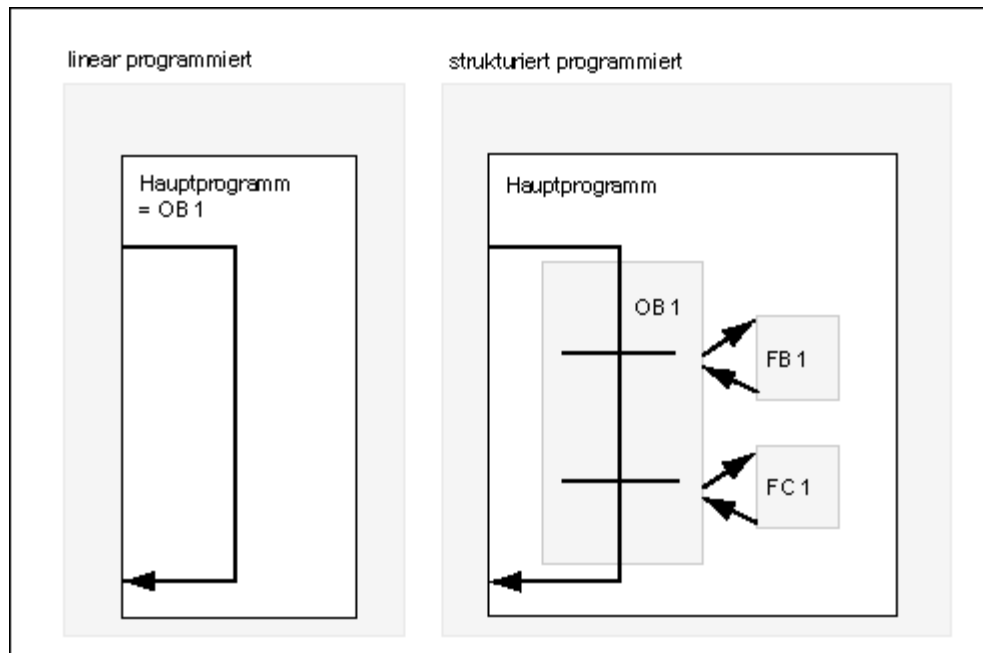


Damit besteht die Möglichkeit, Teile des Anwenderprogramms, die nicht zyklisch bearbeitet werden müssen, nur bei Bedarf zu bearbeiten. Das Anwenderprogramm kann in Teilprogramme zerlegt und auf verschiedene Organisationsbausteine aufgeteilt werden. Soll das Anwenderprogramm auf ein wichtiges Signal reagieren, das relativ selten vorkommt (z. B. Grenzwertgeber zur Füllstandsmessung eines Tanks meldet Füllstand erreicht), kann das Teilprogramm, das bearbeitet werden soll, wenn das Signal ausgegeben wird, in einem OB stehen, der ereignisgesteuert bearbeitet wird.

Linear oder strukturiert programmieren

Sie können Ihr gesamtes Anwenderprogramm in den OB 1 schreiben (lineare Programmierung). Dies ist nur empfehlenswert bei einfachen Programmen, die auf S7-300-CPUs mit kleinem Speicherumfang laufen.

Komplexe Automatisierungsaufgaben können besser bearbeitet werden, wenn sie in kleinere Teilaufgaben untergliedert werden, die den technologischen Funktionen des Automatisierungsprozesses entsprechen oder mehrfach verwendet werden sollen. Im Anwenderprogramm werden diese Teilaufgaben durch entsprechende Programmteile, die Bausteine, repräsentiert (strukturierte Programmierung).



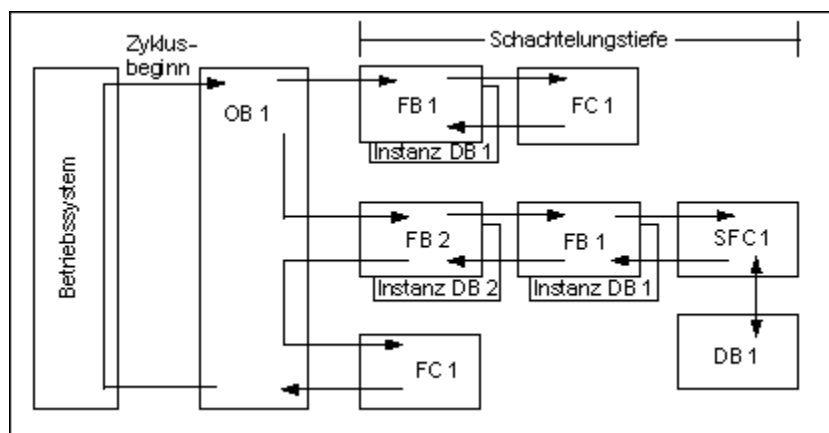
4.2.2 Aufrufhierarchie im Anwenderprogramm

Soll das Anwenderprogramm funktionieren, so müssen die Bausteine, aus denen es besteht, aufgerufen werden. Das geschieht durch spezielle STEP 7-Operationen, den Bausteinaufrufen, die nur innerhalb von Codebausteinen programmiert und gestartet werden können.

Reihenfolge und Schachtelungstiefe

Die Reihenfolge und Schachtelung der Bausteinaufrufe wird Aufrufhierarchie genannt. Die zulässige Schachtelungstiefe ist CPU-abhängig.

Nachfolgendes Bild zeigt anhand eines Beispiels die Reihenfolge und Schachtelung der Bausteinaufrufe innerhalb eines Bearbeitungszyklus.



Regeln für die Reihenfolge der Bausteinerstellung:

- Sie erstellen die Bausteine von oben nach unten, fangen also mit der obersten Bausteinreihe an.
- Jeder Baustein der aufgerufen wird, muss schon vorhanden sein, d. h. innerhalb einer Bausteinreihe ist die Erstellrichtung von rechts nach links.
- Als letztes erstellen Sie den OB 1.

Diese Regeln umgesetzt bedeutet für das Beispiel-Bild folgende Erstellreihenfolge:

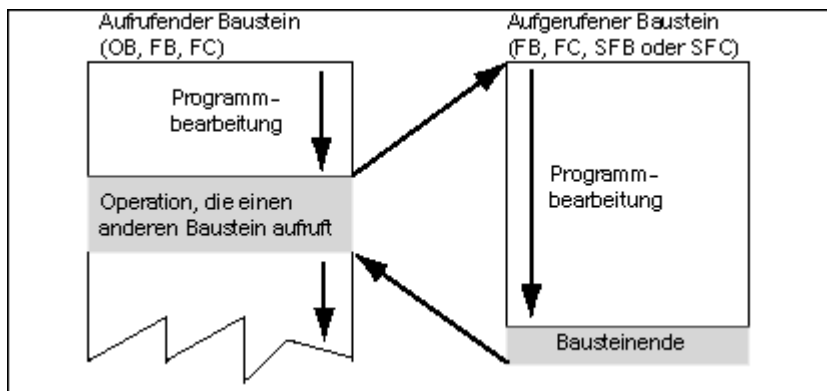
FC 1 > FB 1 + Instanz-DB 1 > DB 1 > SFC 1 > FB 2 + Instanz-DB 2 > OB 1

Hinweis

Bei zu großer Schachtelungstiefe kann der Lokaldaten-Stack überlaufen (siehe auch Lokaldaten-Stack).

Bausteinaufrufe

Nachfolgendes Bild zeigt den Ablauf eines Bausteinaufrufs innerhalb eines Anwenderprogramms: das Programm ruft den zweiten Baustein auf, dessen Operationen dann vollständig bearbeitet werden. Ist die Bearbeitung des aufgerufenen Bausteins beendet, wird die Bearbeitung des aufrufenden Bausteins mit der dem Bausteinaufruf folgenden Operation wiederaufgenommen.



Bevor Sie einen Baustein programmieren, müssen Sie erst festlegen, mit welchen Daten die Bearbeitung Ihres Programms erfolgen soll: Sie müssen die Variablen des Bausteins deklarieren.

Hinweise

- OUT-Parameter müssen bei jedem Bausteinaufruf beschrieben werden.
 - Das Betriebssystem setzt die Instanzen des SFB 3 "TP" bei Kaltstart zurück. Falls Instanzen dieses SFB nach Neustart (Warmstart) initialisiert sein sollen, müssen Sie über den OB 100 die zu initialisierenden Instanzen des SFB mit PT = 0 ms aufrufen. Dies können Sie z.B. über eine Initialisierungsroutine in den Bausteinen erreichen, die Instanzen dieses SFB enthalten.
-

4.2.3 Bausteinarten

4.2.3.1 Organisationsbaustein für zyklische Programmbearbeitung (OB 1)

Die zyklische Programmbearbeitung ist die "normale" Programmbearbeitung bei speicherprogrammierbaren Steuerungen. Das Betriebssystem ruft zyklisch den OB 1 auf und startet damit die zyklische Bearbeitung des Anwenderprogramms.

Ablauf der zyklischen Programmbearbeitung

Nachfolgende Tabelle zeigt die Phasen der zyklischen Programmbearbeitung:

Schritt	Ablauf bei CPUs bis 10/98	Ablauf bei CPUs ab 10/98
1	Das Betriebssystem startet die Zyklusüberwachungszeit.	Das Betriebssystem startet die Zyklusüberwachungszeit.
2	Die CPU liest den Zustand der Eingänge an den Eingabebaugruppen aus und aktualisiert das Prozessabbild der Eingänge.	Die CPU schreibt die Werte aus dem Prozessabbild der Ausgänge in die Ausgabebaugruppen.
3	Die CPU bearbeitet das Anwenderprogramm und führt die im Programm angegebenen Operationen aus.	Die CPU liest den Zustand der Eingänge an den Eingabebaugruppen aus und aktualisiert das Prozessabbild der Eingänge.
4	Die CPU schreibt die Werte aus dem Prozessabbild der Ausgänge in die Ausgabebaugruppen.	Die CPU bearbeitet das Anwenderprogramm und führt die im Programm angegebenen Operationen aus.
5	Am Ende eines Zyklus führt das Betriebssystem anstehende Aufgaben aus, z. B. Laden und Löschen von Bausteinen, Empfangen und Senden von Globaldaten.	Am Ende eines Zyklus führt das Betriebssystem anstehende Aufgaben aus, z. B. Laden und Löschen von Bausteinen, Empfangen und Senden von Globaldaten.
6	Anschließend kehrt die CPU zum Zyklusbeginn zurück und startet erneut die Zykluszeitüberwachung.	Anschließend kehrt die CPU zum Zyklusbeginn zurück und startet erneut die Zykluszeitüberwachung.

Prozessabbilder

Damit der CPU für die Dauer der zyklischen Programmbearbeitung ein konsistentes Abbild der Prozesssignale zur Verfügung steht, greift die CPU beim Ansprechen der Operandenbereiche Eingänge (E) und Ausgänge (A) nicht direkt auf die Signalbaugruppen zu, sondern auf einen internen Speicherbereich der CPU, in dem sich ein Abbild der Ein-/Ausgänge befindet.

Programmieren zyklischer Programmbearbeitung

Sie programmieren die zyklische Programmbearbeitung, indem Sie Ihr Anwenderprogramm mit STEP 7 in den OB 1 und die darin aufgerufenen Bausteine schreiben.

Die zyklische Programmbearbeitung beginnt, sobald das Anlaufprogramm fehlerfrei beendet wurde.

Unterbrechungsmöglichkeiten

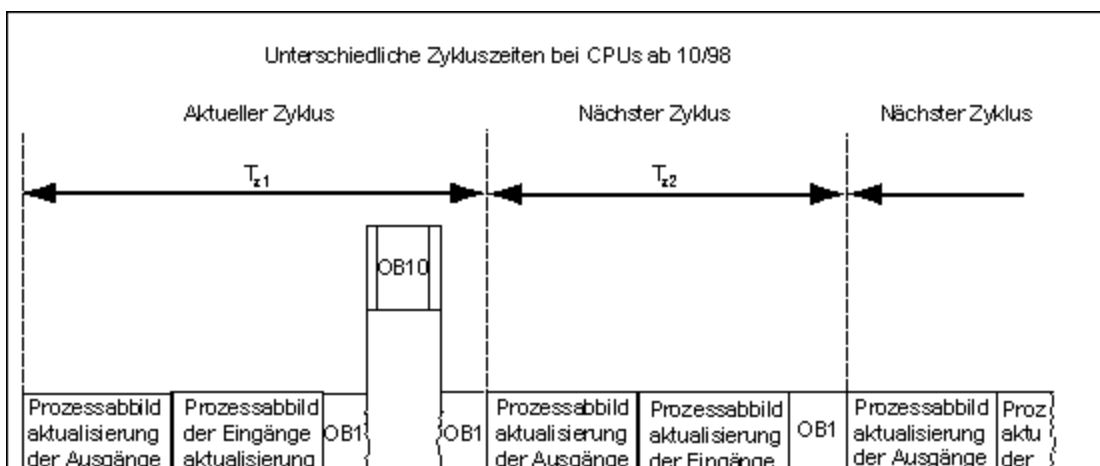
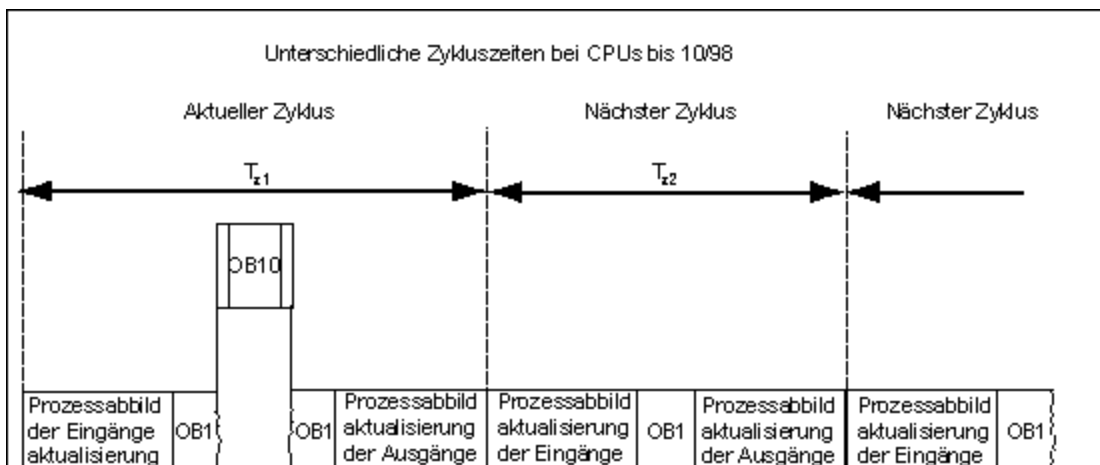
Die zyklische Programmbearbeitung kann unterbrochen werden durch:

- einen Alarm
- einen STOP-Befehl (Betriebsartenschalter, Menübefehl vom PG aus, SFC 46 STP, SFB 20 STOP)
- Netzspannungsausfall
- das Auftreten eines Geräte- oder Programmfehlers

Zykluszeit

Die Zykluszeit ist die Zeit, die das Betriebssystem für die Bearbeitung des zyklischen Programms sowie aller diesen Zyklus unterbrechenden Programmteile (z. B. Bearbeitung anderer Organisationsbausteine) und Systemtätigkeiten (z. B. Prozessabbildaktualisierung) benötigt. Diese Zeit wird überwacht.

Die Zykluszeit (TZ) ist nicht für jeden Zyklus gleich lang. Nachfolgende Bilder zeigen unterschiedliche Zykluszeiten ($TZ1 \neq TZ2$) für CPUs bis 10/98 und CPUs ab 10/98.



Im aktuellen Zyklus wird OB 1 durch einen Uhrzeit-Alarm unterbrochen.

Zyklusüberwachungszeit

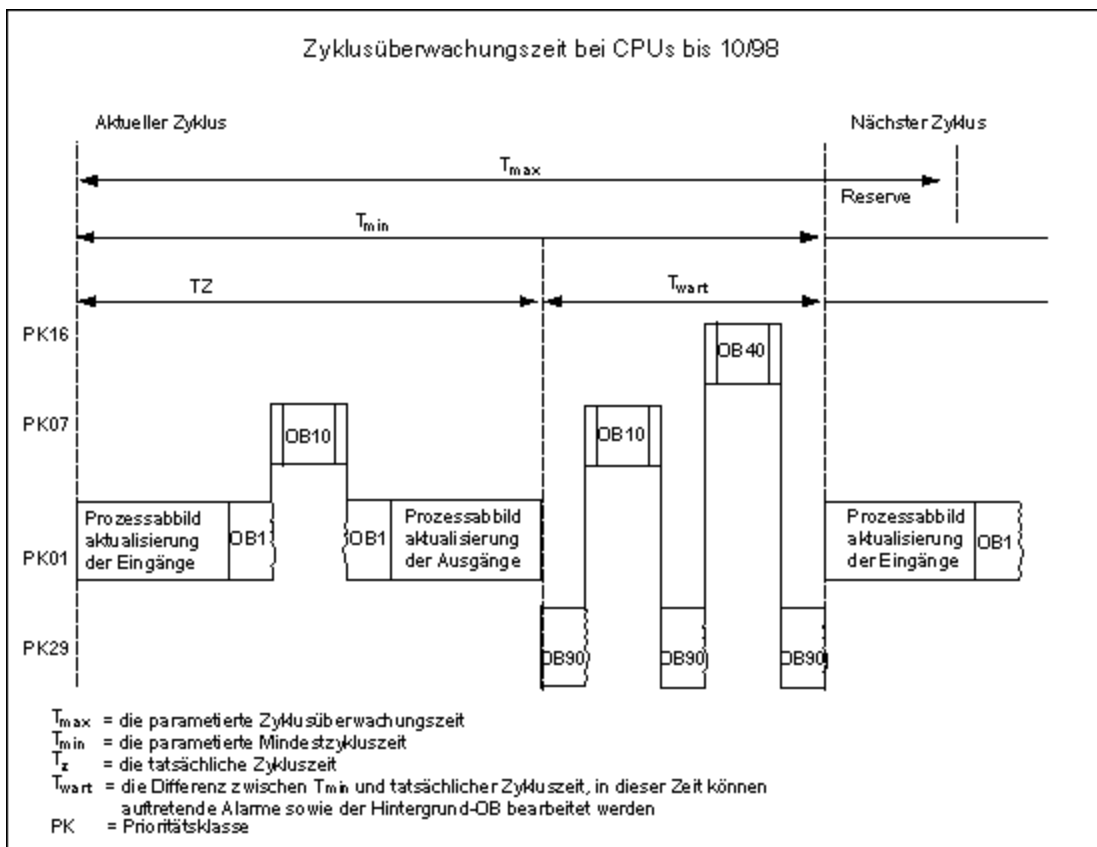
Sie können mit STEP 7 die voreingestellte Zyklusüberwachungszeit ändern. Ist diese Zeit abgelaufen, geht entweder die CPU in STOP, oder es wird der OB 80 aufgerufen, in dem Sie festlegen können, wie die CPU auf den Zeitfehler reagieren soll.

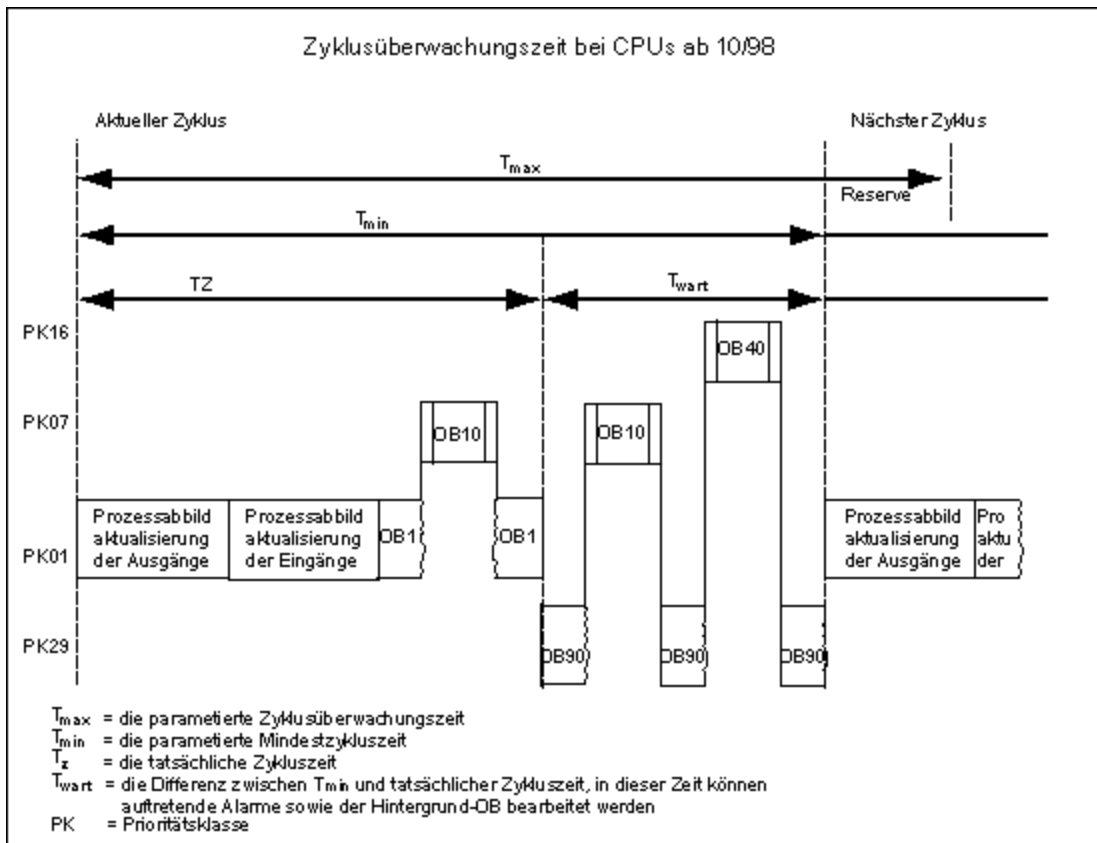
Mindestzykluszeit

Sie können mit STEP 7 bei S7-400-CPU's und bei der CPU 318 eine Mindestzykluszeit einstellen. Dies ist sinnvoll, wenn

- die Zeitabstände zwischen den Starts der Programmbearbeitung des OB 1 (Freier Zyklus) gleich lang sein sollen oder
- bei zu kurzer Zykluszeit die Aktualisierung der Prozessabbilder unnötig oft erfolgen würde.

Nachfolgende Bilder zeigen die Funktion der Zyklusüberwachungszeit im Programmablauf bei CPU's bis 10/98 und CPU's ab 10/98.





Aktualisierung des Prozessabbilds

Bei der zyklischen Programmbearbeitung der CPU wird automatisch das Prozessabbild aktualisiert. Sie können bei S7-400-CPU's und bei der CPU 318 die Aktualisierung des Prozessabbilds abwählen, wenn Sie

- statt dessen direkt auf die Peripherie zugreifen wollen oder
- ein oder mehrere Prozessabbilder der Ein- und Ausgänge zu einem anderen Zeitpunkt mit Hilfe der Systemfunktionen SFC 26 UPDAT_PI und SFC 27 UPDAT_PO aktualisieren wollen.

Kommunikationslast

Mit dem CPU-Parameter "Zyklusbelastung durch Kommunikation" können Sie die Dauer von Kommunikationsprozessen, die immer auch die Zykluszeit verlängern, in einem gewissen Rahmen steuern. Kommunikationsprozesse können z. B. sein: Datenübertragung zu einer anderen CPU über MPI oder Laden von Bausteinen, das über PG angestoßen wurde.

Testfunktionen mit dem PG werden durch diesen Parameter kaum beeinflusst. Sie können aber die Zykluszeit erheblich verlängern. Die für Testfunktionen zur Verfügung gestellte Zeit kann im Prozessbetrieb begrenzt werden (nur S7-300).

Wirkungsweise des Parameters

Das Betriebssystem der CPU stellt laufend der Kommunikation den projektierten Prozentsatz der gesamten CPU-Verarbeitungsleistung zur Verfügung (Zeitscheiben-Technik). Wird diese Verarbeitungsleistung für die Kommunikation nicht benötigt, steht sie der übrigen Verarbeitung zur Verfügung.

Auswirkung auf die tatsächliche Zykluszeit

Ohne zusätzliche asynchrone Ereignisse verlängert sich die OB 1-Zykluszeit um einen Faktor, der sich nach folgender Formel berechnen lässt:

$$\frac{100}{100 - \text{"Zyklusbelastung durch Kommunikation (\%)\"}}$$

Beispiel 1 (keine zusätzlichen asynchrone Ereignisse):

Bei Einstellung der Zyklusbelastung durch Kommunikation auf 50 % kann sich eine Verdopplung der OB 1-Zykluszeit ergeben.

Gleichzeitig wird die OB 1-Zykluszeit auch noch durch asynchrone Ereignisse (z. B. Prozessalarme oder Weckalarne) beeinflusst. Durch die Verlängerung der Zykluszeit durch den Kommunikationsanteil treten statistisch gesehen auch mehr asynchrone Ereignisse innerhalb eines OB 1-Zyklus auf. Dies verlängert der OB 1-Zyklus zusätzlich. Diese Verlängerung ist abhängig davon, wie viele Ereignisse pro OB 1-Zyklus auftreten und der Dauer der Ereignisbearbeitung.

Beispiel 2 (zusätzlichen asynchrone Ereignisse berücksichtigt):

Bei einer reinen OB 1-Ablaufzeit von 500 ms kann sich durch eine Kommunikationslast von 50 % eine tatsächliche Zykluszeit von bis zu 1000 ms ergeben (unter der Voraussetzung, dass die CPU immer genügend Kommunikationsaufträge zum Bearbeiten hat). Wenn nun parallel dazu alle 100 ms ein Weckalarm mit 20 ms Bearbeitungszeit abläuft, dann würde sich dieser ohne Kommunikationslast mit insgesamt $5 \cdot 20 \text{ ms} = 100 \text{ ms}$ auf den Zyklus verlängernd auswirken, d. h. die tatsächliche Zykluszeit wäre 600 ms. Da ein Weckalarm auch die Kommunikation unterbricht, wirkt er sich bei 50% Kommunikationslast mit $10 \cdot 20 \text{ ms}$ auf die Zykluszeit aus, d. h. in diesem Fall beträgt die tatsächliche Zykluszeit nicht 1000 ms sondern 1200 ms.

Hinweise

- Überprüfen Sie die Auswirkungen einer Wertänderung des Parameters "Zyklusbelastung durch Kommunikation" im Anlagenbetrieb.
 - Die Kommunikationslast muss beim Einstellen der minimalen Zykluszeit berücksichtigt werden, da es sonst zu Zeitfehlern kommt.
-

Empfehlungen

- Übernehmen Sie nach Möglichkeit den voreingestellten Wert
- Vergrößern Sie den Wert nur dann, wenn die CPU hauptsächlich zu Kommunikationszwecken eingesetzt wird und das Anwenderprogramm zeitunkritisch ist!
- In allen anderen Fällen den Wert nur verringern!
- Stellen Sie Prozessbetrieb ein (nur bei S7-300) und begrenzen Sie die für Testfunktionen dort benötigte Zeit!

4.2.3.2 Funktionen (FC)

Funktionen gehören zu den Bausteinen, die Sie selbst programmieren. Eine Funktion ist ein Codebaustein "ohne Gedächtnis". Temporäre Variablen der FC werden im Lokaldaten-Stack gespeichert. Diese Daten gehen nach der Bearbeitung der FC verloren. Funktionen können zum Speichern von Daten globale Datenbausteine nutzen.

Weil eine FC keinen zugeordneten Speicher hat, müssen Sie immer Aktualparameter für eine FC angeben. Sie können den Lokaldaten einer FC keine Anfangswerte zuordnen.

Anwendungsbereich

Eine FC enthält ein Programm, das immer dann ausgeführt wird, wenn die FC von einem anderen Codebaustein aufgerufen wird. Funktionen können eingesetzt werden, um

- einen Funktionswert an den aufrufenden Baustein zurückzugeben (Beispiel: mathematische Funktionen).
- eine technologische Funktion auszuführen (Beispiel: Einzelsteuerung mit Binärverknüpfung).

Zuordnen von Aktualparametern zu Formalparametern

Ein Formalparameter ist ein Platzhalter für den "tatsächlichen" Parameter, den Aktualparameter. Aktualparameter ersetzen beim Aufruf einer FC die Formalparameter. Sie müssen den Formalparametern einer FC immer Aktualparameter zuordnen (z. B. dem Formalparameter "Start" einen Aktualparameter "E3.6"). Die Eingangs-, Ausgangs- und Durchgangsparameter, die von der FC verwendet werden, werden als Pointer auf die Aktualparameter des Codebausteins gespeichert, der die FC aufgerufen hat.

Wichtiger Unterschied bei Ausgangsparametern von FC und FB

In Funktionsbausteinen (FB) wird beim Zugriff auf Parameter die Kopie des Aktualparameters im Instanz-DB verwendet. Wird beim Aufruf eines FB ein Eingangsparameter nicht übergeben bzw. im Baustein ein Ausgangsparameter nicht beschrieben, so werden die im Instanz-DB noch vorhandenen älteren Werte weiter verwendet (Instanz-DB = Gedächtnis des FB).

Funktionen (FC) haben kein Gedächtnis. Die Versorgung der Formalparameter ist deshalb im Gegensatz zum FB nicht optional, sondern zwingend erforderlich. Der Zugriff auf FC-Parameter erfolgt über Adressen (bereichsübergreifende Zeiger). Wird als Aktualparameter ein Operand aus dem Bereich Daten (Datenbaustein) oder eine lokale Variable des rufenden Bausteins verwendet, wird für die Parameterübergabe eine Kopie des Aktualparameters in den Lokaldaten des rufenden Bausteins temporär gespeichert.

Achtung

Wird in einem solchen Fall ein OUTPUT Parameter in einem FC nicht beschrieben, können die ausgegebenen Werte zufällig sein!

Der für die Kopie bereitgestellte Bereich in den Lokaldaten des rufenden Bausteins wird mangels Zuweisung an den OUTPUT Parameter nicht beschrieben und bleibt somit unverändert. Damit wird zufällig der in diesem Bereich stehende Wert ausgegeben, da Lokaldaten nicht automatisch mit z. B. 0 vorbelegt sind.

Beachten Sie deshalb Folgendes:

- Initialisieren Sie wenn möglich OUTPUT Parameter.
- Setze- und Rücksetze-Befehle sind VKE-abhängig. Wird mit diesen Befehlen der Wert eines OUTPUT Parameters ermittelt, wird bei VKE = 0 kein Wert gebildet.
- Achten Sie darauf, dass - unabhängig von allen möglichen Programmpfaden innerhalb des Bausteins - die OUTPUT-Parameter auf jeden Fall beschrieben werden. Beachten Sie hierbei insbesondere Sprungbefehle sowie den ENO-Ausgang in KOP und FUP. Denken Sie auch an BEB und die Wirkung der MCR-Befehle.

Hinweis

Bei OUTPUT-Parametern eines FB bzw. INOUT-Parametern eines FC und FB können die ausgegebenen Werte zwar nicht zufällig sein, da hier ohne Beschreibung des Parameters der alte Ausgangswert bzw. der Eingangswert als Ausgangswert erhalten bleibt. Dennoch sollten Sie auch hier die oben gegebenen Hinweise beachten, um nicht unbeabsichtigt "alte" Werte weiter zu verarbeiten.

4.2.3.3 Funktionsbausteine (FB)

Funktionsbausteine gehören zu den Bausteinen, die Sie selbst programmieren. Ein Funktionsbaustein ist ein Baustein "mit Gedächtnis". Er verfügt über einen zugeordneten Datenbaustein als Speicher (Instanz-Datenbaustein). Die Parameter, die an den FB übergeben werden, sowie die statischen Variablen werden im Instanz-DB gespeichert. Die temporären Variablen werden im Lokaldaten-Stack gespeichert.

Daten, die im Instanz-DB gespeichert werden, gehen nicht verloren, wenn die Bearbeitung des FB beendet ist. Daten, die im Lokaldaten-Stack gespeichert werden, gehen nach der Bearbeitung des FB verloren.

Hinweis

Um Fehler beim Arbeiten mit FBs zu vermeiden, lesen Sie Zulässige Datentypen beim Übergeben von Parametern im Anhang.

Anwendungsbereich

Ein FB enthält ein Programm, das immer dann ausgeführt wird, wenn der FB von einem anderen Codebaustein aufgerufen wird. Funktionsbausteine erleichtern das Programmieren häufig wiederkehrender, komplexer Funktionen.

FBs und Instanz-DBs

Jedem Aufruf eines Funktionsbausteins, der Parameter übergibt, ist ein Instanz-Datenbaustein zugeordnet.

Durch das Aufrufen mehrerer Instanzen eines FB können Sie mit einem FB mehrere Geräte steuern. Ein FB für einen Motortyp beispielsweise kann verschiedene Motoren steuern, indem er verschiedene Instanz-Daten für die verschiedenen Motoren verwendet. Die Daten für jeden einzelnen Motor (wie Drehzahl, Ramping, akkumulierte Betriebszeit usw.) können in einem oder mehreren Instanz-DBs gespeichert werden.

Nachfolgende Tabelle zeigt, welchen Variablen des FB Aktualparameter zugeordnet werden müssen.

	Datentyp		
Variablen	Elementarer Datentyp	Zusammengesetzter Datentyp	Parametertyp
Eingang	Parameter nicht erforderlich	Parameter nicht erforderlich	Aktualparameter erforderlich
Ausgang	Parameter nicht erforderlich	Parameter nicht erforderlich	Aktualparameter erforderlich
Durchgang	Parameter nicht erforderlich	Aktualparameter erforderlich	–

Zuordnen von Anfangswerten zu Formalparametern

Sie können den Formalparametern im Deklarationsteil des FB Anfangswerte zuordnen. Diese Werte werden in den Instanz-DB übernommen, der einem FB zugeordnet wird.

Ordnen Sie den Formalparametern in der Aufrufanweisung keine Aktualparameter zu, dann verwendet STEP 7 die Werte, die im Instanz-DB gespeichert sind. Diese Daten können Anfangswerte sein, die in der Variablendeklarationstabelle eines FB eingegeben wurden.

Nachfolgende Tabelle zeigt, welche Variablen einem Anfangswert zugeordnet werden können. Weil die temporären Daten nach der Bearbeitung des Bausteins nicht gespeichert werden, können Sie ihnen keine Werte zuordnen.

	Datentyp		
Variablen	Elementarer Datentyp	Zusammengesetzter Datentyp	Parametertyp
Eingang	Anfangswert zulässig	Anfangswert zulässig	–
Ausgang	Anfangswert zulässig	Anfangswert zulässig	–
Durchgang	Anfangswert zulässig	–	–
Statisch	Anfangswert zulässig	Anfangswert zulässig	–
Temporär	–	–	–

4.2.3.4 Instanz-Datenbausteine

Jedem Aufruf eines Funktionsbausteins, der Parameter übergibt, ist ein Instanz-Datenbaustein zugeordnet. Im Instanz-DB sind die Aktualparameter und die statischen Daten des FB abgelegt. Die im FB deklarierten Variablen bestimmen die Struktur des Instanz-Datenbausteins. Mit Instanz wird der Aufruf eines Funktionsbausteins bezeichnet. Wird z. B. ein Funktionsbaustein im S7-Anwenderprogramm fünfmal aufgerufen, so existieren fünf Instanzen dieses Bausteins.

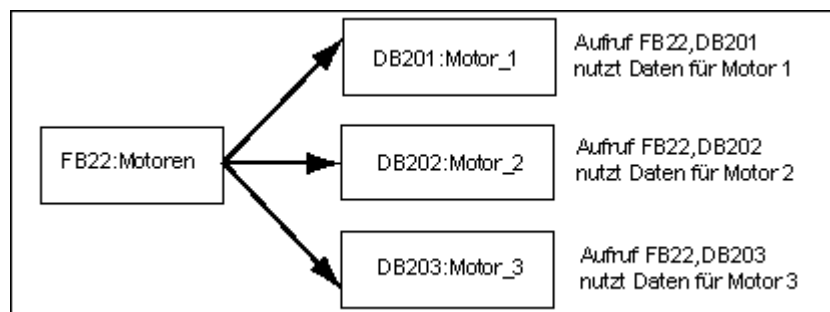
Erzeugen eines Instanz-DB

Bevor Sie einen Instanz-Datenbaustein erzeugen, muss der zuzuordnende FB bereits existieren. Die Nummer dieses FB geben Sie beim Anlegen des Instanz-Datenbausteins an.

Ein Instanz-DB für jede Instanz

Wenn Sie mehrere Instanz-Datenbausteine einem Funktionsbaustein (FB) zuordnen, der einen Motor steuert, so können Sie diesen FB zum Steuern verschiedener Motoren verwenden.

Die unterschiedlichen Daten für die einzelnen Motoren (wie z. B. Drehzahl, Hochlaufzeit, Gesamtbetriebszeit) werden in den verschiedenen Datenbausteinen gespeichert. Je nachdem welcher DB dem FB beim Aufruf zugeordnet wird, kann ein anderer Motor gesteuert werden. Auf diese Weise wird nur ein Funktionsbaustein für mehrere Motoren benötigt (siehe nachfolgendes Bild).

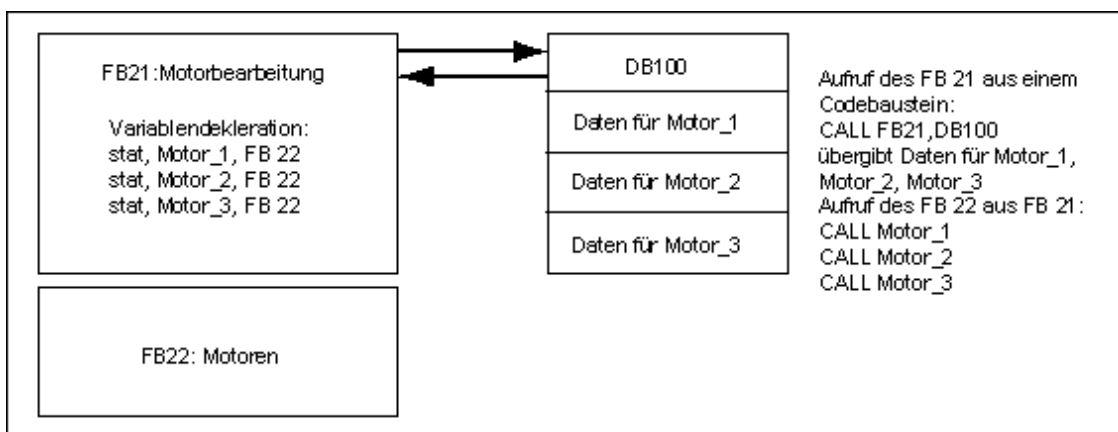


Ein Instanz-DB für mehrere Instanzen eines FB (Multiinstanzen)

Sie können einem FB die Instanzdaten für verschiedene Motoren gemeinsam in einem Instanz-DB übergeben. Dazu müssen Sie den Aufruf der Motorensteuerungen in einem weiteren FB vornehmen und im Deklarationsteil des rufenden FB statische Variablen mit dem Datentyp eines FB für die einzelnen Instanzen deklarieren.

Wenn Sie einen Instanz-DB für mehrere Instanzen eines FB verwenden, sparen Sie Speicherplatz und optimieren den Einsatz von Datenbausteinen.

Im nachfolgenden Bild ist beispielsweise der rufende FB der FB 21 "Motorbearbeitung", die Variablen sind vom Datentyp FB 22 und die Instanzen werden mit Motor_1, Motor_2 und Motor_3 bezeichnet.



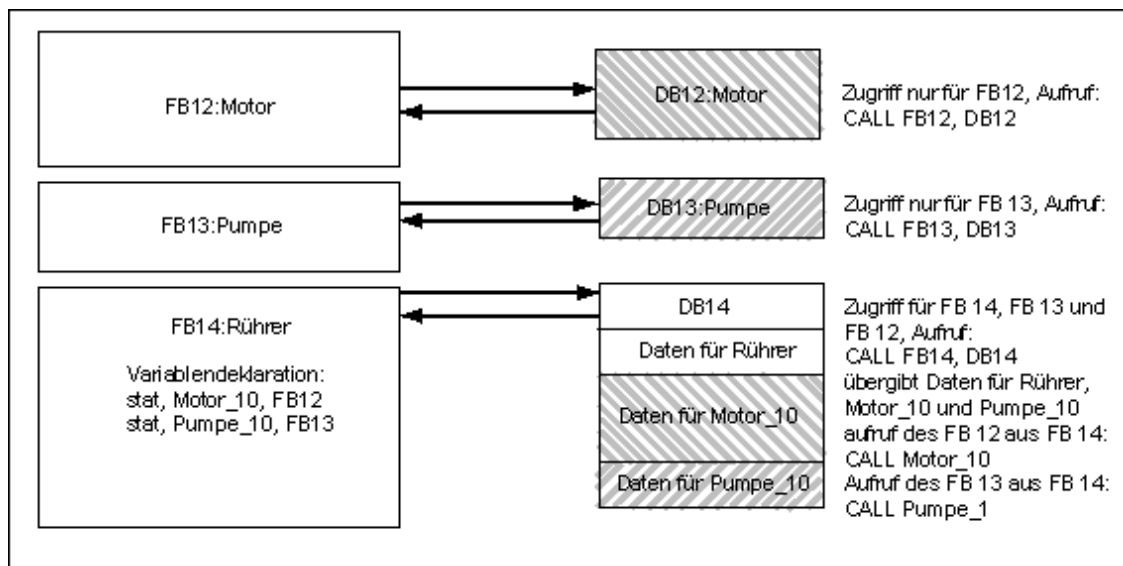
Der FB 22 braucht in diesem Beispiel keinen eigenen Instanz-Datenbaustein, weil seine Instanzdaten im Instanz-Datenbaustein des rufenden FB gespeichert sind.

Ein Instanz-DB für mehrere Instanzen unterschiedlicher FBs (Multiinstanzen)

Sie können in einem Funktionsbaustein Instanzen anderer, bereits erstellter FBs aufrufen. Die dafür benötigten Instanz-Daten können sie dem Instanz-Datenbaustein des aufrufenden FB zuordnen, d. h. Sie brauchen für die aufgerufenen FBs in diesem Fall keine zusätzlichen Datenbausteine.

Für diese Multiinstanzen in einem Instanz-DB müssen Sie im Deklarationsteil des aufrufenden FBs statische Variablen mit dem Datentyp des aufgerufenen FBs für die einzelnen Instanzen deklarieren. Der Aufruf innerhalb des FB erfolgt dann ohne Angabe eines Instanz-DB nur über den Namen der Variable.

In dem Beispiel im Bild werden die zugeordneten Instanzdaten gemeinsam in einem Instanz-DB gespeichert.



4.2.3.5 Globale Datenbausteine (DB)

Datenbausteine enthalten im Gegensatz zu Codebausteinen keine STEP 7-Anweisungen. Sie dienen der Aufnahme von Anwenderdaten, d. h. in den Datenbausteinen stehen variable Daten, mit denen das Anwenderprogramm arbeitet. Globale Datenbausteine dienen der Aufnahme von Anwenderdaten, die von allen anderen Bausteinen aus verwendet werden können.

Die Größe von DBs kann variieren. Die maximal zulässige Größe entnehmen Sie den CPU-Beschreibungen /70/ und /101/.

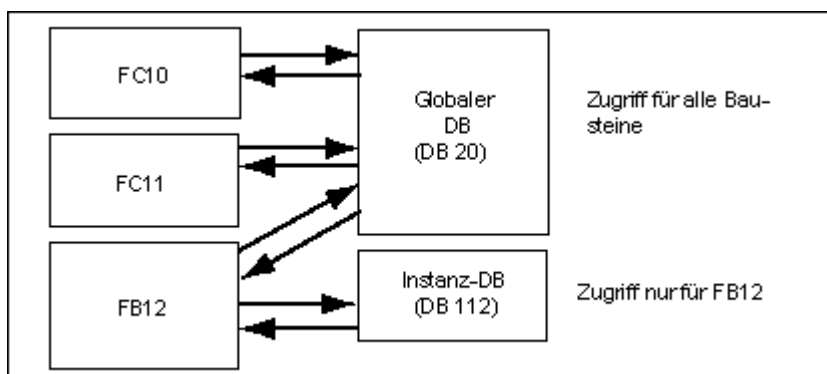
Die Struktur globaler Datenbausteine können Sie beliebig festlegen.

Globale Datenbausteine im Anwenderprogramm

Wird ein Codebaustein (FC, FB oder OB) aufgerufen, so kann er temporär Speicherplatz im Lokaldatenbereich (L-Stack) belegen. Zusätzlich zu diesem Lokaldatenbereich kann ein Codebaustein einen Speicherbereich in Form eines DB öffnen. Im Gegensatz zu den Daten im Lokaldatenbereich werden die in einem DB enthaltenen Daten nicht gelöscht, wenn der DB geschlossen wird, bzw. die Bearbeitung des zugehörigen Codebausteins beendet ist.

Jeder FB, FC oder OB kann die Daten aus einem globalen DB lesen oder selbst Daten in einen globalen DB schreiben. Diese Daten bleiben im DB auch dann erhalten, wenn der DB verlassen wird.

Ein globaler DB und ein Instanz-DB können gleichzeitig geöffnet sein. Nachfolgendes Bild zeigt die verschiedenen Zugriffe auf Datenbausteine.



4.2.3.6 Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)

Vorgefertigte Bausteine

Sie müssen nicht jede Funktion selbst programmieren. S7-CPU's bieten Ihnen vorgefertigte Bausteine, die Sie aus dem Anwenderprogramm heraus aufrufen können.

Nähere Informationen finden Sie in der Referenzhilfe zu Systembausteinen und Systemfunktionen (Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute).

Systemfunktionsbausteine

Ein Systemfunktionsbaustein SFB ist ein Funktionsbaustein, der in die S7-CPU integriert ist. Weil SFBs Teil des Betriebssystems sind, werden sie nicht als Teil des Programms geladen. Wie FBs sind SFBs Bausteine "mit Gedächtnis". Sie müssen auch für SFBs Instanz-Datenbausteine erstellen und als Teil des Programms in die CPU laden.

S7-CPU's bieten SFBs

- zur Kommunikation über projektierte Verbindungen
- für integrierte Sonderfunktionen (z. B. SFB 29 "HS_COUNT" auf der CPU 312 IFM und der CPU 314 IFM)

Systemfunktionen

Eine Systemfunktion ist eine vorprogrammierte Funktion, die in die S7-CPU integriert ist. Sie können die SFC aus Ihrem Programm aufrufen. Weil SFCs Teile des Betriebssystems sind, werden sie nicht als Teil des Programms geladen. Wie FCs sind SFCs Bausteine "ohne Gedächtnis".

S7-CPU's bieten SFCs für:

- Kopier- und Bausteinfunktionen
- die Programmkontrolle
- die Handierung der Uhr und des Betriebsstundenzählers
- die Übertragung von Datensätzen
- Übertragung von Ereignissen im Betriebsmodus Multicomputing, von einer CPU auf alle gesteckten CPU's
- die Handierung von Uhrzeit- und Verzögerungsalarmen
- die Handierung von Synchronfehlerereignissen, Alarm- und Asynchronfehlerereignissen
- die Auskunft über statische und dynamische Systemdaten, z. B. Diagnose
- die Prozessabbildaktualisierung und die Bitfeldbearbeitung
- die Adressierung von Baugruppen
- die dezentrale Peripherie

- die Globaldaten-Kommunikation
- die Kommunikation über nichtprojektierte Verbindungen
- Erzeugung bausteinbezogener Meldungen

Weitere Informationen

Nähere Informationen zu SFBs und SFCs entnehmen Sie dem Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen". Dem Handbuch "Automatisierungssystem S7-300, Aufbau, CPU-Daten" oder dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten" entnehmen Sie, welche SFBs und SFCs verfügbar sind.

4.2.4 Organisationsbausteine für alarmgesteuerte Programmbearbeitung

Die S7-CPU's geben Ihnen durch die Bereitstellung von Alarm-OBs die Möglichkeit

- Programmteile zeitgesteuert zu bearbeiten
- auf externe Signale des Prozesses optimal zu reagieren.

Das zyklische Anwenderprogramm muss nicht ständig abfragen, ob Alarmereignisse eingetreten sind, sondern das Betriebssystem sorgt im Falle eines Alarms dafür, dass der Teil des Anwenderprogramms bearbeitet wird, der im Alarm-OB steht und festlegt, wie das Automatisierungssystem bei diesem Alarm reagieren soll.

Alarmarten und Anwendungen

Nachfolgende Tabelle zeigt, wie die Alarmarten eingesetzt werden können.

Alarmart	Alarm-OBs	Anwendungsbeispiele
Uhrzeitalarm	OB 10 bis OB 17	Errechnen der Durchflussmenge eines Mischprozesses bei Schichtende.
Verzögerungsalarm	OB 20 bis OB 23	Steuern eines Lüfters, der nach dem Abschalten eines Motors noch 20 s laufen soll, bevor er abgeschaltet wird.
Weckalarm	OB 30 bis OB 38	Abtasten eines Signalpegels für eine regelungstechnische Anlage.
Prozessalarm	OB 40 bis OB 47	Melden, dass der maximale Füllstand eines Behälters erreicht ist.

4.2.4.1 Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17)

Die S7-CPU's stellen Uhrzeitalarm-OBs zur Verfügung, die zu einem angegebenen Datum oder in bestimmten Zeitabständen bearbeitet werden können.

Uhrzeitalarme können ausgelöst werden:

- einmalig zu einem bestimmten Zeitpunkt (Absolutzeitangabe mit Datum)
- periodisch mit Angabe des Startzeitpunktes und der Wiederholrate (z. B. minütlich, stündlich, täglich).

Regeln für Uhrzeitalarme

Uhrzeitalarme können nur bearbeitet werden, wenn ein Uhrzeitalarm parametrierbar war und ein entsprechender Organisationsbaustein im Anwenderprogramm enthalten ist. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (OB 80, siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)").

Periodische Uhrzeitalarme müssen einem realen Datum entsprechen. Die monatliche Wiederholung eines OB 10 mit Startzeit 31.1. ist nicht möglich. In diesem Fall würde der OB nur in den Monaten gestartet, die tatsächlich 31 Tage haben (also nicht im Februar, April, Juni etc.).

Ein Uhrzeit-Alarm, der während des Anlaufs (Neustart (Warmstart) oder Wiederanlauf) aktiviert wird, wird erst nach der Beendigung des Anlaufs bearbeitet.

Uhrzeitalarm-OBs, die durch Parametrierung ausgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Nach Neustart (Warmstart) müssen gestellte Uhrzeitalarme neu aktiviert werden (z. B. mit Hilfe der SFC 30 ACT_TINT im Anlaufprogramm).

Starten des Uhrzeitalarms

Damit die CPU einen Uhrzeitalarm starten kann, müssen Sie den Uhrzeitalarm erst stellen und dann aktivieren. Es gibt drei Startmöglichkeiten:

- automatischer Start des Uhrzeitalarms durch Parametrieren mit STEP 7 (Parameterblock "Uhrzeitalarme")
- stellen und aktivieren des Uhrzeitalarms durch SFC 28 SET_TINT und SFC 30 ACT_TINT aus dem Anwenderprogramm heraus
- stellen des Uhrzeitalarms durch Parametrieren mit STEP 7 und aktivieren des Uhrzeitalarms durch die SFC 30 ACT_TINT aus dem Anwenderprogramm heraus.

Abfragen des Uhrzeitalarms

Um abzufragen, ob und zu welchem Zeitpunkt Uhrzeitalarme gestellt sind, können Sie

- die SFC 31 QRY_TINT aufrufen oder
- die Teilliste "Alarmstatus" der Systemzustandsliste anfordern.

Deaktivieren des Uhrzeitalarms

Sie können Uhrzeitalarme, die noch nicht bearbeitet wurden, mit der SFC 29 CAN_TINT deaktivieren. Deaktivierte Uhrzeitalarme lassen sich mit der SFC 28 SET_TINT erneut stellen und mit der SFC 30 ACT_TINT aktivieren.

Priorität der Uhrzeitalarm-OBs

Alle acht Uhrzeitalarm-OBs haben in der Voreinstellung die gleiche Prioritätsklasse (2) und werden dementsprechend in der Reihenfolge ihres Startereignisses bearbeitet. Die Prioritätsklasse kann durch Parametrierung geändert werden.

Ändern der eingestellten Uhrzeit

Es gibt die folgenden Möglichkeiten, die eingestellte Uhrzeit zu ändern:

- ein Uhrzeitmaster synchronisiert die Uhrzeit für Master und Slaves
- im Anwenderprogramm wird durch SFC 0 SET_CLK die Uhrzeit neu gestellt.

Verhalten bei Uhrzeitänderung

Nachfolgende Tabelle zeigt, wie sich Uhrzeitalarme nach dem Ändern der Uhrzeit verhalten.

Wenn...	dann...
durch das Vorstellen der Uhr ein oder mehrere Uhrzeitalarme übersprungen werden,	wird der OB 80 gestartet und in die Startinformation des OB 80 wird eingetragen, welche Uhrzeitalarme übersprungen wurden.
Sie im OB 80 die übersprungenen Uhrzeitalarme deaktiviert haben,	werden die übersprungenen Uhrzeitalarme nicht nachgeholt.
Sie im OB 80 die übersprungenen Uhrzeitalarme nicht deaktiviert haben,	wird der erste übersprungene Uhrzeitalarm nachgeholt, die anderen übersprungenen Uhrzeitalarme werden ignoriert.
durch das Rückstellen der Uhr bereits bearbeitete Uhrzeitalarme erneut anstehen,	wird die Bearbeitung dieser Uhrzeitalarme bei S7-300-CPU s wiederholt bei S7-400-CPU s und CPU 318 nicht wiederholt.

4.2.4.2 Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)

Die S7-CPU's stellen Verzögerungsalarm-OBs zur Verfügung, mit deren Hilfe Sie die zeitverzögerte Bearbeitung von Teilen Ihres Anwenderprogramms programmieren können.

Regeln für Verzögerungsalarne

Verzögerungsalarne können nur bearbeitet werden, wenn sich ein entsprechender Organisationsbaustein im CPU-Programm befindet. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (OB 80, siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)").

Verzögerungsalarm-OBs, die durch Parametrierung ausgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Verzögerungsalarne werden ausgelöst, wenn die in der SFC 32 SRT_DINT angegebene Verzögerungszeit abgelaufen ist.

Starten des Verzögerungsalarms

Um einen Verzögerungsalarm zu starten, müssen Sie in der SFC 32 die Verzögerungszeit festlegen, nach deren Ablauf der entsprechende Verzögerungsalarm-OB aufgerufen werden soll. Die maximal zulässige Länge der Verzögerungszeit entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten".

Priorität der Verzögerungsalarm-OBs

In der Voreinstellung haben die Verzögerungsalarm-OBs die Prioritätsklassen 3 bis 6. Die Prioritätsklassen können durch Parametrierung geändert werden.

4.2.4.3 Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)

Die S7-CPU's stellen Weckalarm-OBs zur Verfügung, welche die zyklische Programmbearbeitung in bestimmten Abständen unterbrechen.

Weckalarme werden in bestimmten Zeitintervallen ausgelöst. Startzeitpunkt des Zeittaktes ist der Betriebszustandswechsel von STOP in RUN.

Regeln für Weckalarme

Achten Sie beim Vorgeben der Zeittakte darauf, dass zwischen den Startereignissen der einzelnen Weckalarme genügend Zeit für die Bearbeitung der Weckalarme bleibt.

Weckalarm-OBs, die durch Parametrierung abgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Starten des Weckalarms

Um einen Weckalarm zu starten, müssen Sie mit STEP 7 im Parameterblock "Weckalarme" einen Zeittakt vorgeben. Der Zeittakt ist immer ein ganzzahliges Vielfaches des Grundtaktes von 1 ms.

$\text{Zeittakt} = n \times \text{Grundtakt } 1 \text{ ms}$

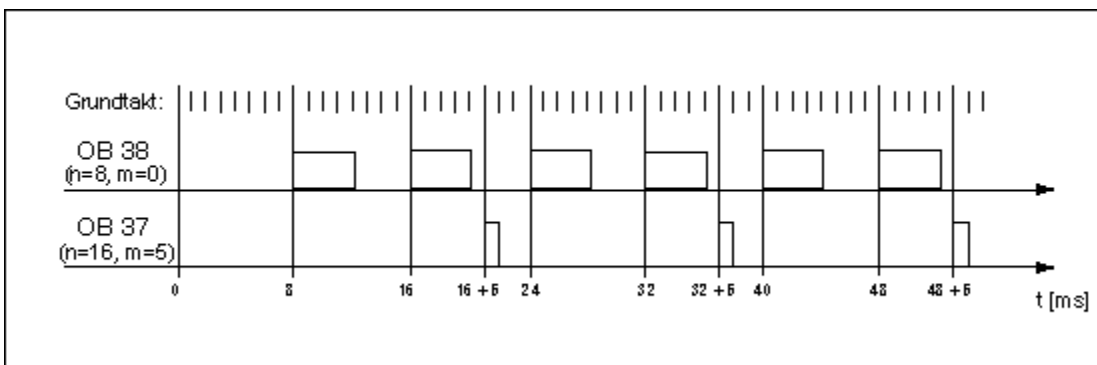
Die neun zur Verfügung stehenden Weckalarm-OBs geben in ihrer Voreinstellung Zeittakte vor (siehe nachfolgende Tabelle). Der Defaultzeittakt wird wirksam, wenn der ihm zugeordnete Weckalarm-OB geladen ist. Sie können jedoch durch Parametrierung die voreingestellten Werte ändern. Die Obergrenze entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten".

Phasenverschiebung bei Weckalarmen

Um zu verhindern, dass die Weckalarme verschiedener Weckalarm-OBs zum gleichen Zeitpunkt eine Startaufforderung erhalten und dadurch möglicherweise ein Zeitfehler (Zykluszeitüberschreitung) entsteht, haben Sie die Möglichkeit, eine Phasenverschiebung vorzugeben. Die Phasenverschiebung sorgt dafür, dass die Bearbeitung eines Weckalarms nach Ablauf des Zeittaktes um einen bestimmten Zeitraum verschoben wird.

$\text{Phasenverschiebung} = m \times \text{Grundtakt (mit } 0 \leq m < n)$

Nachfolgendes Bild zeigt die Bearbeitung eines Weckalarm-OBs mit Phasenverschiebung (OB 37) im Gegensatz zu einem Weckalarm ohne Phasenverschiebung (OB 38).



Priorität der Weckalarm-OBs

Nachfolgende Tabelle zeigt die voreingestellten Zeittakte und Prioritätsklassen der Weckalarm-OBs. Zeittakt und Prioritätsklasse können durch Parametrierung geändert werden.

Weckalarm-OBs	Zeittakt in ms	Prioritätsklasse
OB 30	5000	7
OB 31	2000	8
OB 32	1000	9
OB 33	500	10
OB 34	200	11
OB 35	100	12
OB 36	50	13
OB 37	20	14
OB 38	10	15

4.2.4.4 Organisationsbausteine für den Prozessalarm(OB 40 bis OB 47)

Die S7-CPU's stellen Prozessalarm-OBs zur Verfügung, die auf Signale aus den Baugruppen (z. B. Signalbaugruppen SMs, Kommunikationsprozessoren CPs, Funktionsbaugruppen FMs) reagieren. Mit STEP 7 können Sie für parametrierbare Digital- und Analogbaugruppen einstellen, welches Signal den OB starten soll. Bei CPs und FMs verwenden Sie hierzu die entsprechenden Parametriermasken.

Prozessalarme werden ausgelöst, wenn eine prozessalarmfähige Signalbaugruppe mit parametrierter Prozessalarmfreigabe ein empfangenes Prozesssignal an die CPU weiterleitet oder eine Funktionsbaugruppe der CPU einen Alarm meldet.

Regeln für Prozessalarme

Prozessalarme können nur bearbeitet werden, wenn sich ein entsprechender Organisationsbaustein im CPU-Programm befindet. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)").

Prozessalarm-OBs, die durch Parametrierung abgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Parametrieren prozessalarmfähiger Signalbaugruppen

Jeder Kanal einer prozessalarmfähigen Signalbaugruppe kann einen Prozessalarm auslösen. Deshalb müssen Sie mit STEP 7 in den Parametersätzen prozessalarmfähiger Signalbaugruppen festlegen:

- wodurch ein Prozessalarm ausgelöst werden soll
- welcher Prozessalarm-OB bearbeitet werden soll (die Voreinstellung sieht für die Bearbeitung aller Prozessalarme den OB 40 vor).

Mit STEP 7 aktivieren Sie die Prozessalarmgenerierung der Funktionsbaugruppen. Weitere Parameter weisen Sie in den Parametriermasken dieser Funktionsbaugruppen zu.

Priorität der Prozessalarm-OBs

In der Voreinstellung haben die Prozessalarm-OBs die Prioritätsklassen 16 bis 23. Die Prioritätsklassen können durch Parametrierung geändert werden.

4.2.4.5 Organisationsbausteine für den Anlauf (OB 100 / OB 101 / OB 102)

Anlaufarten

Man unterscheidet die folgende Anlaufarten:

- Wiederanlauf (nicht bei S7-300 und S7-400H)
- Neustart (Warmstart)
- Kaltstart

Der folgenden Tabelle können Sie entnehmen, welcher OB das Betriebssystem im Anlauf jeweils aufruft.

Anlaufart	Zugehöriger OB
Wiederanlauf	OB 101
Neustart (Warmstart)	OB 100
Kaltstart	OB 102

Startereignisse für Anlauf-OBs

Die CPU führt einen Anlauf durch

- nach NETZ-EIN
- wenn Sie den Betriebsartenschalter von STOP "RUN"/"RUN-P" umschalten
- nach Anforderung durch eine Kommunikationsfunktion
- nach Synchronisierung im Multicomputing-Betrieb
- bei einem H-System nach dem Ankoppeln (nur auf Reserve-CPU)

Abhängig vom Startereignis, von der vorliegenden CPU und deren eingestellten Parametern wird der zugehörige Anlauf-OB (OB 100, OB 101 oder OB 102) aufgerufen.

Anlaufprogramm

Sie können die Randbedingungen für das Anlaufverhalten (Initialisierungswerte für RUN, Anlaufwerte für Peripheriebaugruppen) Ihrer CPU bestimmen, indem Sie Ihr Programm für den Anlauf in den Organisationsbausteinen OB 100 für Neustart (Warmstart), OB 101 für Wiederanlauf oder OB 102 für Kaltstart hinterlegen.

Das Anlaufprogramm kann beliebig lang sein, für seine Ausführung besteht keine zeitliche Begrenzung, die Zykluszeitüberwachung ist nicht aktiv. Zeit- oder alarmgesteuerte Bearbeitung ist im Anlaufprogramm nicht möglich. Im Anlauf führen alle Digitalausgänge den Signalzustand 0.

Anlaufart nach manuellem Anlauf

Bei S7-300-CPU ist nur manueller Neustart (Warmstart) oder Kaltstart (nur CPU 318-2) möglich.

Bei einigen S7-400-CPU können Sie mit dem Betriebsartenschalter und dem Anlaufartenschalter (CRST/WRST) einen manuellen Wiederanlauf oder Kaltstart durchführen, wenn dies bei der Parametrierung mit STEP 7 so festgelegt wurde. Manueller Neustart (Warmstart) ist ohne Parametrierung möglich.

Anlaufart nach automatischem Anlauf

Bei S7-300-CPU ist nach NETZ-EIN nur Neustart (Warmstart) möglich.

Bei S7-400-CPU können Sie festlegen, ob ein automatischer Anlauf nach NETZ-EIN zu einem Neustart (Warmstart) oder einem Wiederanlauf führt.

Löschen des Prozessabbilds

Beim Wiederanlauf einer S7-400-CPU wird nach der Bearbeitung des Restzyklus defaultmäßig das Prozessabbild der Ausgänge gelöscht. Sie können das Löschen des Prozessabbildes abwählen, wenn das Anwenderprogramm nach Wiederanlauf mit den Werten weiterarbeiten soll, die vor dem Wiederanlauf aktuell waren.

Soll-Ist-Baugruppenüberwachung

Sie können durch Parametrierung festlegen, dass vor dem Anlauf überprüft wird, ob alle Baugruppen, die in der Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind und der Baugruppentyp stimmt.

Ist die Baugruppenüberprüfung aktiviert, wird der Anlauf nicht durchgeführt, wenn ein Soll-Ist-Unterschied festgestellt wird.

Überwachungszeiten

Um einen fehlerfreien Anlauf des Automatisierungssystems zu gewährleisten, können Sie die folgenden Überwachungszeiten parametrieren:

- die maximal zulässige Zeit für die Übertragung der Parameter zu den Baugruppen
- die maximal zulässige Zeit für die Fertigmeldung der Baugruppen nach NETZ-EIN
- bei S7-400-CPU die maximale Unterbrechungszeit in der noch ein Wiederanlauf zulässig ist.

Nach Ablauf der Überwachungszeiten geht die CPU in STOP, bzw. ist nur Neustart (Warmstart) möglich.

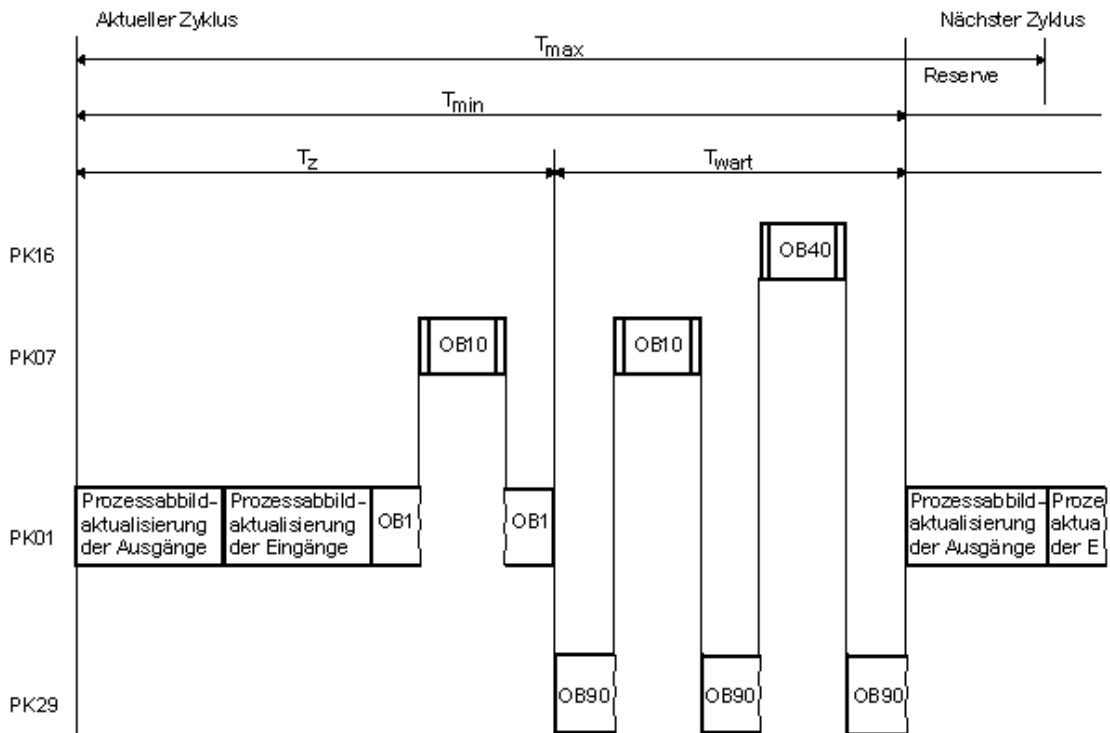
4.2.4.6 Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90)

Falls Sie mit STEP 7 eine Mindestzykluszeit vorgegeben haben und diese größer als die tatsächliche Zykluszeit ist, steht der CPU am Ende des zyklischen Programms Bearbeitungszeit zur Verfügung. Diese wird für die Bearbeitung des Hintergrund-OBs genutzt. Falls der OB 90 auf Ihrer CPU nicht vorhanden ist, wartet die CPU, bis die vorgegebene Mindestzykluszeit abgelaufen ist. Sie können über den OB 90 also zeitunkritische Prozesse ablaufen lassen und damit Wartezeiten vermeiden.

Priorität des Hintergrund-OB

Der Hintergrund-OB hat die Prioritätsklasse 29, was der Priorität 0.29 entspricht. Er ist also der OB mit der niedrigsten Priorität. Die Prioritätsklasse kann durch Parametrierung nicht geändert werden.

Das folgende Bild zeigt ein Beispiel für die Bearbeitung des Hintergrundzyklus, des Freien Zyklus und des OB 10 (bei CPUs ab 10/98).



- Tmax = die parametrierte Zyklusüberwachungszeit
- Tmin = die parametrierte Mindestzykluszeit
- Tz = die tatsächliche Zykluszeit
- Twart = die Differenz zwischen Tmin und tatsächlicher Zykluszeit, in dieser Zeit können auftretende Alarmer sowie der Hintergrund-OB bearbeitet werden
- PK = Prioritätsklasse

Programmieren des OB 90

Die Laufzeit des OB 90 wird vom Betriebssystem der CPU nicht überwacht, so dass Sie im OB 90 Schleifen beliebiger Länge programmieren können. Achten Sie auf die Konsistenz der von Ihnen im Hintergrundprogramm verwendeten Daten, indem Sie folgendes bei Ihrer Programmierung berücksichtigen:

- die Rücksetzereignisse des OB 90 (siehe Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen"),
- die zum OB 90 asynchrone Prozessabbildaktualisierung.

4.2.4.7 Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)

Arten von Fehlern

Die Fehler, die S7-CPU erkennen und auf die Sie mit Hilfe von Organisationsbausteinen reagieren können, lassen sich in zwei Kategorien einteilen:

- **Synchrone Fehler:** Diese Fehler können einem bestimmten Teil des Anwenderprogramms zugeordnet werden. Der Fehler wird während der Bearbeitung von einer bestimmten Operation ausgelöst. Ist der entsprechende synchrone Fehler-OB nicht geladen, geht die CPU in STOP, wenn der Fehler auftritt.
- **Asynchrone Fehler:** Diese Fehler lassen sich nicht direkt dem bearbeiteten Anwenderprogramm zuordnen. Es handelt sich um Prioritätsklassenfehler, Fehler im Automatisierungssystem (z. B. Baugruppendefekte) oder um Redundanzfehler. Ist der entsprechende asynchrone Fehler-OB nicht geladen, geht die CPU in STOP, wenn der Fehler auftritt (Ausnahmen: OB 70, OB 72, OB 81, OB 87).

Nachfolgende Tabelle zeigt die Arten von Fehlern, die auftreten können, aufgeteilt nach der Kategorie der Fehler-OBs.

Asynchrone Fehler / Redundanzfehler	Synchrone Fehler
OB 70 Peripherie-Redundanzfehler (nur H-CPU)	OB 121 Programmierfehler (z. B. DB ist nicht geladen)
OB 72 CPU-Redundanzfehler (nur in H-CPU, z. B. Ausfall einer CPU)	OB 122 Peripheriezugriffsfehler (z. B. Zugriff auf eine Signalbaugruppe, die nicht vorhanden ist)
OB 80 Zeitfehler (z. B. Zykluszeit überschritten)	
OB 81 Stromversorgungsfehler (z. B. Batteriefehler)	
OB 82 Diagnosealarm (z. B. Kurzschluss in der Eingangsbaugruppe)	
OB 83 Ziehen-/Stecken-Alarm (z. B. Ziehen einer Eingabebaugruppe)	
OB 84 CPU-Hardwarefehler (Fehler an der Schnittstelle zum MPI-Netz)	
OB 85 Programmablauffehler (z. B. OB ist nicht geladen)	
OB 86 Baugruppenträgerausfall	
OB 87 Kommunikationsfehler (z. B. falsche Telegrammkennung bei Globaldaten-Kommunikation)	

Verwenden der OBs für synchrone Fehler

Synchrone Fehler werden während der Bearbeitung einer bestimmten Operation verursacht. Wenn diese Fehler auftreten, erstellt das Betriebssystem einen Eintrag im U-Stack und startet den OB für synchrone Fehler.

Die Fehler-OBs, die von synchronen Fehlern aufgerufen werden, werden als Teil des Programms in derselben Prioritätsklasse wie der Baustein bearbeitet, der beim Erkennen des Fehlers bearbeitet wird. Die Details über den Fehler, der den OB-Aufruf ausgelöst hat, finden Sie in der Startinformation des OB. Sie können diese Information dazu verwenden, auf die Fehlerbedingung zu reagieren und dann zu der Bearbeitung Ihres Programms zurückzukehren (z. B. bei Zugriffsfehler auf eine Analogeingabebaugruppe im OB 122 mit der SFC 44 RPL_VAL einen Ersatzwert vorgeben). Damit belasten die Lokaldaten der Fehler-OBs aber auch zusätzlich den L-Stack dieser Prioritätsklasse.

Bei S7-400-CPU kann aus einem synchronen Fehler-OB ein weiterer synchroner Fehler-OB gestartet werden. Bei S7-300 CPU ist dies nicht möglich.

Verwenden der OBs für asynchrone Fehler

Wenn das Betriebssystem der CPU einen asynchronen Fehler entdeckt, dann startet es den entsprechenden Fehler-OB (OB 70 bis OB 73 und OB 80 bis OB 87). Die OBs für asynchrone Fehler haben die höchste Priorität voreingestellt: Sie können von anderen OBs nicht unterbrochen werden, wenn alle asynchronen Fehler-OBs die gleiche Priorität haben. Treten mehrere asynchrone Fehler-OBs gleicher Priorität gleichzeitig auf, werden sie in der Reihenfolge ihres Auftretens bearbeitet.

Maskieren von Startereignissen

Sie können mit Systemfunktionen (SFC) die Startereignisse für einige Fehler-OBs maskieren bzw. verzögern oder sperren. Nähere Informationen hierzu und zu den Organisationsbausteinen im Einzelnen entnehmen Sie dem Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen".

Art des Fehler-OBs	SFC	Funktion der SFC
Synchrone Fehler-OBs	SFC 36 MSK_FLT	Einzelne Synchronfehlerereignisse maskieren. Maskierte Fehlerereignisse starten keinen Fehler-OB und führen nicht zur programmierten Ersatzreaktion.
	SFC 37 DMSK_FLT	Synchronfehlerereignisse demaskieren
Asynchrone Fehler-OBs	SFC 39 DIS_IRT	Alarm- und Asynchronfehlerereignisse pauschal sperren. Gesperrte Fehlerereignisse starten in keinem nachfolgenden CPU-Zyklus. Fehler-OBs und führen nicht zur programmierten Ersatzreaktion.
	SFC 40 EN_IRT	Alarm- und Asynchronfehlerereignisse freigeben
	SFC 41 DIS_AIRT	Höherpriorere Alarm- und Asynchronfehlerereignisse bis zum OB-Ende verzögern
	SFC 42 EN_AIRT	Höherpriorere Alarm- und Asynchronfehlerereignisse freigeben

Hinweis

Um Alarme zu ignorieren, ist es effektiver, diese per SFC im Anlauf zu sperren, anstatt einen leeren OB (mit Inhalt BE) zu laden.

5 Starten und Bedienen

5.1 Starten von STEP 7



Nach dem Start von Windows finden Sie auf der Windows-Oberfläche ein Symbol für den SIMATIC Manager, dem Einstieg in die STEP 7-Software.

STEP 7 wird am schnellsten mit einem Doppelklick auf das Symbol "SIMATIC Manager" gestartet. Daraufhin wird das Fenster des SIMATIC Managers geöffnet. Von hier aus lassen sich alle von Ihnen installierten Funktionen sowohl des Basispakets als auch der Optionspakete ansprechen.

Alternativ können Sie den SIMATIC Manager auch über die Schaltfläche "Start" auf der Taskleiste des Betriebssystems starten. Den Eintrag finden Sie unter "Simatic".

Hinweis

Nähere Informationen über Windows-Standardbedienungen und -Standardoptionen finden Sie in Ihrem Windows-Benutzerhandbuch oder in der Online-Hilfe Ihres Windows Betriebssystems.

SIMATIC Manager

Der SIMATIC Manager ist die Einstiegsoberfläche für die Konfigurierung und Programmierung. Sie können:

- Projekte einrichten
- Hardware konfigurieren und parametrieren
- Hardware-Vernetzungen projektieren
- Bausteine programmieren
- Ihre Programme testen und inbetriebnehmen

Der Zugang zur Funktionalität ist objektorientiert gestaltet und intuitiv und leicht erlernbar.

Sie können mit dem SIMATIC Manager

- offline, d. h. ohne angeschlossene Steuerung oder
- online, d. h. mit angeschlossener Steuerung

arbeiten. (Beachten Sie dabei die entsprechenden Sicherheitshinweise.)

Weiteres Vorgehen

Sie erstellen Automatisierungslösungen in Form von "Projekten". Sie erleichtern sich das Arbeiten, wenn Sie sich zuvor mit einigen grundlegenden Themen vertraut machen:

- der Benutzungsoberfläche,
- einigen Basisbedienungen,
- der Online-Hilfe.

5.2 Starten von STEP 7 mit vorgegebenen Startparametern

Ab STEP 7 V5 können Sie sich mehrere Symbole des SIMATIC Managers erzeugen und in der Aufrufzeile Startparameter angeben. Dadurch können Sie den SIMATIC Manager veranlassen, auf das durch diese Parameter beschriebene Objekt zu positionieren. Damit gelangen Sie einfach per Doppelklick sofort an die entsprechenden Stellen in einem Projekt.

Beim Aufruf von **s7tgotpx.exe** können Sie folgende Startparameter angeben:

/e <vollständiger physikalischer Projektpfad>

/o <logischer Pfad des Objektes, auf welches positioniert werden soll>

/h <ObjektID>

/onl

Der Startparameter **/onl** bewirkt, dass das Projekt online geöffnet wird und der angegebene Pfad aufgerufen wird.

/off

Der Startparameter **/off** bewirkt, dass das Projekt offline geöffnet wird und der angegebene Pfad aufgerufen wird.

/keep

Der Startparameter **/keep** bewirkt, dass bei geöffnetem SIMATIC Manager das über die Kommandozeile neu zu öffnende Projekt zusätzlich zu den bereits angezeigten Projekten geöffnet wird. Wenn der SIMATIC Manager noch nicht geöffnet war, so wird das neu zu öffnende Projekt zusätzlich zu den im Sitzungsgedächtnis des SIMATIC Managers hinterlegten Projekten geöffnet. Wird dieser Startparameter nicht angegeben, werden geöffnete Projekte zunächst geschlossen, das Sitzungsgedächtnis ignoriert und nur das übergebene Projekt geöffnet.

Die geeigneten Parameter lassen sich am einfachsten wie folgt ermitteln.

Ermitteln der Parameter durch Kopieren und Einfügen

Gehen Sie so vor:

1. Legen Sie auf ihrem Desktop eine neue Verknüpfung mit der Datei **s7tgotpx.exe** an. Sie finden diese Datei in Ihrem Installationsverzeichnis unter **S7bin**.
2. Lassen Sie sich das Dialogfeld zu den Eigenschaften anzeigen.
3. Wählen Sie das Register "Verknüpfung". Der Eintrag unter "Ziel" wird nun wie folgt erweitert:
4. Markieren Sie das gewünschte Objekt im SIMATIC Manager.
5. Kopieren Sie das Objekt mit der Tastenkombination **CTRL+ALT+C** in die Zwischenablage.
6. Positionieren Sie den Cursor an das Ende des Eintrags "Ziel" im Register "Verknüpfung".
7. Fügen Sie den Inhalt der Zwischenablage mit der Tastenkombination **CTRL+V** ein.
8. Schließen Sie das Dialogfeld mit "OK".

Beispiel von Parameterangaben:

```
/e F:\SIEMENS\STEP7\S7proj\MyConfig\MyConfig.s7p /keep  
/o "1,8:MyConfig\SIMATIC 400(1)\CPU416-1\S7-Programm(1)\Bausteine\FB1"  
/h T00112001;129;T00116001;1;T00116101;16e /keep
```

Hinweis zum Aufbau des Projektpfads

Der Projektpfad ist der physikalische Pfad im Dateisystem.

Der vollständige logische Pfad ist folgendermaßen aufgebaut:

[Sichtkennung,Onlinekennung]:Projektname\{Objektname}* \ Objektname

Beispiel: /o 1,8:MyConfig\SIMATIC 400(1)\CPU416-1\S7-Programm(1)\Bausteine\FB1

Hinweis zum Aufbau des logischen Pfads

Der vollständige logische Pfad sowie die Objekt-ID können nur mit Hilfe von Kopieren und Einfügen erstellt werden. Es besteht jedoch auch die Möglichkeit, den vom Benutzer lesbaren Pfad anzugeben, d. h. im o. g. Beispiel:

```
/o "MyConfig\SIMATIC 400(1)\CPU416-1\S7-Programm(1)\Bausteine\FB1".
```

Mit **/onl** bzw. **/off** können Sie angeben, ob der Pfad im Online oder Offline-Fenster gültig ist. Diese Angabe ist bei der Vorgehensweise mit Kopieren und Einfügen nicht erforderlich.

Wichtig: Falls Leerzeichen im Pfad vorkommen, muss der Pfad zwischen Hochkommas gesetzt werden.

5.3 Aufrufen der Hilfefunktionen

Online-Hilfe

Die Online-Hilfe bietet Ihnen Informationen für die Stelle an, an der Sie sie benötigen. So können Sie schnell und zielsicher Information nachschlagen, ohne in Handbüchern suchen zu müssen. In der Online-Hilfe finden Sie:

- **Hilfethemen:** bietet verschiedene Zugänge zum Anzeigen von Hilfeinformation.
- **Hilfe zum Kontext** (Taste F1): zeigt Informationen zum markierten Objekt oder zum aktiven Dialogfeld bzw. Fenster an.
- **Einführung:** gibt einen knappen Überblick über Anwendung, wesentliche Merkmale und Funktionsumfang einer Applikation.
- **Erste Schritte:** fasst erste Handlungen zusammen, die Sie durchführen müssen, um zu einem ersten Erfolg zu kommen.
- **Hilfe benutzen:** bietet eine Beschreibung der Möglichkeiten, die Ihnen zur Verfügung stehen, um bestimmte Informationen in der Hilfe zu finden.
- **Info:** liefert Informationen zur aktuellen Version der Applikation.

Über das Menü "Hilfe" können Sie von jedem Fenster aus auf Themen zugreifen, die sich auf die aktuelle Dialogsituation beziehen.

Aufrufen der Online-Hilfe

Sie können die Online-Hilfe auf verschiedene Arten aufrufen:

- Wählen Sie einen Menübefehl aus dem Menü "Hilfe" in der Menüleiste.
- In einem Dialogfeld klicken Sie auf die Schaltfläche "Hilfe". Sie erhalten dann Hilfe zu diesem Dialogfeld.
- Positionieren Sie den Mauszeiger in einem Fenster oder Dialogfeld auf das Thema, zu dem Sie Hilfe benötigen, und drücken Sie die Taste F1 oder wählen Sie den Menübefehl **Hilfe > Hilfe zum Kontext**.
- Fragezeichen-Cursor von Windows.

Die drei letztgenannten Arten werden als kontextabhängige Hilfe bezeichnet.

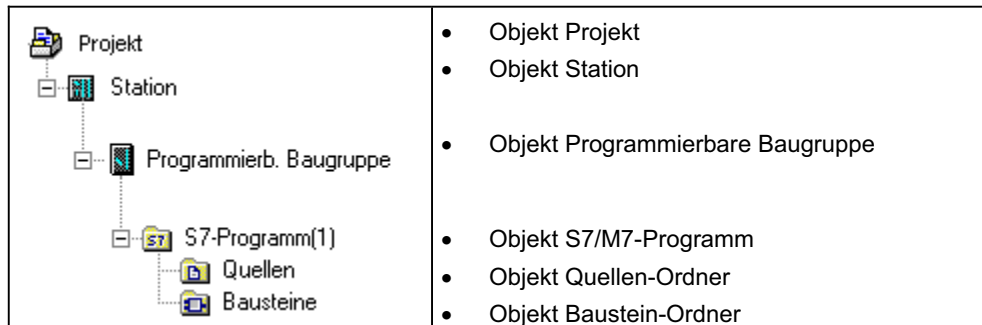
Aufrufen der Kurzhilfe

Eine Kurzhilfe zu Schaltflächen in der Funktionsleiste wird angezeigt, wenn Sie den Cursor auf eine Schaltfläche positionieren und ihn dort kurze Zeit stehen lassen.

5.4 Objekte und Objekthierarchie

Vergleichbar zur Verzeichnisstruktur des Windows-Explorers mit Ordnern und Dateien, ist auch im SIMATIC Manager die Objekt-Hierarchie für Projekte und Bibliotheken aufgebaut.

Nachfolgendes Bild zeigt ein Beispiel für eine Objekt-Hierarchie.



Objekte dienen

- als Träger von Eigenschaften,
- als Ordner,
- als Träger von Funktionen (z. B. zum Start einer bestimmten Applikation).

Objekte als Träger von Eigenschaften

Objekte können sowohl Funktionen als auch Eigenschaften (z. B. Einstellungen) tragen. Nach dem Selektieren eines Objekts können Sie

- mit dem Menübefehl **Bearbeiten > Objekt öffnen** das Objekt bearbeiten.
- mit dem Menübefehl **Bearbeiten > Objekteigenschaften** ein Dialogfeld aufrufen und darin objektspezifische Einstellungen vornehmen.

Auch ein Ordner kann Träger von Eigenschaften sein.

Objekte als Ordner

In einem Ordner können sich weitere Ordner oder Objekte befinden. Diese werden angezeigt, wenn Sie den Ordner öffnen.

Objekte als Träger von Funktionen

Wenn Sie ein Objekt öffnen, wird ein Fenster aufgeblendet, in dem Sie das Objekt bearbeiten können.

Ein Objekt ist entweder Ordner oder Träger von Funktionen. Eine Ausnahme bilden die Stationen: Sie sind sowohl Ordner (für programmierbare Baugruppen) als auch Träger von Funktionen (zum Konfigurieren der Hardware).

- Bei Doppelklick auf eine Station werden die darin enthaltenen Objekte angezeigt: die programmierbaren Baugruppen und die Stationskonfiguration (Station als Ordner).
- Nach dem Öffnen einer Station mit dem Menübefehl **Bearbeiten > Objekt öffnen** können Sie diese Station konfigurieren und parametrieren (Station als Träger einer Funktion). Der Menübefehl hat die gleiche Auswirkung wie ein Doppelklick auf das Objekt Hardware.





5.4.1 Objekt Projekt

Das Projekt repräsentiert die Gesamtheit aller Daten und Programme einer Automatisierungslösung und steht an der Spitze einer Objekthierarchie

Platzierung in der Projektansicht

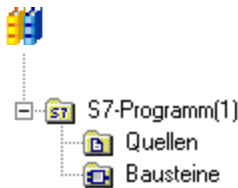
<pre> graph TD Projekt --> Station Station --> Programmierb_Baugruppe[Programmierb. Baugruppe] Programmierb_Baugruppe --> S7_Programm[S7-Programm(1)] S7_Programm --> Quellen S7_Programm --> Bausteine </pre>	<ul style="list-style-type: none"> • Objekt Projekt • Objekt Station • Objekt Programmierbare Baugruppe • Objekt S7/M7-Programm • Objekt Quellen-Ordner • Objekt Baustein-Ordner
---	--


Symbol	Objektordner	Auswahl wichtiger Funktionen
	Projekt	<ul style="list-style-type: none"> • Anlegen eines Projekts • Archivieren von Projekten und Bibliotheken • Drucken einer Projektdokumentation • Texte mehrsprachig verwalten • Projekt auf verwendete Optionspakete prüfen • Reorganisation • Bedienerrelevante Texte übersetzen und bearbeiten • Einfügen von OS-Objekten • Bearbeiten von Projekten durch mehrere Personen • Umsetzen von Version 1-Projekten • Umsetzen von Version 2-Projekten • Einstellen der PG/PC-Schnittstelle



Symbol	Objekte in der Projektebene	Auswahl wichtiger Funktionen
	Station: SIMATIC 300-Station SIMATIC 400-Station	<ul style="list-style-type: none"> • Einfügen von Stationen • Stationen sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Stationsebene). Weitere Funktionen finden Sie bei Objekt Station
 	S7-Programm M7-Programm	<ul style="list-style-type: none"> • S7/M7-Programm ohne Station und CPU • S7-/M7-Programme sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Programmebene). Weitere Funktionen finden Sie bei Objekt S7/M7-Programm
	Netz zum Start der Applikation für die Netzprojektierung und zur Einstellung von Netzeigenschaften	<ul style="list-style-type: none"> • Eigenschaften von Subnetzen und Kommunikationsteilnehmern • Übersicht: Globaldaten-Kommunikation • Vorgehensweise zur Konfigurierung der GD-Kommunikation

5.4.2 Objekt Bibliothek

Eine Bibliothek kann S7-Programme/M7-Programme enthalten und dient zur Ablage von Bausteinen. Sie steht an der Spitze einer Objekthierarchie.

	<ul style="list-style-type: none"> • Objekt Bibliothek • Objekt S7/M7-Programm • Objekt Quellen-Ordner • Objekt Baustein-Ordner
---	---

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Bibliothek	<ul style="list-style-type: none"> • Übersicht der Standardbibliotheken • Arbeiten mit Bibliotheken • Archivieren von Projekten und Bibliotheken

Symbol	Objekte in der Bibliotheksebene	Auswahl wichtiger Funktionen
 	S7-Programm M7-Programm	<ul style="list-style-type: none"> • Einfügen eines S7-/M7-Programms • S7-/M7-Programme sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Programmebene). Weitere Funktionen finden Sie bei Objekt S7/M7-Programm



5.4.3 Objekt Station

Eine SIMATIC 300/400-Station repräsentiert einen S7-Hardware-Aufbau mit einer oder mehreren programmierbaren Baugruppen.

Platzierung in der Projektansicht

	<ul style="list-style-type: none"> • Objekt Projekt • Objekt Station • Objekt Programmierbare Baugruppe • Objekt S7/M7-Programm • Objekt Quellen Ordner • Objekt Baustein-Ordner
--	--

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Station	<ul style="list-style-type: none"> • Einfügen von Stationen • Station laden in PG • Laden einer Konfiguration in ein Zielsystem • Zurückladen einer Konfiguration aus einer Station • CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen • 'Melden von Systemfehlern' projektieren • Diagnose der Hardware und Anzeigen des Baugruppenzustands • Anzeigen und Ändern des Betriebszustands • Anzeigen und Einstellen von Uhrzeit und Datum • Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU
	SIMATIC PC-Station (nicht zugeordnet)	<ul style="list-style-type: none"> • Erzeugen und parametrieren von SIMATIC PC-Stationen • Verbindungen projektieren für eine SIMATIC PC-Station • Laden einer SIMATIC PC-Station
	SIMATIC PC-Station (zugeordnet)	<ul style="list-style-type: none"> • Hervorheben der projektierenden SIMATIC PC-Station in der Netzansicht

Symbol	Objekte in der Stationsebene	Auswahl wichtiger Funktionen
	Hardware	<ul style="list-style-type: none"> • Grundsätzliche Bedienung beim Hardware konfigurieren • Grundsätzliche Schritte beim Konfigurieren einer Station • Überblick: Vorgehensweise beim Konfigurieren und Parametrieren eines zentralen Aufbaus • Grundsätzliche Vorgehensweise zum Konfigurieren eines DP-Mastersystems • Konfigurieren des Multicomputing-Betriebs
	Programmierbare Baugruppe	<ul style="list-style-type: none"> • Programmierbare Baugruppen sind gleichzeitig Objekte (Stationsebene) und Objekt-Ordner (Ebene "Programmierbare Baugruppen"). Weitere Funktionen finden Sie bei Objekt Programmierbare Baugruppe





5.4.4 Objekt Programmierbare Baugruppe

Eine programmierbare Baugruppe repräsentiert die Parametrierungsdaten einer programmierbaren Baugruppe (CPUxxx, FMxxx, CPxxx). Die Systemdaten von Baugruppen, die keinen remanenten Speicher haben (z. B. CP441), werden über die CPU der Station geladen. Solchen Baugruppen ist deshalb kein Objekt "Systemdaten" zugeordnet und sie werden nicht in der Projekthierarchie angezeigt.

Platzierung in der Projektansicht

	<ul style="list-style-type: none"> • Objekt Projekt • Objekt Station • Objekt Programmierbare Baugruppe • Objekt S7/M7-Programm • Objekt Quellen Ordner • Objekt Baustein-Ordner
--	--

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Programmierbare Baugruppe	<ul style="list-style-type: none"> • Überblick: Vorgehensweise beim Konfigurieren und Parametrieren eines zentralen Aufbaus • CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen • 'Melden von Systemfehlern' projektieren • Diagnose der Hardware und Anzeigen des Baugruppenzustands • Laden über EPROM-Memory Cards • Passwortschutz für Zugriff auf Zielsysteme • Forcefenster anzeigen • Anzeigen und Ändern des Betriebszustands • Anzeigen und Einstellen von Uhrzeit und Datum • Einstellen des Betriebsverhaltens • Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU • Diagnosesymbole in der Online-Sicht • Aufteilung der Speicherbereiche • Speichern geladener Bausteine auf integriertem EPROM • Aktualisieren des Betriebssystems auf dem Zielsystem
	Stellvertreter-Objekt für eine programmierbare Baugruppe	<ul style="list-style-type: none"> • Darstellung von Baugruppen, die mit neueren STEP 7-Versionen projektiert sind

Symbol	Objekte in der Ebene "Programmierbare Baugruppen"	Auswahl wichtiger Funktionen
  	<p>Programme:</p> <p>S7-Programm</p> <p>M7-Programm</p> <p>Programm</p>	<ul style="list-style-type: none"> • Einfügen eines S7-/M7-Programms • S7-/M7-Programme sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Programmebene). Weitere Funktionen finden Sie bei Objekt S7/M7-Programm
	<p>Verbindungen zur Definition von Verbindungen im Netzwerk</p>	<ul style="list-style-type: none"> • Vernetzen von Stationen innerhalb eines Projekts • Verbindungstypen und Verbindungspartner • Wissenswertes zu den verschiedenen Verbindungstypen • Eingeben einer neuen Verbindung • Verbindungen projektieren für Baugruppen einer SIMATIC Station





5.4.5 Objekt S7/M7-Programm

Ein (S7-/M7-) Programm ist ein Ordner für Software zu S7/M7-CPU-Baugruppen oder für Software zu Nicht-CPU-Baugruppen (z. B. programmierbare CP- oder FM-Baugruppen).

Platzierung in der Projektansicht

	<ul style="list-style-type: none"> • Objekt Projekt • Objekt Station • Objekt Programmierbare Baugruppe • Objekt S7/M7-Programm • Objekt Quellen Ordner • Objekt Baustein-Ordner
--	--

Symbol	Objektordner	Auswahl wichtiger Funktionen
	S7-Programm	<ul style="list-style-type: none"> • Einfügen eines S7-/M7-Programms • Einstellen des Operandenvorrangs (symbolisch/absolut) • Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen • Vergeben von Meldenummern • Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten (projektweit) • Anwenderdefinierte Diagnosemeldungen anlegen (CPU-weit) • Bedienerrelevante Texte übersetzen und bearbeiten • Texte mehrsprachig verwalten • CPU-Meldungen und anwenderdefinierte Diagnosemeldungen anzeigen • Maßnahmen im Programm zur Störungsbehandlung
	M7-Programm	<ul style="list-style-type: none"> • Vorgehensweise für M7-Systeme
	Programm	<ul style="list-style-type: none"> • Erstellen der Software im Projekt (prinzipiell)

Symbol	Objekte in der Programmebene	Auswahl wichtiger Funktionen
	Quellen-Ordner	<ul style="list-style-type: none"> • Weitere Funktionen finden Sie bei Objekt Quellen-Ordner
	Baustein-Ordner	<ul style="list-style-type: none"> • Weitere Funktionen finden Sie bei Objekt Baustein-Ordner
	Textbibliotheken-Ordner	<ul style="list-style-type: none"> • Anwender-Textbibliotheken
	Symoltabelle zur Zuordnung von Symbolen zu Signalen und anderen Variablen	<ul style="list-style-type: none"> • Absolute und symbolische Adressierung • Struktur und Bestandteile der Symboltabelle • Eingabemöglichkeiten von globalen Symbolen • Allgemeine Hinweise zur Eingabe von Symbolen • Symbolbezogene Meldungen zuordnen und bearbeiten (projektweit) • Symbolbezogene Meldungen zuordnen und bearbeiten (CPU-weit) • Bedienerrelevante Texte übersetzen und bearbeiten • Projektieren von BuB-Attributen über die Symboltabelle • Bearbeiten des Kommunikationsattributs • Exportieren und Importieren von Symboltabellen

5.4.6 Objekt Baustein-Ordner





Ein Baustein-Ordner einer Offline-Ansicht kann enthalten: Codebausteine (OB, FB, FC, SFB, SFC), Datenbausteine (DB), anwenderdefinierte Datentypen (UDT) und Variablentabellen. Das Objekt Systemdaten repräsentiert Systemdatenbausteine.







Der Baustein-Ordner einer Online-Ansicht enthält die ausführbaren Programmteile, die resident im Zielsystem geladen sind.



Platzierung in der Projektansicht

	<ul style="list-style-type: none"> • Objekt Projekt • Objekt Station • Objekt Programmierbare Baugruppe • Objekt S7/M7-Programm • Objekt Quellen Ordner • Objekt Baustein-Ordner
--	--

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Bausteine	<ul style="list-style-type: none"> • Laden mit Projektverwaltung • Laden ohne Projektverwaltung • Übersicht der möglichen Referenzdaten • Umverdrahten • Bausteine vergleichen • Bedienerrelevante Texte übersetzen und bearbeiten • Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute

Symbol	Objekte im Bausteinordner	Auswahl wichtiger Funktionen
	Bausteine allgemein	<ul style="list-style-type: none"> • Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen • Anlegen von Bausteinen • Grundlagen zum Programmieren in AWL-Quellen • Bausteine vergleichen
	OB (Organisationsbaustein)	Zusätzliche Funktionen: <ul style="list-style-type: none"> • Einführung zu Daten- und Parametertypen • Laden • Testen mit Programmstatus • Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte • Umverdrahten • Baustein-Hilfen
	FC (Funktionen)	Zusätzliche Funktionen: <ul style="list-style-type: none"> • Einführung zu Daten- und Parametertypen • Laden • Testen mit Programmstatus • Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte • Umverdrahten • Attribute für Bausteine und Parameter
	FB (Funktionsbausteine)	Zusätzliche Funktionen: <ul style="list-style-type: none"> • Einführung zu Daten- und Parametertypen • Verwendung von Multiinstanzen • Laden • Testen mit Programmstatus • Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte • Umverdrahten • Attribute für Bausteine und Parameter • Bausteinbezogene Meldungen zuordnen und bearbeiten (projektweit) • Bausteinbezogene Meldungen anlegen (CPU-weit) • PCS7-Meldungsprojektierung (projektweit) • PCS 7-Meldungsprojektierung (CPU-weit) • Bedienerrelevante Texte übersetzen und bearbeiten • FB-Parametern BuB-Attribute zuweisen
	UDT (Anwenderdefinierte Datentypen)	<ul style="list-style-type: none"> • Anlegen von Bausteinen • Grundlagen zum Programmieren in AWL- Quellen • Einführung zu Daten- und Parametertypen • Verwenden von anwenderdefinierten Datentypen für den Datenzugriff • Attribute für Bausteine und Parameter

Symbol	Objekte im Bausteinordner	Auswahl wichtiger Funktionen
	DB (Globale Datenbausteine)	<ul style="list-style-type: none"> • Datensicht von Datenbausteinen • Deklarationssicht von Datenbausteinen • Laden • Programmstatus von Datenbausteinen • Einführung zu Daten- und Parametertypen • Verwendung von Multiinstanzen • Attribute für Bausteine und Parameter • Bausteinbezogene Meldungen zuordnen und bearbeiten (projektweit - nur Instanz-DBs) • Bausteinbezogene Meldungen bearbeiten (CPU-weit - nur Instanz-DBs) • PCS7-Meldungsprojektierung (projektweit - nur Instanz-DBs) • PCS 7-Meldungsprojektierung (CPU-weit - nur Instanz-DBs) • Bedienerrelevante Texte übersetzen und bearbeiten (nur Instanz-DBs)
	SFC (Systemfunktionen)	<ul style="list-style-type: none"> • Laden • Attribute für Bausteine und Parameter • Baustein-Hilfen
	SFB (Systemfunktionsbausteine)	<ul style="list-style-type: none"> • Laden • Attribute für Bausteine und Parameter • Bausteinbezogene Meldungen zuordnen und bearbeiten (projektweit) • Bausteinbezogene Meldungen anlegen (CPU-weit) • PCS7-Meldungsprojektierung (projektweit) • PCS 7-Meldungsprojektierung (CPU-weit) • Bedienerrelevante Texte übersetzen und bearbeiten • Baustein-Hilfen
	Baustein mit KNOW HOW Schutz	<ul style="list-style-type: none"> • Regeln zum Festlegen von Bausteineigenschaften in AWL-Quellen • Bausteineigenschaften
	Diagnosefähiger Baustein	Weitere Informationen finden Sie in der Dokumentation zum Optionspaket S7-PDIAG.
	Baustein wurde mit der Erstsprache F-FUP/-KOP/-AWL/-DB erstellt	Weitere Informationen finden Sie in der Dokumentation zum Optionspaket S7 Distributed Safety.

Symbol	Objekte im Bausteinordner	Auswahl wichtiger Funktionen
	VAT (Variablen-tabelle)	<ul style="list-style-type: none"> • Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablen-tabellen • Einführung zum Testen mit der Variablen-tabelle • Einführung zum Beobachten von Variablen • Einführung zum Steuern von Variablen • Einführung zum Forcen von Variablen
	Systemdaten (SDB)	<p>SDBs werden nur indirekt bearbeitet über Funktionen aus:</p> <ul style="list-style-type: none"> • Einführung zum Konfigurieren der Hardware • Eigenschaften von Subnetzen und Kommunikationsteilnehmern • Übersicht: Globaldaten-Kommunikation • Symbolbezogene Meldungen zuordnen und bearbeiten • Laden

5.4.7 Objekt Quellen-Ordner

Ein Quellen-Ordner enthält Quellprogramme in Textform.

Platzierung in der Projektansicht

	<ul style="list-style-type: none"> • Objekt Projekt • Objekt Station • Objekt Programmierbare Baugruppe • Objekt S7/M7-Programm • Objekt Quellen-Ordner • Objekt Baustein-Ordner
--	--

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Quellen-Ordner	<ul style="list-style-type: none"> • Grundlagen zum Programmieren in AWL-Quellen • Exportieren von Quellen • Importieren von Quellen

Symbol	Objekte im Quellen-Ordner	Auswahl wichtiger Funktionen
	Quelle (z. B. AWL-Quelle)	<ul style="list-style-type: none"> • Grundlagen zum Programmieren in AWL-Quellen • Anlegen von AWL-Quellen • Einfügen von Bausteinvorlagen in AWL-Quellen • Einfügen des Quellcodes vorhandener Bausteine in AWL-Quellen • Prüfen der Konsistenz in AWL-Quellen • Übersetzen von AWL-Quellen • Generieren von AWL-Quellen aus Bausteinen • Exportieren von Quellen • Importieren von Quellen
	Netzwerkvorlage	<ul style="list-style-type: none"> • Arbeiten mit Netzwerkvorlagen

5.4.8 S7/M7-Programm ohne Station und CPU

Sie können Programme erstellen, ohne zuvor SIMATIC-Station konfiguriert zu haben. Damit sind Sie zunächst unabhängig von der zu programmierenden Baugruppe und deren Einstellungen.

Anlegen des S7/M7-Programms

1. Öffnen Sie das entsprechende Projekt über den Menübefehl **Datei > Öffnen** bzw. aktivieren Sie das Projektfenster.
2. Markieren Sie das Projekt im Projektfenster der Ansicht offline.
3. Je nachdem, für welches Zielsystem das erstellte Programm bestimmt ist, wählen Sie den entsprechenden Menübefehl:
Einfügen > Programm > S7-Programm, wenn Ihr Programm später auf einer SIMATIC S7 laufen soll.
Einfügen > Programm > M7-Programm, wenn Ihr Programm später auf einer SIMATIC M7 laufen soll.

Der S7/M/-Programm wird eingefügt und im Projektfenster direkt unter dem Projekt angeordnet. Es enthält einen Ordner für die Bausteine und eine leere Symboltabelle. Nun können Sie Bausteine anlegen und programmieren.

Zuordnung zu einer programmierbaren Baugruppe

Baugruppenunabhängig eingefügte Programme können Sie später im Projektfenster einfach einer Baugruppe zuordnen, indem Sie diese Programme auf das Baugruppen-Symbol per Drag & Drop kopieren oder verschieben.

Übernahme in Bibliotheken

Wenn das Programm für das Zielsystem SIMATIC S7 bestimmt ist und gleich einem "Software-Pool" mehrfach verwendet werden soll, kann es auch unter einer Bibliothek eingefügt werden. Zum Testen müssen Programme aber direkt unter einem Projekt liegen, da Sie nur dann eine Verbindung zum Zielsystem herstellen können.

Zugriff auf ein Zielsystem

Wählen Sie die Ansicht online des Projekts. Im Dialogfeld zu den Eigenschaften des Programms können Sie die Adresseinstellungen vornehmen.

Hinweis

Beim Löschen von Stationen oder programmierbaren Baugruppen werden Sie gefragt, ob auch das enthaltene Programm gelöscht werden soll. Wenn Sie diese Frage verneinen, wird das Programm direkt unter dem Projekt als Programm ohne Station eingehängt.

5.5 Benutzungsoberfläche und Bedienung

5.5.1 Bedienphilosophie

Ziel: einfache Bedienung

Die grafische Oberfläche soll Ihnen eine möglichst intuitive Bedienung ermöglichen. Sie finden deshalb Objekte vor, die Sie aus Ihrer täglichen Arbeitswelt kennen, z. B. Stationen, Baugruppen, Programme, Bausteine.

Die Handlungen, die Sie beim Arbeiten mit STEP 7 durchführen, umfassen die Erzeugung, Auswahl und Manipulation solcher Objekte.

Unterschiede zur tool-orientierten Bedienung

Bei der herkömmlichen tool-orientierten Bedienung musste man sich zunächst überlegen, welches Tool zum Lösen einer bestimmten Aufgabe erforderlich ist und dann das Tool aufrufen.

Das prinzipielle Vorgehen bei der objektorientierten Bedienphilosophie ist, sich zu überlegen, welches Objekt bearbeitet werden muss und dann das Objekt zu öffnen und zu bearbeiten.

Bei der objektorientierten Bedienphilosophie ist die Kenntnis einer speziellen Befehlssyntax nicht mehr erforderlich. Objekte werden an der Oberfläche durch grafische Symbole repräsentiert, die sich durch Menübefehle oder Mausklicks öffnen lassen.

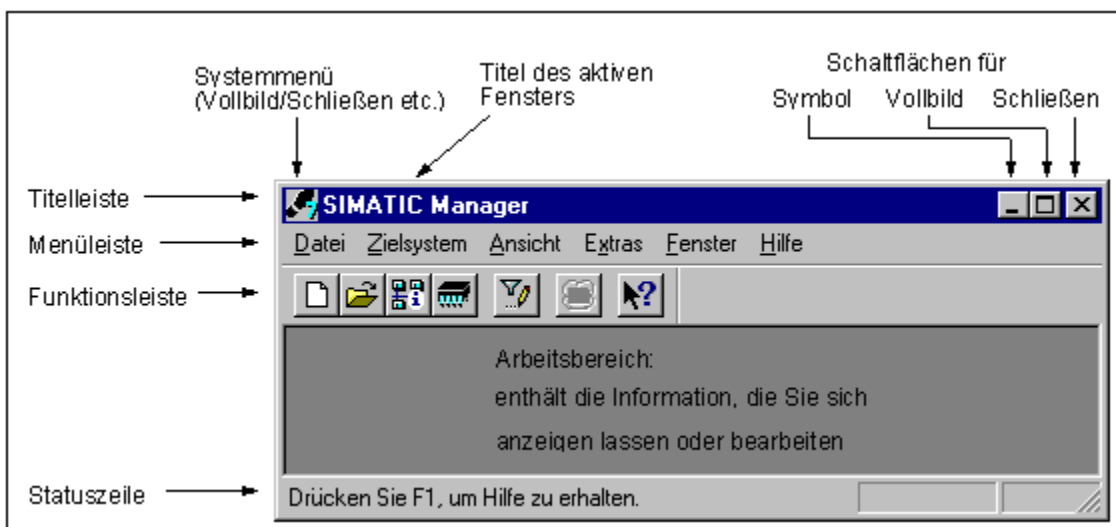
Beim Öffnen eines Objekts wird automatisch die geeignete Softwarekomponente aufgerufen, um den Objektkinhalt anzuzeigen oder zu bearbeiten.

Weiterlesen ...

Nachfolgend sind die grundlegende Aktionen beim Bearbeiten von Objekten beschrieben. Machen Sie sich bitte damit vertraut, da auf diese immer wiederkehrenden Grundbedienungen später aufgebaut wird.

5.5.2 Aufbau des Fensters

Die Standardkomponenten eines Fensters sind im nachfolgenden Bild dargestellt:



Titelleiste und Menüleiste

Titelleiste und die Menüleiste befinden sich immer am oberen Rand des Fensters. Die Titelleiste enthält den Fenstertitel und Symbole für die Fenstersteuerung. Die Menüleiste enthält alle Menüs, die im Fenster zur Verfügung stehen.

Funktionsleiste

Die Funktionsleiste enthält Symbole, über die Sie schnell durch Mausklick auf häufig verwendete und aktuell verfügbare Menübefehle zugreifen können. Eine Kurzinformation über die Funktion eines Symbols wird aufgeblendet, wenn Sie den Cursor kurze Zeit auf dem Symbol positioniert lassen, eine zusätzliche Information wird in der Statuszeile angezeigt.

Wenn in der aktuellen Konfiguration kein Zugriff auf ein Symbol möglich ist, so wird dieses grau dargestellt.

Statuszeile

In der Statuszeile werden kontextabhängige Informationen angezeigt.

5.5.3 Elemente in Dialogfeldern

Eingabe in Dialogfelder

In Dialogfeldern können Sie Informationen eingeben, die für das Ausführen einer bestimmten Aufgabe benötigt werden. Die häufigsten Komponenten von Dialogfeldern sind anhand eines Beispiels im nachfolgenden Bild erklärt:



Listenfelder / Kombinationsfelder

Manchmal sind Textfelder mit einem nach unten gerichteten Pfeil versehen. Dieser Pfeil zeigt an, dass für dieses Feld weitere Auswahlmöglichkeiten zur Verfügung stehen. Klicken Sie auf den Pfeil, um ein Listen- oder Kombinationsfeld aufzuschlagen. Wenn Sie darin einen Eintrag anklicken, wird er automatisch in das Textfeld übernommen.

Registerdialoge

Der Inhalt mancher Dialogfelder ist wegen der besseren Übersicht in Registern organisiert (siehe nachfolgendes Bild).



Die Bezeichnungen der einzelnen Register befinden sich am oberen Rand des Dialogfelds. Um ein bestimmtes Register "nach vorne" zu holen, klicken Sie einfach auf die Registerbezeichnung.

5.5.4 Erzeugen und Manipulieren von Objekten

Einige grundlegende Handlungen an Objekten sind für alle Objekte gleich. Diese prinzipiellen Handlungsfolgen finden Sie hier zusammenfassend dargestellt. Die Kenntnis wird in den nachfolgenden Kapiteln des Handbuchs bei der Beschreibung von Vorgehensweisen als bekannt vorausgesetzt.

Die übliche Handlungsfolge beim Umgang mit Objekten ist:

- Objekt erzeugen,
- Objekt auswählen,
- Aktionen mit dem Objekt durchführen (z. B. kopieren, löschen).

Ablagepfad für neue Projekte/Bibliotheken einstellen

Neue Anwenderprojekte, Bibliotheken und Multiprojekte werden standardmäßig im Verzeichnis "\\Siemens\Step7\S7proj" abgelegt. Wenn Sie ein anderes Verzeichnis als Ablageort wünschen, so sollten Sie vor dem erstmaligen Anlegen neuer Projekte, Bibliotheken oder Multiprojekten den Ablagepfad für diese Objekte einstellen. Wählen Sie dazu den Menübefehl **Extras > Einstellungen**. Im Register "Allgemein" des angezeigten Dialogfelds können Sie dann einen Pfadnamen angeben, unter dem neu erzeugte Projekte oder Bibliotheken abgelegt werden.

Erzeugen von Objekten

Der STEP 7-Assistent "Neues Projekt" unterstützt Sie beim Anlegen eines neuen Projektes und Einfügen von Objekten. Zum Aufruf wählen Sie den Menübefehl **Datei > Assistent 'Neues Projekt'**. In den angezeigten Dialogfeldern können Sie den Aufbau ihres Projektes festlegen und anschließend das Projekt vom Assistenten erzeugen lassen.

Wenn Sie den Assistenten nicht benutzen möchten, können Sie mit dem Menübefehl **Datei > Neu Projekte und Bibliotheken erzeugen**. Diese Objekte bilden die Spitze einer Objekthierarchie. Alle weiteren Objekte der Hierarchie - soweit sie nicht automatisch angelegt werden - können Sie über Befehle im Menü "Einfügen" erzeugen. Eine Ausnahme sind die Baugruppen einer SIMATIC-Station, die nur im Rahmen der Hardware-Konfigurierung und vom Assistenten 'Neues Projekt' angelegt werden.

Öffnen von Objekten

Es gibt mehrere Möglichkeiten, um ein Objekt in der Detailansicht zu öffnen:

- Doppelklick auf das Objektsymbol
- Objekt selektieren und Menübefehl **Bearbeiten > Objekt öffnen**. Dies funktioniert nur bei Objekten, die keine Ordner sind.

Nach dem Öffnen eines Objekts können Sie dessen Inhalt erstellen oder ändern.

Wenn Sie ein Objekt des zuletzt genannten Typs öffnen, wird dessen Inhalt durch eine geeignete Softwarekomponente in einem neuen Fenster zur Bearbeitung dargestellt. Objekte, deren Inhalt bereits anderweitig benutzt wird, können Sie nicht ändern.

Hinweis

Ausnahme: Stationen erscheinen als Ordner für programmierbare Baugruppen (bei Doppelklick) und für die Stationskonfiguration. Wenn Sie auf das Objekt "Hardware" doppelklicken, wird die Applikation für das Konfigurieren der Hardware gestartet. Die gleiche Wirkung hat das Markieren der Station und der anschließende Menübefehl **Bearbeiten > Objekt öffnen**.

Aufbauen einer Objekt-Hierarchie

Lassen Sie sich die Objekthierarchie vom Assistenten 'Neues Projekt' aufbauen. Nach dem Öffnen eines Ordners werden die bereits darin enthaltenen Objekte am Bildschirm dargestellt. Über das Menü "Einfügen" können Sie nun weitere Unterobjekte anlegen, beispielsweise weitere Stationen in einem Projekt. In diesem Menü sind nur die Befehle zum Einfügen solcher Objekte aufrufbar, die im aktuellen Ordner zulässig sind.

Einstellen von Objekteigenschaften

Objekteigenschaften sind Daten des Objekts, die dessen Verhalten festlegen. Der Dialog zur Einstellung von Objekteigenschaften erscheint automatisch, wenn Sie ein Objekt neu erzeugen und dabei Eigenschaften festgelegt werden müssen. Jedoch lassen sich die Eigenschaften auch nachträglich ändern:

Mit dem Menübefehl **Bearbeiten > Objekteigenschaften** wird ein Dialogfeld aufgerufen, in dem sich die Eigenschaften des ausgewählten Objekts ablesen oder einstellen lassen.

Mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften** können Sie Dialogfelder aufrufen und Daten eingeben, die für das Bedienen und Beobachten sowie für die Meldungsprojektierung erforderlich sind.

Um beispielsweise die speziellen Objekteigenschaften eines Bausteins für Bedienen und Beobachten aufrufen zu können, muss der Baustein als BuB-relevant gekennzeichnet sein, d.h. im Register "Attribute" der bas00242 muss das Systemattribut "S7_m_c" mit dem Wert "true" eingetragen sein.

Hinweise

- Eigenschaften des Ordners "Systemdaten" und des Objekts "Hardware" lassen sich nicht anzeigen und verändern.
 - In den Dialogfeldern zu den Eigenschaften von Objekten eines schreibgeschützten Projektes ist kein Schreiben möglich. In diesem Fall sind die Eingabefelder kontrastvermindert dargestellt.
 - Wenn Sie sich die Eigenschaften von programmierbaren Baugruppen anzeigen lassen, dann können Sie die angezeigten Parameter aus Konsistenzgründen nicht bearbeiten. Zum Bearbeiten der Parameter müssen Sie die Applikation "Hardware konfigurieren" aufrufen.
 - Wenn Sie Einstellungen an Objekten auf dem Erstellsystem ändern (z. B. Parametrierungsdaten einer Baugruppe), so sind diese zunächst noch nicht auf dem Zielsystem wirksam. Denn dazu müssen sich die Systemdatenbausteine, in denen diese Einstellungen abgespeichert werden, auf dem Zielsystem befinden.
 - Wenn Sie ein komplettes Anwenderprogramm laden, werden automatisch die Systemdatenbausteine mit übertragen. Wenn Sie nach dem Laden eines Programms Änderungen an den Einstellungen vornehmen, so können Sie das Objekt "Systemdaten" nachladen, um die Einstellungen auf das Zielsystem zu bringen.
 - Es wird dringend empfohlen die Ordner nur mit STEP 7 zu bearbeiten, da sie physikalisch anders strukturiert sein können, als Sie es im SIMATIC Manager sehen.
-

Ausschneiden, Einfügen oder Kopieren

Die meisten Objekte können Sie wie unter Windows üblich ausschneiden, einfügen oder kopieren. Die zugehörigen Menübefehle befinden sich im Menü "Bearbeiten".

Sie können auch durch "Ziehen und Loslassen" (Drag & Drop) Objekte kopieren. Wenn Sie dabei auf ein unzulässiges Ziel zeigen, wird der Cursor als Verbotssymbol dargestellt.

Wenn Sie ein Objekt kopieren, wird damit die gesamte darunter liegende Hierarchie mit kopiert. Dadurch ergeben sich weitreichende Möglichkeiten zur Wiederverwendung von einmal entwickelten Komponenten einer Automatisierungslösung.

Hinweis

Die Verbindungstabelle im Ordner "Verbindungen" können Sie nicht kopieren. Beachten Sie, dass beim Kopieren von Listen bedienerrelevanter Texte nur die Sprachen übernommen werden, die im Zielobjekt installiert sind.

Eine Schritt-für-Schritt-Anleitung zum Kopieren finden Sie in Kopieren von Objekten.

Umbenennen von Objekten

Der SIMATIC Manager vergibt an einige neu eingefügte Objekte standardisierte Namen. Diese Namen ergeben sich in der Regel aus dem Typ des Objekts und (wenn mehrere Objekte dieses Typs im gleichen Ordner angelegt werden können) einer laufenden Nummer.

So heißt z. B. das erste S7-Programm "S7-Programm(1)", das zweite "S7-Programm(2)". Die Symboltabelle hingegen heißt einfach "Symbole", da sie in jedem übergeordneten Ordner nur einmal existiert.

Die Namen der meisten Objekte können Sie ändern und Ihnen dadurch eine selbsterklärende Bezeichnung geben.

Bei Projekten dürfen die im Pfad enthaltenen Verzeichnisnamen nicht länger als 8 Zeichen sein. Ansonsten kann es zu Problemen beim Archivieren und beim Einsatz von "C für M7" (Borland-Compiler) kommen.

Änderbare Objektnamen können Sie direkt editieren oder über die Objekteigenschaften ändern.

Direktes Editieren:

- Klicken Sie im Projektfenster zweimal langsam auf den Namen eines markierten Objekts. Es erscheint ein Rahmen um den Text. Sie können nun die Bezeichnung über die Tastatur editieren.

Ändern über Objekteigenschaften:

- Wählen Sie im Projektfenster das gewünschte Objekt aus, und führen Sie den Menübefehl **Bearbeiten > Umbenennen** aus. Es erscheint ein Rahmen um den Text. Sie können nun die Bezeichnung über die Tastatur editieren.

Wenn Sie den Namen eines Objekts nicht ändern dürfen, ist das Eingabefeld im Dialogfeld grau unterlegt, der aktuelle Name wird angezeigt und Texteingaben sind nicht möglich.

Hinweis

Wenn Sie während des Editierens den Mauszeiger aus dem Namensfeld herausbewegen und eine andere Aktion durchführen (z. B. einen Menübefehl wählen), wird der Editiervorgang beendet. Der geänderte Namen wird, wenn er zulässig ist, übernommen.

Eine Schritt-für-Schritt-Anleitung zum Umbenennen finden Sie in Umbenennen von Objekten.

Verschieben von Objekten

Mit dem SIMATIC Manager können Sie Objekte von einem Ordner zu einem anderen verschieben, auch wenn dieser in einem anderen Projekt liegt. Beim Verschieben eines Ordners wird dessen gesamter Inhalt mitverschoben.

Hinweis

Die folgenden Objekte können Sie nicht verschieben:

- Verbindungen
 - Systemdatenbausteine (SDB) in der Online-Ansicht
 - Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB) in der Online-Ansicht
-

Eine Schritt-für-Schritt-Anleitung zum Verschieben finden Sie in Verschieben von Objekten.

Sortieren von Objekten

Sie können in der Detail-Ansicht (Menübefehl **Ansicht > Details**) die Objekte nach ihren Attributen sortieren lassen. Klicken Sie dazu auf die entsprechende Überschrift des gewünschten Attributs. Bei erneutem Klicken kehrt sich die Sortierreihenfolge um. Bausteine werden entsprechend ihrer numerischen Ergänzung sortiert, z.B. FB 1, FB 2, FB 11, FB 12, FB 21, FC 1.

Voreingestellte Sortier-Reihenfolge (Default-Sortierung):

Wenn Sie ein Projekt neu öffnen, werden die Objekte in der Ansicht "Details" gemäß einer voreingestellten Sortier-Reihenfolge angezeigt. Beispiele:

- Bausteine werden in der Reihenfolge "Systemdaten, OB, FB, FC, DB, UDT, VAT, SFB, SFC" dargestellt.
- Bei Projekt werden erst alle Stationen und dann die S7-Programme angezeigt.

Die Voreinstellung ist also keine alphanumerisch aufsteigende oder absteigende Sortierung für ein Sortierkriterium der Detail-Ansicht.

Wiederherstellen der Default-Sortierung:

Nach einem Umsortieren z. B. durch Klicken auf die Spaltenüberschrift "Objektname", können Sie die voreingestellte Reihenfolge wieder herstellen, wenn Sie wie folgt vorgehen:

- in der Detailansicht auf die Spaltenüberschrift "Typ" klicken.
- das Projekt schließen und erneut öffnen.

Löschen von Objekten

Sie können sowohl Ordner als auch Objekte löschen. Wenn Sie einen Ordner löschen, werden auch alle darin enthaltenen Objekte mitgelöscht.

Der Löschvorgang kann nicht rückgängig gemacht werden. Wenn Sie nicht sicher sind, ob Sie ein Objekt wirklich nicht mehr brauchen, ist es besser, das gesamte Projekt vorher zu archivieren.

Hinweis

Die folgenden Objekte können Sie nicht löschen:

- Verbindungen
 - Systemdatenbausteine (SDB) in der Online-Ansicht
 - Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB) in der Online-Ansicht
-

Eine Schritt-für-Schritt-Anleitung zum Löschen finden Sie in Löschen von Objekten.

5.5.5 Auswahl von Objekten im Dialog

Die Auswahl von Objekten in einem Dialogfeld ist eine bei unterschiedlichen Arbeitsschritten auszuführende Aktion.

Aufruf des Dialogfeldes

Das Dialogfeld wird z. B. in der Hardware-Konfigurierung durch die Menübefehle wie **Station > Neu.../Öffnen...** aufgerufen (Eine Ausnahme bildet das Einstiegsfenster "SIMATIC Manager").

Aufbau des Dialogfeldes

Im Dialogfeld haben Sie die im nachfolgenden Bild gezeigten Auswahlmöglichkeiten.

Einstiegspunkt: Hier wählen Sie den Typ des Objekts aus, in dem Sie die Suche beginnen möchten (z. B. "Projekt", "Bibliothek", aber auch Einträge, die den Zugriff auf Laufwerke oder angeschlossene Automatisierungssysteme ermöglichen).

Ansicht: Hier können Sie zwischen Standardansicht und technologischer Ansicht wählen.

Online/Offline: Hier können Sie - nur bei Einstiegspunkt "Projekte" - zwischen Offline-Sicht (Auswahl aus Projektdaten auf dem P.G.P.C.) und Online-Sicht (Auswahl aus Projektdaten auf dem angeschlossenen Automatisierungssystem) umschalten.

Durchsuchen: Klicken Sie auf diese Schaltfläche, um nach Objekten zu suchen, die nicht in der Liste enthalten sind.

Name: Hier werden die bekannten Objekte des unter Einstieg angegebenen Typs in einem Listefeld angezeigt. Sie können einen Namen aus der Liste auswählen oder direkt über die Tastatur eingeben.

Objektname: Wenn Sie ein Objekt auswählen, wird hier der Objektname eingetragen. Sie können den gewünschten Namen auch direkt eingeben.

Objekttyp: Hier können Sie ein Filterkriterium angeben, nach dem die Objektliste gefiltert wird. Dadurch lässt sich die Menge der angezeigten Objekte einschränken.

Projektansicht: Hier wird der Hierarchybaum der Objekte angezeigt, die Unterobjekte enthalten können.

Detailansicht: Hier wird der Inhalt des links markierten Objekts dargestellt.

5.5.6 Sitzungsgedächtnis

Der SIMATIC Manager kann sich den Fensterinhalt, also die geöffneten Projekte und Bibliotheken, sowie die Fensteranordnung merken.

- Über Menübefehl **Extras > Einstellungen** legen Sie fest, ob Fensterinhalt und Fensteranordnung bei Sitzungsende gespeichert werden. Beim nächsten Sitzungsbeginn werden dieser Fensterinhalt und diese Fensteranordnung wiederhergestellt. In den geöffneten Projekten wird auf den letzten selektierten Behälter positioniert.
- Über Menübefehl **Fenster > Anordnung speichern** werden der aktuelle Fensterinhalt und die aktuelle Fensteranordnung gespeichert.
- Über Menübefehl **Fenster > Anordnung wiederherstellen** werden der mit Menübefehl **Fenster > Anordnung speichern** gespeicherte Fensterinhalt und die Fensteranordnung wiederhergestellt. In den geöffneten Projekten wird auf den letzten selektierten Behälter positioniert.

Hinweis

Der Fensterinhalt von Online-Projekten, der Inhalt des Fensters "Erreichbare Teilnehmer" und der Inhalt des Fensters "S7-Memory Card" werden nicht gespeichert.

Eventuell eingegebene Passwörter für den Zugriff auf Zielsysteme (S7-300/S7-400) werden über das Sitzungsende hinaus nicht gespeichert.

5.5.7 Ändern der Fensteranordnung

Um alle angezeigten Fenster hintereinander gestaffelt anzuordnen, haben Sie folgende Möglichkeiten:

- Wählen Sie den Menübefehl **Fenster > Anordnen > Überlappend**.
- Betätigen Sie SHIFT + F5.

Um alle angezeigten Fenster gleichmäßig untereinander anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Untereinander**.

Um alle angezeigten Fenster gleichmäßig nebeneinander anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Nebeneinander**.

5.5.8 Speichern und Wiederherstellen der Fensteranordnung

Die STEP 7-Applikationen bieten die Möglichkeit, die aktuelle Anordnung von Fenstern zu speichern und zu einem anderen Zeitpunkt wiederherzustellen. Die Einstellung können Sie über den Menübefehl **Extras > Einstellungen** im Register "Allgemein" vornehmen.

Was wird gespeichert?

Wenn Sie die Fensteranordnung speichern, werden folgende Informationen aufgezeichnet:

- Position des Hauptfensters
- Geöffnete Projekte und Bibliotheken und die zugehörigen Fensterpositionen
- Reihenfolge von eventuell übereinanderliegenden Fenstern

Hinweis

Der Fensterinhalt von Online-Projekten, der Inhalt des Fensters "Erreichbare Teilnehmer" und der Inhalt des Fensters "S7-Memory Card" werden nicht gespeichert.

Fensteranordnung speichern

Um die aktuelle Anordnung der Fenster zu speichern, wählen Sie den Menübefehl **Fenster > Anordnung speichern**.

Fensteranordnung wiederherstellen

Um eine gespeicherte Fensteranordnung wieder herzustellen, wählen Sie den Menübefehl **Fenster > Anordnung wiederherstellen**.

Hinweis

Beim Wiederherstellen eines Fensters wird nur der Teil der Hierarchie detailliert dargestellt, in dem das Objekt liegt, das beim Speichern der Fensteranordnung selektiert war.

5.6 Bedienung über Tastatureingabe

Internationale Tastenbezeichnung	Deutsche Tastenbezeichnung
HOME-Taste	POS1-Taste
END-Taste	ENDE-Taste
PAGE-UP-Taste	BILD-AUF-Taste
PAGE-DOWN-Taste	BILD-AB-Taste
CTRL-Taste	STRG-Taste
ENTER-Taste	Eingabetaste
DEL-Taste	ENTF-Taste
INSERT-Taste	EINFG-Taste

5.6.1 Tastenkombinationen für Menübefehle

Jeder Menübefehl kann über das Eingeben der entsprechenden ALT-Tastenkombination ausgelöst werden.

Drücken Sie nacheinander folgende Tasten:

- ALT-Taste
- Buchstabe, der im gewünschten Menü unterstrichen ist (z. B. ALT, D für Menü "Datei" - wenn Menü "Datei" in der Menüleiste eingetragen ist). Das Menü wird aufgeschlagen.
- Buchstabe, der im gewünschten Menübefehl unterstrichen ist (z. B. N für Menübefehl "Neu"). Handelt es sich um einen Menübefehl mit weiteren Untermenüs, so werden diese daraufhin aufgeschlagen. Verfahren Sie in gleicher Weise, bis Sie den gesamten Menübefehl durch Eingabe der jeweiligen Buchstaben ausgewählt haben.

Mit Eingabe des letzten Buchstabens der Tastenkombination wird der Menübefehl ausgelöst.

Beispiele:

Menübefehle	Tasten
Datei > Archivieren	ALT, D, A
Datei > Öffnen	ALT, D, O

Shortcuts für Menübefehle

Funktion		Shortcut
Neu	(Menü "Datei")	CTRL+N
Öffnen	(Menü "Datei")	CTRL+O
Speichern unter	(Menü "Datei")	CTRL+S
Drucken > Objektliste	(Menü "Datei")	CTRL+P
Drucken > Objekttinhalt	(Menü "Datei")	CTRL+ALT+P
Beenden	(Menü "Datei")	ALT+F4
Ausschneiden	(Menü "Bearbeiten")	CTRL+X
Kopieren	(Menü "Bearbeiten")	CTRL+C
Einfügen	(Menü "Bearbeiten")	CTRL+V
Löschen	(Menü "Bearbeiten")	DEL
Alles markieren	(Menü "Bearbeiten")	CTRL+A
Umbenennen	(Menü "Bearbeiten")	F2
Objekteigenschaften	(Menü "Bearbeiten")	ALT+RETURN
Objekt öffnen	(Menü "Bearbeiten")	CTRL+ALT+O
Übersetzen	(Menü "Bearbeiten")	CTRL+B
Laden	(Menü "Zielsystem")	CTRL+L
Diagnose/Einstellung > Baugruppenzustand	(Menü "Zielsystem")	CTRL+D
Diagnose/Einstellung Betriebszustand	(Menü "Zielsystem")	CTRL+I
Aktualisieren	(Menü "Ansicht")	F5
aktualisiert die Statusanzeige der sichtbaren CPUs in der Online-Sicht		CTRL+F5
Einstellungen	(Menü "Extras")	CTRL+ALT+E
Referenzdaten > Anzeigen	(Menü "Extras")	CTRL+ALT+R
Anordnen > Überlappend	(Menü "Fenster")	SHIFT+F5
Anordnen > Untereinander	(Menü "Fenster")	SHIFT+F2
Anordnen > Nebeneinander	(Menü "Fenster")	SHIFT+F3
Hilfe zum Kontext	(Menü "Hilfe")	F1 (Besteht ein aktueller Kontext, z. B. markierter Menübefehl, wird das entsprechende Hilfethema aufgerufen, andernfalls das Inhaltsverzeichnis der Hilfe.)

5.6.2 Tastenkombinationen für das Bewegen des Cursors

Bewegen des Cursors in der Menüleiste / im Kontextmenü

Funktion	Tasten
Zur Menüleiste	F10
Zum Kontextmenü	SHIFT+F10
Zu dem Menü, das den unterstrichenen Buchstaben X enthält	ALT+X
Zugeordneter Menübefehl	Unterstrichener Buchstabe im Menübefehl
Einen Menübefehl nach links	PFEIL LINKS
Einen Menübefehl nach rechts	PFEIL RECHTS
Einen Menübefehl nach oben	PFEIL OBEN
Einen Menübefehl nach unten	PFEIL UNTEN
Aktiviert den markierten Menübefehl	ENTER
Verlässt das Menü bzw. zurück zum Text	ESC

Bewegen des Cursors beim Bearbeiten von Texten

Funktion	Tasten
Eine Zeile nach oben oder ein Zeichen nach links in einem Text, der nur aus einer Zeile besteht	PFEIL OBEN
Eine Zeile nach unten oder ein Zeichen nach rechts in einem Text, der nur aus einer Zeile besteht	PFEIL UNTEN
Ein Zeichen nach rechts	PFEIL RECHTS
Ein Zeichen nach links	PFEIL LINKS
Ein Wort nach rechts	CTRL+PFEIL RECHTS
Ein Wort nach links	CTRL+PFEIL LINKS
Zum Anfang der Zeile	HOME
Zum Ende der Zeile	ENDE
Einen Bildschirm nach oben	PAGE-UP
Einen Bildschirm nach unten	PAGE-DOWN
Zum Textanfang	CTRL+HOME
Zum Textende	CTRL+END

Bewegen des Cursors beim Bearbeiten von Tabellen

Funktion	Tasten
Eine Zelle nach oben	PFEIL OBEN
Eine Zelle nach unten	PFEIL UNTEN
Ein Zeichen oder eine Zelle nach links	PFEIL RECHTS
Ein Zeichen oder eine Zelle nach rechts	PFEIL LINKS
Zum Anfang der Zeile	CTRL+PFEIL RECHTS
Zum Ende der Zeile	CTRL+PFEIL LINKS
Zum Anfang der Zelle	HOME
Zum Ende der Zelle	ENDE
Einen Bildschirm nach oben	PAGE-UP
Einen Bildschirm nach unten	PAGE-DOWN
Zum Anfang der Tabelle	CTRL+HOME
Zum Ende der Tabelle	CTRL+END
Nur in der Symboltabelle: Zur Spalte Symbol	SHIFT+HOME
Nur in der Symboltabelle: Zur Spalte Kommentar	SHIFT+END

Bewegen des Cursors in Dialogfeldern

Funktion	Tasten
Zum nächsten Eingabefeld (von links nach rechts und von oben nach unten)	TAB
Ein Eingabefeld zurück	SHIFT+TAB
Zu dem Eingabefeld, das den unterstrichenen Buchstaben X enthält	ALT+X
Markiert in Auswahlliste	RICHTUNGSTASTEN
Öffnet eine Auswahlliste	ALT+PFEIL UNTEN
Markiert ein Objekt bzw. macht Markierung rückgängig	LEERTASTE
Bestätigt Einträge und schließt das Dialogfeld (Schaltfläche "OK")	ENTER
Schließt das Dialogfeld, ohne die Auswahl zu speichern (Schaltfläche "Abbrechen")	ESC

5.6.3 Tastenkombinationen für das Markieren von Texten

Funktion	Tasten
Ein Zeichen nach rechts	SHIFT+PFEIL RECHTS
Ein Zeichen nach links	SHIFT+PFEIL LINKS
Bis zum Anfang der Kommentarzeile	SHIFT+HOME
Bis zum Ende der Kommentarzeile	SHIFT+END
Eine Zeile in einer Tabelle	SHIFT+LEERTASTE
Eine Zeile nach oben	SHIFT+PFEIL OBEN
Eine Zeile nach unten	SHIFT+PFEIL UNTEN
Einen Bildschirm nach oben	SHIFT+PAGE-UP
Einen Bildschirm nach unten	SHIFT+PAGE-DOWN
Bis zum Dateianfang	CTRL+SHIFT+HOME
Bis zum Dateiende	CTRL+SHIFT+END

5.6.4 Tastenkombinationen für den Zugriff auf Online-Hilfe

Funktion	Tasten
Öffnet die Hilfe	F1 (Besteht ein aktueller Kontext, z. B. markierter Menübefehl, wird das entsprechende Hilfethema aufgerufen, andernfalls das Inhaltsverzeichnis der Hilfe.)
Aktivieren des Fragezeichen-Symbols für kontextsensitive Hilfe	SHIFT+F1
Schließt das Hilfefenster, zurück zum Applikationsfenster	ALT+F4

5.6.5 Tastenkombinationen für das Umschalten zwischen Fenstertypen

Funktion	Tasten
Umschalten zwischen Teilfenstern	F6
Zurückschalten auf das vorherige Teilfenster, wenn kein andockbares Fenster vorhanden ist	Shift+F6
Umschalten zwischen Dokumentfenster und andockbarem Fenster des Dokuments (z. B. Variablendeklarationsfenster) Wenn es keine andockbaren Fenster gibt, wird auf das vorherige Teilfenster zurückgeschaltet.	Shift+F6
Umschalten zwischen Dokumentfenstern	Ctrl+F6
Zurückschalten auf das vorherige Dokumentfenster	Shift+Ctrl+F6
Umschalten zwischen Nichtdokumentfenstern (Applikationsrahmen und andockbare Fenster des Applikationsrahmens; bei Rückkehr zum Rahmen wird zu dem Dokumentfenster umgeschaltet, das zuletzt aktiv war)	Alt+F6
Zurückschalten auf das vorherige Nichtdokumentfenster	Shift+Alt+F6
Schließen des aktiven Fensters	Ctrl+F4

6 Einrichten und Bearbeiten des Projekts

6.1 Projektstruktur

Projekte dienen dazu, die bei der Erstellung einer Automatisierungslösung anfallenden Daten und Programme geordnet abzulegen. Die in einem Projekt zusammengefassten Daten umfassen insbesondere:

- Konfigurationsdaten über den Hardware-Aufbau und Parametrierungsdaten für Baugruppen,
- Projektierungsdaten für die Kommunikation über Netze und
- Programme für programmierbare Baugruppen.

Die Hauptaufgabe bei der Erstellung eines Projekts sind die Bereitstellung dieser Daten und die Programmierung.

Daten werden in einem Projekt in Form von Objekten abgelegt. Die Objekte sind innerhalb eines Projekts in einer Baumstruktur (Projekthierarchie) angeordnet. Die Darstellung der Hierarchie im Projektfenster ähnelt der des Windows-Explorers. Lediglich die Objektsymbole haben ein anderes Aussehen.

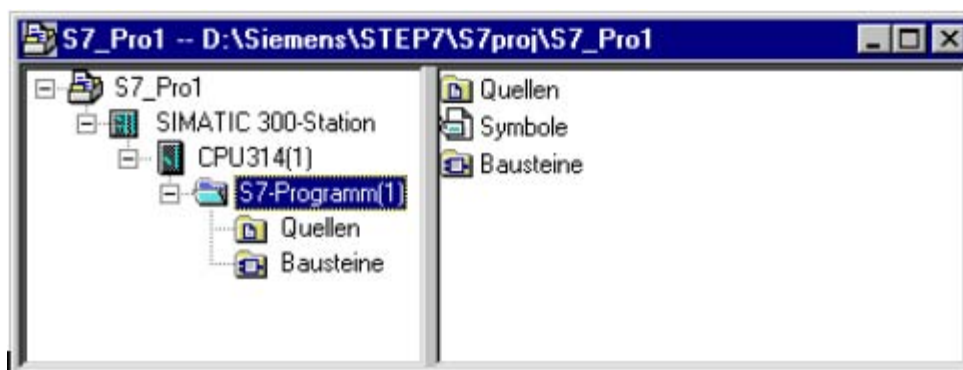
Die Spitze der Projekthierarchie ist wie folgt aufgebaut:

1. Stufe : Projekt
2. Stufe : Subnetze, Stationen oder S7-/M7-Programme
3. Stufe: abhängig vom zugehörigen Objekt der 2. Stufe.

Projektfenster

Das Projektfenster ist zweigeteilt. In der linken Hälfte wird die Baumstruktur des Projekts dargestellt. In der rechten Hälfte wird der Inhalt des links markierten Objekts in der gewählten Ansicht angezeigt (große Symbole, kleine Symbole, Liste oder Details).

Klicken Sie im linken Teil des Fensters auf die Kästchen mit einem Pluszeichen, um die vollständige Baumstruktur des Projekts sichtbar zu machen. Sie erhalten dann eine Darstellung ähnlich der im nachfolgenden Bild.



An der Spitze der Objekt-Hierarchie steht hier das Objekt "S7_Pro1" als Symbol für das gesamte Projekt. Es kann zum Anzeigen der Projekteigenschaften benutzt werden und dient als Ordner für Netze (zum Projektieren von Netzen), Stationen (zum Konfigurieren der Hardware) sowie für S7- oder M7-Programme (zur Software-Erstellung). Die enthaltenen Objekte werden rechts im Projektfenster angezeigt, wenn Sie das Projektsymbol markieren. Die Objekte an der Spitze der Hierarchie (dazu gehören neben Projekten auch Bibliotheken) bilden die Einstiegspunkte in Dialogfeldern zur Auswahl von Objekten.

Projektsicht

Sie können sich die Projektstruktur für den Datenbestand auf dem Erstellsystem in der Ansicht "offline" und für den zugehörigen Datenbestand auf dem Zielsystem in der Ansicht "online" in Projektfenstern anzeigen lassen.

Als weitere einstellbare Sicht steht, wenn das entsprechende Optionspaket geladen ist, die des Anlagenmanagers zur Verfügung.

Hinweis

Das Konfigurieren von Hardware und das Projektieren von Netzen ist nur in der Ansicht "offline" durchführbar.

6.2 Wissenswertes zum Zugriffsschutz

Ab STEP 7 V5.4 gibt es die Möglichkeit, durch die Vergabe eines Projektpasswortes einen Zugriffsschutz für Projekte und Bibliotheken einzurichten. Diese Funktionalität setzt jedoch die Installation von SIMATIC Logon voraus.

Außerdem können Sie dazu auch ein Änderungsprotokoll aktivieren, deaktivieren und anzeigen.

Falls SIMATIC Logon auf Ihrem Rechner installiert ist, erhalten Sie dynamisch folgende Menübefehle im SIMATIC Manager, mit denen Sie den Zugriffsschutz zu einem Projekt oder einer Bibliothek verwalten können:

- Zugriffsschutz, Aktivieren
- Zugriffsschutz, Deaktivieren
- Zugriffsschutz, Benutzer verwalten
- Zugriffsschutz, Im Multiprojekt abgleichen
- Zugriffsschutz und Änderungsprotokoll entfernen

Sie aktivieren den Zugriffsschutz im SIMATIC Manager über den Menübefehl **Extras > Zugriffsschutz, Aktivieren**. Wenn Sie den Zugriffsschutz über den o.g. Menübefehl aktivieren, öffnet sich ein Dialogfeld, in dem Sie ein Projektpasswort vergeben. Das entsprechende Projekt oder die Bibliothek sind dann nur noch nach Eingabe des Projektpasswortes zu bearbeiten.

Mit dem Menübefehl **Zugriffsschutz und Änderungsprotokoll entfernen** entfernen Sie sowohl den Zugriffsschutz als auch das Änderungsprotokoll eines passwortgeschützten Projektes oder einer passwortgeschützten Bibliothek. Nach dem Entfernen des Zugriffsschutzes können Sie Projekte auch wieder mit einer STEP 7-Version kleiner als V5.4 bearbeiten.

Hinweise

- Um den Zugriffsschutz zu aktivieren oder zu deaktivieren, müssen Sie in SIMATIC Logon als Projekt-Administrator authentifiziert sein.
 - Beim erstmaligen Aktivieren des Zugriffsschutzes wird das Projektformat geändert. Sie erhalten einen Hinweis, dass das geänderte Projekt nicht mehr mit älteren STEP 7-Versionen bearbeitet werden kann.
 - Die Funktion **Extras->Zugriffsschutz->Zugriffsschutz und Änderungsprotokoll entfernen** erlaubt es, dass das Projekt bzw. die Bibliothek wieder mit einer STEP 7-Version kleiner als V5.4 bearbeitet werden kann. Sie verlieren jedoch die Information über die Benutzer, die auf dieses Projekt bzw. Bibliothek Zugriff haben und sämtliche Änderungsprotokolle.
 - Der aktuell angemeldete Benutzer wird in der Statuszeile des SIMATIC Managers angezeigt.
 - Der aktuell angemeldete Projekt-Bearbeiter, der den Zugriffsschutz aktiviert, wird als Projekt-Administrator eingetragen und wird aufgefordert, das Projektpasswort zu vergeben.
 - Um ein zugriffsgeschütztes Projekt öffnen zu können, müssen Sie in SIMATIC Logon als Projekt-Administrator oder Projekt-Bearbeiter authentifiziert sein und der Zugriffsschutz muss aktiviert sein oder Sie kennen das Projektpasswort.
-

6.3 Wissenswertes zum Änderungsprotokoll

Ab STEP 7 V5.4 gibt es die Möglichkeit, nach dem Einrichten eines Zugriffsschutzes für Projekte und Bibliotheken ein Änderungsprotokoll zu führen, das Online-Aktionen mitprotokolliert.

Beispiele sind:

- Aktivierung / Deaktivierung / Konfiguration von Zugriffsschutz und Änderungsprotokoll
- Öffnen / Schließen von Projekten und Bibliotheken
- Laden ins Zielsystem (Systemdaten)
- Ausgewählte Operationen zum Laden und Kopieren von Bausteinen
- Aktivitäten zur Änderung des Betriebszustands
- Urlöschen

Sie können sich das Änderungsprotokoll anzeigen lassen und dieses mit Begründungen für die Änderung sowie mit Kommentaren versehen. Diese Funktionalität setzt jedoch die Installation von SIMATIC Logon voraus.

Sie aktivieren das Änderungsprotokoll im SIMATIC Manager über den Menübefehl **Extras > Änderungsprotokoll, Aktivieren**. Nachdem Sie das Änderungsprotokoll aktiviert haben, können Sie es sich über den entsprechenden Menübefehl anzeigen lassen bzw. wieder deaktivieren.

Abhängig davon, welches Objekt Sie in der Projektstruktur angewählt haben (z.B. Projektordner oder untergeordnete Station), wird das zugehörige Änderungsprotokoll angezeigt.

Hinweise

- Die Funktion **Extras->Zugriffsschutz->Zugriffsschutz und Änderungsprotokoll entfernen** erlaubt es, dass das Projekt bzw. die Bibliothek wieder mit einer STEP 7-Version kleiner als V5.4 bearbeitet werden kann. Sie verlieren jedoch die Information über die Benutzer, die auf dieses Projekt bzw. Bibliothek Zugriff haben und sämtliche Änderungsprotokolle.
 - Um diese Funktion ausführen zu können, müssen Sie in SIMATIC Logon als Projekt-Administrator authentifiziert sein und der Zugriffsschutz zu diesem Projekt muss aktiviert sein.
-

6.4 Fremdsprachige Zeichensätze verwenden

Ab STEP 7 V5.3 SP2 können Sie in Projekten und Bibliotheken fremdsprachige Texte eingeben, die nicht der eingestellten STEP 7-Sprache entsprechen. Dazu muss in der Systemsteuerung des Betriebssystems die entsprechende Windows-Sprache eingestellt werden. So ist es z. B. möglich, auf einem chinesischen Windows STEP 7 in der STEP 7-Sprache Englisch zu betreiben, aber dennoch chinesische Texte einzugeben.

Folgende Spracheinstellungsmöglichkeiten sind dabei zu unterscheiden:

Windows-Spracheinstellung

Diese Spracheinstellung nehmen Sie in der Systemsteuerung von Windows vor. In der ausgewählten Sprache werden Texte des Betriebssystems angezeigt und Sie können Texte in fremdsprachigen Zeichensätzen eingeben.

Projektsprache

Die Projektsprache ist die Sprache, die beim Erstellen eines Projektes in der Systemsteuerung von Windows eingestellt ist. Ein Ändern der Projektsprache auf eine andere Sprache ist nicht möglich. Unter der Einstellung "sprachneutral" kann jedoch ein Projekt auf Rechnern mit anderer Windows-Spracheinstellung geöffnet werden. Vergewissern Sie sich, bevor Sie die Projektsprache auf "sprachneutral" ändern, dass im Projekt bei der Texteingabe nur Zeichen des englischen Zeichensatzes (ASCII-Zeichen 0x2a - 0x7f) verwendet wurden.

In welcher Projektsprache ein Projekt oder eine Bibliothek erstellt wurde, können Sie mit dem Menübefehl **Bearbeiten > Objekteigenschaften** anzeigen lassen. Im gleichen Dialogfeld können Sie auch die Option "Kann unter jeder Windows-Spracheinstellung geöffnet werden (sprachneutral)" auswählen.

Kopieren Sie ein Projekt mit dem Menübefehl **Speichern unter** und die Projektsprache stimmt nicht mit der aktuellen Windows-Spracheinstellung überein, können Sie die Projektsprache im kopierten Projekt auf die aktuelle Windows-Spracheinstellung ändern. Dies ist dann von Nutzen, wenn Sie z. B. von einem Projekt sprachspezifische Varianten erstellen wollen. Das Master-Projekt sollte in diesem Fall nur Zeichen des englischen Zeichensatzes (ASCII-Zeichen 0x2a - 0x7f) enthalten, so dass sichergestellt ist, dass es nicht zu Datenverfälschungen kommt, wenn das Projekt in der jeweiligen Sprache weiter bearbeitet wird.

STEP 7-Sprache

Die STEP 7-Sprache stellen Sie im SIMATIC Manager über den Menübefehl **Extras > Einstellungen** ein. In dieser Sprache werden die Oberflächenelemente, Menübefehle, Dialogfelder und Fehlermeldungen von STEP 7 ausgegeben.

Verwenden Sie eine andere Windows-Sprache als Deutsch, Englisch, Französisch, Italienisch oder Spanisch, können Sie sicherstellen, dass die STEP 7-Oberfläche einwandfrei dargestellt wird, wenn Sie als STEP 7-Sprache Englisch wählen.

Regeln

Wenn Sie beabsichtigen, Ihre Projekte oder Bibliotheken auf Rechnern mit unterschiedlicher Spracheinstellung zu bearbeiten, beachten Sie die folgenden Regeln, um Inkompatibilitäten bzw. Datenverfälschungen beim Verwenden fremdsprachiger Zeichensätze zu verhindern:

- Installieren Sie STEP 7 nur in Verzeichnisse, die Zeichen des englischen Zeichensatzes (ASCII-Zeichen 0x2a - 0x7f) enthalten.
- Verwenden Sie nur Projektnamen und Projektpfade, die Zeichen des englischen Zeichensatzes (ASCII-Zeichen 0x2a - 0x7f) enthalten. Verwenden Sie z. B. deutsche Umlaute bzw. kyrillische oder chinesische Zeichen, kann das Projekt nur auf Rechnern geöffnet werden, auf denen die entsprechende Windows-Spracheinstellung vorgenommen wurde.
- Verwenden Sie in Multiprojekten nur Projekte und Bibliotheken mit der selben Projektsprache oder solche, die als sprachneutral gekennzeichnet sind. Das Multiprojekt selbst ist sprachneutral.
- Erstellen Sie Bibliotheken immer sprachneutral, um sicherzustellen, dass sie auf Rechnern mit unterschiedlichen Windows-Spracheinstellungen verwendet werden können. Verwenden Sie bei der Namensvergabe der Bibliotheksobjekte, bei Kommentaren, Symbolnamen etc. nur ASCII-Zeichen (0x2a - 0x7f), damit die Bibliotheken ohne Probleme verwendet werden können.
- Beachten Sie beim Import/Export von Hardwarekonfigurationen oder Symboltabellen, dass Sie nur sprachkompatible Dateien importieren/exportieren.
- Verwenden Sie in Namen anwenderdefinierter Attribute nur Zeichen des englischen Zeichensatzes (ASCII-Zeichen 0x2a - 0x7f).
- Verwenden Sie in einer AWL-Quelle in Einträgen für die Bausteineigenschaften TITLE, AUTHOR, FAMILY oder NAME Zeichen, die nicht im englischen Zeichensatz (ASCII-Zeichen 0x2a - 0x7f) enthalten sind, setzen Sie diese Einträge in einfache Hochkommata.

Hinweise

- Ändern oder Kopieren Sie Projekte oder Bibliotheken, die auf einem Rechner erstellt wurden, der bezüglich der Windows-Spracheinstellung zwar als sprachneutral gekennzeichnet, aber nicht kompatibel zum aktuellen Rechner ist, kann es zu Datenverfälschungen kommen, wenn im Projekt oder in der Bibliothek Zeichen verwendet wurden, die nicht im englischen Zeichensatz (ASCII-Zeichen 0x2a - 0x7f) enthalten sind. Vergewissern Sie sich deshalb vor dem Bearbeiten "fremder" Projekte oder Bibliotheken, ob die Windows-Spracheinstellung auf Ihrem Rechner mit der Projektsprache übereinstimmt.
- Exportieren Sie Hardwarekonfigurationen oder Symboltabellen, die unter einer anderen Windows-Spracheinstellung importiert werden sollen, stellen Sie sicher, dass nur ASCII-Zeichen (0x2a - 0x7f) verwendet wurden und keine sprachspezifischen Zeichen wie z. B. deutsche Umlaute, japanische oder kyrillische Zeichen.
- Exportierte Hardwarekonfigurationen oder Symboltabellen, die sprachspezifischen Zeichen wie z. B. deutsche Umlaute, japanische oder kyrillische Zeichen enthalten, dürfen nur unter der Windows-Spracheinstellung importiert werden, unter der sie exportiert wurden. Importieren Sie z. B. ältere Symboltabellen, die möglicherweise sprachspezifische Zeichen enthalten, prüfen Sie das Ergebnis sorgfältig: die Symbole sollten eindeutig sein, keine Frage- oder Schmierzeichen enthalten und plausibel sein.
- In Symboltabellen mit Sonderzeichen, die unter der jeweiligen Windows-Spracheinstellung nicht definiert sind, können durch Frage- oder Schmierzeichen veränderte Symbolnamen und Fehler beim Sortieren nach Namen und Kommentaren auftreten.
- Bitte beachten Sie, dass bei symbolischer Adressierung die symbolischen Namen in Anführungszeichen zu schreiben sind ("`<symbolischer Name>`").

Prinzipielle Vorgehensweise

Um in Projekten und Bibliotheken Texte in fremdsprachigen Zeichensätzen eingeben zu können, gehen Sie folgendermaßen vor:

1. Stellen Sie über die Systemsteuerung die Windows-Spracheinstellung auf die gewünschte Sprache ein.
2. Erstellen Sie ein Projekt.
3. Geben Sie die gewünschten fremdsprachigen Texte ein.

Bei Projekten und Bibliotheken, die bereits vor STEP 7 V5.3 SP2 erstellt wurden, ist die Projektsprache "noch nicht festgelegt". Sie können bei diesen Projekten mit dem Menübefehl **Bearbeiten > Objekteigenschaften** die Projektsprache auf die angezeigte, momentan eingestellte Windows-Sprache umstellen. Vergewissern Sie sich vorher, dass keine Zeichen verwendet wurden, die unter der aktuellen Windows-Spracheinstellung nicht definiert sind.

6.5 Einstellen der Windows-Sprache

Um die Windows-Sprache einzustellen, gehen Sie folgendermaßen vor:

Spracheinstellung unter Windows XP und Windows Server 2003 :

1. Stellen Sie über **Systemsteuerung > Regions- und Sprachoptionen > Erweitert > Sprache für Programme**, die Unicode nicht unterstützen die gewünschte Anzeigesprache ein.
2. Stellen Sie über **Systemsteuerung > Regions- und Sprachoptionen > Sprachen > Details** die Eingabesprache (Standard-Eingabebereichsschema) ein.
3. Stellen Sie über **Systemsteuerung > Regions- und Sprachoptionen > Regionale Einstellungen (Standards und Formate)** die gewünschte Sprache ein.

Nur wenn Sie alle Einstellungen gemacht haben, können Sie Texte in der gewünschten Sprache eingeben und korrekt anzeigen.

Spracheinstellung unter Windows 2000:

1. Stellen Sie über **Systemsteuerung > Ländereinstellungen > Allgemein > Standard** festlegen die gewünschte Anzeigesprache ein.
2. Stellen Sie über **Systemsteuerung > Ländereinstellungen > Allgemein > Gebietsschema (Standort)** die gewünschte Anzeigesprache ein.
3. Stellen Sie über **Systemsteuerung > Ländereinstellungen > Eingabe** die Eingabesprache (Systemgebietsschema) ein.

Nur wenn Sie alle Einstellungen gemacht haben, können Sie Texte in der gewünschten Sprache eingeben und korrekt anzeigen.

6.6 Einrichten eines Projekts

6.6.1 Anlegen eines Projekts

Um Ihre Automatisierungsaufgabe im Rahmen einer Projektverwaltung zu lösen, legen Sie zunächst ein neues Projekt an. Das neue Projekt wird in dem Verzeichnis angelegt, das Sie nach Anwahl des Menübefehls **Extras > Einstellungen** im Register "Allgemein" für die Projekte eingestellt haben.

Hinweis

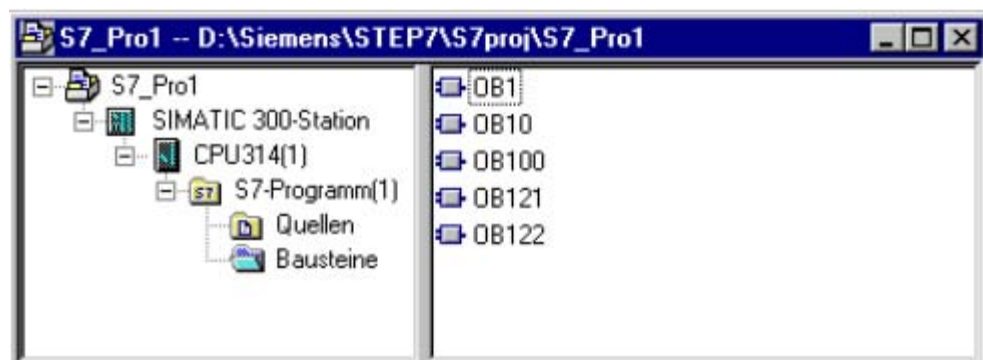
Der SIMATIC Manager lässt Namen zu, die länger als 8 Zeichen sind. Der Name des Projektverzeichnisses wird jedoch auf 8 Zeichen abgeschnitten. Projektnamen müssen sich daher in den ersten 8 Zeichen unterscheiden. Zwischen Groß- und Kleinschreibung wird nicht unterschieden.

Eine Schritt-für-Schritt-Anleitung zum Anlegen eines Projekts finden Sie in Anlegen eines Projekts manuell oder in Anlegen eines Projekts über Assistent.

Anlegen eines Projekts über Assistenten

Ein neues Projekt können Sie am einfachsten mit dem Assistenten 'Neues Projekt' anlegen. Zum Aufruf wählen Sie den Menübefehl **Datei > Assistent 'Neues Projekt'**. Der Assistent fragt die benötigten Angaben mit Hilfe von Dialogfeldern ab und legt dann das Projekt für Sie an. Neben Station, CPU, Programm-, Quellen-, Bausteinordner und OB1 können Sie dort bereits OBs für die Fehler- und Alarmbearbeitung auswählen.

Nachfolgendes Bild zeigt ein mit dem Assistenten angelegtes Projekt.



Manuelles Anlegen eines Projekts

Es besteht aber auch die Möglichkeit, ein neues Projekt über den Menübefehl **Datei > Neu** im SIMATIC Manager anzulegen. Es enthält bereits das Objekt "MPI-Subnetze".

Alternativen für die Weiterarbeit

Bei der Wahl der weiteren Bearbeitungsreihenfolge eines Projekts haben Sie best mögliche Flexibilität. Nach dem Anlegen eines Projekts können Sie beispielsweise

- zunächst die Hardware konfigurieren und anschließend die Software dafür erstellen oder aber
- unabhängig von einer konfigurierten Hardware mit der Software-Erstellung beginnen.

Alternative 1: Zunächst Hardware konfigurieren

Wenn Sie zunächst die Hardware konfigurieren möchten, gehen Sie wie in Band 2 des Handbuchs "Hardware-Konfiguration mit STEP 7" beschrieben vor. Nach dem Konfigurieren sind die für die Software-Erstellung erforderlichen Ordner "S7-Programm" bzw. "M7-Programm" bereits eingefügt. Dann fahren Sie mit dem Einfügen der zur Programmerstellung erforderlichen Objekte fort. Erstellen Sie anschließend die Software für die programmierbaren Baugruppen.

Alternative 2: Zunächst Software erstellen

Sie können auch ohne vorherige Hardware-Konfigurierung Software erstellen und die Konfigurierung zu einem späteren Zeitpunkt nachholen. Der Hardware-Aufbau einer Station braucht zur Eingabe von Programmen noch nicht festgelegt sein.

Die prinzipielle Vorgehensweise ist wie folgt:

1. Fügen Sie die erforderlichen Software-Ordner (S7 / M7-Programm ohne Station und CPU) in Ihr Projekt ein.
Dabei entscheiden Sie lediglich, ob der Ordner "Programme" für S7-Hardware oder M7-Hardware enthalten soll.
2. Erstellen Sie anschließend die Software für die programmierbaren Baugruppen.
3. Konfigurieren Sie Ihre Hardware.
4. Ordnen Sie das M7- bzw. S7-Programm einer CPU zu, nachdem Sie die Hardware konfiguriert haben.

6.6.2 Einfügen von Stationen

In einem Projekt repräsentiert die Station den Hardware-Aufbau des Automatisierungssystems und beinhaltet die Daten zur Konfigurierung und Parametrierung der einzelnen Baugruppen.

Neue Projekte, die mit dem Assistenten 'Neues Projekt' angelegt wurden, enthalten bereits eine Station. Andernfalls können Sie die Station über den Menübefehl **Einfügen > Station** erzeugen.

Folgende Stationen stehen zur Auswahl:

- SIMATIC 300-Station
- SIMATIC 400-Station
- SIMATIC H-Station
- SIMATIC PC-Station
- PC/PG
- SIMATIC S5
- Andere Stationen, d. h. nicht SIMATIC S7/M7, SIMATIC S5

Die Station wird mit einer voreingestellten Bezeichnung (z. B. SIMATIC 300-Station (1), SIMATIC 300-Station (2) usw.) eingefügt. Sie können die Bezeichnung von Stationen durch aussagekräftige Namen ersetzen.

Eine Schritt-für-Schritt-Anleitung zum Einfügen finden Sie in Einfügen einer Station.

Hardware-Konfigurierung durchführen

Über die Hardware-Konfigurierung legen Sie mit Hilfe eines Baugruppenkatalogs die CPU und alle in Ihrer Steuerung enthaltenen Baugruppen fest. Die Hardware-Konfigurierung starten Sie durch Doppelklick auf die Station.

Nach dem Speichern und Beenden der Hardware-Konfigurierung wird zu jeder programmierbaren Baugruppe, die Sie bei der Konfigurierung angelegt haben, automatisch ein S7- bzw. M7-Programm als Software-Ordner sowie eine Verbindungstabelle (Objekt "Verbindungen") angelegt. Projekte, die mit dem Assistenten 'Neues Projekt' angelegt wurden, enthalten diese Objekte bereits von Anfang an.

Eine Schritt-für-Schritt-Anleitung zum Konfigurieren finden Sie in Konfigurieren der Hardware, ausführliche Informationen in Grundsätzliche Schritte beim Konfigurieren einer Station.

Verbindungstabelle erstellen

Eine (leere) Verbindungstabelle (Objekt "Verbindungen") wird automatisch zu jeder programmierbaren Baugruppe angelegt. Die Verbindungstabelle wird zur Definition von Kommunikationsverbindungen zwischen programmierbaren Baugruppen in einem Netz verwendet. Nach dem Öffnen wird ein Fenster aufgeblendet und darin eine Tabelle zur Definition von Verbindungen zwischen programmierbaren Baugruppen.

Ausführliche Informationen finden Sie in Vernetzen von Stationen innerhalb eines Projekts.

Nächste Schritte

Nach dem Erstellen der Hardware-Konfiguration können Sie die Software für Ihre programmierbaren Baugruppen erstellen (siehe auch Einfügen eines S7-/M7-Programms).

6.6.3 Einfügen eines S7- / M7-Programms

Die Software für programmierbare Baugruppen wird in Objektordnern abgelegt. Für SIMATIC S7-Baugruppen heißt ein solcher Objektordner "S7-Programm", für SIMATIC M7-Baugruppen "M7-Programm".

Nachfolgendes Bild zeigt als Beispiel ein S7-Programm in einer programmierbaren Baugruppe einer SIMATIC 300-Station.



Bereits angelegte Komponenten

Zu jeder programmierbaren Baugruppe wird automatisch ein S7- / M7-Programm als Ordner für die Software angelegt.

Das S7-Programm enthält bereits:

- Symboltabelle (Objekt "Symbole"),
- einen Ordner "Bausteine" für Bausteine mit erstem Baustein,
- einen Ordner "Quellen" für Quellprogramme.

Das M7-Programm enthält bereits:

- Symboltabelle (Objekt "Symbole"),
- einen Ordner "Bausteine".

Erstellen von S7-Bausteine

Sie wollen AWL-, KOP- oder FUP-Programme erstellen. Dazu markieren Sie das bereits angelegte Objekt "Bausteine" und klicken dann auf den Menübefehl **Einfügen > S7-Baustein**. In einem Folgemenü können Sie den Bausteintyp auswählen (Datenbaustein, Datentyp (UDT), Funktion, Funktionsbaustein, Organisationsbaustein, Variablentabelle (VAT)).

Nach dem Öffnen des (leeren) Bausteins können Sie das AWL-, KOP- oder FUP-Programm eingeben. Weitere Information dazu finden Sie in Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen und in den Handbüchern zu AWL, KOP und FUP.

Hinweis

Das Objekt Systemdaten (SDB), das Sie eventuell in Anwenderprogrammen vorfinden, wird vom System erzeugt. Sie können das Objekt zwar öffnen, den Inhalt aus Konsistenzgründen aber nicht ändern. Es dient dazu, nach dem Laden eines Programms Konfigurationsänderungen vornehmen und diese auf das Zielsystem nachladen zu können.

Verwenden von Bausteinen aus Standardbibliotheken

Für die Erstellung von Anwenderprogrammen können Sie auch Bausteine aus den Standardbibliotheken benutzen, die im Lieferumfang enthalten sind. Auf Bibliotheken greifen Sie über den Menübefehl **Datei > Öffnen** zu. Weitere Hinweise zum Benutzen von Standardbibliotheken sowie zum Anlegen eigener Bibliotheken finden Sie in Arbeiten mit Bibliotheken in der Online-Hilfe.

Erstellen von Quellen/CFC-Pläne

Sie wollen eine Quelle in einer bestimmten Programmiersprache oder einen CFC-Plan erstellen. Dazu markieren Sie im S7-Programm das Objekt "Quellen" bzw. "Pläne" und klicken anschließend auf den Menübefehl **Einfügen > S7-Software**. Wählen Sie im Folgemenu die zur Programmiersprache passende Quelle aus. Nach dem Öffnen der leeren Quelle können Sie das Programm eingeben. Weitere Hinweise finden Sie in Grundlagen zum Programmieren in AWL-Quellen.

Erstellen von Programmen für M7

Sie wollen Programme für das Betriebssystem RMOS einer programmierbaren Baugruppe aus dem M7-Spektrum erstellen. Dazu markieren Sie das M7-Programm und klicken anschließend auf den Menübefehl **Einfügen > M7-Software**. Wählen Sie im Folgemenu das zur Programmiersprache bzw. zum Betriebssystem passende Objekt aus. Nach dem Öffnen des angelegten Objekts gelangen Sie in die zugehörige Erstellungsumgebung.

Erstellen einer Symboltabelle

Eine (leere) Symboltabelle (Objekt "Symbole") wird beim Erzeugen des S7-/M7-Programms automatisch angelegt. Nach dem Öffnen wird das Fenster "Symbol Editor" aufgeblendet und darin die Symboltabelle angezeigt. Weitere Hinweise finden Sie in Eingeben mehrerer globaler Symbole in der Symboltabelle.

Einfügen externer Quellen

Sie können Quelldateien mit beliebigen ASCII-Editoren erstellen und bearbeiten. Diese Dateien können Sie anschließend in ein Projekt importieren und in einzelne Bausteine übersetzen.

Die bei der Übersetzung einer importierten Quelle entstehenden Bausteine werden im Ordner "Bausteine" abgelegt.

Weitere Hinweise finden Sie in Einfügen von externen Quellen.

6.7 Bearbeiten eines Projekts

Öffnen eines Projekts

Um ein Projekt zu öffnen, geben Sie zunächst den Menübefehl **Datei > Öffnen** ein. Wählen Sie dann ein Projekt in den Folgedialogen aus. Danach wird das Projektfenster geöffnet.

Hinweis

Wenn Sie das Projekt nicht in der Projektliste angezeigt bekommen, betätigen Sie die Schaltfläche "Durchsuchen". In dem gleichnamigen Dialogfeld können Sie nach weiteren Projekten suchen und gefundene Projekte in die Projektliste aufnehmen. Die Einträge in der Projektliste können Sie mit dem Menübefehl **Datei > Verwalten** ändern.

Kopieren eines Projekts

Ein Projekt kopieren Sie, indem Sie dieses über den Menübefehl **Datei > Speichern unter** unter einen anderen Namen abspeichern.

Projektteile wie Stationen, Programme, Bausteine usw. kopieren Sie mit dem Menübefehl **Bearbeiten > Kopieren**.

Eine Schritt-für-Schritt-Anleitung zum Kopieren eines Projekts finden Sie in Kopieren eines Projekts und Kopieren eines Projektteils.

Löschen eines Projekts

Ein Projekt löschen Sie über den Menübefehl **Datei > Löschen**.

Projektteile wie Stationen, Programme, Bausteine usw. löschen Sie mit dem Menübefehl **Bearbeiten > Löschen**.

Eine Schritt-für-Schritt-Anleitung zum Löschen eines Projekts finden Sie in Löschen eines Projekts und Löschen eines Projektteils.

6.7.1 Projekt auf verwendete Softwarepakete prüfen

Wird ein Projekt bearbeitet, das Objekte enthält, die mit einem anderen Softwarepaket erstellt wurden, so ist dieses Softwarepaket zur Bearbeitung notwendig.

STEP 7 hilft Ihnen, egal an welchem Erstellsystem Sie mit den Multiprojekten, Projekten oder Bibliotheken arbeiten, indem es Ihnen die benötigten Softwarepakete in der notwendig Version anzeigt.

Die Informationen über die benötigten Softwarepakete sind vollständig, wenn:

- das Projekt (bzw. alle Projekte im Multiprojekt) oder die Bibliothek mit STEP 7 ab V5.2 erstellt wurde.
- Sie das Projekt auf verwendete Softwarepakete geprüft haben. Markieren Sie hierzu das Projekt im SIMATIC Manager und wählen Sie den Menübefehl **Bearbeiten > Objekteigenschaften**. Wählen Sie im angezeigten Dialog das Register "Notwendige Softwarepakete". Den Hinweisen in diesem Dialog können Sie entnehmen, ob Sie das Projekt auf verwendete Softwarepakete prüfen sollten.

6.8 Texte mehrsprachig verwalten

STEP 7 bietet die Möglichkeit, Texte, die in einem Projekt einsprachig abgelegt sind, zu exportieren, übersetzen zu lassen, wieder zu importieren und in der übersetzten Sprache anzeigen zu lassen.

Die folgenden Arten von Texten können mehrsprachig verwaltet werden.

- Titel und Kommentare
 - Bausteintitel und Bausteinkommentare
 - Netzwerktitel und Netzwerkkommentare
 - Zeilenkommentare aus AWL-Programmen
 - Kommentare aus Symboltabellen, Variablendeklarationstabellen, anwenderdefinierten Datentypen und Datenbausteinen
 - Kommentare, Zustandsnamen und Transitionsnamen in HiGraph-Programmen
 - Erweiterungen zu Schrittnamen und Schrittkommentare in S7-GRAPH-Programmen
- Anzeigetexte
 - Meldungstexte, die von STEP 7, S7-GRAPH, S7-HiGraph, S7-PDIAG oder ProTool generiert werden
 - System-Textbibliotheken
 - Anwender-Textbibliotheken
 - Bedienerrelevante Texte
 - Anwendertexte

Export

Der Export wird für alle Bausteine und Symboltabellen ausgeführt, die unter dem angewählten Objekt liegen. Für jeden Texttyp wird eine Exportdatei erzeugt. Diese enthält je eine Spalte für die Quell- und für die Zielsprache. Die Texte in der Quellsprache dürfen nicht geändert werden.

Import

Beim Import wird der Inhalt der Spalten für die Zielsprache (rechte Spalte) in das angewählte Objekt übernommen. Es werden nur diejenigen Übersetzungen übernommen, deren Quelltexte (exportierte Texte) mit einem vorhandenen Eintrag in der Spalte "Quellsprache" übereinstimmen.

Sprachwechsel

Beim Sprachwechsel können alle Sprachen ausgewählt werden, die beim Import in das angewählte Projekt angegeben wurden. Der Sprachwechsel für "Titel und Kommentare" wird nur für das angewählte Objekt ausgeführt. Der Sprachwechsel für "Anzeigetexte" wird immer für das gesamte Projekt ausgeführt.

Sprache löschen

Beim Löschen einer Sprache werden alle Texte in dieser Sprache aus der internen Datenhaltung gelöscht.

Es sollte immer eine Sprache als Bezugssprache in Ihrem Projekt vorhanden sein. Das kann beispielsweise Ihre Landessprache sein. Diese Sprache sollten Sie nicht löschen. Beim Exportieren und Importieren stellen Sie diese Bezugssprache immer als Quellsprache ein. Die Zielsprache stellen Sie nach Wunsch ein.

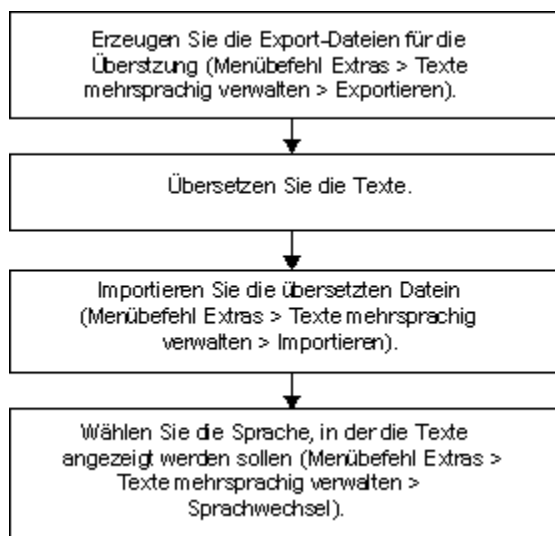
Reorganisieren

Beim Reorganisieren wird ein Sprachwechsel in die aktuell eingestellte Sprache durchgeführt. Die aktuell eingestellte Sprache ist die Sprache, die Sie in den Einstellungen für Kommentarbehandlung als "Sprache für zukünftige Bausteine" gewählt haben. Das Reorganisieren betrifft nur Titel und Kommentare.

Kommentarbehandlung

Sie können einstellen, wie Bausteinkommentare in Projekten mit mehrsprachig verwalteten Texten behandelt werden sollen.

Prinzipielle Vorgehensweise



6.8.1 Texttypen mehrsprachig verwalteter Texte

Für jeden Texttyp wird beim Export eine eigene Datei angelegt. Diese hat den Texttyp als Namen und das Exportformat als Endung (Texttyp.Format: z. B. SymbolComment.CSV oder SymbolComment.XLS). Dateien, die nicht der Namenskonvention genügen, können nicht als Quelle oder Ziel angegeben werden.

Die übersetzbaren Texte innerhalb eines Projekts sind in folgende Texttypen aufgeteilt:

Texttyp	Bedeutung
BlockTitle	Bausteintitel
BlockComment	Bausteinkommentar
NetworkTitle	Netzwerktitel
NetworkComment	Netzwerkkommentar
LineComment	Zeilenkommentar in AWL
InterfaceComment	Var_Section-Kommentar (Deklarationstabelle in Codebausteinen) UDT-Kommentar (anwenderdefinierte Datentypen) Datenbausteinkommentar
SymbolComment	Symbolkommentar
S7UserTexts	Vom Anwender eingegebene Texte, die auf Anzeigegeräten ausgegeben werden können
S7SystemTextLibrary	Texte aus System-Textbibliotheken, die in Meldungen integriert, dynamisch zur Laufzeit aktualisiert und am PG oder anderen Anzeigegeräten angezeigt werden können

Texttyp	Bedeutung
S7UserTextLibrary	Texte aus Anwender-Textbibliotheken, die in Meldungen integriert, dynamisch zur Laufzeit aktualisiert und am PG oder anderen Anzeigegeräten angezeigt werden können
HiGraphStateName HiGraphStateComment	S7-HiGraph Zustandsname Zustandskommentar
HiGraphTransitionName HiGraphTransitionComment	Transitionsname Transitionskommentar
S7GraphStateName S7GraphStateComment	S7-GRAPH Schritt-Namenserweiterung Schrittkommentar

Aus Editoren anderer Optionspakete (z.B Protocol, WinCC,.....) kann es noch weitere applikationsspezifische Texttypen geben, die hier nicht beschrieben sind.

6.8.2 Aufbau der Export-Datei

Die Export-Datei ist prinzipiell folgendermaßen aufgebaut:

Beispiel:

\$ Languages		
7(1) Deutsch (Deutschland)	9(1) Englisch (USA)	
\$ Typ(NetworkTitle)		
// Maximale Textlänge: 64 Zeichen		
#\$ _Export am 16.11.2005 13:14:34		
Erste zu übersetzende Zeichenfolge	Übersetzung	test\S7-Programm(1)\Bausteine\OB1
Zweite zu übersetzende Zeichenfolge	Übersetzung	test\S7-Programm(1)\Bausteine\OB1
Zeichenfolge, die in der Übersetzung nicht angezeigt werden soll	\$_hide	test\S7-Programm(1)\Bausteine\OB1

Quellsprache

Zielsprache

Verwendungsstelle

Grundsätzlich gilt:

1. Nicht geändert, überschrieben oder gelöscht werden dürfen:
 - die Felder mit einem vorangestellten "\$_" (das sind Schlüsselwörter)
 - die Nummern für die Sprache (im Beispiel oben: 7(1) für die Quellsprache Deutsch (Deutschland) und 9(1) für die Zielsprache Englisch)
2. In der Datei sind immer nur Texte eines Texttyps hinterlegt. Im Beispiel ist der Texttyp Netzwerktitel (\$_Typ(NetworkTitle)). Die Regeln für den Übersetzer, der in dieser Datei editiert, sind im einleitenden Text der Export-Datei selbst hinterlegt.
3. Zusatzinformationen zu den Texten oder Kommentare müssen immer vor der Typdefinition (\$_Typ...) oder hinter der letzten Spalte stehen.

Hinweise

Wenn die Spalte für die Zielsprache mit "512(32) \$_Undefined" überschrieben ist, dann wurde beim Exportieren keine Zielsprache angegeben. Zur besseren Übersichtlichkeit können Sie diesen Text ersetzen durch die Zielsprache; z. B. durch "9(1) Englisch (USA)". Beim Importieren müssen Sie dann die vorgeschlagene Zielsprache kontrollieren und ggf. neu wählen.

Texte, die in der Zielsprache nicht angezeigt werden sollen, können mit dem Schlüsselwort \$_hide ausgeblendet werden. Dies gilt nicht für Kommentare zu Variablen (InterfaceComment) und zu Symbolen (SymbolComment).

Format der Exportdatei

Sie können wählen, in welchem Format die Exportdateien abgelegt werden sollen.

Haben Sie sich für das CSV-Format entschieden, beachten Sie bei der Bearbeitung mit EXCEL, dass EXCEL eine CSV-Datei nur dann korrekt öffnet, wenn dazu der Öffnen-Dialog verwendet wird. **Beim Öffnen einer CSV-Datei durch Doppelklick im Explorer wird diese Datei häufig unbrauchbar.** Durch folgendes Vorgehen kann das Bearbeiten von CSV-Dateien mit EXCEL erleichtert werden:

1. Öffnen der Exportdateien mit EXCEL
2. Abspeichern der Dateien als XLS-Dateien
3. Übersetzen der Texte in den XLS-Dateien
4. Abspeichern der XLS-Dateien mit EXCEL im CSV-Format.

Hinweis

Die Exportdateien dürfen nicht umbenannt werden!

6.8.3 Verwalten von Anwendertexten, deren Sprach-Font nicht installiert ist

Sie haben auch die Möglichkeit, Anwendertexte in einer Sprache, deren Sprach-Font Sie nicht auf Ihrem Betriebssystem installiert haben, zu exportieren, übersetzen zu lassen, zu importieren und anschließend in Ihrem Projekt abzuspeichern.

Allerdings können diese Texte nur auf einem Rechner angezeigt werden, auf dem der entsprechende Sprach-Font installiert ist.

Gehen Sie z. B. bei Anwendertexten, die ins Russische übersetzt werden soll, folgendermaßen vor, wenn Sie die kyrillische Schrift nicht installiert haben:

1. Exportieren Sie die zu übersetzenden Anwendertexte mit Quellsprache Deutsch und Zielsprache Russisch.
2. Schicken Sie die Exportdateien an den Übersetzer; der auf seinem Rechner natürlich die kyrillische Schrift installiert hat.
3. Importieren Sie die übersetzten Exportdateien.
Ergebnis: Das Projekt ist jetzt mehrsprachig (Deutsch und Russisch) auf Ihrem Rechner angelegt.
4. Speichern Sie das gesamte Projekt und schicken Sie es an den Endkunden, der sich die russischen Texte auf seinem Rechner mit installierter kyrillischer Schrift anzeigen lassen kann.

6.8.4 Informationen zur Protokoll-Datei

Fehlermeldungen und Warnungen, die beim Arbeiten mit mehrsprachig verwalteten Texten entstehen, werden in einer Protokoll-Datei im TXT-Format ausgegeben. Diese Datei wird im gleichen Verzeichnis abgelegt wie die Export-Dateien.

Die Meldungen sind im Allgemeinen selbsterklärend. Einige weiterführende Erklärungen finden Sie hier:



Warnung

Der Text 'xyz' in der Datei 'xyz' ist bereits vorhanden. Das weitere Vorkommen des Textes wurde ignoriert.

Erläuterung

Ein Text wird unabhängig von seiner Sprache als Schlüssel für die Übersetzung verwendet. Wird ein identischer Text in mehr als einer Sprache oder in einer Sprache mehrmals für unterschiedliche Begriffe verwendet, kann er nicht mehr eindeutig zugeordnet werden und wird deshalb nicht übersetzt.

Beispiel:

\$ Languages	
7(1) Deutsch (Deutschland)	9(1) Englisch (USA)
kein	none
keine	none
keiner	none

/
/

Quellsprache
Zielsprache

Dies trifft nur auf Titel und Kommentare zu.

Abhilfe

Benennen Sie die betreffenden Texte in der exportierten Textdatei um (im Beispiel muss statt der 3 verschiedenen deutschen Begriffe ein einziger verwendet werden) und importieren Sie die Texte wieder.

6.8.5 Optimieren der Vorlage für die Übersetzung

Sie können das sprachliche "Quellmaterial" für die Übersetzung bereinigen, indem Sie unterschiedliche Begriffe bzw. Ausdrücke zusammenführen.

Beispiel

Vor der Bereinigung (Export-Datei):

\$ Languages		
7(1) Deutsch (Deutschland)	7(1) Deutsch (Deutschland)	
\$ Typ(SymbolComment)		
Auto-Freig.		I
Automatische Freigabe		
Auto-Freigabe		

/
/

Quellsprache
Zielsprache

Zusammenführen zu einem Ausdruck:

\$ _Languages		
7(1) Deutsch (Deutschland)	7(1) Deutsch (Deutschland)	
\$ _Typ(SymbolComment)		
Auto-Freig.	Auto-Freigabe	
Automatische Freigabe	Auto-Freigabe	
Auto-Freigabe	Auto-Freigabe	

/
/

Quellsprache
Zielsprache

Nach der Bereinigung (d. h. nach Import und anschließendem Export):

\$ _Languages		
7(1) Deutsch (Deutschland)	7(1) Deutsch (Deutschland)	
\$ _Typ(SymbolComment)		
Auto-Freigabe	Auto-Freigabe	

/
/

Quellsprache
Zielsprache

6.8.6 Optimierung des Übersetzungsvorgangs

Bei Projekten, deren Aufbau und Texte ähnlich einem Vorgängerprojekt sind, können Sie den Übersetzungsvorgang optimieren.

Insbesondere für Projekte, die durch Kopieren und anschließendes Anpassen entstanden sind, ist das im folgenden beschriebene Vorgehen empfehlenswert.

Voraussetzung

Es liegt ein bereits übersetztes Exportziel vor.

Vorgehensweise

1. Kopieren Sie die Exportdateien in das Projektverzeichnis für das zu übersetzende neue Projekt.
2. Öffnen Sie das neue Projekt und exportieren Sie die Texte (Menübefehl **Extras > Texte mehrsprachig verwalten > Exportieren**). Weil das Exportziel bereits existiert, werden Sie gefragt, ob das Exportziel erweitert oder überschrieben werden soll.
3. Klicken Sie auf die Schaltfläche "Erweitern".
4. Lassen Sie die Exportdateien übersetzen (es müssen nur die neuen Texte übersetzt werden).
5. Importieren Sie anschließend die übersetzten Dateien.

6.9 Micro Memory Card (MMC) als Datenträger

6.9.1 Wissenswertes zu Micro Memory Cards (MMC)

Micro Memory Cards (MMC) sind steckbare Speicherkarten für z. B. eine CPU 31xC oder eine IM 151/CPU (ET 200S), die sich durch eine extrem kompakte Bauform auszeichnen.

Für die MMCs gilt ein neues Speicherkonzept, das im Folgenden kurz beschrieben ist.

Inhalt der MMCs

Die MMC übernimmt sowohl die Funktion des Ladespeichers als auch die Funktion eines Datenträgers.

MMC als Ladespeicher

Die MMCs enthalten den kompletten **Ladespeicher** einer für MMCs geeigneten CPU. Er enthält das Programm mit den Bausteinen (OBs, DBs, FCs, ...) sowie die HW-Konfiguration. Diese Inhalte beeinflussen die Funktionsweise der CPU. In der Funktion als Ladespeicher können Bausteine und die HW-Konfiguration mit Ladefunktionen transferiert werden (z. B. **Laden** in CPU). In die CPU geladene Bausteine werden sofort wirksam, die HW-Konfiguration nach Anlauf der CPU.

Reaktion nach Urlöschen

Bausteine auf der MMC bleiben nach dem Urlöschen erhalten.

Laden und Löschen

Die Bausteine auf der MMC können überschrieben werden.

Die Bausteine auf der MMC können gelöscht werden.

Überschriebene und gelöschte Bausteine können nicht mehr wiederhergestellt werden.

Zugriff auf Datenbausteine auf der MMC

Für große Datenmengen bzw. für Daten, die eher selten im Anwenderprogramm benötigt werden, können Sie Datenbausteine und Datenbaustein-Inhalte auf der MMC hantieren. Dafür stehen neue Systemfunktionen zur Verfügung:

- SFC 82: Datenbaustein im Ladespeicher erzeugen
- SFC 83: Lesen aus Datenbaustein im Ladespeicher
- SFC 84: Schreiben in einen Datenbaustein im Ladespeicher

MMC und Passwortschutz

Wird eine CPU, die mit einer MMC ausgestattet ist (z. B. eine CPU aus der 300-C-Familie), passwortgeschützt, so wird dieses Passwort auch beim Öffnen dieser MMC im SIMATIC Manager (am PG oder PC) abgefragt.

Anzeige der Speicherbelegung mit STEP 7

Die Anzeige der Belegung des Ladespeichers im Baugruppenzustandsdialog (Register "Speicher") fasst EPROM- und RAM-Bereich zusammen.

Die Bausteine auf der MMC zeigen zu 100% EPROM-Verhalten.

6.9.2 Micro Memory Card als Datenträger verwenden

Die SIMATIC Micro Memory Card (MMC) lässt sich mit STEP 7 wie ein ganz normaler externer Datenträger einsetzen.

Unter der Voraussetzung, dass die MMC ausreichend groß gewählt wurde, lassen sich alle im Datei-Explorer des Betriebssystems sichtbaren Dateien auf die MMC übertragen.

Sie können somit an Ihrer Anlage zusätzliche Zeichnungen, Serviceanweisungen, Funktionsbeschreibungen etc. für andere Mitarbeiter zur Verfügung stellen.

6.9.3 Memory Card-Datei

Memory Card-Dateien (*.wld) werden erzeugt für die

- Software PLC **WinLC** (WinAC Basis und WinAC RTX) und
- SlotPLCs **CPU 41x-2 PCI** (WinAC Slot 412 und WinAC Slot 416).

In einer Memory Card-Datei können die Bausteine und Systemdaten für eine WinLC oder CPU 41x-2 PCI wie in einer S7-Memory Card abgelegt werden. Der Inhalt dieser Dateien entspricht dann dem Inhalt einer entsprechenden Memory Card für eine S7-CPU.

Diese Datei kann dann über einen Menübefehl des Bedienpanels der WinLC oder CPU 41x-2 PCI in deren Ladespeicher geladen werden, was dem Laden des Anwenderprogramms mit STEP 7 entspricht.

Bei den CPUs 41x-2 PCI kann diese Datei automatisch beim Hochlauf des PC-Betriebssystems geladen werden, wenn die CPU 41x-2 PCI ungepuffert und nur mit einer RAM-Card betrieben wird (Funktion "Autoload").

Memory Card-Dateien sind "normale" Dateien im Sinne von Windows, die mit dem Explorer verschoben, gelöscht oder mit einem Datenträger transportiert werden können.

Weitere Informationen entnehmen Sie bitte der entsprechenden Dokumentation zu den WinAC Produkten.

6.9.4 Projektdaten auf Micro Memory Card (MMC) ablegen

Mit STEP 7 können Sie die Projektdaten Ihres STEP 7-Projektes und beliebige andere Daten (z. B. WORD- oder Excel-Dateien) auf der SIMATIC Micro Memory Card (MMC) einer geeigneten CPU, bzw. eines PG/PC ablegen. Sie haben damit den Vorteil auch mit Programmiergeräten, die das Projekt nicht gespeichert haben, auf die Projektdaten zugreifen zu können.

Voraussetzungen

Das Ablegen von Projektdaten auf einer MMC ist nur möglich, wenn sich die MMC im Schacht einer geeigneten CPU bzw. eines PG/PC befindet und eine Onlineverbindung besteht.

Die Größe der MMC muss so bemessen sein, dass die abgelegten Daten auf dieser Platz finden.

Folgende Daten können auf MMC abgelegt werden

Unter der Voraussetzung, dass die MMC ausreichend groß gewählt wurde, lassen sich alle im Datei-Explorer des Betriebssystems sichtbaren Dateien auf die MMC übertragen, wie z. B.:

- Komplette Projektdaten zu STEP 7
- Stationskonfigurationen
- Symboltabellen
- Bausteine und Quellen
- Mehrsprachig verwaltete Texte
- Beliebige andere Daten, wie z. B. WORD- oder Excel-Dateien

7 Bearbeiten von Projekten mit unterschiedlichen STEP 7-Versionen

7.1 Version 2-Projekte und Bibliotheken bearbeiten

Die Version V5.2 von STEP 7 unterstützt **Änderungen in V2-Projekten** und -Bibliotheken **nicht** mehr. Wenn Sie V2-Projekte oder -Bibliotheken bearbeiten, dann können Inkonsistenzen auftreten, die dazu führen, dass V2-Projekte oder -Bibliotheken nicht mehr mit älteren STEP 7 Versionen bearbeitet werden können.

Um V2-Projekte oder Bibliotheken weiterhin bearbeiten zu können, muss eine STEP 7-Version verwendet werden, die älter als V5.1 ist.

7.2 Erweitern von DP-Slaves, die mit Vorgänger-Versionen von STEP 7 erstellt wurden

Konstellationen, die durch das Einspielen neuer GSD-Dateien entstehen können

Neue DP-Slaves können durch die Installation neuer GSD-Dateien in den Hardware-Katalog von HW Konfig aufgenommen werden. Nach der Installation sind sie im Ordner "Weitere Feldgeräte" verfügbar.

Sie können einen modularen DP-Slave nicht mehr wie gewohnt umkonfigurieren bzw. erweitern, wenn er

- mit einer Vorgänger-Version von STEP 7 konfiguriert wurde und
- im Hardware Katalog nicht durch eine GSD-, sondern durch eine Typdatei repräsentiert wurde und
- durch eine neue GSD-Datei überinstalliert wurde

Abhilfe

Wenn Sie den DP-Slave mit **neuen Baugruppen**, die in der GSD-Datei beschrieben sind, einsetzen wollen:

- Löschen Sie den DP-Slave und projektieren Sie ihn neu - der DP-Slave wird dann nicht durch die Typ-, sondern komplett durch die GSD-Datei beschrieben.

Wenn Sie **keine neuen Baugruppen**, die nur in der GSD-Datei beschrieben sind, einsetzen wollen:

- Wählen Sie den Ordner "Weitere FELDKERÄTE/Kompatible Profibus-DP-Slaves" unter PROFIBUS-DP im Fenster "Hardware Katalog". In diesen Ordner verschiebt STEP 7 die "alten" Typdateien, wenn sie durch neue GSD-Dateien ersetzt werden. In diesem Ordner finden Sie die Baugruppen, mit denen Sie den bereits konfigurierten DP-Slave erweitern können.

Konstellation nach Austausch von Typ- durch GSD-Dateien in STEP 7 V5.1 Servicepack 4

Ab STEP 7 V5.1, Servicepack 4 sind die Typdateien weitgehend durch GSD-Dateien ersetzt bzw. aktualisiert worden. Die Ersetzung betrifft nur die von STEP 7 gelieferten Katalogprofile, nicht die selbst erstellten Katalogprofile.

Die DP-Slaves, deren Eigenschaften vorher von Typdateien und nun von GSD-Dateien bestimmt werden, sind an derselben Stelle im Hardware Katalog geblieben.

Die "alten" Typdateien wurden nicht gelöscht, sondern an eine andere Stelle des Hardware Kataloges verschoben. Sie befinden sich nun unter "**Weitere Feldgeräte\Kompatible Profibus-DP-Slaves\...**".

Erweitern einer bestehenden DP-Konfiguration mit STEP 7 ab V5.1, Servicepack 4

Wenn Sie ein Projekt bearbeiten, das mit einer Vorgänger-Version von STEP 7 (kleiner V5.1, SP4) angelegt wurde und Sie wollen einen modularen DP-Slave erweitern, dann können Sie die Baugruppen bzw. Module nicht von der gewohnten Stelle des Hardware Katalogs verwenden. In diesem Fall verwenden Sie den DP-Slave unter "**Weitere FELDKERÄTE\Kompatible Profibus-DP-Slaves\...**".

Bearbeiten einer DP-Konfiguration mit einer STEP 7-Version kleiner V5.1, SP4

Wenn Sie einen "aktualisierten" DP-Slave, mit STEP 7 ab V5.1, Servicepack 4 konfigurieren und anschließend das Projekt mit einer Vorgänger-Version bearbeiten (STEP 7-Version kleiner V5.1, SP4), dann ist dieser DP-Slave nicht bearbeitbar, da die GSD-Datei der STEP 7-Version unbekannt ist.

Abhilfe: Sie können die benötigte GSD-Datei in der Vorgänger-Version von STEP 7 nachinstallieren. In diesem Fall wird die GSD-Datei im Projekt gespeichert. Wenn das Projekt anschließend mit der aktuellen Version von STEP 7 bearbeitet wird, verwendet STEP 7 die nachinstallierte GSD-Datei für die Projektierung.

7.3 Aktuelle Konfigurationen mit Vorgänger-Versionen von STEP 7 bearbeiten

Projektierung von Direktem Datenaustausch (Querverkehr)

Projektierung von direktem Datenaustausch bei einem DP-Master **ohne** DP-Mastersystem:

- nicht möglich mit STEP 7 V5.0, Servicepack 2 (oder älter)
- möglich mit STEP 7 V5.0, ab Servicepack 3 und ab STEP 7 V5.1

Wenn Sie einen DP-Master ohne eigenes DP-Mastersystem mit projektierten Zuordnungen für den direkten Datenaustausch speichern und dieses Projekt mit einer älteren STEP 7 V5-Version weiterbearbeiten (STEP 7 V5.0, Servicepack 2 (oder älter)), dann können folgende Effekte auftreten:

- Ein DP-Mastersystem wird angezeigt mit DP-Slaves, die für eine STEP 7-interne Datenablage der Zuordnungen für den direkten Datenaustausch genutzt werden. Diese DP-Slaves gehören nicht zum angezeigten DP-Mastersystem.
- Sie können weder ein neues noch ein verwaistes DP-Mastersystem an diesen DP-Master anschließen.

Online-Verbindung zur CPU über PROFIBUS-DP-Schnittstelle

Projektierung der PROFIBUS-DP-Schnittstelle **ohne** DP-Mastersystem:

- STEP 7 V5.0, Servicepack 2 (oder älter): eine Verbindung über diese Schnittstelle zur CPU ist nicht möglich.
- ab STEP 7 V5.0, Servicepack 3: Beim Übersetzen werden Systemdaten für die PROFIBUS-DP-Schnittstelle erzeugt; eine Verbindung über diese Schnittstelle zur CPU ist nach dem Laden möglich.

7.4 SIMATIC PC - Konfigurationen aus Vorgänger-Versionen nutzen

PC-Konfigurationen aus STEP 7 V5.1-Projekten (bis SP 1)

Ab STEP 7 V5.1, Servicepack 2 können Sie Verbindungen in die PC-Station laden wie in eine S7-300 oder S7-400-Station (ohne Umweg über eine Konfigurationsdatei). Trotzdem wird beim Speichern und übersetzen immer eine Konfigurationsdatei erzeugt, um auch über diesen Weg die Projektierung auf die Ziel-PC-Station übertragen zu können.

Das hat zur Folge, dass die neu erzeugten Konfigurationsdateien Informationen enthalten, die von "älteren" PC-Stationen nicht interpretiert werden können. STEP 7 stellt sich automatisch auf diesen Umstand ein:

- Wenn Sie eine SIMATIC PC-Station mit STEP 7 ab V5.1, Servicepack 2 **neu** projektieren, dann geht STEP 7 davon aus, dass die Ziel-PC-Station mittels SIMATIC NET CD ab 7/2001 konfiguriert wurde, d. h. dass der S7RTM (Runtime Manager) installiert ist. Die Konfigurationsdaten werden so erzeugt, dass sie von einer "neuen" PC-Station interpretiert werden können.
- Wenn Sie eine projektierte SIMATIC PC-Station aus einer Vorgänger-Version verwenden (z. B. eine PC-Station, die mit STEP 7 V5.1, Servicepack 1 projektiert wurde), dann geht STEP 7 **nicht** davon aus, dass die Ziel-PC-Station mittels SIMATIC NET CD ab 7/2001 konfiguriert wurde. Die Konfigurationsdaten werden so erzeugt, dass sie von einer "alten" PC-Station interpretiert werden können.

Wenn dieses voreingestellte Verhalten nicht passend ist, können Sie das Verhalten ändern wie im Folgenden beschrieben:

Einstellung im Kontext "Hardware konfigurieren":

1. Öffnen Sie die Hardware-Konfiguration der PC-Station.
2. Klicken Sie mit der rechten Maustaste in das Stationsfenster (in die weiße Fläche).
3. Wählen Sie das Kontextmenü "Stationseigenschaften".
4. Aktivieren oder deaktivieren Sie das Kontrollkästchen unter "Kompatibilität".

Einstellung im Kontext "Netze konfigurieren"

1. Öffnen Sie die Netzkonfiguration.
2. Markieren Sie die PC-Station.
3. Wählen Sie den Menübefehl **Bearbeiten > Objekteigenschaften**.
4. Wählen Sie im Dialog das Register "Konfiguration".
5. Aktivieren oder deaktivieren Sie das Kontrollkästchen unter "Kompatibilität".

PC-Konfigurationen aus STEP 7 V5.0-Projekten

Wenn Sie eine projektierte SIMATIC PC-Station mit STEP 7 ab V5.0, Servicepack 3 weiter bearbeiten wollen, um neue Komponenten zu konfigurieren, die erst ab Servicepack 3 unterstützt werden, dann müssen Sie die Station konvertieren:

1. Markieren Sie im SIMATIC Manager die SIMATIC PC-Station und wählen Sie den Menübefehl **Bearbeiten > Objekteigenschaften**.
2. Im Register "Funktionalität" des Eigenschaftsdialogs klicken Sie auf die Schaltfläche "Erweitern".
Anschließend wird die SIMATIC PC-Station konvertiert und ist nur noch mit STEP 7 V5.0, Servicepack 3 oder späteren Versionen zu bearbeiten.

7.5 Darstellung von Baugruppen, die mit neueren STEP 7-Versionen oder einem Optionspaket projiziert sind




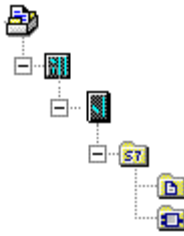

Ab STEP 7 V5.1, Servicepack 3 werden alle Baugruppen dargestellt; auch die, die mit einer neueren STEP 7-Version projiziert wurden und daher dem "älteren" STEP 7 unbekannt sind. Es werden auch die Baugruppen dargestellt, die mit einem Optionspaket projiziert wurden - auch wenn beim projekt-öffnenden PG das notwendige Optionspaket nicht installiert ist.

In vorangegangenen STEP 7-Versionen sind die Baugruppen und darunter liegenden Objekte nicht dargestellt worden. In der aktuellen Version sind diese Objekte sichtbar und zum Teil bearbeitbar. Sie können z. B. mit Hilfe dieser Funktion Anwenderprogramme auch dann ändern, wenn das Projekt auf einem anderen Rechner mit einer neueren STEP 7-Version erstellt wurde und die Baugruppe (z. B. eine CPU) nicht mit der vorliegenden STEP 7-Version projiziert werden kann, weil sie z. B. neue Eigenschaften und neue Parameter hat.

Die für STEP 7 "unbekannte" Baugruppe erscheint als Stellvertreter-Baugruppe mit folgender Darstellung:



Wenn Sie das Projekt mit der geeigneten STEP 7-Version bzw. mit dem geeigneten Optionspaket öffnen, werden alle Baugruppen in der gewohnten Darstellung angezeigt und sind auch wieder uneingeschränkt bearbeitbar.

PG mit neuestem STEP 7 / mit Optionspaket		PG mit älterem STEP 7 / ohne Optionspaket
		
	>>>---Projektdaten--->>>	
Darstellung mit "bekannt", neuester Baugruppe		Darstellung der neuesten Baugruppe als "unbekannte" Baugruppe
		

Arbeiten mit der Stellvertreter-Baugruppe im SIMATIC Manager

Die Stellvertreter-Baugruppe ist unterhalb der Station sichtbar. Alle unterlagerten Objekte wie z. B. Anwenderprogramm, Systemdaten und Verbindungstabelle sind sichtbar und aus dem SIMATIC Manager ladbar.

Das Anwenderprogramm (z. B. Bausteine) können Sie öffnen, bearbeiten, übersetzen und laden. Es sind jedoch folgende Einschränkungen bei Projekten mit Stellvertreter-Baugruppen vorhanden:

- Das Kopieren einer Station mit Stellvertreter-Baugruppe ist nicht möglich.
- Bei "Projekt Speichern unter..." kann die Option "mit Reorganisation" nicht vollständig ausgeführt werden.
Die Stellvertreter-Baugruppe sowie alle Bezüge und unterlagerten Objekte dieser Baugruppe fehlen anschließend im kopierten und reorganisierten Projekt (z. B. das Anwenderprogramm).

Arbeiten mit der Stellvertreter-Baugruppe in der Hardware-Konfiguration

Die Stellvertreter-Baugruppe wird auf dem Steckplatz dargestellt, auf dem sie projektiert wurde.

Sie können die Baugruppe öffnen, aber nicht umparametrieren und nicht laden. Die Baugruppeneigenschaften sind auf das Register "Stellvertreter" beschränkt. Die Stationskonfiguration kann nicht geändert werden (z. B. können keine neuen Baugruppen hinzugefügt werden).

Die Diagnose der Hardware (z. B. Station online öffnen) ist ebenfalls möglich (eingeschränkt; neue Diagnosemöglichkeiten und -texte werden nicht berücksichtigt).

Arbeiten mit der Stellvertreter-Baugruppe in der Netzkonfiguration

Die Stellvertreter-Baugruppe wird auch in NetPro dargestellt. Dabei wird dem Namen der Baugruppe in der betreffenden Station ein Fragezeichen vorangestellt.

Ein Projekt mit einer Stellvertreter-Baugruppe kann in NetPro nur schreibgeschützt geöffnet werden.

Wenn Sie das Projekt schreibgeschützt öffnen, können Sie sich die Netzkonfiguration anzeigen lassen und drucken. Verbindungsstatus ist ebenfalls möglich; zumindest werden die Informationen angezeigt, die in der vorliegenden STEP 7-Version unterstützt werden.

Sie können aber generell keine Änderungen vornehmen und diese Änderungen speichern, übersetzen und laden.

Nachinstallieren von Baugruppen

Falls die Baugruppe aus einer späteren Version von STEP 7 stammt und ein HW-Update für diese Baugruppe verfügbar ist, können Sie den Stellvertreter durch die "tatsächliche" Baugruppe ersetzen. Bereits beim Öffnen der Station werden Sie über die notwendigen HW-Updates bzw. Optionspakete informiert und können aus diesem Dialog heraus HW-Updates installieren. Alternativ können Sie über den Menübefehl **Extras > HW-Updates installieren** die Baugruppen nachinstallieren.

8 Festlegen von Symbolen

8.1 Absolute und symbolische Adressierung

In einem STEP 7-Programm arbeiten Sie mit Operanden wie E/A-Signalen, Merkern, Zählern, Zeiten, Datenbausteinen und Funktionsbausteinen. Sie können diese Operanden in Ihrem Programm absolut adressieren, die Lesbarkeit von Programmen steigt jedoch erheblich, wenn Sie dafür Symbole (z. B. Motor_A_Ein oder Bezeichnungen gemäß eines in Ihrer Branche üblichen Kennzeichnungssystems) benutzen. In Ihrem Anwenderprogramm lässt sich dann ein Operand über dieses Symbol ansprechen.

Absolute Adresse

Eine absolute Adresse besteht aus einem Operandenkennzeichen und einer Adresse (z. B. A 4.0, E 1.1, M 2.0, FB 21).

Symbolische Adressierung

Sie können Ihr Programm übersichtlicher gestalten und die Fehlerbehebung vereinfachen, wenn Sie die absoluten Adressen mit symbolischen Namen versehen.

STEP 7 kann automatisch die symbolischen Namen in die erforderlichen absoluten Adressen übersetzen. Wenn Sie auf ARRAYS, STRUCTs, Datenbausteine, Lokaldaten, Codebausteine und anwenderdefinierte Datentypen vorzugsweise über symbolische Namen zugreifen, dann müssen Sie den absoluten Adressen jedoch zunächst symbolische Namen zuordnen, bevor Sie die Daten symbolisch adressieren können.

Sie können beispielsweise dem Operanden A 4.0 den symbolischen Namen MOTOR_EIN zuordnen und dann MOTOR_EIN als Adresse in einer Programmanweisung verwenden. Über symbolische Adressen können Sie leichter erkennen, inwieweit Elemente des Programms mit den Komponenten Ihres Prozesssteuerungsprojekts übereinstimmen.

Hinweis

In einem symbolischen Namen (Variablenbezeichnung) sind zwei Unterstriche nacheinander (wie z. B. MOTOR__EIN) nicht erlaubt.

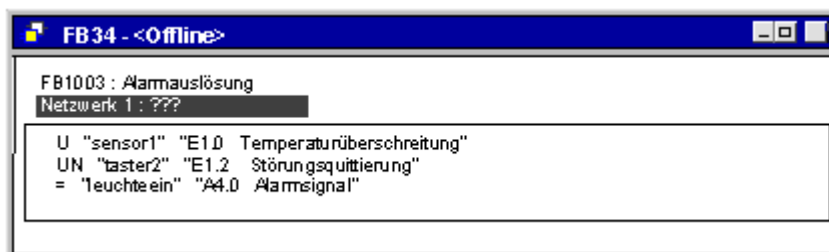
Unterstützung bei der Programmeingabe

In den Programmiersprachen KOP, FUP und AWL können Sie Adressen, Parameter und Bausteinnamen absolut oder als Symbol angeben.

Mit dem Menübefehl **Ansicht > Symbolische Darstellung** schalten Sie zwischen der Darstellung absoluter und symbolischer Adressierung um.

Zur leichteren Programmierung mit symbolischer Adressierung können Sie sich zu den vorkommenden Symbolen die zugehörige absolute Adresse mit dem Symbolkommentar anzeigen lassen. Dieses erreichen Sie über den Menübefehl **Ansicht > Symbolinformation**. Dadurch wird hinter jeder AWL-Anweisung der Zeilenkommentar entsprechend ersetzt. Die Anzeige können Sie nicht bearbeiten; Änderungen müssen Sie in der Symboltabelle bzw. in der Variablendeklarationstabelle vornehmen.

Nachfolgendes Bild zeigt Ihnen eine Symbolinformation in AWL.



Beim Ausdruck des Bausteins wird die aktuelle Bildschirmdarstellung mit Anweisungskommentar oder Symbolkommentar wiedergegeben.

8.2 Globale und lokale Symbole

Ein Symbol ermöglicht Ihnen, anstelle von absoluten Adressen mit einer aussagekräftigen Bezeichnung zu arbeiten. Mit der Kombination aus kurzen Symbolen und ausführlicheren Kommentaren können Sie sowohl die Anforderung einer effektiven Programmerstellung als auch einer guten Programmdokumentation erfüllen.

Man unterscheidet zwischen lokalen und globalen Symbolen:

	Globale Symbole	Lokale Symbole
Gültigkeitsbereich	<ul style="list-style-type: none"> • Sie gelten im gesamten Anwenderprogramm. • Sie können von allen Bausteinen benutzt werden. • Sie haben in allen Bausteinen die gleiche Bedeutung. • Die Bezeichnung muss im gesamten Anwenderprogramm eindeutig sein. 	<ul style="list-style-type: none"> • Sind nur in dem Baustein bekannt, in dem sie definiert wurden. • Die gleiche Bezeichnung kann in verschiedenen Bausteinen für unterschiedliche Zwecke verwendet werden.
Zulässige Zeichen	<ul style="list-style-type: none"> • Buchstaben, Ziffern, Sonderzeichen • Umlaute außer 0x00, 0xFF und Anführungszeichen • Symbol muss bei Verwendung von Sonderzeichen in Anführungs- und Schlusszeichen gesetzt werden 	<ul style="list-style-type: none"> • Buchstaben • Ziffern • Unterstrich (_)
Einsatz	<p>Sie können globale Symbole definieren für:</p> <ul style="list-style-type: none"> • E/A-Signale (E, EB, EW, ED, A, AB, AW, AD) • Peripherie-Eingänge / -Ausgänge (PE, PA) • Merker (M, MB, MW, MD) • Zeiten (T)/ Zähler (Z) • Codebausteine (OB, FB, FC, SFB, SFC) • Datenbausteine (DB) • Anwenderdefinierte Datentypen • Variablentabelle 	<p>Lokale Symbole können Sie definieren für:</p> <ul style="list-style-type: none"> • Bausteinparameter (Eingangs-, Ausgangs- und Durchgangparameter), • statische Daten eines Bausteins • temporäre Daten eines Bausteins
Ort der Definition	Symbole Tabelle	Variablendeklarationstabelle des Bausteins

8.3 Darstellung von globalen oder lokalen Symbolen

Sie können die globalen und lokalen Symbole im Anweisungsteil eines Programms folgendermaßen auseinanderhalten:

- Symbole aus der Symboltabelle (global) werden in Anführungszeichen ".." dargestellt.
- Symbole aus der Variablendeklarationstabelle des Bausteins (bausteinlokal) werden mit vorangestelltem "#" dargestellt.

Die Kennzeichnung in Anführungszeichen oder mit "#" müssen Sie nicht eingeben. Bei der Programmeingabe in KOP, FUP oder AWL wird die Kennzeichnung nach der Syntaxprüfung entsprechend ergänzt.

Wenn jedoch Verwechslungen denkbar sind, weil z. B. gleiche Symbole in Symboltabelle und Variablendeklaration verwendet wurden, müssen Sie, wenn Sie das globale Symbol verwenden wollen, dieses explizit kennzeichnen. Nicht gekennzeichnete Symbole werden in diesem Fall als bausteinlokale Variablen interpretiert.

Die Kennzeichnung globaler Symbole ist zudem erforderlich, wenn das Symbol Leerzeichen oder Sonderzeichen enthält.

Bei der Programmierung in einer AWL-Quelle gelten die gleichen Sonderzeichen und Regeln ihrer Verwendung. Die Kennzeichnungen bei der quellorientierten Eingabe werden hier nicht automatisch ergänzt, sind jedoch nur bei Verwechslungsgefahr erforderlich.

Hinweis

Mit dem Menübefehl **Ansicht > Symbolische Darstellung** schalten Sie zwischen der Anzeige der vereinbarten globalen Symbole und der zugehörigen absoluten Adresse um.

8.4 Einstellen des Operandenvorrangs (symbolisch/absolut)

Der Operandenvorrang hilft Ihnen bei Änderungen in der Symboltabelle, Änderung der Parameternamen von Datenbausteinen oder Funktionsbausteinen oder bei Änderung von Komponentennamen referenzierender UDTs oder Multiinstanzen, den Programmcode nach Wunsch nachzuziehen.

Stellen Sie den Operandenvorrang für die folgenden Änderungssituationen gezielt ein. Um die Vorteile des Operandenvorrangs nutzen zu können, sollte jeder Änderungsvorgang in sich abgeschlossen werden, bevor eine Änderung eines anderen Typs gestartet wird.

Markieren Sie im SIMATIC Manager den Bausteinordner und wählen Sie den Menübefehl **Bearbeiten > Objekteigenschaften**, um den Operandenvorrang einzustellen. Im Register "Operandenvorrang" können Sie die gewünschten Einstellungen wählen.

Zur Einstellung des optimalen Operandenvorrangs wird zwischen folgenden Änderungssituationen unterschieden:

- Korrektur einzelner Namen
- Tausch von Namen bzw. Zuordnungen
- Neue Symbole, Variable, Parameter oder Komponenten

Hinweis

Bitte beachten Sie, dass bei Bausteinaufrufen ("Call FC" oder "Call FB, DB") für den Codebaustein immer nur die absolute Bausteinnummer maßgeblich ist - auch bei eingestelltem symbolischen Operandenvorrang!

Korrektur einzelner Namen

Beispiele

In der Symboltabelle oder im Programm-/Bausteineditor muss ein Name wegen eines Schreibfehlers korrigiert werden. Dies gilt für alle Namen in der Symboltabelle, als auch für alle Namen von Parametern, Variablen oder Komponenten, die mit dem Programm-/Bausteineditor geändert werden können.

Einstellung des Operandenvorrangs



Änderungsverfolgung

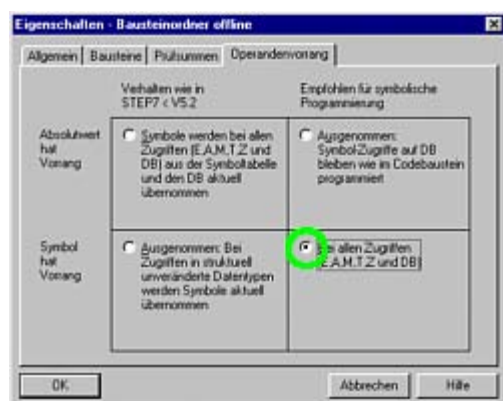
Markieren Sie im SIMATIC Manager den Bausteinordner und wählen Sie den Menübefehl **Bearbeiten > Bausteinkonsistenz prüfen**. Die Funktion "Bausteinkonsistenz prüfen" führt die notwendigen Änderungen in den einzelnen Bausteinen nach.

Tausch von Namen bzw. Zuordnungen

Beispiele

- Die Namen von bestehenden Zuordnungen in der Symboltabelle werden getauscht.
- Bestehenden Zuordnungen in der Symboltabelle werden neue Adressen zugewiesen.
- Variablen-, Parameter- oder Komponentennamen werden im Programm-/Bausteineditor getauscht.

Einstellung des Operandenvorrangs



Änderungsverfolgung

Markieren Sie im SIMATIC Manager den Bausteinordner und wählen Sie den Menübefehl **Bearbeiten > Bausteinkonsistenz prüfen**. Die Funktion "Bausteinkonsistenz prüfen" führt die notwendigen Änderungen in den einzelnen Bausteinen nach.

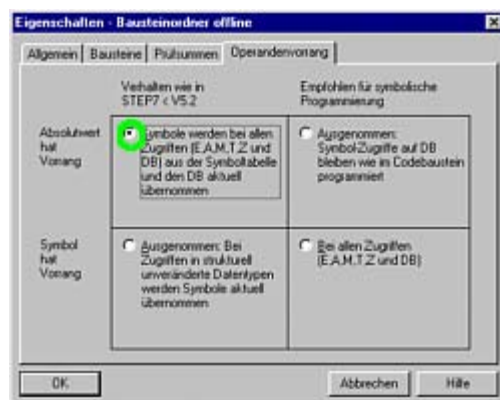
Neue Symbole, Variable, Parameter oder Komponenten

Beispiele

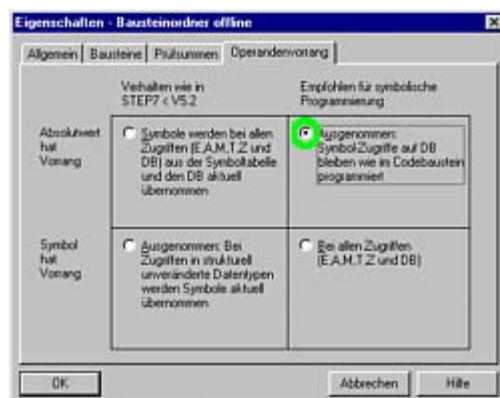
- Sie legen für im Programm verwendete Adressen neue Symbole an.
- Sie erweitern Datenbausteine, UDTs oder Funktionsbausteine um neue Variablen bzw. Parameter.

Einstellung des Operandenvorrangs

- Bei Änderungen in der Symboltabelle.



- Bei Änderungen im Programm-/Bausteineditor.



Änderungsverfolgung

Markieren Sie im SIMATIC Manager den Bausteinordner und wählen Sie den Menübefehl **Bearbeiten > Bausteinkonsistenz prüfen**. Die Funktion "Bausteinkonsistenz prüfen" führt die notwendigen Änderungen in den einzelnen Bausteinen nach.

8.5 Symboltabelle für globale Symbole

In der Symboltabelle definieren Sie globale Symbole.

Eine leere Symboltabelle (Objekt "Symbole") wird automatisch erzeugt, wenn Sie ein S7- oder M7-Programm anlegen.

Gültigkeitsbereich

Die Symboltabelle gilt genau für die Baugruppe, der Sie das Programm zuordnen. Wenn Sie in verschiedenen CPUs die gleichen Symbole verwenden wollen, müssen Sie selbst dafür sorgen, dass die entsprechenden Einträge in den zugehörigen Symboltabellen übereinstimmen (z. B. durch Kopieren).

8.5.1 Struktur und Bestandteile der Symboltabelle

Struktur der Symboltabelle

	Status	Ü	B	M	K	BK	Symbol ▲	Adresse	Datent	Kommentar
1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Automatik Ein	E 0.5	BOOL	für die Speicherfunktion ...
2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Automatikbetr...	A 4.2	BOOL	Ausgang mit speichernd...
3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Benzin	DB 1	FB ...	Daten für Benzinmotor
4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	BM_ausschal...	E 1.1	BOOL	Benzinmotor ausschalten
5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	BM_Drehzahl...	MWV 2	INT	tatsächliche Drehzahl fü...
6		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	BM_Ein	A 5.0	BOOL	Kommando für Benzinm...
7		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	BM_einschalten	E 1.0	BOOL	Benzinmotor einschalten
8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	BM_Lüfter_ein	A 5.2	BOOL	Kommando für Benzinm...

Zeilenspalte

	Wurden die Spalten für die "Speziellen Objekteigenschaften" ausgeblendet (Menübefehl Ansicht > Spalten Ü, B, M, K, BK abgewählt), so erscheint in der Zeilenspalte dieses Symbol, wenn in der betreffenden Zeile mindestens eine "Spezielle Objekteigenschaft" gesetzt wurde.
--	---

Spalte "Status"

	Der Symbolname oder Operand ist mit einem anderen Eintrag in der Symboltabelle identisch.
	Das Symbol ist noch unvollständig (es fehlt der Symbolname, bzw. die Adresse).

Spalten "Ü/B/M/K/BK"

Die Spalten Ü/B/M/K/BK zeigen an, ob einem Symbol spezielle Objekteigenschaften (Attribute) zugewiesen wurden:

- Ü bedeutet, für das Symbol wurden Fehlerdefinitionen zur Prozessdiagnose mit dem Optionspaket S7-PDIAG (V5) angelegt.
- B bedeutet, das Symbol kann mit WinCC bedient und beobachtet werden.
- M bedeutet, dem Symbol wurde eine symbolbezogene Meldung (SCAN) zugeordnet.
- K bedeutet, dem Symbol wurden Kommunikationseigenschaften zugeordnet.
- BK bedeutet, das Symbol kann im Programmeditor schnell und direkt beobachtet und gesteuert werden (Bedienen am Kontakt).

Klicken Sie auf die Kontrollkästchen, um die "speziellen Objekteigenschaften" zu aktivieren bzw. zu deaktivieren. Sie können die "speziellen Objekteigenschaften" auch über den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften** ändern.

Spalte "Symbol"

Der Name des Symbols darf maximal 24 Zeichen lang sein.

Für Operanden aus Datenbausteinen (DBD, DBW, DBB, DBX) können Sie in der Symboltabelle keine Namen vergeben. Deren Namen sind durch die Deklaration in den Datenbausteinen festgelegt.

Für Organisationsbausteine (OBs) und einige Systemfunktionsbausteine (SFBs) und Systemfunktionen (SFCs) existieren vordefinierte Symboltabelleneinträge, die Sie beim Bearbeiten der Symboltabelle Ihres S7-Programms in diese importieren können. Die Importdatei finden Sie im STEP 7-Verzeichnis unter ...\\S7data\\Symbol\\Symbol.sdf.

Spalte "Adresse"

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden.

Beispiel: Eingang E 12.1

Die Syntax der Adresse wird bei der Eingabe überprüft.

Spalte "Datentyp"

Sie können unter den verschiedenen Datentypen wählen, die Ihnen STEP 7 zur Verfügung stellt. Das Feld wird mit einem Default-Datentyp vorbesetzt, den Sie jedoch ändern können. Falls Ihre Änderung nicht zur Adresse passt oder die Syntax falsch ist, erscheint beim Verlassen des Feldes eine Fehlermeldung.

Spalte "Kommentar"

Allen Symbolen können Sie Kommentare zuordnen. Mit einer Kombination aus kurzen Symbolen und ausführlichen Kommentaren können Sie sowohl Anforderungen an eine gute Programmdokumentation als auch an eine effektive Programmerstellung erfüllen. Ein Kommentar kann maximal 80 Zeichen lang sein.

Umsetzung in C-Variable

Aus der Symboltabelle eines M7-Programms können Sie Symbole auswählen und in Verbindung mit der Optionsoftware ProC/C++ in korrespondierende C-Variablen umsetzen.


8.5.2 Zulässige Adressen und Datentypen in der Symboltabelle

Es ist nur eine Notation über die gesamte Symboltabelle möglich. Die Umschaltung zwischen Deutsch (vormals SIMATIC) und Englisch (vormals IEC) muss im SIMATIC Manager erfolgen, unter **Extras > Einstellungen** im Register "Sprache".


Englisch	Deutsch	Erläuterung:	Datentyp:	Adressbereich:
I	E	Eingangsbit	BOOL	0.0..65535.7
IB	EB	Eingangsbyte	BYTE, CHAR	0..65535
IW	EW	Eingangswort	WORD, INT, S5TIME, DATE	0..65534
ID	ED	Eingangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
Q	A	Ausgangsbit	BOOL	0.0..65535.7
QB	AB	Ausgangsbyte	BYTE, CHAR	0..65535
QW	AW	Ausgangswort	WORD, INT, S5TIME, DATE	0..65534
QD	AD	Ausgangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
M	M	Merkerbit	BOOL	0.0..65535.7
MB	MB	Merkerbyte	BYTE, CHAR	0..65535
MW	MW	Merkerwort	WORD, INT, S5TIME, DATE	0..65534
MD	MD	Merkerdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
PIB	PEB	Peripherieeingangs- byte	BYTE, CHAR	0..65535
PQB	PAB	Peripherieausgangs- byte	BYTE, CHAR	0..65535
PIW	PEW	Peripherieeingangs- wort	WORD, INT, S5TIME, DATE	0..65534
PQW	PAW	Peripherieausgangs- wort	WORD, INT, S5TIME, DATE	0..65534
PID	PED	Peripherieeingangs- doppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
PQD	PAD	Peripherieausgangs- doppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
T	T	Zeit	TIMER	0..65535
C	Z	Zähler	COUNTER	0..65535
FB	FB	Funktionsbaustein	FB	0..65535
OB	OB	Organisationsbaustein	OB	1..65535
DB	DB	Datenbaustein	DB, FB, SFB, UDT	1..65535
FC	FC	Funktion	FC	0..65535
SFB	SFB	Systemfunktions- baustein	SFB	0..65535
SFC	SFC	Systemfunktion	SFC	0..65535
VAT	VAT	Variablentabelle		0..65535
UDT	UDT	Anwenderdefinierter Datentyp	UDT	0..65535

8.5.3 Unvollständige und mehrdeutige Symbole in der Symboltabelle

Unvollständige Symbole

Sie haben die Möglichkeit, auch unvollständige Symbole abzuspeichern. Dadurch können Sie beispielsweise zunächst nur den Namen festlegen und die Adressangaben zu einem späteren Zeitpunkt nachtragen. Insbesondere können Sie die Arbeiten an der Symboltabelle zu einem beliebigen Zeitpunkt unterbrechen und den Zwischenstand speichern. Unvollständige Symbole werden in der Spalte "Status" mit dem Symbol  gekennzeichnet. Um das Symbol bei der Software-Erstellung (ohne Fehlermeldung) benutzen zu können, müssen jedoch die Bezeichnung des Symbols, die Adresse und der Datentyp eingetragen sein.

Entstehung mehrdeutiger Symbole

Mehrdeutige Symbole entstehen, wenn Sie ein Symbol in die Symboltabelle einfügen, dessen Name (Symbol) und/oder Adresse bereits in einem anderen Symbol vorhanden sind. Dadurch werden sowohl das ursprünglich vorhandene und das eingefügte Symbol mehrdeutig. Dieser Zustand wird in der Spalte "Status" mit dem Symbol  gekennzeichnet.

Dieser Fall tritt z. B. ein, wenn Sie ein Symbol kopieren und einfügen, um den Eintrag in der Kopie danach leicht zu modifizieren.

Kennzeichnung mehrdeutiger Symbole

In der Symboltabelle sind mehrdeutige Symbole durch grafische Hervorhebung (Farbe, Schriftart) gekennzeichnet. Die veränderte Darstellung weist Sie auf eine erforderliche Nachbearbeitung hin. Sie können alle Symbole anzeigen oder die Anzeige so filtern, dass nur eindeutige oder nur mehrdeutige Symbole angezeigt werden.

Beseitigung der Mehrdeutigkeit

Ein mehrdeutiges Symbol wird eindeutig, wenn Sie die Komponente (Name und/oder Adresse) ändern, die die Mehrdeutigkeit verursacht hat. Sind zwei Symbole mehrdeutig, und Sie machen ein Symbol eindeutig, so wird das andere ebenfalls wieder eindeutig.

8.6 Eingabemöglichkeiten von globalen Symbolen

Für die Eingaben von Symbolen, die bei der späteren Programmierung weiterverwendet werden, gibt es drei Möglichkeiten:

- Eingabe über Dialogfeld
Sie können in einem Fenster zur Programmeingabe ein Dialogfeld öffnen und darin ein neues Symbol definieren. Das Vorgehen eignet sich zur Definition einzelner Symbole, beispielsweise wenn Sie während der Programmierung feststellen, dass ein Symbol fehlt oder korrigiert werden muss. Sie ersparen sich so das Anzeigen der Symboltabelle.
- Eingabe direkt in Symboltabelle
Sie können Symbole und ihre zugeordneten Adressen direkt in eine "Symboltabelle" eintragen. Dieses Verfahren empfiehlt sich zur Eingabe mehrerer Symbole und zur Ersterstellung der Symboltabelle, weil Sie die bereits definierten Symbole am Bildschirm angezeigt bekommen und so leichter die Übersicht behalten.
- Importieren von Symboltabellen aus anderen Tabelleneditoren
Sie können die Daten für die Symboltabelle auch mit dem von ihnen bevorzugten Tabelleneditor (z. B. Microsoft Excel) erstellen und die erzeugte Datei dann in die Symboltabelle importieren.

8.6.1 Allgemeine Hinweise zur Eingabe von Symbolen

Um neue Symbole in die Symboltabelle einzutragen, navigieren Sie in die erste leere Zeile der Tabelle und füllen dann die Felder aus. Neue Zeilen können Sie mit dem Menübefehl **Einfügen > Symbol** in die Symboltabelle vor der aktuellen Zeile einfügen. Wenn die Zeile vor der Cursor-Position bereits einen Operanden enthält, werden Sie beim Einfügen von neuen Symbolen unterstützt durch eine Vorbelegung der Spalten "Adresse" und "Datentyp". Die Adresse wird abgeleitet aus der vorhergehenden Zeile, als Datentyp wird der Default-Datentyp eingetragen.

Vorhandene Einträge können Sie mit Befehlen im Menü "Bearbeiten" kopieren und anschließend ändern. Speichern und schließen Sie dann die Symboltabelle. Sie können auch Symbole speichern, die noch nicht vollständig definiert sind.

Bei der Eingabe der Symbole sind folgende Besonderheiten zu beachten:

Spalte	Hinweis
Symbol	Der Name muss innerhalb der gesamten Symboltabelle eindeutig sein. Wenn Sie die Eingabe in diesem Feld übernehmen oder das Feld verlassen, wird ein mehrdeutiges Symbol gekennzeichnet. Das Symbol darf maximal 24 Zeichen lang sein. Es sind keine Anführungszeichen " zugelassen.
Adresse	Wenn Sie die Eingabe in diesem Feld übernehmen oder das Feld verlassen, wird geprüft, ob die eingegebene Adresse zulässig ist.
Datentyp	Nach Eingabe der Adresse wird dieses Feld mit einer Default-Einstellung vorbelegt. Wenn Sie diese Einstellung ändern, wird geprüft, ob der neue Datentyp zur Adresse passt.
Kommentar	Hier können Sie Anmerkungen eingeben, um die Funktionen der Symbole kurz zu erklären (max. 80 Zeichen). Die Angabe des Kommentars ist optional.

8.6.2 Eingeben einzelner globaler Symbole im Dialog

Die hier beschriebene Vorgehensweise zeigt Ihnen, wie Sie beim Programmieren von Bausteinen Symbole über Dialogfelder ändern oder neu definieren können, ohne dass dabei die Symboltabelle angezeigt werden muss.

Die Vorgehensweise ist nützlich, wenn Sie nur ein einzelnes Symbol bearbeiten wollen. Um mehrere Symbole zu ändern, sollten Sie die Symboltabelle öffnen und direkt in der Symboltabelle arbeiten.

Aktivieren der Anzeige von Symbolen im Baustein

Aktivieren Sie bei geöffnetem Baustein die Anzeige der Symbole im Bausteinfenster mit dem Menübefehl **Ansicht > Symbolische Darstellung**. Ein Häkchen vor dem Menübefehl kennzeichnet, dass die symbolische Darstellung aktiv ist.

Definieren von Symbolen bei der Programmeingabe

1. Stellen Sie sicher, dass im Bausteinfenster die symbolische Darstellung eingeschaltet ist (Menübefehl **Ansicht > Symbolische Darstellung**).
2. Markieren Sie im Anweisungsteil Ihres Programms die absolute Adresse, der Sie ein Symbol zuweisen wollen.
3. Wählen Sie den Menübefehl **Bearbeiten > Symbol**.
4. Füllen Sie das angezeigte Dialogfeld aus. Tragen Sie insbesondere ein Symbol ein und schließen Sie den Dialog ab.

Das definierte Symbol wird in die Symboltabelle eingetragen. Angaben, die zu mehrdeutigen Symbolen führen würden, werden mit Fehlermeldung abgewiesen.

Editieren in der Symboltabelle

Über den Menübefehl **Extras > Symboltabelle** können Sie die Symboltabelle zum Bearbeiten öffnen.

8.6.3 Eingeben mehrerer globaler Symbole in der Symboltabelle

Symboltabelle öffnen

Sie haben mehrere Möglichkeiten zum Öffnen einer Symboltabelle:

- Doppelklick auf die Symboltabelle im Projektfenster,
- Markieren der Symboltabelle im Projektfenster und Menübefehl **Bearbeiten > Objekt öffnen**.

Die Symboltabelle zum aktuellen Programm wird in einem eigenen Fenster angezeigt. Nun können Sie Symbole erzeugen oder ändern. Beim ersten Öffnen nach dem Erzeugen der Symboltabelle ist die Symboltabelle noch leer.

Eingeben von Symbolen

Um neue Symbole in die Symboltabelle einzutragen, navigieren Sie in die erste leere Zeile der Tabelle und füllen dann die Felder aus. Neue Leerzeilen können Sie mit dem Menübefehl **Einfügen > Symbol** in die Symboltabelle vor der aktuellen Zeile einfügen. Vorhandene Einträge können Sie mit Befehlen im Menü "Bearbeiten" kopieren und anschließend ändern. Speichern und schließen Sie dann die Symboltabelle. Sie können auch Symbole speichern, die noch nicht vollständig definiert sind.

Sortieren von Symbolen

Die Datensätze in der Symboltabelle lassen sich alphabetisch nach Symbol, Adresse, Datentyp oder Kommentar sortieren.

Die Sortierung kann z. B. in dem Dialogfeld geändert werden, das Sie mit dem Menübefehl **Ansicht > Sortieren...** aufrufen können.

Filtern von Symbolen

Mit einem Filter wählen Sie aus allen Datensätzen einer Symboltabelle eine Teilmenge aus.

Über den Menübefehl **Ansicht > Filtern** öffnen Sie das Dialogfeld "Filtern".

Hier können Sie Kriterien definieren, die die Datensätze erfüllen müssen, um herausgefiltert zu werden. Sie können filtern nach

- Namen, Adressen, Datentypen, Kommentaren
- Symbolen mit BuB-Atribut, Symbolen mit Kommunikationseigenschaften, Symbolen für Binärvariable bei Meldungen (Merker oder Prozesseingang)
- Symbolen mit Status "gültig", "ungültig (mehrdeutig, unvollständig)"

Die einzelnen Kriterien sind mit UND verknüpft. Die herausgefilterten Datensätze beginnen mit den angegebenen Zeichenfolgen.

Wenn Sie mehr über die Möglichkeiten in dem Dialogfeld "Filtern" wissen möchten, öffnen Sie die Online-Hilfe durch Betätigen der Funktionstaste F1.

8.6.4 Groß-/Kleinschreibung bei Symbolen

Keine Unterscheidung mehr zwischen Groß- und Kleinschreibung

Bisher konnten in STEP 7 Symbole definiert werden, die sich nur durch die Groß- und Kleinschreibung einzelner Zeichen unterschieden. Ab STEP 7 V4.02 wurde dieses Verhalten geändert. Eine Unterscheidung der Symbole nach Groß- und Kleinschreibung ist ab dieser Version nicht mehr möglich.

Mit dieser Änderung wurde den Wünschen unserer Kunden Rechnung getragen, da damit mögliche Fehlerquellen in einem Programm deutlich reduziert werden. Mit der jetzt vorgenommenen Einschränkung bei der Symboldefinition werden ebenfalls die Ziele der PLCopen für die Definition eines Standards für übertragbare Programme unterstützt.

Eine unterschiedliche Symboldefinition, die sich nur aus der verwendeten Groß- oder Kleinschrift ergibt, wird zukünftig nicht mehr unterstützt. Bisher war z. B. folgende Definition in der Symboltabelle möglich:

Motor1 = E 0.0

motor1 = E 1.0

Die Symbole wurden aufgrund der Schreibweise (Groß-/Kleinschreibung) des ersten Buchstabens unterschieden. Bei dieser Art der Unterscheidung besteht eine große Verwechslungsgefahr. Mit der jetzigen Definition wurde diese mögliche Fehlerquelle ausgeschlossen.

Auswirkung auf bestehende Programme

Wenn Sie bei der Symboldefinition bisher dieses Kriterium zur Unterscheidung genutzt haben, kann es mit der neuen Definition zu Konflikten kommen wenn sich:

- Symbole **nur** durch Groß-/Kleinschreibung unterscheiden
- Parameter **nur** durch Groß-/Kleinschreibung unterscheiden
- Symbole **nur** durch Groß-/Kleinschreibung von Parametern unterscheiden

Alle drei Fälle können jedoch wie im Anschluss beschrieben analysiert und bereinigt werden.

Symbole, die sich nur durch Groß-/Kleinschreibung unterscheiden

Konflikt:

Falls die Symboltabelle noch nicht mit der aktuellen Softwareversion bearbeitet wurde, so wird beim Übersetzen von Quelldateien das oberste der mehrdeutigen Symbole aus der Symboltabelle verwendet.

Wurde die Symboltabelle bereits bearbeitet, so sind solche Symbole ungültig, d. h. beim Öffnen von Bausteinen wird keine Symbolik angezeigt und Quelldateien, die diese Symbole benutzen, sind nicht mehr fehlerfrei übersetzbar.

Abhilfe:

Überprüfen Sie ihre Symboltabelle auf Konfliktfälle, indem Sie die Symboltabelle öffnen und neu sichern. Durch diese Aktion werden die mehrdeutigen Symbole erkannt. Sie können sich die mehrdeutigen Symbole nun über den Filter "mehrdeutige Symbole" anzeigen lassen und diese korrigieren. Korrigieren Sie anschließend die Quelldateien mit Konflikten. Bei Bausteinen ist keine weitere Änderung nötig, da beim Öffnen automatisch die aktuelle (nun konfliktfreie) Symboltabelle verwendet bzw. angezeigt wird.

Parameter, die sich nur durch Groß-/Kleinschreibung unterscheiden

Konflikt:

Quelldateien, die solche Schnittstellen beinhalten, sind nicht mehr fehlerfrei übersetzbar. Bausteine mit solchen Schnittstellen können zwar noch geöffnet werden, jedoch ist kein Zugriff auf den zweiten dieser Parameter mehr möglich. Zugriffe auf den zweiten der Parameter werden beim Speichern automatisch auf den ersten Parameter umgesetzt.

Abhilfe:

Zur Überprüfung, welche Bausteine solche Konflikte enthalten, empfiehlt es sich, mit der Funktion "Quelle generieren" für alle Bausteine eines Programms eine Quelldatei zu generieren. Wenn beim Versuch, die erzeugte Quelldatei neu zu übersetzen, Fehler auftreten, so liegt ein Konfliktfall vor.

Korrigieren Sie Ihre Quelldateien indem Sie die Parameter z. B. mittels der Funktion Suchen/Ersetzen eindeutig machen und übersetzen Sie diese neu.

Symbole, die sich nur durch Groß-/Kleinschreibung von Parametern unterscheiden

Konflikt:

Unterscheiden sich globale und lokale Symbole einer Quelldatei nur durch Groß-/Kleinschreibung und wurden keine Einleitungszeichen zur Kennzeichnung von globaler ("Symbolname") bzw. lokaler (#Symbolname) Symbolik verwendet, so wird bei der Compilierung immer das lokale Symbol verwendet. Das führt zu einem veränderten Maschinencode.

Abhilfe:

In diesem Fall empfiehlt es sich, aus allen Bausteinen eine Quelle neu zu generieren. Dadurch werden automatisch die lokalen und globalen Zugriffe mit den entsprechenden Einleitungszeichen versehen und bei anschließenden Übersetzungsvorgängen korrekt behandelt.

8.6.5 Exportieren und Importieren von Symboltabellen

Sie können die angezeigte Symboltabelle in eine Textdatei exportieren, um sie beispielsweise mit einem beliebigen Texteditor weiterzubearbeiten.

Sie können Tabellen, die Sie mit einem anderen Werkzeug erzeugt haben, in Ihre Symboltabelle importieren und hier weiterbearbeiten. Die Import-Funktion lässt sich beispielsweise benutzen, um unter STEP5/ST erzeugte Zuordnungslisten nach Konvertierung in die Symboltabelle aufzunehmen.

Zur Auswahl stehen die Dateiformate *.SDF, *.ASC, *.DIF und *.SEQ.

Regeln zum Exportieren

Sie können die gesamte Symboltabelle, eine durch Filter begrenzte Teilmenge oder in der Tabellendarstellung markierte Zeilen exportieren.

Die Eigenschaften von Symbolen, die Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften...** einstellen können, werden nicht exportiert.

Regeln zum Importieren

- Für häufig verwendete Systemfunktionsbausteine (SFBs), Systemfunktionen (SFCs) und Organisationsbausteine (OBs) sind in der Datei ...\\S7DATA\\SYMBOL\\SYMBOL.SDF Symbole vordefiniert, die Sie bei Bedarf importieren können.
- Eigenschaften von Symbolen, die Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften...** einstellen können, werden beim Exportieren und Importieren nicht berücksichtigt.

8.6.6 Dateiformate für den Import/Export einer Symboltabelle

Folgende Dateiformate können in die Symboltabelle importiert bzw. aus ihr exportiert werden:

- ASCII-Dateiformat (ASC)
- Data Interchange Format (DIF)
DIF-Dateien können Sie in Microsoft Excel öffnen, bearbeiten und wieder speichern.
- System Data Format (SDF)
SDF-Dateien können Sie in Microsoft Access öffnen, bearbeiten und wieder speichern.
 - Benutzen Sie für den Import und Export von Daten der Microsoft-Applikation ACCESS das SDF-Format.
 - Wählen Sie in ACCESS als Dateiformat "Text (mit Trennzeichen)".
 - Benutzen Sie als Texttrennzeichen das doppelte Hochkomma ("").
 - Benutzen Sie als Feldtrennzeichen das Komma (,).
- Zuordnungsliste (SEQ)
Achtung: Beim Exportieren der Symboltabelle in eine Datei vom Typ .SEQ werden Kommentare, die länger als 40 Zeichen sind, nach dem 40. Zeichen abgeschnitten !

ASCII-Dateiformat (ASC)

Dateityp:	*.ASC
Aufbau:	Datensatzlänge, Trennzeichen Komma, Datensatz
Beispiel:	126,grün_phase_fußg T 2 TIMER Dauer der Grünphase für Fußgänger 126,rot_fußg A 0.0 BOOL Rot für Fußgänger

Data Interchange Format (DIF)

Dateityp:	*.DIF
Aufbau:	Eine DIF-Datei gliedert sich in Dateikopf (Header) und Daten:

Header	TABLE	Beginn einer DIF-Datei
	0,1	
	"<Titel>"	Kommentierende Zeichenkette
	VECTORS	Anzahl der Datensätze in der Datei
	0,<Datensatzzahl>	
	""	
	TUPLES	Anzahl der Datenfelder eines Datensatzes
	0,<Spaltenanzahl>	
	""	
	DATA	Kennung für Header-Ende und Datenbeginn
	0,0	
	""	
Daten (je Datensatz)	<Typ>,<numerischer Wert>	Kennung für den Datentyp, numerischer Wert
	<Zeichenkette>	Alphanummerischer Teil oder
	V	Falls alphanummerischer Teil unbenutzt

Header: Der Dateikopf muss die Satztypen TABLE, VECTORS, TUPLES und DATA in der angegebenen Reihenfolge enthalten. Vor DATA können in DIF-Dateien noch weitere, optionale Satztypen stehen. Diese werden vom Symbol Editor jedoch ignoriert.

Daten: Im Datenteil besteht jeder Eintrag aus 3 Teilen: der Kennung für den Typ, einem numerischen Wert und einem alphanummerischen Teil.

DIF-Dateien können Sie in Microsoft Excel öffnen, bearbeiten und wieder speichern. Sie sollten dabei allerdings auf die Verwendung von deutschen Umlauten und anderen sprachabhängigen Sonderzeichen verzichten.

System Data Format (SDF)

Dateityp:	*.SDF
Aufbau:	Zeichenketten in Hochkommata, Teile durch Kommata getrennt
Beispiel:	"grün_phase_fußg","T 2","TIMER","Dauer der Grünphase für Fußgänger" "rot_fußg","A 0.0","BOOL","Rot für Fußgänger"

Zum Öffnen einer SDF-Datei in Microsoft Access wählen Sie das Dateiformat "Text (mit Trennzeichen)". Geben Sie als Texttrennzeichen das doppelte Hochkomma (") und als Feldtrennzeichen das Komma (,) an.

Zuordnungsliste (SEQ)

Dateityp:	*.SEQ
Aufbau:	TAB Adresse TAB Symbol TAB Kommentar CR
Beispiel:	T 2 grün_phase_fußg Dauer der Grünphase für Fußgänger A 0.0 rot_fußg Rot für Fußgänger

TAB steht für das Tabulatorzeichen (09H),
CR für einen Zeilenumbruch mit der RETURN-Taste (0DH).

8.6.7 Bearbeiten von Bereichen in Symboltabellen

Ab der Version V5.3 von STEP 7 ist es möglich, zusammenhängende Bereiche einer Symboltabelle zu markieren und zu bearbeiten. Sie können diese Bereiche z.B. in eine andere Symboltabelle kopieren, ausschneiden, einfügen oder löschen.

So können Daten aus einer Symboltabelle komfortabel in eine andere übertragen und Symboltabellen einfacher aktualisiert werden.

Markierbare Bereiche:

Sie können komplette Zeilen markieren, sobald Sie diese in der ersten Spalte anklicken. Wenn Sie alle Felder von der Spalte "Status" bis zur Spalte "Kommentar" markieren, so gilt dies ebenfalls als Zeilenselektion.

- Sie können einzelne oder mehrere zusammenhängende Felder als Bereiche markieren. Um diesen markierten Bereich bearbeiten zu können, müssen alle Felder zu den Spalten "Symbol", "Adresse", "Datentyp" und "Kommentar" gehören. Falls Ihre Auswahl ungültig ist, sind die Menübefehle zum Bearbeiten nicht aktiv.
- Die Spalten Ü, B, M, K, BK enthalten die Speziellen Objekteigenschaften zu den jeweiligen Symbolen und werden nur dann mitkopiert, wenn im Dialogfeld "Einstellungen", aufrufbar über den Menübefehl **Extras > Einstellungen**, das Kontrollkästchen "Spezielle Objekteigenschaften mitkopieren" aktiviert ist und wenn Sie komplette Zeilen als Bereich markiert haben.
- Das Kopieren der Inhalte der Spalten Ü, B, M, K, BK ist unabhängig davon, ob diese Spalten gerade angezeigt werden oder nicht. Die Ansicht ist über den Menübefehl **Ansicht > Spalten Ü, B, M, K, BK** umschaltbar.

Gehen Sie folgendermaßen vor:

1. Markieren Sie den gewünschten Bereich in der Symboltabelle, den Sie bearbeiten wollen:
 - Klicken Sie dazu mit der **Maus** in die gewünschte Startzelle und bewegen Sie die Maus bei gedrückter linker Maustaste über den Bereich, den Sie markieren wollen oder
 - Markieren Sie den gewünschten Bereich über die **Tastatur** durch Drücken der Tast Shift + Cursorstasten.
2. Der markierte Bereich wird in der Selektionsfarbe dargestellt. Die zuerst selektierte Zelle der Symboltabelle wird mit einem Rahmen markiert.
3. Bearbeiten Sie den markierten Bereich wie gewünscht.

9 Anlegen von Bausteinen und Bibliotheken

9.1 Auswahl der Erstellmethode

Je nach Programmiersprache, die Sie für die Programmerstellung benutzen, können Sie Ihr Programm inkrementell und/oder quellorientiert eingeben.

Inkrementelle Editoren für die Programmiersprachen KOP, FUP, AWL und S7-GRAPH

Mit den inkrementellen Editoren für KOP, FUP, AWL und S7-GRAPH erstellen Sie Bausteine, die im Anwenderprogramm abgelegt werden. Entscheiden Sie sich für die inkrementelle Eingabe, wenn Sie Ihre Eingaben sofort kontrollieren lassen möchten. Diese Eingabeart kommt auch Programmieranfängern entgegen. Bei der inkrementellen Eingabe wird jede Zeile bzw. jedes Element sofort auf syntaktische Korrektheit überprüft. Eventuell vorhandene Fehler werden angezeigt und müssen vor dem Beenden der Eingabe beseitigt werden. Syntaktisch korrekte Eingaben werden automatisch übersetzt und im Anwenderprogramm abgelegt.

Verwendete Symbole müssen vor dem Editieren der Anweisungen definiert sein. Falls bestimmte Symbole nicht vorhanden sind, kann der Baustein nicht vollständig übersetzt werden; dieser "inkonsistente Zwischenstand" kann jedoch abgespeichert werden.

Quellorientierte (Text-) Editoren für die Programmiersprachen AWL, S7-SCL oder S7-HiGraph

In quellorientierten Editoren erstellen Sie **Quellen**, die anschließend in Bausteine übersetzt werden.

Wählen Sie die quellorientierte Eingabe, um ein Programm schnell einzugeben oder zu erfassen.

Bei der quellorientierten Eingabe wird das Programm bzw. ein Baustein in einer Textdatei editiert und die Textdatei anschließend kompiliert.

Die Textdateien (Quellen) werden im Quellordner Ihres S7-Programms z. B. als **AWL-Quelle** bzw. **SCL-Quelle** abgelegt. Eine Quelldatei kann dabei den Code für einen oder mehrere Bausteine enthalten. Mit den Texteditoren für AWL bzw. SCL können Sie Code für **OBs, FBs, FCs, DBs und UDTs** (anwenderdefinierte Datentypen), also auch für ein komplettes Anwenderprogramm erstellen. Das gesamte Programm einer CPU (d. h. alle Bausteine) kann in einer einzigen Textdatei stehen.

Durch Übersetzung der Quelldatei werden die entsprechenden Bausteine erzeugt und im Anwenderprogramm abgelegt. Verwendete Symbole müssen vor dem Übersetzen definiert sein. Eventuelle Fehler werden erst bei der Übersetzung durch den entsprechenden Compiler gemeldet.

Für die Übersetzung ist es wichtig, dass Sie die vorgeschriebene Syntax der Programmiersprache berücksichtigen. Eine Syntaxprüfung erfolgt erst durch Ihren Befehl zur Konsistenzprüfung oder bei der Übersetzung in Bausteine.

9.2 Auswahl der Programmiersprache

Festlegen der Programmiersprache des Editors

In welcher Programmiersprache und welchem Editor Sie einen Baustein bzw. eine Quelldatei erstellen wollen, legen Sie bei der Erzeugung des Bausteins bzw. der Quelle in den Objekteigenschaften fest. Diesem Eintrag gemäß wird der entsprechende Editor beim Öffnen des Bausteins/der Quelldatei gestartet.

Spracheditor starten

Den jeweiligen Spracheditor starten Sie im SIMATIC Manager durch einen Doppelklick auf das entsprechende Objekt (Baustein, Quelldatei, etc.), über den Menübefehl **Bearbeiten > Objekt öffnen** oder über das entsprechende Symbol in der Funktionsleiste.

Zur Erstellung des S7-Programms stehen Ihnen die in der Tabelle angegebenen Programmiersprachen zur Verfügung. Die Programmiersprachen KOP, FUP und AWL sind in der Basissoftware STEP 7 enthalten. Die übrigen Programmiersprachen können Sie als Optionssoftware erwerben.

Damit haben Sie die Auswahl zwischen unterschiedlichen Programmierphilosophien (Kontaktplan, Funktionsplan, Anweisungsliste, Hochsprache, Ablaufsteuerung oder Zustandsgraph) und zwischen textueller oder grafischer Programmierung.

Durch die Wahl der Programmiersprache sind auch die zulässigen Eingabemethoden (•) festgelegt.

Programmiersprache	Anwendergruppe	Anwendungsfall	Inkrementelle Eingabe	Quellorientierte Eingabe	Baustein aus der CPU rückdokumentierbar
Anweisungsliste AWL	Anwender, die maschinennah programmieren wollen.	Laufzeit- und speicherplatz-optimierte Programme	•	•	•
Kontaktplan KOP	Anwender, die Umgang mit Stromlaufplänen gewohnt sind.	Programmierung von Verknüpfungssteuerungen	•	–	•
Funktionsplan FUP	Anwender, die mit den logischen Boxen der Booleschen Algebra vertraut sind.	Programmierung von Verknüpfungssteuerungen	•	–	•
F-KOP, F-FUP Optionspaket	Anwender, die mit den Programmiersprachen KOP und FUP vertraut sind.	Programmierung von Sicherheitsprogrammen für F-Systeme	•	–	•

Programmiersprache	Anwendergruppe	Anwendungsfall	Inkrementelle Eingabe	Quellorientierte Eingabe	Baustein aus der CPU rückdokumentierbar
SCL (Structured Control Language) Optionspaket	Anwender, die in Hochsprachen wie PASCAL oder C programmiert haben.	Programmieren von Datenverarbeitungsaufgaben	–	•	–
GRAPH Optionspaket	Anwender, die technologieorientiert ohne tiefere Programmier-/SPS-Kenntnisse arbeiten wollen.	komfortable Beschreibung sequenzieller Vorgänge	•	–	•
HiGraph Optionspaket	Anwender, die technologieorientiert ohne tiefere Programmier-/SPS-Kenntnisse arbeiten wollen.	komfortable Beschreibung asynchroner, nichtsequenzieller Vorgänge.	–	•	–
CFC Optionspaket	Anwender, die technologieorientiert ohne tiefere Programmier-/SPS-Kenntnisse arbeiten wollen.	Beschreibung kontinuierlicher Vorgänge.	–	–	–

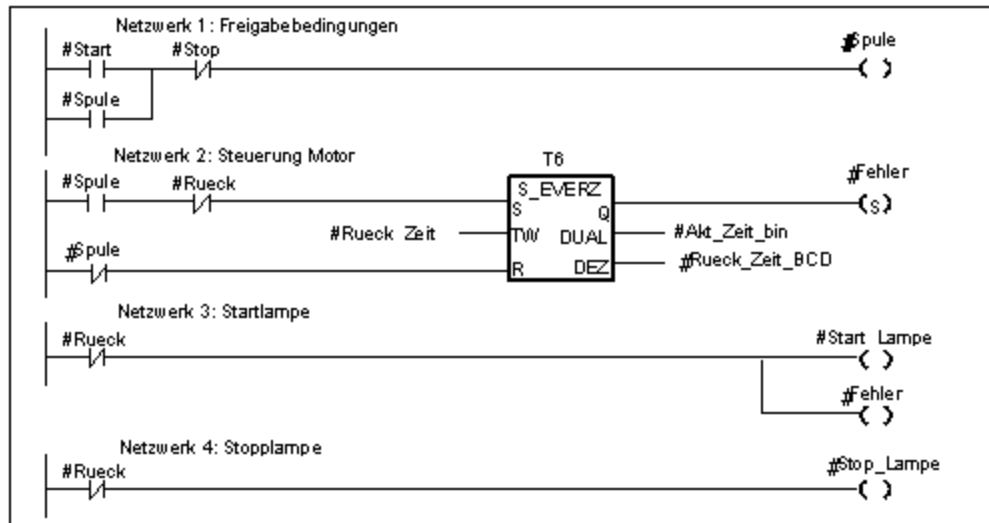
Bei fehlerfreien Bausteinen können Sie zwischen den Darstellungen des Bausteins in den Programmiersprachen KOP, FUP und AWL wechseln. Dabei werden Programmteile, die in der Zielsprache nicht darstellbar sind, in AWL dargestellt.

Bausteine können Sie aus Quelldateien in AWL erzeugen und hieraus auch wieder Quelldateien generieren.

9.2.1 Programmiersprache KOP (Kontaktplan)

Die Darstellung in der grafischen Programmiersprache KOP (Kontaktplan) ist Stromlaufplänen nachempfunden. Die Elemente eines Stromlaufplans, wie z. B. Öffner und Schließer, werden zu Netzwerken zusammengefügt. Ein oder mehrere Netzwerke ergeben den gesamten Anweisungsteil eines Codebausteins.

Beispiel für Netzwerke in KOP:



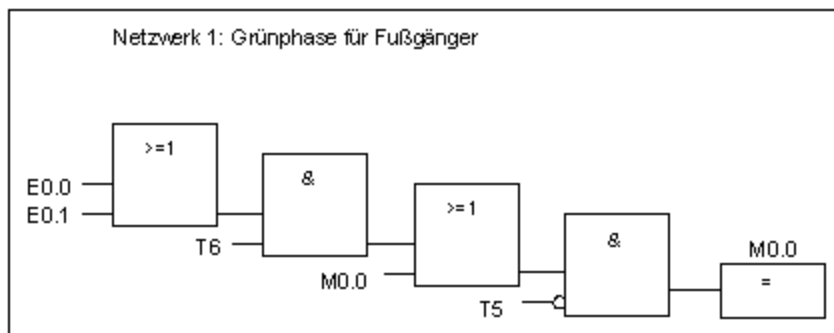
Die Programmiersprache KOP ist in der Basissoftware STEP 7 enthalten. Die Programmerstellung in KOP erfolgt mit einem inkrementellen Editor.

9.2.2 Programmiersprache FUP (Funktionsplan)

Die Programmiersprache FUP (Funktionsplan) benutzt zur Darstellung der Logik die von der Booleschen Algebra bekannten grafischen Logiksymbole. Außerdem können komplexe Funktionen, wie z. B. mathematische Funktionen, direkt in Verbindung mit den logischen Boxen dargestellt werden.

Die Programmiersprache FUP ist in der Basissoftware STEP 7 enthalten.

Beispiel für ein Netzwerk in FUP:



Die Programmerstellung in FUP erfolgt mit einem inkrementellen Editor.

9.2.3 Programmiersprache AWL (Anweisungsliste)

Die Programmiersprache AWL (Anweisungsliste) ist eine maschinennahe textuelle Sprache. Die einzelnen Anweisungen entsprechen den Arbeitsschritten, mit denen die CPU die Abarbeitung des Programms durchführt. Mehrere Anweisungen können zu Netzwerken zusammengefügt werden.

Beispiel für Netzwerke in AWL:

```
Netzwerk 1: Steuerung Abflussventil
U(
O
O #Spule
)
UN #Schliessen
= #Spule

Netzwerk 2: Anzeige "Ventil offen"
U #Spule
= #Anz_offen

Netzwerk 3: Anzeige "Ventil geschlossen"
UN #Spule
= #Anz_geschl
```

Die Programmiersprache AWL ist in der Basissoftware STEP 7 enthalten. Sie können in dieser Programmiersprache mit inkrementellen Editoren S7-Bausteine bearbeiten oder Ihr Programm mit einem quellorientierten Editor in einer AWL-Quelle erstellen und anschließend in Bausteine übersetzen.

9.2.4 Programmiersprache S7-SCL

Die als Optionssoftware erhältliche Programmiersprache SCL (Structured Control Language) ist eine textuelle Hochsprache und entspricht in ihrer Sprachdefinition im wesentlichen der IEC 1131-3. Die PASCAL-artige Sprache vereinfacht durch ihre Hochsprachenbefehle im Gegensatz zu AWL z. B. die Programmierung von Schleifen und bedingten Verzweigungen. SCL eignet sich daher u. a. für Formelberechnungen, komplexe Optimierungsalgorithmen oder die Verwaltung größerer Datenmengen.

Die Programmerstellung in S7-SCL erfolgt mit einem quellorientierten Editor in einer SCL-Quelle.

Beispiel:

```
FUNCTION_BLOCK FB 20
VAR_INPUT
ENDWERT : INT;
END_VAR
VAR_IN_OUT
IQ1 : REAL;
END_VAR
VAR
INDEX : INT;
END_VAR

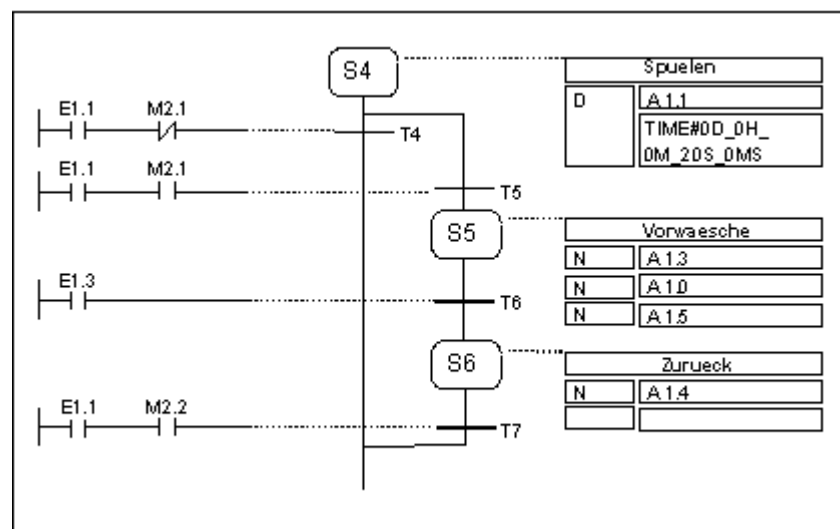
BEGIN
CONTROL := FALSE;
FOR INDEX := 1 TO ENDWERT DO
    IQ1 := IQ1 * 2;
    IF IQ1 > 10000 THEN
        CONTROL = TRUE
    END_IF;
END_FOR;
END_FUNCTION_BLOCK
```

9.2.5 Programmiersprache S7-GRAPH (Ablaufsteuerung)

Die als Optionssoftware erhältliche grafische Programmiersprache S7-GRAPH ermöglicht Ihnen die Programmierung von Ablaufsteuerungen. Dazu gehören die Erstellung einer Schrittkette, die Festlegung der jeweiligen Schrittinhalte und der Weiterschaltbedingungen (Transitionen). Den Schrittinhalt legen Sie in einer speziellen Programmiersprache (ähnlich AWL) fest, die Weiterschaltbedingungen werden in einer Kontaktplandarstellung eingegeben (eingeschränkter Umfang der Programmiersprache KOP).

S7-GRAPH stellt auch komplexe Abläufe sehr übersichtlich dar und erlaubt damit eine effektive Programmierung und Fehlersuche.

Beispiel für eine Ablaufsteuerung in S7-GRAPH



Erzeugte Bausteine

Mit dem Editor für S7-GRAPH programmieren Sie den Funktionsbaustein, der die Schrittkette beinhaltet. Die Daten für die Schrittkette, z. B. FB-Parameter, Schritt- und Transitionsbedingungen, enthält ein dazugehöriger Instanz-DB. Eine automatische Erzeugung dieses Instanz-DB veranlassen Sie im Editor für S7-GRAPH.

Quelldatei

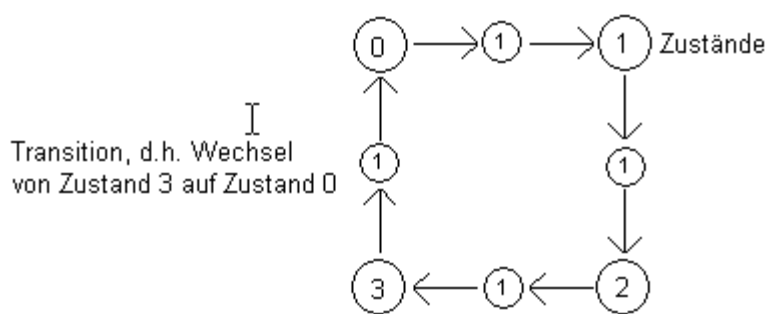
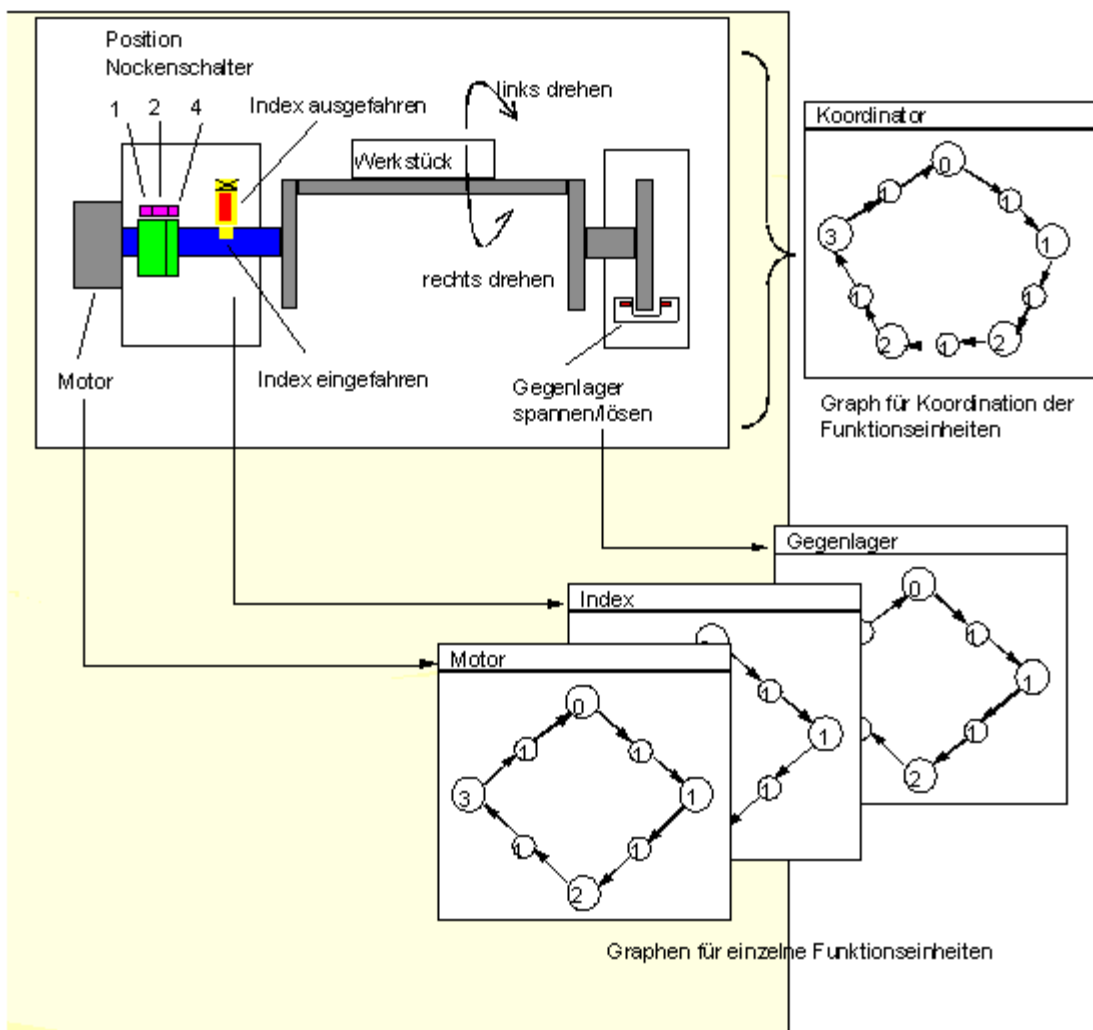
Aus einem mit S7-GRAPH erstellten FB kann eine textuelle Quelldatei (GRAPH-Quelle) erzeugt werden, die von Operator Panels oder Textdisplays zur Anzeige der Ablaufsteuerung interpretiert werden kann.

9.2.6 Programmiersprache S7-HiGraph (Zustandsgraph)

Die als Optionssoftware erhältliche grafische Programmiersprache S7-HiGraph ermöglicht Ihnen die Programmierung einiger Bausteine Ihres Programmes als Zustandsgraph. Dabei zerlegen Sie Ihre Anlage in eigenständige Funktionseinheiten, die jeweils verschiedene Zustände annehmen können. Für den Wechsel zwischen den Zuständen definieren Sie Transitionen. Die Aktionen, die den Zuständen zugeordnet sind, und die Bedingungen für die Transitionen zwischen den Zuständen beschreiben Sie in einer an AWL angelehnten Lupensprache.

Für jede Funktionseinheit erstellen Sie einen Graphen, der das Verhalten dieser Funktionseinheit beschreibt. Die Graphen einer Anlage werden zu Graphengruppen zusammengefasst. Zwischen den Graphen können Nachrichten zur Synchronisation der Funktionseinheiten ausgetauscht werden.

Die übersichtliche Darstellung der Zustandsübergänge einer Funktionseinheit erlaubt eine systematische Programmierung und erleichtert die Fehlersuche. Im Unterschied zu S7-GRAPH ist bei S7-HiGraph zu einem Zeitpunkt immer nur ein Zustand (bei S7-GRAPH "Schritt") aktiv. Nachfolgendes Bild zeigt das Erstellen von Graphen für Funktionseinheiten (Beispiel).



Eine Graphengruppe wird in einer HiGraph-Quelle im Ordner "Quellen" unter dem S7-Programm abgespeichert. Diese Quelle wird dann in die S7-Bausteine für das Anwenderprogramm übersetzt.

Syntax und Formalparameter werden bei der letzten Eingabe für einen Graph (mit Schließen des Arbeitsfensters) geprüft. Die Adressen und Symbole werden erst beim Übersetzen der Quelle geprüft.

9.2.7 Programmiersprache S7-CFC

Die Optionssoftware CFC (*C*ontinuous *F*unction *C*hart) ist eine Programmiersprache zur grafischen Verschaltung komplexer Funktionen.

Mit der Programmiersprache S7-CFC programmieren Sie, indem Sie vorhandene Funktionen grafisch verschalten. Viele Standardfunktionen brauchen Sie nicht selbst programmieren, sondern Sie können auf Bibliotheken mit Standardbausteinen (z. B. für Logik-, Arithmetik-, Regler- und Datenverarbeitungsfunktionen) zurückgreifen. Sie benötigen für die Anwendung von CFC keine tieferen Programmierkenntnisse oder spezifische SPS-Kenntnisse und können sich auf die branchenorientierte Technologie konzentrieren.

Das erstellte Programm wird in Form von CFC-Plänen gespeichert. Diese werden im Ordner "Pläne" unter dem S7-Programm abgelegt. Aus diesen Plänen werden dann die S7-Bausteine für das Anwenderprogramm übersetzt.

Zu verschaltende Bausteine, die Sie selbst erstellen möchten, programmieren Sie für SIMATIC S7 mit einer der S7-Programmiersprachen, für SIMATIC M7 mit C/C++.

9.3 Anlegen von Bausteinen

9.3.1 Bausteinordner

Das Programm für eine S7-CPU können Sie erstellen in Form von:

- Bausteinen
- Quellen

Für die Ablage von Bausteinen steht der Ordner "Bausteine" unter dem S7-Programm zur Verfügung.

Im Bausteinordner sind die Bausteine enthalten, die Sie für die Realisierung Ihrer Automatisierung auf die S7-CPU laden. Zu diesen ladbaren Bausteinen gehören Codebausteine (OBs, FBs, FCs) und Datenbausteine (DB). Ein leerer Codebaustein OB 1 wird automatisch mit dem Ordner "Bausteine" angelegt, da Sie diesen Baustein auf jeden Fall für die Bearbeitung Ihres Programms in der S7-CPU benötigen.

Zusätzlich enthält der Ordner "Bausteine" noch folgende Objekte:

- Die von Ihnen erstellten anwenderdefinierten Datentypen. Sie vereinfachen Ihnen die Programmierung, werden aber nicht auf die CPU geladen.
- Die Variablen Tabellen (VATs), die Sie für den Test Ihres Programms zum Beobachten und Steuern von Variablen erstellen können. Variablen Tabellen werden nicht auf die CPU geladen.
- Das Objekt "Systemdaten" (Systemdatenbausteine), das Systeminformationen (Systemkonfiguration, Systemparameter) enthält. Diese Systemdatenbausteine werden durch das Konfigurieren der Hardware angelegt und mit Daten versorgt.
- Die Systemfunktionen (SFCs) und Systemfunktionsbausteine (SFBs), die Sie in Ihrem Anwenderprogramm aufrufen wollen. Die SFCs und SFBs selbst können Sie nicht bearbeiten.

Mit Ausnahme der Systemdatenbausteine (die nur über die Konfiguration des Automatisierungssystems erstellt und bearbeitet werden) erfolgt die Bearbeitung der Bausteine des Anwenderprogramms jeweils mit dem zugeordneten Editor. Dieser wird automatisch bei Doppelklick auf den entsprechenden Baustein gestartet.

Hinweis

Im Ordner "Bausteine" werden auch die Bausteine gespeichert, die Sie in Form von Quellen programmiert und anschließend übersetzt haben.

9.3.2 Anwenderdefinierte Datentypen (UDT)

Anwenderdefinierte Datentypen sind von Ihnen erzeugte spezielle Datenstrukturen, die Sie nach Ihrer Definition im gesamten S7-Anwenderprogramm verwenden können.

- UDTs können wie elementare Datentypen oder zusammengesetzte Datentypen in der Variablendeklaration von Codebausteinen (FC, FB, OB) oder als Datentyp für Variablen in einem Datenbaustein (DB) verwendet werden. Damit haben Sie den Vorteil, dass Sie eine mehrmals zu verwendende, spezielle Datenstruktur nur einmal definieren und sie anschließend beliebig vielen Variablen zuweisen können.
- UDTs können als Vorlage für die Erstellung von Datenbausteinen mit gleicher Datenstruktur dienen, d. h. Sie erstellen nur einmal die Struktur und erzeugen anschließend die benötigten Datenbausteine durch einfache Zuweisung des UDTs (Beispiel Rezepturen: Die Struktur des DB ist immer gleich, nur die Mengenangaben sind unterschiedlich).

Anwenderdefinierte Datentypen werden - wie die übrigen Bausteine auch - im SIMATIC Manager oder im inkrementellen Editor angelegt.

Aufbau eines UDT

Nach dem Öffnen eines UDT wird Ihnen in einem neuen Arbeitsfenster die Tabellendarstellung für diesen anwenderdefinierten Datentyp in Deklarationsicht angezeigt.

- Die erste und die letzte Zeile beinhalten bereits die Deklarationen STRUCT bzw. END_STRUCT für Beginn und Ende des anwenderdefinierten Datentyps. Diese Zeilen können Sie nicht editieren.
- Sie bearbeiten den anwenderdefinierten Datentyp, indem Sie Ihre Eingaben ab der zweiten Zeile der Deklarationstabelle in den entsprechenden Spalten durchführen.
- Sie können anwenderdefinierte Datentypen strukturieren aus:
 - elementaren Datentypen
 - zusammengesetzten Datentypen
 - bereits vorhandenen anwenderdefinierten Datentypen

Die anwenderdefinierten Datentypen des S7-Anwenderprogramms werden nicht auf die S7-CPU geladen. Sie werden entweder direkt durch einen inkrementellen Editor erstellt und bearbeitet oder sie entstehen durch die Übersetzung von Quellen.

9.3.3 Bausteineigenschaften

Mit Hilfe von Bausteineigenschaften können Sie die erstellten Bausteine besser identifizieren (z. B. bei der Versionspflege) oder Sie schützen Bausteine vor unberechtigten Veränderungen.

Die Bausteineigenschaften sollten Sie bei geöffnetem Baustein bearbeiten. Neben editierbaren Eigenschaften werden in dem zugehörigen Dialogfeld auch Daten nur zu Ihrer Information angezeigt: Diese können Sie nicht bearbeiten.

Die Bausteineigenschaften und Systemattribute werden zwar auch im SIMATIC Manager in den Objekteigenschaften für einen Baustein angezeigt. Dort können Sie jedoch nur die Eigenschaften NAME, FAMILIE, AUTHOR und VERSION editieren.

Die Objekteigenschaften editieren Sie nach dem Einfügen des Bausteins über den SIMATIC Manager. Wurde ein Baustein nicht im SIMATIC Manager, sondern über einen der Editoren erzeugt, so sind diese Eintragungen (z. B. Programmiersprache) automatisch in den Objekteigenschaften gespeichert.

Hinweis

Die Mnemonik, in der Sie Ihre S7-Bausteine programmieren wollen, legen Sie im SIMATIC Manager über den Menübefehl **Extras > Einstellungen** im Register "Sprache" fest.

Tabelle der Bausteineigenschaften

Wenn Sie Bausteineigenschaften eingeben möchten, dann ist die Reihenfolge der Eingabe nach folgender Tabelle zu beachten.

Schlüsselwort / Eigenschaft	Bedeutung	Beispiel
[KNOW_HOW_PROTECT]	Bausteinschutz; ein Baustein, der mit dieser Option übersetzt wurde, lässt keinen Einblick in den Anweisungsteil zu. Die Schnittstelle des Bausteins kann eingesehen, aber nicht verändert werden.	KNOW_HOW_PROTECT
[AUTHOR:]	Name des Autors, Firmenname, Abteilungsname oder andere Namen (max. 8 Zeichen, ohne Leerzeichen)	AUTHOR : Siemens, aber kein Schlüsselwort
[FAMILY:]	Name der Bausteinfamilie: z.B. Regler. (max. 8 Zeichen, ohne Leerzeichen).	FAMILY : Regler, aber kein Schlüsselwort
[NAME:]	Bausteinname (max. 8 Zeichen)	NAME : PID, aber kein Schlüsselwort
[VERSION: int1 . int2]	Versionsnummer des Bausteins (beide Zahlen zwischen 0..15, d. h.0.0 - 15.15))	VERSION : 3.10

Schlüsselwort / Eigenschaft	Bedeutung	Beispiel
[CODE_VERSION1]	Kennung, ob ein FB multiinstanzfähig ist oder nicht. Wenn Sie Multiinstanzen deklarieren möchten, darf der FB diese Eigenschaft nicht haben.	CODE_VERSION1
[UNLINKED] nur für DBs!	Datenbausteine mit der Eigenschaft UNLINKED werden nur im Ladespeicher abgelegt, belegen keinen Platz im Arbeitsspeicher und werden nicht in das Programm eingebunden. Auf sie kann nicht mit MC7-Befehlen zugegriffen werden. Die Inhalte solcher DB können, abhängig von der jeweiligen CPU, nur mit SFC 20 BLKMOV oder SFC 83 READ_DBL in den Arbeitsspeicher transferiert werden.	
[Non-Retain]	Datenbausteine mit diesem Attribut werden nach jedem Netz-Aus- und Netz-Einschalten und nach jedem STOP-RUN-Übergang der CPU auf die Ladewerte zurückgesetzt.	
[READ_ONLY] nur für DBs!	Schreibschutz für Datenbausteine; seine Daten können nur gelesen u. nicht verändert werden.	READ_ONLY
Baustein schreibgeschützt	Kopie eines Bausteins, der zu Referenzzwecken schreibgeschützt abgelegt wurde. Diese Eigenschaft kann nur im Programmierer über den Menübefehl Datei > Schreibgeschützt ablegen zugewiesen werden.	

Der Bausteinschutz KNOW_HOW_PROTECT hat folgende Konsequenzen:

- Wenn Sie später einen übersetzten Baustein im inkrementellen AWL-, FUP- oder KOP-Editor ansehen wollen, kann der Anweisungsteil des Bausteins nicht eingesehen werden.
- In der Variablendeklarationsliste des Bausteins werden nur die Variablen der Deklarationstypen var_in, var_out und var_in_out angezeigt. Die Variablen der Deklarationstypen var_stat und var_temp bleiben verborgen.

Zuordnung Bausteineigenschaften zu Bausteinararten

Die nachfolgenden Tabelle zeigt, welche Bausteineigenschaften bei welchen Bausteinararten vereinbart werden können.

Eigenschaft	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	•	•	•	•	–
AUTHOR	•	•	•	•	–
FAMILY	•	•	•	•	–
NAME	•	•	•	•	–
VERSION	•	•	•	•	–
UNLINKED	–	–	–	•	–
READ_ONLY	–	–	–	•	–
Non-Retain	–	–	–	•	–
Baustein schreibgeschützt	•	•	•	•	•

Die Eigenschaft KNOW_HOW_PROTECT kann beim Programmieren des Bausteins in einer Quelle eingestellt werden. Im Dialogfeld zu den Baustein-Eigenschaften wird sie angezeigt, ist aber nicht veränderbar.

9.3.4 Anzeige von Baustein-Längen

Baustein-Längen werden in der Einheit "Byte" angezeigt.

Anzeige in den Eigenschaften eines Bausteinordners

In den Eigenschaften des Bausteinordners werden in Offline-Ansicht folgende Längen angezeigt:

- Größe (Summe aller Bausteine ohne Systemdaten) im Ladespeicher des Zielsystems
- Größe (Summe aller Bausteine ohne Systemdaten) im Arbeitsspeicher des Zielsystems
- Bausteinlängen auf dem Erstellsystem (PG/PC) werden in den Eigenschaften des Bausteinordners nicht angezeigt.

Anzeige in den Eigenschaften eines Bausteins

In den Eigenschaften eines Bausteins werden angezeigt:

- Benötigte Anzahl an Lokaldaten: Größe der Lokaldaten in Byte
- MC7: Größe des MC7-Codes in Byte bzw. Größe der DB-Nutzdaten
- Größe im Ladespeicher des Zielsystems
- Größe im Arbeitsspeicher des Zielsystems: wird nur bei bekannter Hardware-Zuordnung angezeigt.

Die Anzeigen sind unabhängig davon, ob der Baustein im Fenster einer Online-Ansicht oder einer Offline-Ansicht liegt.

Anzeige im SIMATIC Manager (Ansicht Details)

Ist ein Baustein-Behälter geöffnet und "Ansicht Details" eingestellt, so wird im Projektfenster der Arbeitsspeicherbedarf angezeigt, unabhängig davon, ob der Baustein-Behälter im Fenster einer Online-Ansicht oder einer Offline-Ansicht liegt.

Sie können die Bausteinlängen summieren, indem Sie alle relevanten Bausteine markieren. In diesem Fall wird die Summe für die markierten Bausteine in der Statuszeile des SIMATIC Managers angezeigt.

Für nicht auf das Zielsystem ladbare Bausteine (z. B. VATs) wird keine Länge dargestellt.

Bausteinlängen auf dem Erstellsystem (PG/PC) werden in der Ansicht Details nicht angezeigt.

9.3.5 Bausteine vergleichen

Einführung

Wenn Sie Bausteine vergleichen wollen, die sich an verschiedenen Orten befinden, dann haben Sie folgende Möglichkeiten, den Bausteinvergleich aufzurufen:

- Aufruf des Bausteinvergleichs im SIMATIC Manager mit dem Menübefehl **Extras > Bausteine vergleichen**. Mit Klicken der Schaltfläche "Gehe zu" im Folgedialog "Bausteine vergleichen - Ergebnisse" wird das Register "Vergleich" im Programmeditor (KOP/FUP/AWL) mit Inhalten gefüllt.
- Aufruf des Bausteinvergleichs im Programmeditor über den **Menübefehl Extras > On-/Offline Partner vergleichen**.

Im folgenden wird die Funktionsweise erläutert. Dabei wird nach Codebausteinen (OBs, FBs, FCs) und Datenbausteinen (DBs) unterschieden.

Funktionsweise des Bausteinvergleichs: Codebausteine

Im ersten Schritt vergleicht STEP 7 die Zeitstempel der Schnittstellen der zu vergleichenden Codebausteine. Wenn diese Zeitstempel gleich sind, geht STEP 7 von einer Gleichheit der Schnittstellen aus.

Wenn die Zeitstempel der Schnittstellen unterschiedlich sind, vergleicht STEP 7 abschnittsweise die Datentypen der Schnittstellen. Bei Differenzen ermittelt STEP 7 jeweils den ersten Unterschied eines Abschnitts, d. h. jeweils den **ersten** Unterschied in den jeweiligen Deklarationsbereichen. Multiinstanzen und UDTs werden in den Vergleich mit einbezogen. Bei Gleichheit der Datentypen in den Abschnitten vergleicht STEP 7 anschließend die Anfangswerte der Variablen. Es werden alle Unterschiede angezeigt.

Im zweiten Schritt prüft STEP 7 den Code netzwerkweise (für den Fall, daß die Option "Codevergleich durchführen" nicht aktiviert war, wird ein Codevergleich dennoch beim Aktivieren der Schaltfläche "Gehe zu" im Programmeditor durchgeführt!)

Zunächst werden eingefügte bzw. gelöschte Netzwerke ermittelt. Im Vergleichsergebnis werden Netzwerke, die nur in einem Baustein vorhanden sind, mit dem Kommentar "nur enthalten in" versehen.

Danach werden die übrigen Netzwerke jeweils bis zur **ersten** unterschiedlichen Anweisung verglichen. Anweisungen werden wie folgt verglichen:

- Bei der Einstellung "Absolutoperand hat Vorrang" anhand der Absolutoperanden
- Bei der Einstellung "Symbol hat Vorrang" anhand des Symbols

Gleich sind Anweisungen dann, wenn Operator und Operand gleich sind.

Wenn die zu vergleichenden Bausteine in unterschiedlichen Erstsprachen programmiert wurden, vergleicht STEP 7 auf der Basis von AWL.

Besonderheit bei Offline-Offline-Vergleich:

Im Unterschied zu einem Offline-Online-Vergleich ermittelt STEP 7 bei einem Offline-Offline-Vergleich zusätzlich unterschiedliche Variablennamen. Das ist bei einem Offline-Online-Vergleich nicht möglich, da online nur eine Ersatzsymbolik zur Verfügung steht.

Vom Vergleich ausgeschlossen sind Baustein- Netzwerk- und Zeilenkommentare sowie weitere Baustein-Attribute (z. B. S7-PDIAG-Informationen und Meldungen).

Funktionsweise des Bausteinvergleichs: Datenbausteine

Im ersten Schritt vergleicht STEP 7 wie bei Codebausteinen die Zeitstempel der Schnittstellen der zu vergleichenden Datenbausteine. Wenn diese Zeitstempel gleich sind, geht STEP 7 von einer Gleichheit der Datenstrukturen aus.

Wenn die Zeitstempel der Schnittstellen unterschiedlich sind, vergleicht STEP 7 die Strukturen bis zum **ersten** Unterschied in der Datenstruktur. Bei Gleichheit der Datenstruktur in den Abschnitten vergleicht STEP 7 anschließend die Anfangs- und Aktualwerte. Es werden alle Unterschiede angezeigt.

Besonderheit bei Offline-Offline-Vergleich:

Im Unterschied zu einem Offline-Online-Vergleich ermittelt STEP 7 bei einem Offline-Offline-Vergleich zusätzlich unterschiedliche Variablennamen. Das ist bei einem Offline-Online-Vergleich nicht möglich, da online nur eine Ersatzsymbolik zur Verfügung steht.

Vom Vergleich ausgenommen sind Kommentare sowie die Strukturen von UDTs, die im Datenbaustein verwendet wurden.

Funktionsweise des Bausteinvergleichs: Datentypen (UDT)

Im ersten Schritt vergleicht STEP 7 wie bei Datenbausteinen die Zeitstempel der Schnittstellen der zu vergleichenden Datentypen. Wenn diese Zeitstempel gleich sind, geht STEP 7 von einer Gleichheit der Datenstrukturen aus.

Wenn die Zeitstempel der Schnittstellen unterschiedlich sind, vergleicht STEP 7 die Strukturen bis zum **ersten** Unterschied in der Datenstruktur. Bei Gleichheit der Datenstruktur in den Abschnitten vergleicht STEP 7 anschließend die Anfangswerte. Es werden alle Unterschiede angezeigt.

Vorgehensweise: Bausteinvergleich im Programmeditor

1. Öffnen Sie einen Baustein, den Sie mit der geladenen Version vergleichen wollen.
2. Wählen Sie den Menübefehl **Extras > On-/Offline Partner vergleichen**.

Wenn der Online Partner erreichbar ist, dann werden im Register "7:Vergleich" im unteren Bereich des Programmeditor-Fensters die Vergleichsergebnisse angezeigt.

Tipp: Wenn zwei Netzwerke als "ungleich" ermittelt wurden, dann können Sie durch Doppelklick auf diese Zeile das entsprechende Netzwerk öffnen.

Vorgehensweise: Bausteinvergleich im SIMATIC Manager

1. Markieren Sie im SIMATIC Manager den Baustein-Ordner oder nur die Bausteine, die sie vergleichen möchten.
2. Wählen Sie den Menübefehl Extras > Bausteine vergleichen.
3. Wählen Sie im Dialogfeld "Bausteine vergleichen" die Art des Vergleichs (ONLINE/offline oder Pfad1/Pfad2).
4. Bei Pfad1/Pfad2-Vergleich: Wählen Sie im SIMATIC Manager den Baustein-Ordner oder die Bausteine aus, mit denen verglichen werden soll. Die Bausteine werden dann automatisch in das Dialogfeld eingetragen.
5. Aktivieren Sie das Kontrollkästchen "einschließlich SDBs", wenn auch SDBs verglichen werden sollen.
6. Aktivieren Sie das Kontrollkästchen "Codevergleich durchführen", wenn Sie zusätzlich einen Codevergleich durchführen möchten. Ist dieses Kontrollkästchen aktiviert, so kann nachfolgend festgelegt werden, ob ein "Detailvergleich" durchgeführt wird. Beim Detailvergleich werden zusätzlich zu den ablaufrelevanten Teilen der Bausteine (Schnittstellen und Code) Änderungen von Namen lokaler Variablen und Parameternamen angezeigt. Außerdem können Sie festlegen, ob "auch Bausteine mit unterschiedlichen Erstsprachen" verglichen werden (z. B. AWL, FUP) sollen. In diesem Fall werden die Bausteine auf Basis von AWL verglichen.
7. Betätigen Sie im Dialogfeld die Schaltfläche "OK".
Die Ergebnisse des Vergleichs werden im weiteren Dialogfeld "Bausteine vergleichen - Ergebnisse" angezeigt.
8. Wenn Sie in diesem Dialogfeld auf die Schaltfläche "Details" klicken, werden die Eigenschaften der verglichenen Bausteine angezeigt (z. B. Zeitpunkte der letzten Änderungen, Prüfsummen etc.).
Wenn Sie auf die Schaltfläche "Gehe zu" klicken, öffnet sich der Programmierer und die Ergebnisse des Vergleichs sind im unteren Teil des Fensters sichtbar.

Hinweis

Beim Vergleich eines Offline Bausteinordners mit einem Online Bausteinordner werden nur ladbare Bausteintypen (OB, FB, ...) verglichen.

Beim Vergleich Offline/Online oder Pfad1/Pfad2 einer Multiselektion werden alle selektierten Bausteine verglichen, auch wenn es sich um nicht ladbare Bausteine (z. B. Variablentabellen oder UDTs) handelt.

9.3.6 Umverdrahten

Folgende Bausteine und Operanden können umverdrahtet werden:

- Eingänge, Ausgänge
- Merker, Zeiten, Zähler
- Funktionen, Funktionsbausteine

Gehen Sie folgendermaßen vor:

1. Markieren Sie im SIMATIC Manager den Ordner "Bausteine", in dem sich die Bausteine befinden, die Sie umverdrahten möchten.
2. Wählen Sie den Menübefehl **Extras > Umverdrahten**.
3. Tragen Sie im angezeigten Dialogfeld "Umverdrahten" die gewünschten Ersetzungen (Alter Operand / Neuer Operand) in die Tabelle ein.
4. Wählen Sie die Option "Alle Zugriffe innerhalb der angegebenen Operanden", wenn Sie Operandenbereiche (BYTE, WORD, DWORD) umverdrahten möchten.
Beispiel: Sie geben als Operandenbereiche EW0 und EW4 an. Dann werden die Operanden E0.0 - E1.7 auf die Operanden E4.0 - E5.7 umverdrahtet. Operanden aus dem umverdrahteten Bereich (z.B. E0.1) können danach nicht mehr einzeln in die Tabelle eingetragen werden.
5. Betätigen Sie die Schaltfläche "OK".

Damit starten Sie das Umverdrahten. Nach der Verdrahtung können Sie über ein Dialogfeld entscheiden, ob Sie die Infodatei zur Umverdrahtung ansehen wollen. Die Infodatei enthält die Operandenliste, "Alter Operand" und "Neuer Operand". Dazu werden die einzelnen Bausteine aufgeführt mit der Anzahl Verdrahtungen, welche jeweils im Baustein durchgeführt worden sind.

Beim Umverdrahten ist Folgendes zu beachten:

- Wenn Sie einen Baustein umverdrahten (d. h. umbenennen), dann darf der neue Baustein nicht bereits existieren. Falls der Baustein bereits existiert, wird der Vorgang abgebrochen.
- Wenn Sie einen Funktionsbausteine (FB) umverdrahten, dann wird automatisch sein Instanz-DB dem umverdrahteten FB zugeordnet. Der Instanz-DB ändert sich nicht (d. h. die DB-Nummer bleibt erhalten).

9.3.7 Attribute für Bausteine und Parameter

Eine Beschreibung der Attribute finden Sie in der Referenzhilfe zu Systemattributen:

- Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute

9.4 Arbeiten mit Bibliotheken

Bibliotheken dienen zur Ablage von wiederverwendbaren Programmkomponenten für die SIMATIC S7/M7. Die Programmkomponenten können aus vorhandenen Projekten in eine Bibliothek kopiert oder direkt in der Bibliothek unabhängig von Projekten erstellt werden.

Sie können sich viel Programmieraufwand sparen, wenn Sie Bausteine, die Sie immer wieder verwenden wollen, in einem S7-Programm unter einer Bibliothek ablegen. Von dort können Sie diese immer in das entsprechende Anwenderprogramm kopieren.

Für die Erstellung von S7/M7-Programmen in einer Bibliothek steht Ihnen - mit Ausnahme des Testens - die gleiche Funktionalität wie in Projekten zur Verfügung.

Anlegen von Bibliotheken

Bibliotheken können Sie wie Projekte mit dem Menübefehl **Datei > Neu** anlegen. Die neue Bibliothek wird in dem Verzeichnis angelegt, das Sie nach Auswahl des Menübefehls **Extras > Einstellungen** im Register "Allgemein" für die Bibliotheken eingestellt haben.

Hinweis

Der SIMATIC Manager lässt Namen zu, die länger als 8 Zeichen sind. Der Name des Bibliotheksverzeichnisses wird jedoch auf 8 Zeichen abgeschnitten. Bibliotheksnamen müssen sich daher in den ersten 8 Zeichen unterscheiden. Zwischen Groß- und Kleinschreibung wird nicht unterschieden. Beim Öffnen wird im Browser wieder der vollständige Name angezeigt, beim Durchsuchen nur der gekürzte.

Beachten Sie, dass Sie in Projekten einer älteren STEP 7-Version keine Bausteine aus Bibliotheken einer neueren STEP 7-Version verwenden können.

Öffnen von Bibliotheken

Um eine Bibliothek zu öffnen, geben Sie zunächst den Menübefehl **Datei > Öffnen** ein. Wählen Sie dann eine Bibliothek in den Folgedialogen aus. Danach wird das Bibliotheksfenster geöffnet.

Hinweis

Falls Sie die gewünschte Bibliothek nicht in der Bibliotheksliste angezeigt bekommen, betätigen Sie im Dialogfeld "Öffnen" die Schaltfläche "Durchsuchen". Daraufhin wird Ihnen im Windows-Standarddialogfeld die Verzeichnisstruktur angezeigt, in der Sie die Bibliothek suchen können.

Beachten Sie hierbei, dass der Name der Datei immer dem ursprünglichen Namen der angelegten Bibliothek entspricht, d. h. Namensänderungen im SIMATIC Manager werden auf Dateiebene nicht nachgezogen.

Sobald Sie die Bibliothek ausgewählt haben, wird diese in die Bibliotheksliste aufgenommen. Die Einträge in der Bibliotheksliste können Sie mit dem Menübefehl **Datei > Verwalten** ändern.

Kopieren von Bibliotheken

Eine Bibliothek kopieren Sie, indem Sie diese über den Menübefehl **Datei > Speichern unter** unter einen anderen Namen abspeichern.

Bibliotheksteile wie Programme, Bausteine, Quellen usw. kopieren Sie mit dem Menübefehl **Bearbeiten > Kopieren**.

Löschen von Bibliothek

Eine Bibliothek löschen Sie über den Menübefehl **Datei > Löschen**.

Bibliotheksteile wie Programme, Bausteine, Quellen usw. löschen Sie mit dem Menübefehl **Bearbeiten > Löschen**.

9.4.1 Hierarchischer Aufbau von Bibliotheken

Ebenso wie Projekte sind Bibliotheken hierarchisch aufgebaut:

- Bibliotheken können S7/M7-Programme enthalten.
- Ein S7-Programm kann genau einen Ordner "Bausteine" (Anwenderprogramm), einen Ordner "Quellen", einen Ordner "Pläne" sowie ein Objekt "Symbole" (Symboltabelle) enthalten.
- Ein M7-Programm kann Pläne und C-Programme für programmierbare M7-Baugruppen sowie ein Objekt "Symbole" (Symboltabelle) und einen Bausteinordner für DBs und VATs (Variablentabellen) enthalten.
- Der Ordner "Bausteine" enthält die Bausteine, die auf die S7-CPU geladen werden können. Die enthaltenen Variablentabellen und anwenderdefinierten Datentypen werden nicht auf die CPU geladen.
- Der Ordner "Quellen" enthält die Quellen für die in den diversen Programmiersprachen erstellten Programmen.
- Der Ordner "Pläne" enthält die CFC-Pläne (nur bei Optionsoftware S7-CFC).

Wenn Sie ein neues S7-/M7-Programm einfügen, wird darin automatisch je ein Ordner "Bausteine" und "Quellen" (nur bei S7) sowie ein Objekt "Symbole" eingefügt.

9.4.2 Übersicht der Standardbibliotheken

Das STEP 7-Basispaket enthält folgende Standardbibliotheken:

- **System Function Blocks:** Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB)
- **S5-S7 Converting Blocks:** Bausteine für die Konvertierung von STEP 5-Programmen
- **IEC Function Blocks:** Bausteine für IEC-Funktionen, wie z. B. für die Bearbeitung von Zeit- und Datumsangaben, für Vergleichsoperationen, für String-Bearbeitung und zur Auswahl von Maximum und Minimum
- **Organization Blocks:** Standard-Organisationsbausteine (OB)
- **PID Control Blocks:** Funktionsbausteine (FB) zur PID-Regelung
- **Communication Blocks:** Funktionen (FC) und Funktionsbausteine (FB) für SIMATIC NET-CPs.
- **TI-S7 Converting Blocks:** allgemein verwendbare Standardfunktionen
- **Miscellaneous Blocks:** Bausteine zur Zeitstempelung und Uhrzeitsynchronisation

Weitere Bibliotheken können bei der Installation von Optionsoftware dazukommen.

Löschen und Installieren von mitgelieferten Bibliotheken

Die mitgelieferten Bibliotheken können Sie im SIMATIC Manager löschen und danach wieder neu installieren. Für die Installation führen Sie das Programm Setup von STEP 7 komplett neu aus.

Hinweis

Die mitgelieferten Bibliotheken werden bei einer STEP 7-Installation immer kopiert. Falls Sie mitgelieferte Bibliotheken bearbeitet haben, werden bei einer erneuten STEP 7-Installation die von Ihnen geänderten Bibliotheken mit dem Original überschrieben.

Deswegen sollten Sie mitgelieferte Bibliotheken vor Änderungen kopieren und nur die Kopie bearbeiten.

10 Erstellen von Codebausteinen

10.1 Grundlagen zum Erstellen von Codebausteinen

10.1.1 Aufbau des Programmeditor-Fensters

Das Programmeditor-Fenster ist aufgeteilt in die folgenden Bereiche:

Übersichten

Das Register "Programmelemente" zeigt eine komplette Übersicht über die Programmelemente, die Sie in Ihr KOP-, FUP- oder AWL-Programm einfügen können. Das Register "Aufrufstruktur" zeigt die Aufrufhierarchie der Bausteine im aktuellen S7-Programm.

Variablendeklaration

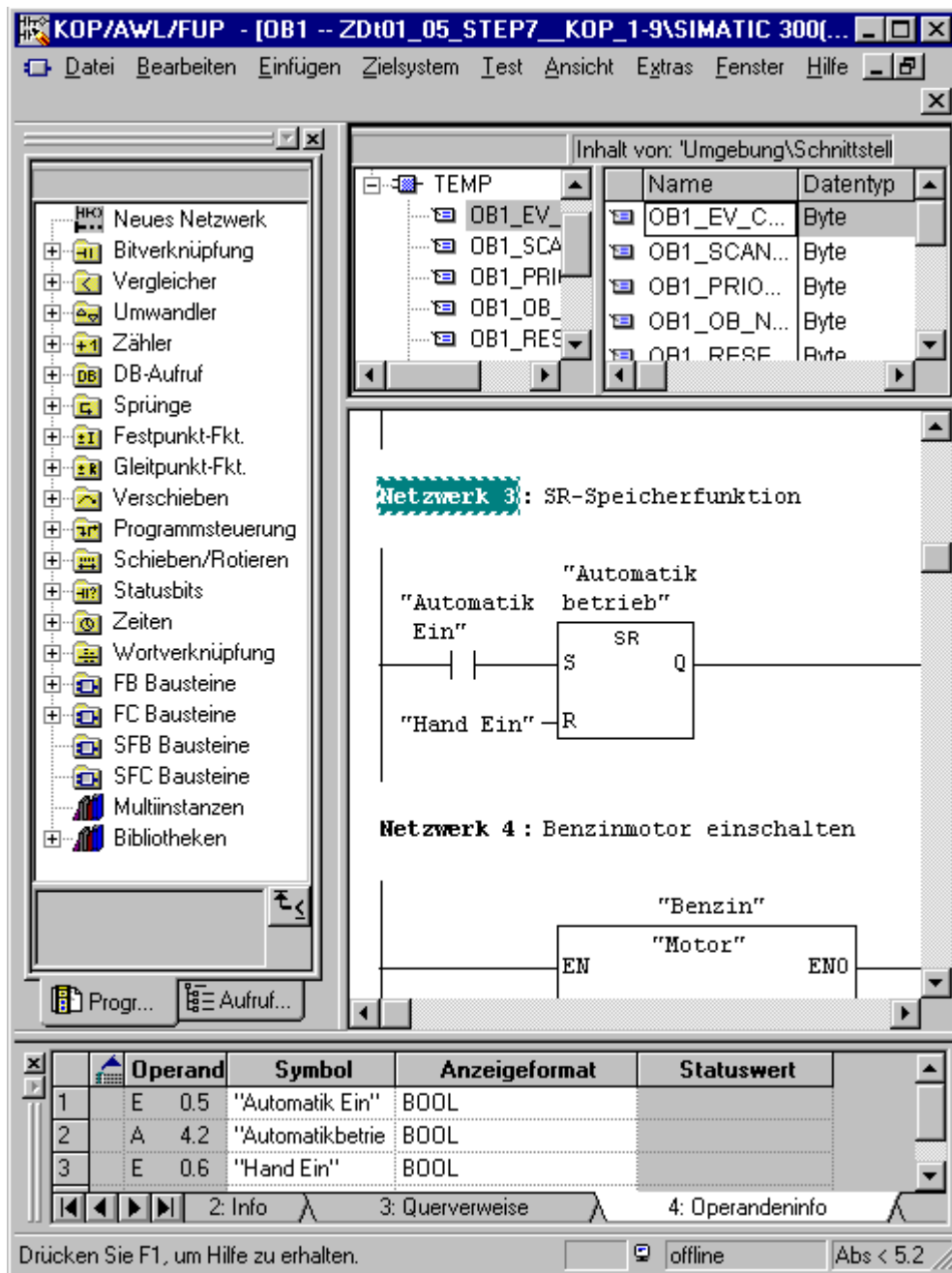
Die Variablendeklaration ist unterteilt in die Variablenübersicht und die Variablendetailsicht.

Anweisungen

Im Anweisungsteil wird der Bausteincode angezeigt, der vom Automatisierungssystem bearbeitet werden soll. Er besteht aus einem oder mehreren Netzwerken.

Details

Die verschiedenen Register des Fensters "Details" ermöglichen z. B. das Anzeigen von Fehlermeldungen, Informationen zu Operanden, die Bearbeitung von Symbolen, das Steuern von Operanden, den Vergleich von Bausteinen und das Bearbeiten von Fehlerdefinitionen für die Prozessdiagnose.

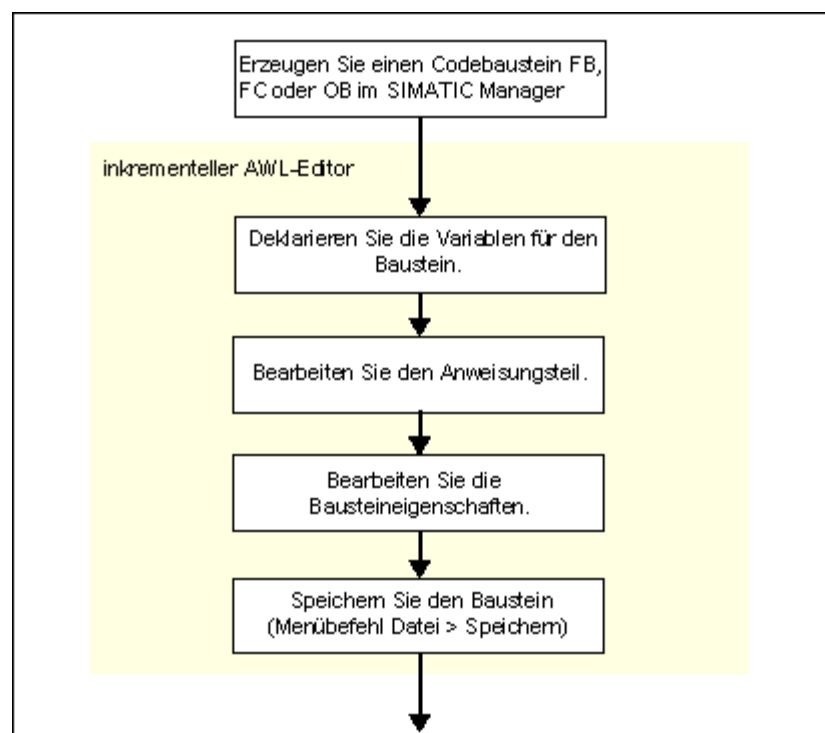


10.1.2 Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen

Codebausteine (OBs, FBs, FCs) bestehen aus einem Variablendeklarationsteil und einem Anweisungsteil. Zusätzlich besitzen Codebausteine Eigenschaften. Bei der Programmierung müssen Sie also folgende drei Teile bearbeiten:

- **Variablendeklaration:** In der Variablendeklaration legen Sie die Parameter, Systemattribute für Parameter sowie lokale Variablen des Bausteins fest.
- **Anweisungsteil:** Im Anweisungsteil programmieren Sie den Bausteincode, der vom Automatisierungssystem bearbeitet werden soll. Er besteht aus einem oder mehreren Netzwerken. Für die Erstellung der Netzwerke stehen Ihnen z. B. die Programmiersprachen AWL (Anweisungsliste), KOP (Kontaktplan) und FUP (Funktionsplan) zur Verfügung.
- **Bausteineigenschaften:** Die Bausteineigenschaften beinhalten zusätzliche Informationen wie Zeitstempel oder Pfadangabe, die vom System eingetragen werden. Daneben können Sie selbst Angaben zu Namen, Familie, Version und Autor machen und Sie können Systemattribute für Bausteine vergeben.

Prinzipiell ist es gleich, in welcher Reihenfolge Sie die Teile eines Codebausteins bearbeiten. Selbstverständlich können Sie auch korrigieren/ergänzen.



Hinweis

Wenn Sie auf Symbole der Symboltabelle zurückgreifen wollen, sollten Sie diese zuerst auf Vollständigkeit überprüfen und ggf. ergänzen.

10.1.3 Voreinstellungen für Programmeditor KOP/FUP/AWL

Bevor Sie mit der Programmierung beginnen, sollten Sie die Einstellmöglichkeiten kennen lernen, um möglichst komfortabel und Ihren Gewohnheiten entsprechend arbeiten zu können.

Über den Menübefehl **Extras > Einstellungen** öffnen Sie einen Registerdialog. In den verschiedenen Registern können Sie Voreinstellungen für das Programmieren von Bausteinen einstellen, z. B. im Register "Allgemein":

- die Schrift (Art und Größe) in Text und Tabellen
- ob Sie bei einem neuen Baustein zunächst Symbolik und Kommentar angezeigt bekommen möchten.

Die Einstellungen für Sprache, Kommentar und Symbolik können Sie aktuell während der Bearbeitung über Befehle des Menüs **Ansicht > ...** ändern.

Die Farbe, in der Markierungen z. B. von Netzwerken oder Anweisungszeilen dargestellt werden, ändern Sie im Register "KOP/FUP".

10.1.4 Zugriffsrechte auf Bausteine bzw. Quellen

Bei der Bearbeitung eines Projekts wird häufig eine gemeinsame Datenhaltung benutzt; dadurch kann es vorkommen, dass mehrere Bearbeiter auf den gleichen Baustein oder die gleich Datenquelle zugreifen wollen.

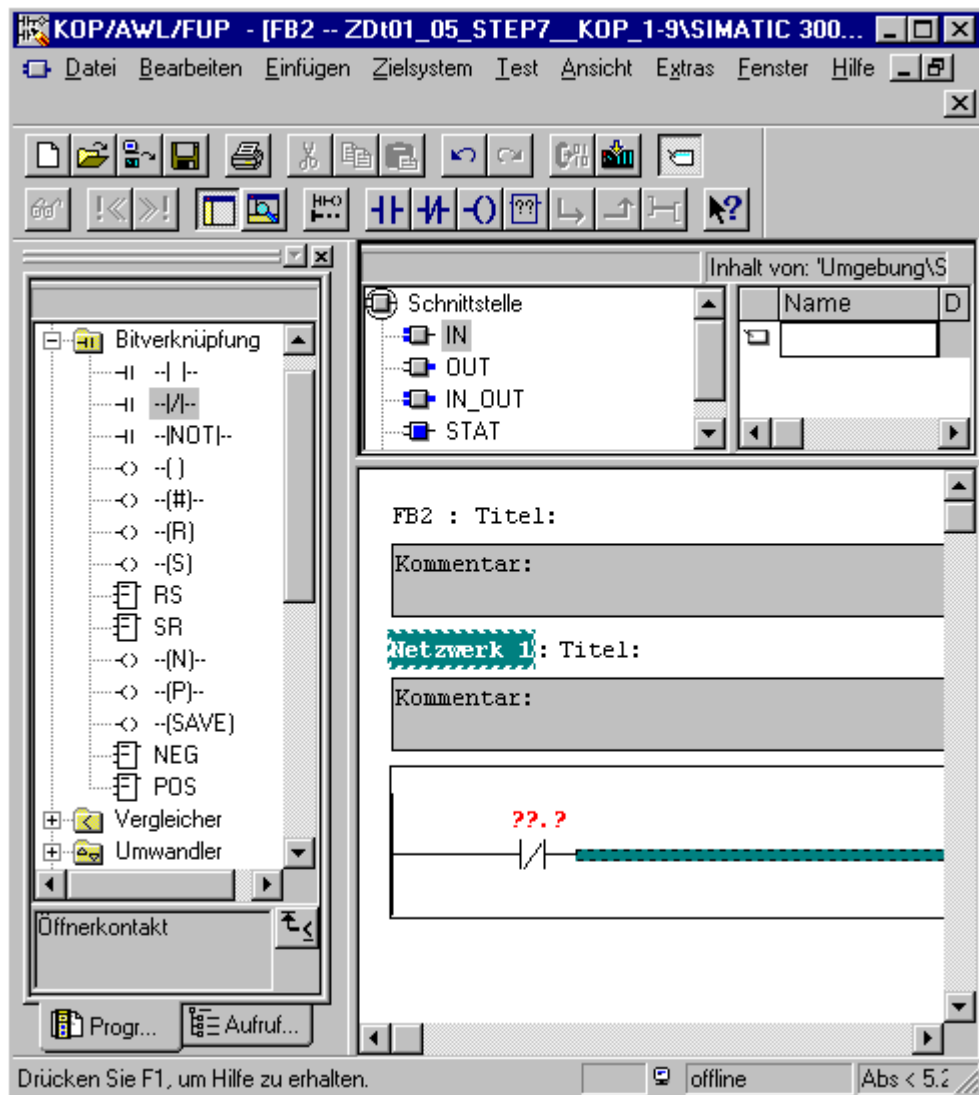
Die Schreib- / Leserechte sind wie folgt vergeben:

- **Offline-Bearbeitung:**
Beim Öffnen eines Bausteines/Quelle wird geprüft, ob dieser mit dem Recht 'Schreiben' geöffnet werden kann. Ist der Baustein/Quelle bereits geöffnet kann nur mit einer Kopie gearbeitet werden. Wollen Sie die Kopie speichern wird abgefragt, ob Sie das Original überschreiben oder die Kopie unter eine anderen Namen speichern wollen.
- **Online-Bearbeitung:**
Öffnet der Bearbeiter über eine projektierte Verbindung einen Online-Baustein, so wird der zugehörige Offline-Baustein gesperrt und somit dessen gleichzeitige Änderung unterbunden.

10.1.5 Anweisungen aus der Programmelemente-Übersicht

Das Register "Programmelemente" im Übersichtsfenster stellt KOP- und FUP-Sprachelemente sowie bereits deklarierte Multiinstanzen, fertige Bausteine und Bausteine aus Bibliotheken zur Verfügung. Das Register lässt sich über den Menübefehl **Ansicht > Übersichten** aufrufen. Programmelemente können auch mit dem Menübefehl **Einfügen > Programmelemente** in den Anweisungsteil übernommen werden.

Beispiel des Registers "Programmelemente" bei KOP



10.2 Editieren der Variablendeklaration

10.2.1 Verwendung der Variablendeklaration in Codebausteinen

Nach dem Öffnen eines Codebausteins erscheint ein Fenster mit der Variablenübersicht und der Variablendetailsicht des Bausteins im oberen Teil sowie dem Anweisungsteil im unteren Teil, in dem Sie den eigentlichen Bausteincode bearbeiten.

Beispiel: Variablensichten und Anweisungsteil in AWL

The screenshot shows the SIMATIC Manager interface for editing a function block (FB1) named 'KOP/AWL/FUP'. The window title is 'KOP/AWL/FUP - [FB1 -- ZDI01_02_STEP7_AWL_1-10\SIMATIC 30...'. The menu bar includes 'Datei', 'Bearbeiten', 'Einfügen', 'Zielsystem', 'Test', 'Ansicht', 'Extras', 'Fenster', and 'Hilfe'. The main window is divided into two sections.

The upper section displays the variable declaration for the 'Schnittstelle' (Interface). On the left, a tree view shows the structure: 'Schnittstelle' (expanded) contains 'IN' (expanded) with sub-elements 'Switch_On', 'Switch_Off', 'Failure', and 'Actual_Speed'; 'OUT'; 'IN_OUT'; 'STAT'; and 'TEMP'. On the right, a table titled 'Inhalt von: 'Umgebung\Schnittstelle\IN'' lists the variables:

Name	Datentyp	Adresse	Anfangswert
Switch_On	Bool	0.0	
Switch_Off	Bool	0.1	
Failure	Bool	0.2	
Actual_Speed	Int	2.0	

The lower section shows the instruction part of the function block. The text reads: 'FB1 : Funktionsbaustein für Motorsteuerung' followed by 'Netzwerk 1: Motor einschalten, Negation von Signalen'. Below this, the instruction is defined as:

```

U      #Switch_On
UN     "Automatikbetrieb"
S      #Engine_On
O      #Switch_Off
ON     #Failure
R      #Engine_On

```

At the bottom of the window, there is a status bar with the text 'Drücken Sie F1, um Hilfe zu erhalten.', a status indicator 'offline', and a page number 'Abs < 5.2'.

In der Variablendetailsicht legen Sie die lokalen Variablen einschließlich der Formalparameter des Bausteins und der Systemattribute für Parameter fest. Das hat u. a. folgende Auswirkungen:

- Durch das Deklarieren wird entsprechend Speicherplatz reserviert, für temporäre Variablen im Lokaldaten-Stack, im Fall von Funktionsbausteinen, für statische Variablen im später zugeordneten Instanz-DB.
- Durch die Festlegung von Eingangs-, Ausgangs- und Durchgangsparametern legen Sie auch die "Schnittstelle" für den Aufruf des Bausteins im Programm fest.
- Wenn Sie die Variablen in einem Funktionsbaustein deklarieren, bestimmen diese Variablen (mit Ausnahme der temporären) auch die Datenstruktur für jeden Instanz-DB, der dem FB zugeordnet wird.
- Durch die Festlegung von Systemattributen weisen Sie Parametern spezielle Eigenschaften z. B. für Meldungs- und Verbindungsprojektierung, Bedien- und Beobachtungsfunktionen und die Leittechnikprojektierung zu.

10.2.2 Zusammenspiel zwischen Variablendeklaration und Anweisungsteil

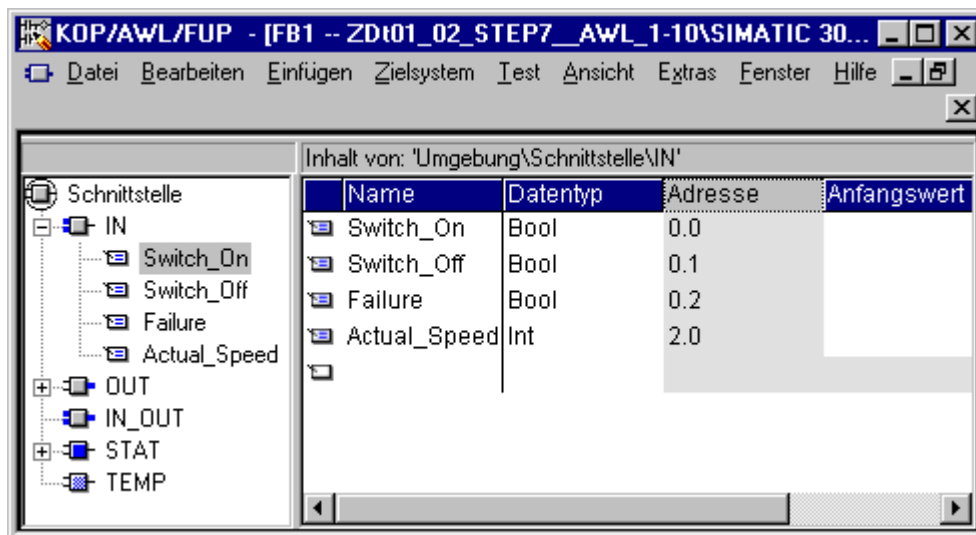
Variablendeklaration und Anweisungsteil von Codebausteinen sind eng miteinander verbunden, da beim Programmieren im Anweisungsteil die Namen aus der Variablendeklaration verwendet werden. Änderungen in der Variablendeklaration wirken sich daher im gesamten Anweisungsteil aus.

Aktion in der Variablendeklaration	Reaktion im Anweisungsteil
Korrekte Neueingabe	Falls ungültiger Code vorhanden, wird zuvor nicht deklarierte Variable nun gültig.
Korrekte Namensänderung ohne Typänderung	Symbol wird sofort überall mit neuem Namen dargestellt.
Korrechter Name wird in einen ungültigen Namen geändert	Code wird nicht geändert.
Ungültiger Name wird in einen korrekten geändert	Falls ungültiger Code vorhanden, wird dieser gültig.
Typänderung	Falls ungültiger Code vorhanden, wird dieser gültig, und falls gültiger Code vorhanden, wird dieser unter Umständen ungültig.
Löschen einer Variable (Symbol), die im Code verwendet wird.	Gültiger Code wird ungültig.

Kommentaränderungen, die fehlerhafte Eingabe einer neuen Variablen, eine Anfangswertänderung oder das Löschen einer nicht verwendeten Variablen haben keine Auswirkung auf den Anweisungsteil.

10.2.3 Aufbau des Variablendeklarationsfensters

Das Variablendeklarationsfenster besteht aus der Variablenübersicht und der Variablendetailsicht.



Nachdem Sie einen neu erstellten Codebaustein geöffnet haben, wird eine voreingestellte Variablenübersicht angezeigt. Diese listet nur die für die gewählte Bausteinart zulässigen Deklarationstypen (in, out, in_out, stat, temp) auf und zwar in der vorgeschriebenen Reihenfolge. Bei der Neuerstellung eines OB wird eine Standard-Variablendeklaration angezeigt, deren Werte Sie ändern können.

Zulässige Datentypen für die Lokaldaten der verschiedenen Bausteinarten finden Sie in Zuordnen von Datentypen zu Lokaldaten von Codebausteinen.

10.3 Multiinstanzen in der Variablendeklaration

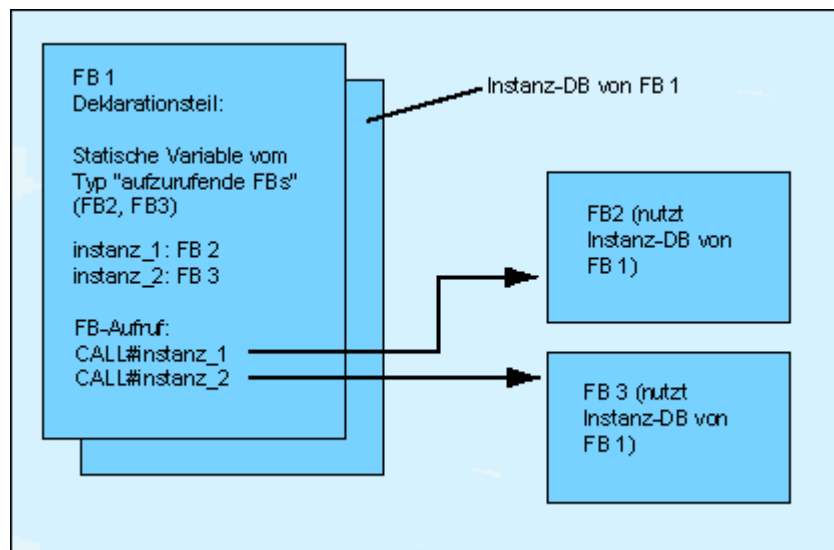
10.3.1 Verwendung von Multiinstanzen

Möglicherweise wollen oder können Sie aufgrund der Leistungsdaten (z. B. Speicherplatz) der verwendeten S7-CPU nur eine beschränkte Anzahl von Datenbausteinen für Instanzdaten spendieren. Wenn in Ihrem Anwenderprogramm in einem FB weitere, bereits vorhandene Funktionsbausteine aufgerufen werden (Aufrufhierarchie von FBs), dann können Sie diese weiteren Funktionsbausteine ohne eigene (d. h. zusätzliche) Instanz-DBs aufrufen.

Dazu bietet sich folgende Lösung an:

- Nehmen Sie die aufzurufenden FBs als statische Variablen in die Variablendeklaration des aufrufenden FB auf.
- In diesem Funktionsbaustein rufen Sie weitere Funktionsbausteine auf ohne eigene (d. h. zusätzliche) Instanz-DBs.
- Damit erreichen Sie eine Konzentrierung der Instanzdaten in einem Instanz-Datenbaustein, d. h. Sie können die verfügbare Anzahl der DBs besser ausnutzen.

Das folgende Beispiel zeigt die beschriebene Lösung: FB 2 und FB 3 nutzen den Instanz-DB des Funktionsbausteins FB 1, von dem aus sie aufgerufen wurden.



Einzigste Voraussetzung: Sie müssen dem aufrufenden Funktionsbaustein "mitteilen", welche Instanzen Sie aufrufen und von welchem (FB-) Typ diese Instanzen sind. Die Angaben sind im Deklarationsfenster des aufrufenden FB zu machen. Der zu verwendende FB muss mindestens eine Variable oder einen Parameter aus dem Datenbereich (also nicht VAR_TEMP) haben.

Verwenden Sie keine Multiinstanzdatenbausteine, solange Online-Änderungen bei laufender CPU zu erwarten sind. Ein stoßfreies Nachladen ist nur bei Verwendung von Instanzdatenbausteinen sichergestellt.

10.3.2 Regeln für die Bildung von Multiinstanzen

Für die Deklaration von Multiinstanzen gelten folgende Regeln:

- Die Deklaration von Multiinstanzen ist nur in Funktionsbausteinen möglich, die mit STEP 7 ab Version 2 erstellt wurden (siehe Bausteinattribut in den Eigenschaften des FB).
- Für das Deklarieren von Multiinstanzen muss der Funktionsbaustein als multiinstanzfähiger FB erzeugt worden sein (Voreinstellung ab STEP 7 Version x.x; abschaltbar im Editor über **Extras > Einstellungen**).
- Dem Funktionsbaustein, in dem eine Multiinstanz deklariert wird, muss ein Instanz-DB zugeordnet werden.
- Eine Multiinstanz kann nur als statische Variable deklariert werden (Deklarationstyp "stat").

Hinweis

- Sie können auch Multiinstanzen zu Systemfunktionsbausteinen erzeugen.
 - Wurde der FB als nicht multiinstanzfähig angelegt, und Sie benötigen nun doch diese Eigenschaft, so können Sie aus dem FB eine Quelle generieren, in dieser die Bausteineigenschaft CODE_VERSION1 löschen und daraus wieder den FB übersetzen.
-

10.3.3 Eingeben der Multiinstanz im Variablendeklarationsfenster

1. Öffnen Sie den FB, von dem aus die unterlagerten FB aufgerufen werden sollen.
2. Legen Sie in der Variablendeklaration des aufrufenden FBs eine statische Variable für jeden Aufruf eines Funktionsbausteins fest, für dessen Instanz Sie keinen Instanz-Datenbaustein angeben wollen:
 - Selektieren Sie in der Variablenübersicht die Hierarchieebene "STAT".
 - Geben Sie in der Variablendetailsicht in der Spalte "Name" eine Bezeichnung für den FB-Aufruf ein.
 - Geben Sie in Spalte "Datentyp" den aufzurufenden FB absolut oder mit seinem symbolischen Namen an.
 - Eventuelle Erläuterungen können Sie in die Kommentarspalte eintragen.

Aufrufe im Anweisungsteil

Wenn Sie Multiinstanzen deklariert haben, können Sie FB-Aufrufe ohne Angabe eines Instanz-DB verwenden.

Beispiel: Ist die statische Variable "Name: Motor_1 , Datentyp: FB20" definiert, lässt sich die Instanz wie folgt aufrufen:

```
Call Motor_1      // Aufruf des FB 20 ohne Instanz-DB
```

10.4 Allgemeine Hinweise zum Editieren von Anweisungen und Kommentaren

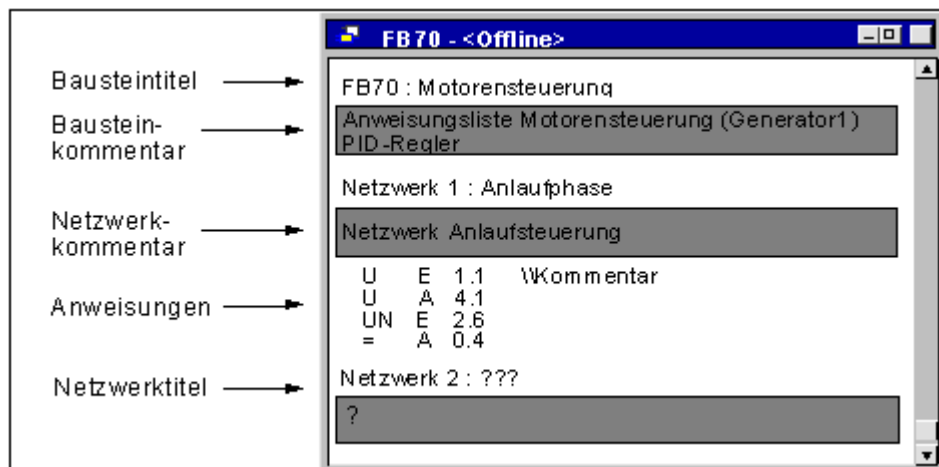
10.4.1 Aufbau des Anweisungsteils

Im Anweisungsteil beschreiben Sie den Programmablauf Ihres Codebausteins. Sie geben dazu die jeweiligen Anweisungen in Abhängigkeit der gewählten Programmiersprache in Netzwerken ein. Der Editor führt nach der Eingabe einer Anweisung sofort eine Syntaxprüfung durch und zeigt Ihnen Fehler rot und kursiv an.

Der Anweisungsteil für einen Codebaustein besteht in den häufigsten Fällen aus mehreren Netzwerken, die wiederum aus einer Liste von Anweisungen bestehen.

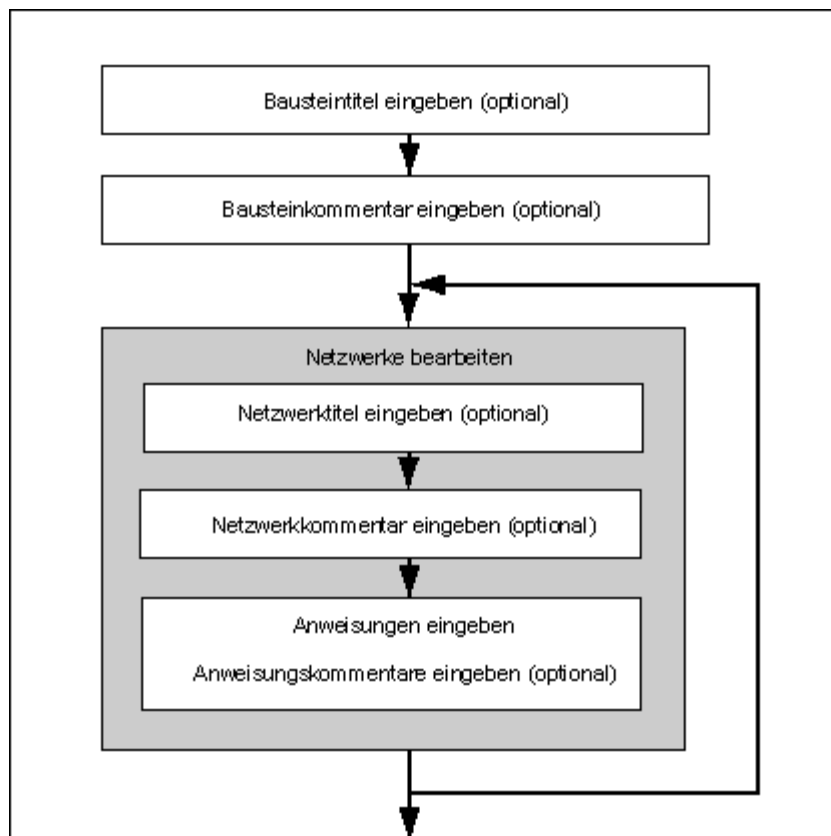
In einem Anweisungsteil können Sie Bausteintitel, Bausteinkommentare, Netzwerktitel, Netzwerkkommentare und Anweisungszeilen innerhalb der Netzwerke bearbeiten.

Aufbau des Anweisungsteils am Beispiel der Programmiersprache AWL



10.4.2 Vorgehensweise beim Eingeben von Anweisungen

Die Bestandteile des Anweisungsteils können Sie grundsätzlich in beliebiger Reihenfolge bearbeiten. Wenn Sie zum ersten Mal einen Baustein programmieren, empfiehlt sich folgendes Vorgehen:



Änderungen können Sie zusätzlich zum Einfügemodus auch im Überschreibmodus vornehmen. Zwischen beiden Modi wechseln Sie mit der INSERT-Taste.

10.4.3 Eingeben von globalen Symbolen in ein Programm

Mit dem Menübefehl **Einfügen > Symbol** können Sie Symbole in den Anweisungsteil Ihres Programms einfügen. Steht der Cursor am Beginn, am Ende oder innerhalb einer Zeichenkette, so ist bereits das Symbol markiert, das mit dieser Zeichenkette beginnt - falls ein solches Symbol existiert. Wenn Sie die Zeichenkette ändern, wird die Markierung in der Liste nachgeführt.

Trennzeichen für Beginn und Ende einer Zeichenkette sind z. B. Leerzeichen, Punkt, Doppelpunkt. Innerhalb von globalen Symbolen werden keine Trennzeichen interpretiert.

Beim Einfügen von Symbolen können Sie wie folgt vorgehen:

1. Geben Sie die Anfangsbuchstaben des gewünschten Symbols im Programm ein.
2. Drücken Sie gleichzeitig die CTRL-Taste und die J-Taste, um die Liste der Symbole anzuzeigen. Das erste Symbol mit den eingegebenen Anfangsbuchstaben ist bereits markiert.
3. Übernehmen Sie das Symbol durch Drücken der RETURN-Taste oder wählen Sie ein anderes Symbol aus.

Daraufhin wird das in doppelte Hochkommas eingeschlossene Symbol anstelle der Anfangsbuchstaben eingefügt.

Generell gilt: Steht der Cursor am Beginn, am Ende oder innerhalb einer Zeichenkette, so wird beim Einfügen eines Symbols diese Zeichenkette durch das in doppelte Hochkommas eingeschlossene Symbol ersetzt.

10.4.4 Titel und Kommentare zu Bausteinen und Netzwerken

Kommentare erhöhen die Lesbarkeit Ihres Anwenderprogramms und ermöglichen somit eine effektivere Inbetriebnahme und evtl. Fehlersuche. Sie sind ein wichtiger Bestandteil der Programmdokumentation und sollten auf jeden Fall verwendet werden.

Kommentare für KOP, FUP und AWL-Programme

Folgende Kommentare stehen Ihnen zur Verfügung:

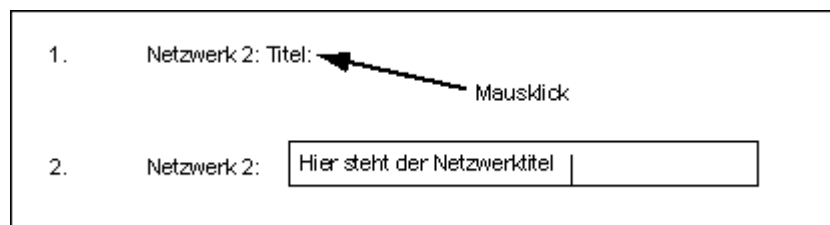
- Bausteintitel: Überschrift eines Bausteins (max. 64 Zeichen).
- Bausteinkommentar: Dokumentieren des gesamten Codebausteins, wie z. B. der Zweck des Bausteins.
- Netzwerktitel: Überschrift eines Netzwerks (max. 64 Zeichen).
- Netzwerkkommentar: Dokumentieren der Funktion einzelner Netzwerke.
- Kommentarspalte der Variablendetailsicht: Kommentare für die deklarierten Lokaldaten.
- Symbolkommentar: Kommentare, die bei der Definition des Namens in der Symboltabelle für einen Operanden angegeben worden sind.
Sie können sich diese Kommentare mit dem Menübefehl **Ansicht > Anzeigen mit > Symbolinformation** anzeigen lassen.

Bausteintitel und Netzwerktitel sowie Bausteinkommentare oder Netzwerkkommentare können Sie im Anweisungsteil eines Codebausteins eingeben.

Baustein- oder Netzwerktitel

Um Baustein- oder Netzwerktitel einzugeben, positionieren Sie den Cursor auf das Wort "Titel:" rechts neben dem Bausteinnamen bzw. Netzwerknamen (z. B. Netzwerk 1: Titel:). Es öffnet sich ein Textfeld, in das Sie den Titel eingeben. Dieser kann maximal 64 Zeichen lang sein.

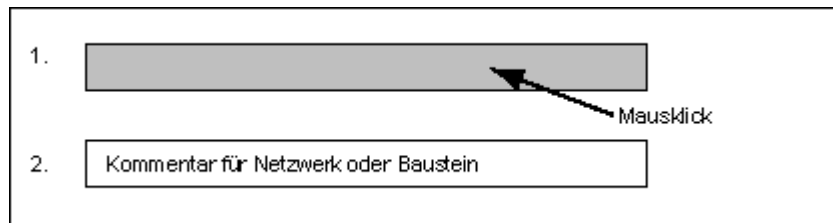
Bausteinkommentare beziehen sich auf den gesamten Codebaustein. Dort können Sie die Funktion desselben beschreiben. Netzwerkkommentare beziehen sich auf einzelne Netzwerke und beschreiben deren Einzelheiten.



Um Netzwerktitel automatisch zu vergeben, wählen Sie den Menübefehl **Extras > Einstellungen** und klicken Sie im Register "Allgemein" die Option "Automatische Netzwerktitelvergabe" an. Dann wird der Symbolkommentar des ersten eingegebenen Operanden als Netzwerktitel übernommen.

Baustein- und Netzwerkkommentare

Sie können die Ansicht der grauen Kommentarfelder über den Menübefehl **Ansicht > Anzeigen mit > Kommentar** ein- bzw. ausblenden. Durch einen Doppelklick auf ein solches Kommentarfeld wird das Textfeld geöffnet, in das Sie nun Ihre Erläuterungen eingeben können. Pro Baustein stehen Ihnen 64 KByte für Baustein- und Netzwerkkommentare zur Verfügung.



10.4.5 Eingeben von Baustein-/Netzwerkkommentaren

1. Schalten Sie mit dem Menübefehl **Ansicht > Anzeigen mit > Kommentar** die Anzeige der Kommentare ein (Häckchen vor dem Menübefehl).
2. Positionieren Sie den Cursor mit einem Mausclick in das graue Feld unter dem Bausteinnamen bzw. Netzwerknamen. Das zuvor graue Kommentarfeld wird weiß und umrandet dargestellt.
3. Geben Sie im geöffneten Textfeld den Kommentar ein. Je Baustein stehen Ihnen insgesamt 64 kBytes für Baustein- und Netzwerkkommentare zur Verfügung.
4. Verlassen Sie das Textfeld durch einen Mausclick außerhalb des Textfeldes, durch Betätigen der TAB-Taste oder mit der Tastenkombination SHIFT + TAB.
5. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Anzeigen mit > Kommentar** die Anzeige der Kommentare wieder aus.

10.4.6 Arbeiten mit Netzwerkvorlagen

Wenn Sie beim Programmieren von Bausteinen Netzwerke mehrfach verwenden möchten, so können Sie diese als Netzwerkvorlage ggf. mit Platzhaltern (z. B. für Operanden) in einer Bibliothek ablegen. Die Bibliothek muss vor dem Erstellen einer Netzwerkvorlage bereits vorhanden sein.

Netzwerkvorlage erstellen

Legen Sie ggf. eine neue Bibliothek im SIMATIC Manager an und fügen Sie mit dem Menübefehl **Einfügen > Programm > S7-Programm** ein Programm in die Bibliothek ein.

1. Öffnen Sie den Baustein, der das oder die Netzwerke enthält, aus denen Sie eine Netzwerkvorlage erstellen möchten.
2. Ersetzen Sie im geöffneten Baustein ggf. Titel, Kommentar oder Operanden durch Platzhalter. Als Platzhalter können Sie die Zeichenketten %00 bis %99 verwenden. Platzhalter für Operanden werden rot dargestellt. Dies stört aber hier nicht, da Sie den Baustein nach der Erstellung der Netzwerkvorlage nicht speichern.
Die Platzhalter können Sie später beim Einfügen der Netzwerkvorlage in einen Baustein wieder durch geeignete Operanden ersetzen.
3. Markieren Sie "Netzwerk <Nr>" des Netzwerks oder der Netzwerke, die in die Netzwerkvorlage aufgenommen werden sollen.
4. Wählen Sie den Menübefehl **Bearbeiten > Netzwerkvorlage** erstellen.
5. Geben Sie im angezeigten Dialogfeld einen aussagefähigen Kommentar zu den verwendeten Platzhaltern ein.
6. Beenden Sie das Dialogfeld durch Klicken auf die Schaltfläche "OK".
7. Markieren Sie im nächsten Dialogfeld den **Quellen-Ordner** des S7-Programms in Ihrer Netzvorlagen-Bibliothek und geben Sie einen Namen für die Netzwerkvorlage ein.
8. Bestätigen Sie anschließend die Eingabe durch Klicken auf die Schaltfläche "OK". Die Netzwerkvorlage wird in der ausgewählten Bibliothek abgelegt.
9. Schließen Sie den Baustein, ohne ihn zu speichern.

Netzwerk anhand einer Netzwerkvorlage einfügen

1. Öffnen Sie den Baustein, in dem Sie das neue Netzwerk einfügen möchten.
2. Klicken Sie im Baustein in das Netzwerk, hinter dem Sie das Netzwerk entsprechend der Netzwerkvorlage einfügen möchten.
3. Öffnen Sie das Register "Programmelemente" (Menübefehl **Einfügen > Programmelemente**).
4. Öffnen Sie den Ordner "S7-Programm" der betreffenden Bibliothek.
5. Doppelklicken Sie auf die Netzwerkvorlage.
6. Geben Sie im angezeigten Dialogfeld die gewünschten Ersetzungen für die Platzhalter in der Netzwerkvorlage an.
7. Klicken Sie auf die Schaltfläche "OK". Die Netzwerkvorlage wird im Anschluss an das aktuelle Netzwerk eingefügt.

Hinweis

Sie können die Vorlage auch mit Drag & Drop aus dem Register in das Editor-Fenster ziehen.

10.4.7 Suchfunktion für Fehler im Anweisungsteil

Aufgrund der roten Darstellung sind Fehler im Anweisungsteil leicht zu erkennen. Um leichter zu Fehlern zu navigieren, die außerhalb des gerade sichtbaren Bereiches liegen, bietet der Editor die zwei Suchfunktionen **Bearbeiten > Gehe zu > Vorheriger Fehler/Nächster Fehler** an.

Die Fehlersuche ist netzwerkübergreifend. Das heißt, es wird im gesamten Anweisungsteil gesucht, nicht nur innerhalb eines Netzwerkes oder des gerade sichtbaren Bereichs.

Wenn Sie die Statuszeile mit dem Menübefehl **Ansicht > Statuszeile** einschalten, werden Ihnen dort Hinweise zu den Fehlern angezeigt.

Fehlerkorrekturen und Änderungen können Sie auch im Überschreibmodus vornehmen. Zwischen Einfüge- und Überschreibmodus wechseln Sie mit der INSERT-Taste.

10.5 Editieren von KOP-Anweisungen im Anweisungsteil

10.5.1 Einstellungen für Programmiersprache KOP

Einstellen des Layout für KOP

Sie können für die Programmerstellung in der Programmiersprache KOP das Layout festlegen. Das ausgewählte Format (DIN A4 Hochformat/Querformat/maximale Größe) wirkt sich auf die Anzahl der in einem Zweig darstellbaren KOP-Elemente aus.

1. Wählen Sie den Menübefehl **Extras > Einstellungen**.
2. Wählen Sie im nachfolgenden Dialogfeld das Register "KOP/FUP".
3. Wählen Sie im Listenfeld Layout das gewünschte Format. Geben Sie die gewünschte Formatgröße an.

Einstellungen zum Drucken

Wenn Sie den KOP-Anweisungsteil ausdrucken möchten, so sollten Sie bereits vor der Erstellung des Anweisungsteils das geeignete Seitenformat einstellen.

Einstellungen im Register "KOP/FUP"

Im Register "KOP/FUP", auf das Sie über **Extras > Einstellungen** zugreifen, können Sie Grundeinstellungen vornehmen, z. B. zum Layout und zur Operandenfeldbreite.

10.5.2 Regeln für die Eingabe von KOP-Anweisungen

Die Beschreibung der Sprache "KOP" finden Sie im Handbuch "KOP für S7-300/400 - Bausteine programmieren" oder in der Online-Hilfe zu KOP.

Ein KOP-Netzwerk kann aus mehreren Elementen in mehreren Zweigen bestehen. Alle Elemente und Zweige müssen untereinander verbunden sein, wobei die linke Stromschiene nicht als Verbindung zählt (IEC 1131-3).

Bei der Programmierung in KOP müssen Sie einige Regeln beachten. Auf eventuelle Fehler werden Sie durch Fehlermeldungen hingewiesen.

Abschluss eines KOP-Netzwerks

Jedes KOP-Netzwerk muss einen Abschluss in Form einer Spule oder einer Box haben. Folgende KOP-Elemente dürfen Sie jedoch nicht als Abschluss eines Netzwerkes verwenden:

- Vergleichsboxen
- Spulen für Konnektoren $_/(#)_ /$
- Spulen für positive $_/(P)_ /$ oder negative $_/(N)_ /$ Flankenauswertung

Platzierung von Boxen

Der Ausgangspunkt des Zweiges für einen Boxanschluss muss immer die linke Stromschiene sein. Im Zweig vor der Box dürfen jedoch logische Verknüpfungen oder andere Boxen vorhanden sein.

Platzierung von Spulen

Spulen werden automatisch an den rechten Rand des Netzwerks platziert, wo sie den Abschluss eines Zweiges bilden.

Ausnahmen: Spulen für Konnektoren $_/(#)_ /$ und positive $_/(P)_ /$ oder negative $_/(N)_ /$ Flankenauswertung können weder ganz links noch ganz rechts im Zweig platziert werden. Auch in Parallelzweigen sind sie nicht zulässig.

Unter den Spulen gibt es solche, die eine boolesche Verknüpfung verlangen, und solche, die keine boolesche Verknüpfung haben dürfen.

- Spulen mit erforderlicher boolescher Verknüpfung:
 - Ausgang $_()$, Ausgang setzen $_(S)$, Ausgang rücksetzen $_(R)$
 - Konnektor $_(#)_ /$, positive Flanke $_/(P)_ /$, negative Flanke $_/(N)_ /$
 - alle Zähler- und Timerspulen
 - Springe wenn 0 $_/(JMPN)$
 - Master Control Relais einschalten $_/(MCR<)$
 - Verknüpfungsergebnisses ins BR-Register laden $_/(SAVE)$
 - Springe zurück $_/(RET)$
- Spulen, die keine boolesche Verknüpfung erlauben:
 - Master Control Relais Anfang $_/(MCRA)$
 - Master Control Relais Ende $_/(MCRD)$
 - Datenblock Öffnen $_/(OPN)$
 - Master Control Relais ausschalten $_/(MCR>)$

Alle anderen Spulen können wahlweise eine boolesche Verknüpfung haben oder auch nicht.

Folgende Spulen dürfen Sie **nicht als Parallelausgang** verwenden:

- Springe wenn 0 $_/(JMPN)$
- Springe wenn 1 $_/(JMP)$
- Bausteinaufruf $_/(CALL)$
- Rücksprung $_/(RET)$

Freigabeeingang/ Freigabeausgang

Die Beschaltung des Freigabeeingangs "EN" bzw. des Freigabeausgangs "ENO" von Boxen kann erfolgen, ist aber nicht zwingend erforderlich.

Entfernen und Ändern

Besteht eine Verzweigung aus nur einem Element, so wird mit dem Entfernen dieses Elementes der gesamte Zweig entfernt.

Mit dem Entfernen einer Box werden mit Ausnahme des Hauptzweiges auch alle Zweige, die mit booleschen Eingängen der Box verbunden sind, entfernt.

Zum einfachen Austauschen von Elementen gleichen Typs steht Ihnen der Überschreibmodus zur Verfügung.

Parallele Verzweigungen

- Ergänzen Sie ODER-Verzweigungen von links nach rechts.
- Parallelzweige werden nach unten geöffnet und nach oben geschlossen.
- Parallelzweige werden immer hinter dem markierten KOP-Element geöffnet.
- Parallelzweige werden immer hinter dem markierten KOP-Element geschlossen.
- Um einen Parallelzweig zu löschen, müssen Sie alle KOP-Elemente dieses Zweiges löschen. Beim Entfernen des letzten KOP-Elements der Verzweigung wird der Rest der Verzweigung ebenfalls entfernt.

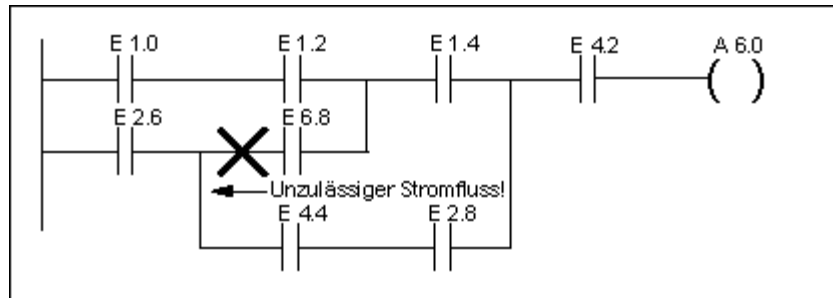
Konstanten

Binäre Verknüpfungen können nicht mit Konstanten (z. B. TRUE oder FALSE) belegt werden. Verwenden Sie stattdessen Operanden mit dem Datentyp BOOL.

10.5.3 Unzulässige Verschaltungen in KOP

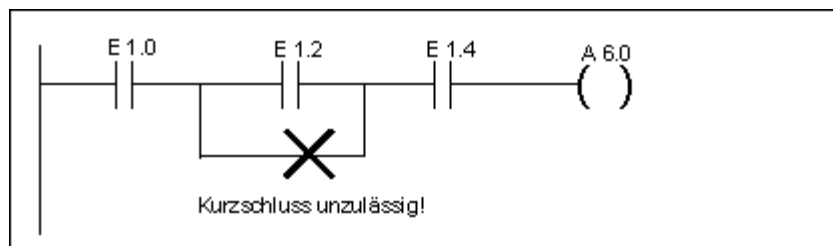
Stromfluss von rechts nach links

Es können keine Zweige editiert werden, die einen Stromfluss in umgekehrter Richtung verursachen könnten. Das folgende Bild zeigt ein Beispiel. Beim Signalzustand "0" an E 1.4 käme es an E 6.8 zu einem Stromfluss von rechts nach links. Dieses ist nicht erlaubt.



Kurzschluss

Es können keine Zweige editiert werden, die einen Kurzschluss verursachen. Das folgende Bild zeigt ein Beispiel:



10.6 Editieren von FUP-Anweisungen im Anweisungsteil

10.6.1 Einstellungen für die Programmiersprache FUP

Einstellen des Layout für FUP

Sie können für die Programmerstellung in der Programmiersprache FUP das Layout festlegen. Das ausgewählte Format (DIN A4 Hochformat/Querformat/maximale Größe) wirkt sich auf die Anzahl der in einem Zweig darstellbaren FUP-Elemente aus.

1. Wählen Sie den Menübefehl **Extras > Einstellungen**.
2. Wählen Sie im nachfolgenden Dialogfeld das Register "KOP/FUP".
3. Wählen Sie im Listenfeld "Layout" das gewünschte Format. Geben Sie die gewünschte Formatgröße an.

Hinweise zum Drucken

Wenn Sie den FUP-Anweisungsteil ausdrucken möchten, so sollten Sie bereits vor der Erstellung des Anweisungsteils das geeignete Seitenformat einstellen.

Register KOP/FUP unter Extras > Einstellungen

Im Register "KOP/FUP", auf das Sie über **Extras > Einstellungen** zugreifen, können Sie Grundeinstellungen vornehmen, z. B. zum Layout und zur Operandenfeldbreite.

10.6.2 Regeln für die Eingabe von FUP-Anweisungen

Die Beschreibung der Sprache "FUP" finden Sie im Handbuch "FUP für S7-300/400 - Bausteine programmieren" oder in der Online-Hilfe zu FUP.

Ein FUP-Netzwerk kann aus mehreren Elementen bestehen. Alle Elemente müssen untereinander verbunden sein (IEC 1131-3).

Bei der Programmierung in FUP müssen Sie einige Regeln beachten. Auf eventuelle Fehler werden Sie durch Fehlermeldungen hingewiesen.

Einfügen und Editieren von Adressen und Parametern

Beim Einfügen eines FUP-Elements werden die Zeichenfolgen "???" und "..." als Platzhalter für Adressen bzw. Parameter eingesetzt.

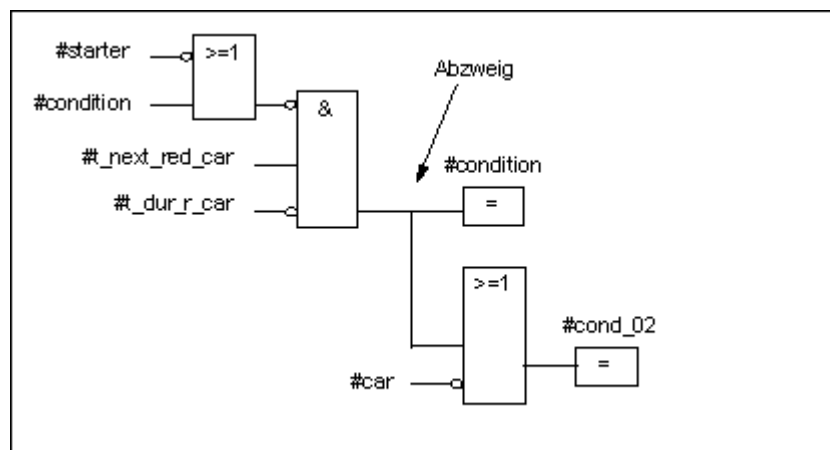
- Die rot dargestellte Zeichenfolge "???" kennzeichnet Adressen und Parameter, die beschaltet werden müssen.
- Die schwarz dargestellte Zeichenfolge "..." kennzeichnet Adressen und Parameter, die beschaltet werden können.

Wenn Sie den Mauszeiger über den Platzhalter bewegen, wird der erwartete Datentyp angezeigt.

Platzierung von Boxen

An Boxen mit binären Verknüpfungen (&, >=1, XOR) können Standard-Boxen (Flipflops, Zähler, Zeiten, Rechenoperationen usw.) angefügt werden. Ausgenommen von dieser Regelung sind Vergleichsboxen.

In einem Netzwerk dürfen keine voneinander getrennten Verknüpfungen mit separaten Ausgängen programmiert werden. Sie können jedoch mit Hilfe des Abzweigs einer Verknüpfungskette mehrere Zuweisungen zuordnen. Nachfolgendes Bild zeigt ein Netzwerk mit zwei Zuweisungen.



Die folgenden Boxen dürfen nur am rechten Rand der Verknüpfungskette platziert werden, wo sie den Abschluss der Kette bilden:

- Zähleranfangswert setzen
- Vorwärtszählen, Rückwärtszählen
- Zeit als Impuls starten, Zeit als verlängerten Impuls starten
- Zeit als Einschaltverzögerung/Ausschaltverzögerung starten

Unter den Boxen gibt es solche, die eine boolesche Verknüpfung verlangen, und solche, die keine boolesche Verknüpfung haben dürfen.

Boxen mit erforderlicher boolescher Verknüpfung:

- Ausgang, Ausgang setzen, Ausgang rücksetzen `_[R]`
- Konnektor `_[#]_`, positive Flanke `_[P]_`, negative Flanke `_[N]_`
- alle Zähler- und Timerboxen
- Springe wenn 0 `_[JMPN]`
- Master Control Relais einschalten `_[MCR<]`
- Verknüpfungsergebnis ins BIE-Register laden `_[SAVE]`
- Springe zurück `_[RET]`

Boxen, die keine boolesche Verknüpfung erlauben:

- Master Control Relais Anfang `[MCRA]`
- Master Control Relais Ende `[MCRD]`
- Datenblock öffnen `[OPN]`
- Master Control Relais ausschalten `[MCR>]`

Alle anderen Boxen können wahlweise eine boolesche Verknüpfung haben oder auch nicht.

Freigabeeingang/-Freigabeausgang

Die Beschaltung des Freigabeeingangs "EN" bzw. des Freigabeausgangs "ENO" von Boxen kann erfolgen, ist aber nicht zwingend erforderlich.

Entfernen und Ändern

Mit dem Entfernen einer Box werden mit Ausnahme des Hauptzweiges auch alle Zweige, die mit booleschen Eingängen der Box verbunden sind, entfernt.

Zum einfachen Austauschen von Elementen gleichen Typs steht Ihnen der Überschreibmodus zur Verfügung.

Konstanten

Binäre Verknüpfungen können nicht mit Konstanten (z. B. TRUE oder FALSE) belegt werden. Verwenden Sie stattdessen Operanden mit dem Datentyp BOOL.

10.7 Editieren von AWL-Anweisungen im Anweisungsteil

10.7.1 Einstellungen für die Programmiersprache AWL

Einstellen der Mnemonik

Sie können zwischen zwei Einstellungen der Mnemonik wählen:

- Deutsch oder
- Englisch.

Die Mnemonik stellen Sie vor dem Öffnen eines Bausteins im SIMATIC Manager mit dem Menübefehl **Extras > Einstellungen** im Register "Sprache" ein. Während der Bausteinbearbeitung kann die Mnemonik nicht geändert werden.

Die **Bausteineigenschaften** bearbeiten Sie über einen eigenen Dialog.

Im Editor können Sie mehrere Bausteine geöffnet haben und nach Wunsch abwechselnd bearbeiten.

10.7.2 Regeln für die Eingabe von AWL-Anweisungen

Die Beschreibung der Sprache "AWL" finden Sie im Handbuch "AWL für S7-300/400 - Bausteine programmieren" oder in der Online-Hilfe zu AWL (Sprachbeschreibungen).

Bei der inkrementellen Eingabe von AWL-Anweisungen müssen Sie folgende Grundregeln beachten:

- Achten Sie auf die Reihenfolge der Programmierung der Bausteine. Aufgerufene Bausteine müssen vor den aufrufenden Bausteinen programmiert werden.
- Eine Anweisung besteht aus der Angabe von Sprungmarke (optional), Operation, Operand und Kommentar (optional).
Beispiel: M001: U E1.0 //Kommentar
- Jede Anweisung steht in einer eigenen Zeile.
- Sie können je Baustein maximal 999 Netzwerke eingeben.
- Je Netzwerk können Sie etwa 2000 Zeilen eingeben. Bei vergrößerter oder verkleinerter Ansicht sind entsprechend mehr oder weniger Zeilen möglich.
- Bei der Eingabe von Operationen oder absoluten Adressen wird nicht zwischen Groß- und Kleinschreibung unterschieden.

10.8 Aktualisieren von Bausteinaufrufen

Mit dem Menübefehl **Bearbeiten > Aufruf > Aktualisieren** in "KOP/AWL/FUP-Bausteine programmieren" können Sie ungültig gewordene Bausteinaufrufe automatisch aktualisieren. Nach folgenden Schnittstellenänderungen muss eine Aktualisierung ausgeführt werden:

- Einfügen neuer Formalparameter
- Löschen von Formalparametern
- Namensänderung von Formalparametern
- Typänderung von Formalparametern
- Änderung der Reihenfolge (Umkopieren) von Formalparametern

Bei der Zuordnung von Formal- und Aktualeite wird nach folgenden Regeln in der angegebenen Reihenfolge vorgegangen:

1. **Gleiche Parameternamen:**
Die Aktualparameter werden automatisch zugeordnet, wenn der Name des Formalparameters gleich geblieben ist.
Sonderfall: Die Vorverknüpfung von binären Eingangsparametern kann in KOP und FUP nur bei gleichem Datentyp (BOOL) automatisch zugeordnet werden. Wurde der Datentyp in einem solchen Fall geändert, so bleibt die Vorverknüpfung als offener Zweig bestehen.
2. **Gleiche Parameterdatentypen:**
Nachdem die Parameter mit gleichen Namen zugeordnet wurden werden nun noch nicht zugeordnete Aktualparameter zu Formalparametern mit gleichem Datentyp wie der "alte" Formalparameter zugeordnet.
3. **Gleiche Parameterposition:**
Die nach der Zuordnung gemäß Regel 1 und 2 noch nicht zugeordnete Aktualparameter werden nun gemäß ihrer Parameterposition in der "alten" Schnittstelle den neuen Formalseiten zugeordnet.
4. Können Aktualparameter nach den drei genannten Regeln nicht zugeordnet werden, so werden diese gelöscht bzw. bleiben im Falle von binären Vorverknüpfungen in KOP oder FUP als offene Zweige erhalten.

Überprüfen Sie nach Ausführung dieser Funktion die vorgenommenen Änderungen in der Variablendeklarationstabelle und im Anweisungsteil des Programms.

10.8.1 Ändern von Schnittstellen

Bei Offlinebausteinen, die mit STEP 7 Version 5 bearbeitet wurden, können Sie Schnittstellenänderungen auch im inkrementellen Editor durchführen:

1. Stellen Sie sicher, dass alle Bausteine mit STEP 7 Version 5 übersetzt wurden. Generieren Sie dazu eine Quelle aller Bausteine und übersetzen Sie diese.
2. Ändern Sie die Schnittstelle des betroffenen Bausteins.
3. Öffnen Sie nun nacheinander alle Aufrufer - die entsprechenden Aufrufe sind rot dargestellt.
4. Nutzen Sie die Funktion **Bearbeiten > Aufruf > Aktualisieren**.
5. Generieren Sie die betroffenen Instanzdatenbausteine neu.

Hinweis

- Schnittstellenänderungen an einem Online geöffneten Baustein können zum Stop der CPU führen.
 - Umverdrahten von Bausteinaufrufen
Ändern Sie zuerst die Nummern der aufgerufenen Bausteine und führen Sie anschließend die Funktion Umverdrahten aus, um die Aufrufe anzupassen.
-

10.9 Speichern von Codebausteinen

Um neu erstellte Bausteine bzw. Änderungen im Anweisungsteil von Codebausteinen oder in Deklarationstabellen in die PG-Datenhaltung zu übernehmen, müssen Sie den entsprechenden Baustein abspeichern. Dazu werden die Daten auf die Festplatte des PG geschrieben.

Speichern von Bausteinen auf die Festplatte des PG:

1. Aktivieren Sie das Arbeitsfenster des zu speichernden Bausteins.
2. Wählen Sie:
 - den Menübefehl **Datei > Speichern**, wenn Sie den Baustein unter gleichem Namen abspeichern möchten.
 - den Menübefehl **Datei > Speichern unter...**, wenn Sie den Baustein unter einem anderen S7-Anwenderprogramm bzw. unter anderem Namen abspeichern möchten. Geben Sie im folgenden Dialogfeld den neuen Pfad bzw. den neuen Baustein an.

In beiden Fällen wird der Baustein nur bei fehlerloser Syntax gespeichert. Syntaktische Fehler werden bei der Erstellung sofort erkannt und rot dargestellt. Diese Fehler müssen vor dem Speichern des Bausteins behoben sein.

Hinweis

- Das Speichern von Bausteinen oder Quellen unter anderen Projekten oder Bibliotheken können Sie auch im SIMATIC Manager durchführen (z. B. durch Drag&Drop).
 - Das Speichern von Bausteinen oder kompletten Anwenderprogrammen auf eine Memory Card wird nur im SIMATIC Manager durchgeführt.
 - Wenn beim Speichern oder Übersetzen großer Bausteine Probleme auftreten, sollten Sie das Projekt reorganisieren. Rufen Sie dazu den Menübefehl **Datei > Reorganisieren** im SIMATIC Manager auf. Versuchen Sie anschließend das Speichern bzw. Übersetzen erneut.
-

11 Erstellen von Datenbausteinen

11.1 Grundlagen zum Erstellen von Datenbausteinen

Der Datenbaustein ist ein Baustein, indem Sie z. B. Werte hinterlegen, auf die Ihre Maschine oder Anlage zurückgreift. Im Gegensatz zum Codebaustein, der mit einer der Programmiersprachen KOP/FUP oder AWL programmiert wurde, enthält der Datenbaustein nur den Variablendeklarationsteil. Der Anweisungsteil entfällt und damit auch das Programmieren von Netzwerken.

Nach dem Öffnen eines Datenbausteins können Sie sich den Baustein entweder in der Deklarationssicht oder in der Datensicht anzeigen lassen. Umschalten können Sie mit den Menübefehlen **Ansicht > Deklarationssicht** und **Ansicht > Datensicht**.

Deklarationssicht

Wählen Sie die Deklarationssicht, um

- die Datenstruktur globaler DBs abzulesen oder festzulegen,
- die Datenstruktur von DBs mit zugeordnetem anwenderdefinierten Datentyp (UDT) abzulesen,
- die Datenstruktur von DBs mit zugeordnetem Funktionsbaustein abzulesen.

Die Struktur von Datenbausteinen, die einem FB oder einem anwenderdefinierten Datentyp zugeordnet sind, kann nicht geändert werden. Dazu müsste zunächst der entsprechende FB bzw. UDT geändert und anschließend der DB neu angelegt werden.

Datensicht

Wählen Sie die Datensicht, um Daten zu ändern. Nur in der Datensicht können Sie sich für jedes Element den aktuellen Wert anzeigen lassen, bzw. ihn eingeben oder ändern. In der Datensicht von Datenbausteinen werden bei Variablen mit zusammengesetzten Datentypen die Elemente einzeln mit vollständigem Namen aufgelistet.

Unterschied zwischen Instanz-Datenbaustein und globalen Datenbaustein

Der globale Datenbaustein ist keinem Codebaustein zugeordnet. Er enthält Werte, die von der Anlagen oder Maschine benötigt werden, und lässt sich direkt an jeder Stelle des Programms aufrufen.

Der Instanz-Datenbaustein ist ein Baustein, der einem Codebaustein, z. B. einem Funktionsbaustein, unmittelbar zugeordnet ist. Der Instanz-Datenbaustein enthält die Daten, die in einem Funktionsbaustein in der Variablendeklarationstabelle hinterlegt wurden.

11.2 Deklarationssicht von Datenbausteinen

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Deklaration	Diese Spalte wird nur für Instanz-Datenbausteine dargestellt. Dieser Spalte können Sie entnehmen, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: <ul style="list-style-type: none">• Eingangsparameter (IN)• Ausgangsparameter (OUT)• Durchgangsparameter (IN_OUT)• statische Daten (STAT)
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangswert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert für die Variable festlegen. Bitte beachten Sie: Anfangswerte können nicht in die CPU geladen werden!
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variablen eingeben. Der Kommentar darf 79 Zeichen lang sein.

11.3 Datensicht von Datenbausteinen

Die Datensicht zeigt Ihnen die aktuellen Werte aller Variablen des Datenbausteins. Sie können diese Werte nur in der Datensicht ändern. Die Tabellendarstellung dieser Sicht ist für alle Global-Datenbausteine gleich. Für Instanz-Datenbausteine wird zusätzlich die Spalte "Deklaration" angezeigt.

Für Variablen mit zusammengesetzten Datentypen oder anwenderdefinierten Datentypen werden in der Datensicht alle Elemente einzeln in einer eigenen Zeile mit vollständigem Namen dargestellt. Wenn die Elemente im IN_OUT-Bereich eines Instanz-Datenbausteins liegen, wird in der Spalte "Aktualwert" der Zeiger auf den zusammengesetzten oder anwenderdefinierten Datentyp angezeigt.

Die Datensicht zeigt folgende Spalten:

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch vergibt.
Deklaration	Diese Spalte wird nur für Instanz-DBs dargestellt. Dieser Spalte können Sie entnehmen, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: <ul style="list-style-type: none"> • Eingangparameter(IN) • Ausgangparameter (OUT) • Durchgangparameter (IN_OUT) • statische Daten (STAT)
Name	Hierbei handelt es sich um den Namen, der für die Variable festgelegt ist. Dieses Feld können Sie in der Datensicht nicht bearbeiten.
Typ	Hierbei handelt es sich um den Datentyp, der für die Variable festgelegt ist. Bei einem globalen Datenbaustein stehen hier nur noch elementare Datentypen, da in der Datensicht für Variablen mit zusammengesetzten oder anwenderdefinierten Datentypen die Elemente einzeln aufgelistet werden. Bei einem Instanz-Datenbaustein werden zusätzlich Parameterdatentypen angezeigt, bei Durchgangsparemtern (INOUT) mit zusammengesetzten oder anwenderdefinierten Datentyp wird in der Spalte "Aktualwert" ein Zeiger auf den Datentyp angezeigt.
Anfangswert	Hierbei handelt es sich um den Anfangswert, den Sie für die Variable festgelegt haben, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen sollte. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen. Bitte beachten Sie: Im Gegensatz zu den Aktualwerten können Anfangswerte nicht in die CPU geladen werden!

Spalte	Erläuterung
Aktualwert	<p>Offline: Hierbei handelt es sich um den Wert, den die Variable beim Öffnen des Datenbausteins bzw. nach Ihrer letzten gespeicherten Änderung hatte (auch wenn Sie den DB online geöffnet haben, wird diese Anzeige nicht aktualisiert!).</p> <p>Online: Der aktuelle Wert beim Öffnen des Datenbausteins wird angezeigt, jedoch nicht automatisch aktualisiert. Zum Aktualisieren der Anzeige drücken Sie die Taste F5.</p> <p>Dieses Feld können Sie editieren, wenn es nicht zu einem Durchgangsparameter (INOUT) mit einem zusammengesetzten oder anwenderdefinierten Datentyp gehört. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein.</p> <p>Bitte beachten Sie: Nur Aktualwerte können in die CPU geladen werden!</p>
Kommentar	<p>Hierbei handelt es sich um den Kommentar, der in der zur Dokumentation der Variable vergeben wurde. Dieses Feld können Sie in der Datensicht nicht bearbeiten.</p>

11.4 Eingeben in Datenbausteine und Speichern

11.4.1 Eingeben der Datenstruktur von globalen Datenbausteinen

Haben Sie einen Datenbaustein geöffnet, der keinem UDT oder FB zugeordnet ist, so können Sie in der Deklarationssicht des Datenbausteins seine Struktur festlegen. Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

1. Öffnen Sie einen globalen Datenbaustein, d. h. einen Baustein, der keinem UDT oder FB zugeordnet ist.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird
3. Legen Sie die Struktur fest, indem Sie die angezeigte Tabelle anhand der nachstehenden Angaben ausfüllen.

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangswert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert für die Variable festlegen.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variable eingeben. Der Kommentar darf 80 Zeichen lang sein.

11.4.2 Eingeben / Anzeige der Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs)

Eingabe

Wenn Sie einen Datenbaustein einem FB zuordnen (Instanz-DB), dann definiert die Variablendeklaration des FB die Struktur des DB. Änderungen sind nur im zugeordneten FB möglich:

1. Öffnen Sie den zugeordneten Funktionsbaustein.
2. Bearbeiten Sie die Variablendeklaration des Funktionsbausteins.
3. Legen Sie den Instanz-Datenbaustein erneut an!

Anzeige

Sie können sich in der Deklarationssicht des Instanz-DBs anzeigen lassen, wie die Variablen im FB deklariert wurden.

1. Öffnen Sie den Datenbaustein.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird.
3. Erläuterungen zur angezeigten Tabelle finden Sie im Anschluss.

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch vergibt.
Deklaration	Diese Spalte zeigt an, wie die Variablen in der Variablendeklaration des FB vereinbart worden sind: <ul style="list-style-type: none"> • Eingangsparameter (IN) • Ausgangsparameter (OUT) • Durchgangsparameter (IN_OUT) • statische Daten (STAT) Die deklarierten temporären Daten des FB sind nicht im Instanz-Datenbaustein.
Name:	Anzeige des Namens, der durch die Variablendeklaration des FB für die Variable festgelegt ist.
Typ	Anzeige des Datentyps, der durch die Variablendeklaration des FB für die Variable festgelegt ist. Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben. Werden innerhalb des FB weitere Funktionsbausteine aufgerufen, für deren Aufruf statische Variablen vereinbart worden sind, dann kann hier auch ein FB oder Systemfunktionsbaustein (SFB) als Datentyp angegeben sein.
Anfangswert:	Hierbei handelt es sich um den Vorbelegungswert, den Sie in der Variablendeklaration des FB für die Variable festgelegt haben, wenn die Software nicht den Defaultwert des Datentyps übernehmen sollte. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.

Spalte	Erläuterung
Kommentar	Hierbei handelt es sich um den Kommentar, der in der Variablendeklaration des FB zur Dokumentation des Datenelements vergeben wurde. Dieses Feld können Sie nicht bearbeiten.

Hinweis

Bei Datenbausteinen, die einem FB zugeordnet sind, können Sie nur die aktuellen Werte für die Variablen bearbeiten. Das Eingeben von aktuellen Werten für die Variablen erfolgt in der Datensicht von Datenbausteinen.

11.4.3 Eingeben der Struktur von anwenderdefinierten Datentypen (UDT)

1. Öffnen Sie den anwenderdefinierten Datentyp (UDT).
2. Lassen Sie sich die Deklarationssicht anzeigen, falls diese nicht bereits angezeigt wird.
3. Legen Sie die Struktur des anwenderdefinierten Datentyps fest, indem Sie die Reihenfolge der Variablen, ihren Datentyp und ggf. einen Anfangswert anhand der nachstehenden Angaben festlegen.
4. Sie beenden die Eingabe einer Variablen, indem Sie die Zeile mit der TAB- oder RETURN-Taste verlassen.

Spalte	Erläuterung
Adresse:	Anzeige der Adresse, die STEP 7 für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder wiederum anwenderdefinierte Datentypen haben.
Anfangswert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Default-Wert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Speichern einer Instanz des UDT (einer Variablen bzw. eines Datenbausteins) für die Variable als aktueller Wert übernommen, sofern Sie nicht explizit einen anderen aktuellen Wert vorgeben.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variablen eingeben. Der Kommentar darf 80 Zeichen lang sein.

11.4.4 Eingeben / Anzeige der Struktur von Datenbausteinen mit zugeordnetem UDT

Eingabe

Wenn Sie einen Datenbaustein einem UDT zuordnen, dann definiert die Datenstruktur des UDT die Struktur des DB. Änderungen sind nur im zugeordneten UDT möglich:

1. Öffnen Sie den anwenderdefinierten UDT.
2. Bearbeiten Sie die Struktur des anwenderdefinierten Datentyps.
3. Legen Sie den Datenbaustein neu an!

Anzeige

Sie können sich in der Deklarationssicht des DBs nur anzeigen lassen, wie die Variablen im UDT deklariert wurden.

1. Öffnen Sie den Datenbaustein.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird.
3. Erläuterungen zur angezeigten Tabelle finden Sie im Anschluss.

Die Deklarationssicht kann nicht verändert werden. Änderungen sind nur im zugeordneten UDT möglich.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch vergibt.
Name	Anzeige des Namens, der durch den UDT für die Variable festgelegt ist.
Typ:	Anzeige der im UDT festgelegten Datentypen. Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangswert	Anzeige des Vorbelegungswerts, den Sie im UDT für die Variable festgelegt haben, wenn die Software nicht den Default-Wert des Datentyps übernehmen sollte. Dieser Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.
Kommentar	Hierbei handelt es sich um den Kommentar, der durch den UDT zur Dokumentation des Datenelements festgelegt ist.

Hinweis

Bei Datenbausteinen, die einem UDT zugeordnet sind, können Sie nur die aktuellen Werte für die Variablen bearbeiten. Das Eingeben von aktuellen Werten für die Variablen erfolgt in der Datensicht von Datenbausteinen.

11.4.5 Ändern von Datenwerten in der Datensicht

Das Bearbeiten von aktuellen Werten ist nur in der Datensicht von Datenbausteinen möglich.

1. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Datensicht** die Tabellendarstellung in die Datensicht um.
2. Geben Sie die gewünschten aktuellen Werte für die Datenelemente in den Feldern der Spalte "Aktualwert" ein. Die Aktualwerte müssen kompatibel zum Datentyp der Datenelemente sein.

Fehlerhafte Eingaben bei der Bearbeitung werden sofort erkannt und rot dargestellt (z. B. wenn ein eingegebener aktueller Wert nicht kompatibel zum Datentyp ist). Diese Fehler müssen vor dem Speichern des Datenbausteins behoben sein.

Hinweis

Änderungen der Datenwerte werden erst beim Speichern von Datenbausteinen übernommen.

11.4.6 Rücksetzen von Datenwerten auf die Anfangswerte

Das Rücksetzen ist nur in der Datensicht von Datenbausteinen möglich.

1. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Datensicht** die Tabellendarstellung in die Datensicht um.
2. Wählen Sie hierfür den Menübefehl **Bearbeiten > Datenbaustein** initialisieren.

Alle Variablen werden wieder mit ihrem vorgesehenen Anfangswert versorgt, d.h. die aktuellen Werte aller Variablen werden durch den jeweiligen Anfangswert überschrieben.

Hinweis

Änderungen der Datenwerte werden erst beim Speichern von Datenbausteinen übernommen.

11.4.7 Speichern von Datenbausteinen

Um neu erstellte Datenbausteine bzw. Änderungen bei Datenwerten in Datenbausteinen in die PG-Datenhaltung zu übernehmen, müssen Sie den entsprechenden Baustein abspeichern. Dazu werden die Daten auf die Festplatte des PG geschrieben.

Speichern von Bausteinen auf die Festplatte des PG:

1. Aktivieren Sie das Arbeitsfenster des zu speichernden Bausteins.
2. Wählen Sie:
 - den Menübefehl **Datei > Speichern**, wenn Sie den Baustein unter gleichem Namen abspeichern möchten.
 - den Menübefehl **Datei > Speichern unter**, wenn Sie den Baustein unter einem anderen S7-Anwenderprogramm bzw. unter anderem Namen abspeichern möchten. Geben Sie im folgenden Dialogfeld den neuen Pfad bzw. den neuen Baustein an. Sie dürfen als DB-Nummer nicht DB 0 vergeben, da diese Nummer für das System reserviert ist.

In beiden Fällen wird der Baustein nur bei fehlerloser Syntax gespeichert. Syntaktische Fehler werden bei der Erstellung sofort erkannt und rot dargestellt. Diese Fehler müssen vor dem Speichern des Bausteins behoben sein.

Hinweis

- Das Speichern von Bausteinen oder Quellen unter anderen Projekten oder Bibliotheken können Sie auch im SIMATIC Manager durchführen (z. B. durch Drag&Drop).
 - Das Speichern von Bausteinen oder kompletten Anwenderprogrammen auf eine Memory Card wird nur im SIMATIC Manager durchgeführt.
 - Wenn beim Speichern oder Übersetzen großer Bausteine Probleme auftreten, sollten Sie das Projekt reorganisieren. Rufen Sie dazu den Menübefehl **Datei > Reorganisieren** im SIMATIC Manager auf. Versuchen Sie anschließend das Speichern bzw. Übersetzen erneut.
-

12 Parametrieren von Datenbausteinen

Mit der Funktion "Parametrieren von Datenbausteinen" können Sie außerhalb des Programmeditors KOP/AWL/FUP

- Aktualwerte von Instanz-Datenbausteinen bearbeiten und in das Zielsystem laden, ohne den gesamten Datenbaustein laden zu müssen.
- Instanz-Datenbausteine online beobachten.
- in der Parametriersicht Instanz-Datenbausteine und Multiinstanzen mit dem Systemattribut S7_techparam (Technologische Funktionen) komfortabel parametrieren und online beobachten.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie im SIMATIC Manager den Instanz-Datenbaustein über Doppelklick.
2. Bestätigen Sie die Abfrage, ob Sie die Funktion "Parametrieren von Datenbausteinen" aufrufen wollen, mit "Ja". **Ergebnis:** Der Instanz-DB wird in der Applikation "Parametrieren von Datenbausteinen" geöffnet.
3. Wählen Sie über **Ansicht > Datensicht** bzw. **Ansicht > Deklarationssicht** in welcher Sicht der Datenbaustein angezeigt werden soll. Bei Instanz-Datenbausteinen oder Multiinstanzen mit dem Systemattribut "S7_techparam" wird automatisch die technologische Parametriersicht geöffnet.
4. Bearbeiten Sie den Instanz-Datenbaustein nach Ihren Vorstellungen. Eventuelle Informationen, Warnungen und Fehler werden im Meldungsfenster angezeigt. Ein Doppelklick auf eine Warnung oder einen Fehler führt Sie zur entsprechenden Fehlerstelle.
5. Laden Sie geänderte Aktualwerte über **Zielsystem > Parametrierdaten laden** aus dem PG in die CPU, die Sie dem aktuellen S7-Programm zugewiesen haben.
6. Schalten Sie über den Menübefehl **Test > Beobachten** die Anzeige des Programmstatus für den geöffneten Baustein, ein und beobachten Sie die Bearbeitung der geladenen Aktualwerte online.

Hinweis

Bausteine, die das Systemattribut "S7_techparam" aufweisen, erkennen Sie, wenn Sie im SIMATIC Manager den Baustein selektieren, den Menübefehl **Bearbeiten > Objekteigenschaften** wählen und das Register "Attribute" öffnen.

12.1 Parametrieren von Technologischen Funktionen

Mit der Funktion "Parametrieren von Datenbausteinen" können Sie die Temperaturreglerbausteine FB 58 "TCONT_CP" und FB 59 "TCONT_S" aus der Standard Library "PID Control Blocks" komfortabel parametrieren und online beobachten.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie im SIMATIC Manager über **Datei > Öffnen > Bibliotheken** die Standard Library von STEP 7.
2. Wählen Sie "PID Control Blocks" an und klicken Sie auf "Blocks". Hier finden Sie folgende Funktionsbausteine mit dem Attribut "S7_techparam":
 - **FB 58 "TCONT_CP"**: Temperaturregler für Stellglieder mit kontinuierlichem oder impulsförmigem Eingangssignal
 - **FB 59 "TCONT_S"**: Temperaturregler für integralwirkende Stellglieder
3. Kopieren Sie den gewünschten Funktionsbaustein (FB 58 bzw. FB 59) aus der Standard Library in Ihr Projekt.
4. Legen Sie über **Einfügen > S7-Baustein > Datenbaustein** einen Instanz-DB zum gewünschten FB an.
5. Öffnen Sie im SIMATIC Manager den Instanz-DB über Doppelklick, um die Funktion "Parametrieren von Datenbausteinen" zu starten.
Ergebnis: Der Instanz-DB wird in der technologischen Sicht geöffnet. Sie können den Instanz-DB nun auf komfortable Art parametrieren und online beobachten.
6. Tragen Sie in der Technologischen Sicht die gewünschten Reglerwerte ein. Eventuelle Informationen, Warnungen und Fehler werden im Meldungsfenster angezeigt. Ein Doppelklick auf eine Warnung oder einen Fehler führt Sie zur entsprechenden Fehlerstelle.

Hinweis

Bausteine, die das Systemattribut "S7_techparam" aufweisen, erkennen Sie, wenn Sie im SIMATIC Manager den Baustein selektieren, den Menübefehl **Bearbeiten > Objekteigenschaften** wählen und das Register "Attribute" öffnen.

13 Erstellen von AWL-Quellen

13.1 Grundlagen zum Programmieren in AWL-Quellen

Sie haben die Möglichkeit, Ihr Programm bzw. Teile davon, als AWL-Quelle einzugeben und diese anschließend in Bausteine zu übersetzen. Die Quelle kann den Code für mehrere Bausteine enthalten, die anschließend in einem Lauf als Bausteine übersetzt werden.

Das Erstellen des Programms über eine Quelle hat folgende Vorteile:

- Sie können Ihre Quelle mit beliebigen ASCII-Editoren erstellen und bearbeiten, anschließend importieren und mit dieser Applikation in einzelne Bausteine übersetzen. Durch das Übersetzen werden die einzelnen Bausteine erzeugt und im S7-Anwenderprogramm abgelegt.
- Sie können mehrere Bausteine in einer Quelle programmieren.
- Sie können eine Quelle trotz Syntaxfehler abspeichern. Das ist bei der Erstellung von Codebausteinen mit inkrementeller Syntaxprüfung nicht möglich. Allerdings werden Sie erst beim Übersetzen der Quelle auf Syntaxfehler aufmerksam gemacht.

Die Quelle wird in der Syntax der Programmiersprache "Anweisungsliste (AWL)" erstellt. Die Strukturierung der Quelle in Bausteine, Variablendeklaration oder Netzwerke erfolgt über Schlüsselwörter.

Beim Erstellen von Bausteinen in AWL-Quellen sind zu beachten:

- Regeln zum Programmieren von AWL-Quellen
- Zulässige Syntax und Formate in AWL-Quellen
- Zulässige Bausteinstruktur von AWL-Quellen

13.2 Regeln zum Programmieren in AWL-Quellen

13.2.1 Regeln zur Eingabe von Anweisungen in AWL-Quellen

Eine AWL-Quelle besteht prinzipiell aus fortlaufendem Text. Damit sie sich in Bausteine übersetzen lässt, müssen bestimmte Strukturen und Syntaxvorschriften eingehalten werden.

Für die Erstellung von Anwenderprogrammen als AWL-Quelle gelten folgende allgemeine Regeln:

Thema	Regel
Syntax	Die Syntax der AWL-Anweisungen ist die gleiche wie im inkrementellen AWL-Editor. Eine Ausnahme besteht beim Aufrufbefehl CALL.
CALL	<p>In einer Quelle geben Sie die Parameter in Klammern an. Die einzelnen Parameter werden dabei durch ein Komma voneinander getrennt.</p> <p>Beispiel Aufruf FC (einzeilig): CALL FC 10 (param1 :=E0.0,param2 :=E0.1);</p> <p>Beispiel Aufruf FB (einzeilig): CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1);</p> <p>Beispiel Aufruf FB (mehrzeilig): CALL FB10, DB100 (para1 :=E0.0, para2 :=E0.1);</p> <p>Hinweis: Übergeben Sie im ASCII-Editor bei einem Bausteinaufruf die Parameter in definierter Reihenfolge. Ansonsten stimmen u. U. die Kommentaruordnung dieser Zeilen bei AWL- und Quell-Ansicht nicht überein.</p>
Groß- oder Kleinschreibung	Der Editor dieser Applikation berücksichtigt allgemein keine Groß- oder Kleinschreibung. Ausgenommen davon sind Systemattribute und Sprungmarken. Bei Eingabe von Zeichenketten (Datentyp STRING) muss ebenfalls auf Groß- und Kleinschreibung geachtet werden. Schlüsselwörter werden in Großbuchstaben dargestellt. Beim Übersetzen wird Groß- und Kleinschreibung jedoch nicht berücksichtigt, so dass Sie Schlüsselwörter in Groß- oder Kleinbuchstaben oder gemischt angeben können.
Semikolon	Kennzeichnen Sie das Ende jeder AWL-Anweisung und jeder Variablendeklaration mit einem Semikolon. Sie können mehr als eine Anweisung pro Zeile eingeben.
zwei Schrägstriche (//)	Beginnen Sie jeden Kommentar mit zwei Schrägstrichen (//) und beenden Sie jede Kommentareingabe mit der RETURN-Taste.

13.2.2 Regeln für Variablendeklaration in AWL-Quellen

Zu jedem Baustein der Quelle müssen Sie die entsprechenden Variablen deklarieren.

Die Variablendeklaration steht vor dem Anweisungsteil des Bausteins.

Die Variablen müssen - soweit vorhanden - in der Reihenfolge der Deklarationstypen deklariert werden. Damit stehen alle Variablen eines Deklarationstyps zusammen.

Während Sie in KOP, FUP oder AWL eine entsprechende Variablendeklarationstabelle ausfüllen, müssen Sie hier mit entsprechenden Schlüsselwörtern arbeiten.

Schlüsselwörter für die Variablendeklaration

Deklarationstyp	Schlüsselwörter	Gültig für ...
Eingangsparameter	"VAR_INPUT" Deklarationsliste "END_VAR"	FBs, FCs
Ausgangsparameter	"VAR_OUTPUT" Deklarationsliste "END_VAR"	FBs, FCs
Durchgangsparameter	"VAR_IN_OUT" Deklarationsliste "END_VAR"	FBs, FCs
Statische Variablen	"VAR" Deklarationsliste "END_VAR"	FBs
Temporäre Variablen	"VAR_TEMP" Deklarationsliste "END_VAR"	OBs, FBs, FCs

Das Schlüsselwort END_VAR kennzeichnet das Ende einer Deklarationsliste.

Die Deklarationsliste ist Liste der Variablen eines Deklarationstyps. Darin können den Variablen Vorbelegungswerte zugeordnet werden (Ausnahme: VAR_TEMP). Das folgende Beispiel zeigt den Aufbau eines Eintrags in der Deklarationsliste:

```
Dauer_Motor1      :      S5TIME      :=      S5T#1H_30M ;
```

Variable **Datentyp** **Vorbelegungswert**

Hinweis

- Der Variablenname muss mit einem Buchstaben oder einem Unterstrich beginnen. Sie dürfen keinen Variablennamen vergeben, der einem der reservierten Schlüsselwörter entspricht.
- Sind Variablennamen in den lokalen Deklarationen und in der Symboltabelle identisch, dann kennzeichnen Sie lokale Variablen mit einem # vor dem Namen und setzen Sie Variablen der Symboltabelle in Anführungszeichen. Andernfalls interpretiert der Baustein die Variable als lokale Variable.

13.2.3 Regeln zur Reihenfolge der Bausteine in AWL-Quellen

Aufgerufene Bausteine stehen vor den aufrufenden Bausteinen, d. h.:

- Der in den häufigsten Fällen verwendete OB 1, der andere Bausteine aufruft, steht zuletzt. Bausteine, die wiederum von den aus OB 1 aufgerufenen Bausteinen aufgerufen werden, müssen vor diesen stehen.
- Anwenderdefinierte Datentypen (UDTs) stehen vor den Bausteinen, in denen Sie verwendet werden.
- Datenbausteine mit zugeordnetem anwenderdefinierten Datentyp (UDT) stehen hinter dem UDT.
- Globale Datenbausteine stehen vor allen Bausteinen, aus denen Sie aufgerufen werden.
- Ein Instanz-Datenbausteine steht hinter dem zugeordneten Funktionsbaustein.
- Der DB 0 ist vorbelegt. Sie können also keinen DB dieses Namens erzeugen.

13.2.4 Regeln zum Festlegen von Systemattributen in AWL-Quellen

Systemattribute können Bausteinen und Parametern zugewiesen werden. Sie steuern die Meldungs- und Verbindungsprojektierung, Bedien- und Beobachtungsfunktionen und die Leittechnikprojektierung.

Für die Eingabe in der Quelle gilt Folgendes:

- Die Schlüsselwörter für Systemattribute beginnen immer mit S7_.
- Die Systemattribute stehen in geschweiften Klammern.
- Syntax: {S7_Identifier := 'string'}
Mehrere Identifier sind durch ";" getrennt.
- Systemattribute für Bausteine stehen vor den Bausteineigenschaften und nach dem Schlüsselworten ORGANIZATION_ bzw. TITLE.
- Systemattribute für Parameter stehen in der Parameterdeklaration, d. h. vor dem Doppelpunkt der Datendeklaration.
- Es wird zwischen Groß- und Kleinbuchstaben unterschieden, d. h. die Groß- und Kleinschreibung ist bei der Eingabe von Systemattributen relevant!

Die Systemattribute für Bausteine können bei der inkrementellen Eingabe mit dem Menübefehl **Datei > Eigenschaften** im Register "Attribute" kontrolliert bzw. verändert werden.

Die Systemattribute für Parameter können bei der inkrementellen Eingabe mit dem Menübefehl **Bearbeiten > Objekteigenschaften** kontrolliert bzw. verändert werden. Dabei muss der Cursor im Namensfeld der Parameterdeklaration stehen.

13.2.5 Regeln zum Festlegen von Bausteineigenschaften in AWL-Quellen

Mit Hilfe von Bausteineigenschaften können Sie die erstellten Bausteine besser identifizieren (z. B. bei der Versionspflege) oder Sie schützen Bausteine vor unberechtigten Veränderungen.

Die Bausteineigenschaften können bei der inkrementellen Eingabe mit dem Menübefehl **Datei > Eigenschaften** in den Registern "Allgemein Teil 1" und "Allgemein Teil 2" kontrolliert bzw. verändert werden.

Alle anderen Bausteineigenschaften können nur in der Quelle eingegeben werden.

Dabei gilt in der Quelle folgendes:

- Bausteineigenschaften werden vor dem Variablendeklarationsteil angegeben.
- Jede Bausteineigenschaft steht in einer eigenen Zeile.
- Am Ende der Zeile steht kein Semikolon.
- Die Bausteineigenschaften werden durch die Verwendung von Schlüsselwörtern angegeben.
- Wenn Sie Bausteineigenschaften eingeben möchten, dann ist die Reihenfolge dieser nach Tabelle der Bausteineigenschaften einzuhalten.
- Die jeweils möglichen Bausteineigenschaften einer Bausteinart sind der Zuordnung Bausteineigenschaften zu Bausteinarten zu entnehmen

Hinweis

Die Bausteineigenschaften werden auch im SIMATIC Manager in den Objekteigenschaften für einen Baustein angezeigt. Dort können auch die Eigenschaften AUTHOR, FAMILY, NAME und VERSION editiert werden.

Bausteineigenschaften und Reihenfolge

Wenn Sie Bausteineigenschaften eingeben möchten, dann ist die Reihenfolge der Eingabe nach folgender Tabelle zu beachten.

Reihenfolge	Schlüsselwort / Eigenschaft	Bedeutung	Beispiel
1.	[KNOW_HOW_PROTECT]	Bausteinschutz; ein Baustein, der mit dieser Option übersetzt wurde, lässt keinen Einblick in den Anweisungsteil zu. Die Schnittstelle des Bausteins kann eingesehen, aber nicht verändert werden.	KNOW_HOW_PROTECT
2.	[AUTHOR:]	Name des Autors, Firmenname, Abteilungsname od. andere Namen (max. 8 Zeichen, ohne Leerzeichen)	AUTHOR : Siemens, aber kein Schlüsselwort
3.	[FAMILY:]	Name der Bausteinfamilie: z. B. Regler. (max. 8 Zeichen, ohne Leerzeichen).	FAMILY : Regler, aber kein Schlüsselwort
4.	[NAME:]	Bausteinname (max. 8 Zeichen)	NAME : PID, aber kein Schlüsselwort
5.	[VERSION: int1 . int2]	Versionsnummer des Bausteins (beide Zahlen zwischen 0..15, d. h. 0.0 - 15.15)	VERSION : 3.10
6.	[CODE_VERSION1]	Kennung, ob ein FB multiinstanzfähig ist oder nicht. Wenn Sie Multiinstanzen deklarieren möchten, darf der FB diese Eigenschaft nicht haben.	CODE_VERSION1
7.	[UNLINKED] nur für DBs!	Datenbausteine mit der Eigenschaft UNLINKED werden nur im Ladespeicher abgelegt, belegen keinen Platz im Arbeitsspeicher und werden nicht in das Programm eingebunden. Auf sie kann nicht mit MC7-Befehlen zugegriffen werden. Die Inhalte solcher DB können nur mit SFC 20 BLKMOV (S7-300/S7-400) oder SFC 83 READ_DBL (S7-300C) in den Arbeitsspeicher transferiert werden.	
8.	[READ_ONLY] nur für DBs!	Schreibschutz für Datenbausteine; seine Daten können nur gelesen u. nicht verändert werden.	FAMILY= Beispiele VERSION= 3.10 READ_ONLY

13.2.6 Zulässige Bausteineigenschaften je Bausteinart

Die nachfolgenden Tabelle zeigt, welche Bausteineigenschaften bei welchen Bausteinarten vereinbart werden können.

Eigenschaft	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	•	•	•	•	–
AUTHOR	•	•	•	•	–
FAMILY	•	•	•	•	–
NAME	•	•	•	•	–
VERSION	•	•	•	•	–
UNLINKED	–	–	–	•	–
READ_ONLY	–	–	–	•	–

Bausteinschutz einrichten mit KNOW_HOW_PROTECT

Sie können für Bausteine einen Bausteinschutz einrichten, indem Sie das Schlüsselwort `KNOW_HOW_PROTECT` bei der Programmierung des Bausteins in der AWL-Quelle angeben.

Der Bausteinschutz hat folgende Konsequenzen:

- Wenn Sie später einen übersetzten Baustein im inkrementellen AWL-, FUP- oder KOP-Editor ansehen wollen, kann der Anweisungsteil des Bausteins nicht eingesehen werden.
- In der Variablendeklarationsliste des Bausteins werden nur die Variablen der Deklarationstypen `var_in`, `var_out` und `var_in_out` angezeigt. Die Variablen der Deklarationstypen `var_stat` und `var_temp` bleiben verborgen.
- Das Schlüsselwort `KNOW_HOW_PROTECT` wird vor allen anderen Bausteineigenschaften eingegeben.

Schreibschutz auf Datenbausteine einrichten mit READ_ONLY

Für Datenbausteine können Sie einen Schreibschutz einrichten, so dass sie beim Programmablauf nicht überschrieben werden können. Der Datenbaustein muss dazu als AWL-Quelle vorliegen.

Geben Sie in der Quelle das Schlüsselwort `READ_ONLY` ein. Dieses muss direkt vor den Variablendeklarationen in einer eigenen Zeile stehen.

13.3 Struktur von Bausteinen in AWL-Quellen

Die Strukturierung von Bausteinen in AWL-Quelle erfolgt über Schlüsselwörter. Abhängig von der Art des Bausteins unterscheidet man die Struktur von

- Codebausteinen
- Datenbausteinen
- anwenderdefinierten Datentypen

13.3.1 Struktur von Codebausteinen in AWL-Quellen

Ein Codebaustein besteht aus folgenden Bereichen, die jeweils durch entsprechende Schlüsselwörter gekennzeichnet werden:

- Bausteinanfang,
- gekennzeichnet durch Schlüsselwort und Bausteinnummer oder Bausteinnamen, also z. B.:
 - "ORGANIZATION_BLOCK OB 1" für einen Organisationsbaustein.
 - "FUNCTION_BLOCK FB6" für einen Funktionsbaustein.
 - "FUNCTION FC 1 : INT" für eine Funktion. Bei Funktionen wird zusätzlich der Funktionstyp angegeben. Dieser kann von elementarem oder zusammengesetztem Datentyp (außer ARRAY und STRUCT) sein und bestimmt den Datentyp des Rückgabewertes (RET_VAL). Soll kein Wert zurückgegeben werden, ist das Schlüsselwort VOID anzugeben.
- Optionaler Bausteintitel, eingeleitet mit dem Schlüsselwort "TITLE" (max. Eingabe des Titels: 64 Zeichen).
- Weiterer Kommentar, beginnend mit zwei Schrägstrichen // am Zeilenanfang.
- Eintragung der Bausteineigenschaften (optional).
- Variablendeklarationsteil.
- Anweisungsteil, beginnend mit BEGIN. Der Anweisungsteil besteht wiederum aus ein oder mehreren Netzwerken, die mit NETWORK gekennzeichnet werden. Die Angabe einer Netzwerknummer ist nicht erlaubt.
- Optionaler Netzwerktitel für jedes implementierte Netzwerk, eingeleitet mit dem Schlüsselwort "TITLE =" (max. Eingabe des Titels: 64 Zeichen).
- Weitere Kommentare je Netzwerk, beginnend mit zwei Schrägstrichen "/" am Zeilenanfang.
- Bausteinende, gekennzeichnet durch END_ORGANIZATION_BLOCK, END_FUNCTION_BLOCK oder END_FUNCTION.
- Zwischen Bausteinart und Bausteinnummer steht ein Leerzeichen. Der symbolische Bausteinname kann durch Anführungszeichen gekennzeichnet werden, um die Eindeutigkeit zwischen Namen lokaler Variablen und Namen der Symboltabelle sicherzustellen.

13.3.2 Struktur von Datenbausteinen in AWL-Quellen

Ein Datenbaustein besteht aus folgenden Bereichen, die jeweils durch entsprechende Schlüsselwörter eingeleitet werden:

- Bausteinanfang, gekennzeichnet durch Schlüsselwort und Bausteinnummer oder Bausteinnamen, also z. B. DATA_BLOCK DB 26
- Angabe des zugeordneten UDT oder des FB, dem der DB zugeordnet ist (optional).
- Optionaler Bausteintitel, eingeleitet mit dem Schlüsselwort "TITLE =" (Eingaben > 64 Zeichen werden abgeschnitten).
- Optionaler Bausteinkommentar, beginnend mit "//".
- Eintragung der Bausteineigenschaften (optional).
- Variablendeklarationsteil (optional).
- Zuweisungsteil mit Vorbelegungswerten, beginnend mit BEGIN (optional).
- Bausteinende, gekennzeichnet durch END_DATA_BLOCK.

Es gibt drei Typen von Datenbausteinen:

- Datenbausteine, benutzerdefiniert
- Datenbausteine, mit zugeordnetem anwenderdefinierten Datentyp (UDT)
- Datenbausteine, die einem FB zugeordnet sind (Instanz-DB)

13.3.3 Struktur von anwenderdefinierten Datentypen in AWL-Quellen

Ein anwenderdefinierter Datentyp besteht aus folgenden Bereichen, die jeweils durch entsprechende Schlüsselwörter eingeleitet werden:

- Bausteinanfang, gekennzeichnet durch Schlüsselwort TYPE und Nummer oder Namen, also z. B. TYPE UDT 20
- Angabe eines strukturierten Datentyps
- Bausteinende, gekennzeichnet durch END_TYPE.

Bei der Eingabe von anwenderdefinierten Datentypen müssen Sie beachten, dass sie vor den Bausteinen stehen, in denen Sie verwendet werden.

13.4 Syntax und Formate für Bausteine in AWL-Quellen

Die Formattabellen stellen die Syntax und die Formate dar, die Sie bei der Programmierung von AWL-Quellen beachten müssen. Die Syntax ist folgendermaßen dargestellt:

- Jedes Element ist in der rechten Spalte beschrieben.
- Unbedingt erforderliche Elemente werden in Anführungszeichen dargestellt.
- Die eckigen Klammern [...] bedeuten, dass der Inhalt dieser Klammern nicht unbedingt angegeben werden muss.
- Schlüsselwörter werden in Großbuchstaben angegeben.

13.4.1 Formattabelle von OBs

Nachfolgende Tabelle zeigt in Kurzform das Format für Organisationsbausteine in einer AWL-Quelle:

Aufbau	Beschreibung
"ORGANIZATION_BLOCK" ob_nr oder ob_name	ob_nr ist die Bausteinnummer, z. B. OB 1; ob_name ist der symbolische Name des Bausteins laut Symboltabelle
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten
[Bausteinkommentar]	Kommentare können hinter "/" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Variablendeklarationsteil	Deklaration von temporären Variablen
"BEGIN"	Schlüsselwort zur Trennung der AWL-Anweisungen vom Variablendeklarationsteil
NETWORK	Beginn eines Netzwerks
[TITLE=]	Netzwerktitel (max. 64 Zeichen)
[Netzwerkkommentar]	Kommentare können hinter "/" eingegeben werden
Liste der AWL-Anweisungen	Bausteinanweisungen
"END_ORGANIZATION_BLOCK"	Schlüsselwort zum Beenden des Organisationsbausteins

13.4.2 Formattabelle von FBs

Nachfolgende Tabelle zeigt in Kurzform das Format für Funktionsbausteine in einer AWL-Quelle:

Aufbau	Beschreibung
"FUNCTION_BLOCK" fb_nr oder fb_name	fb_nr ist die Bausteinnummer, z. B. FB 6; fb_name ist der symbolische Name des Bausteins laut Symboltabelle
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten
[Bausteinkommentar]	Kommentare können hinter "/" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Variablendeklarationsteil	Deklaration von Eingangs-, Ausgangs- und Durchgangsparemtern sowie von temporären oder statischen Variablen Die Deklaration der Parameter enthält ggf. die Deklarationen der Systemattribute für Parameter
"BEGIN"	Schlüsselwort zur Trennung der AWL-Anweisungen vom Variablendeklarationsteil
NETWORK	Beginn eines Netzwerks
[TITLE=]	Netzwerktitle (max. 64 Zeichen)
[Netzwerkcommentar]	Kommentare können hinter "/" eingegeben werden
Liste der AWL-Anweisungen	Bausteinanweisungen
"END_FUNCTION_BLOCK"	Schlüsselwort zum Beenden des Funktionsbausteins

13.4.3 Formattabelle von FCs

Nachfolgende Tabelle zeigt in Kurzform das Format für Funktionen in einer AWL-Quelle:

Aufbau	Beschreibung
"FUNCTION" fc_nr : fc_type oder fc_name : fc_type	fc_nr ist die Bausteinnummer, z. B. FC 5; fc_name ist der symbolische Name des Bausteins laut Symboltabelle fc_type gibt den Datentyp des Rücksprungwerts (RET_VAL) der Funktion an. Es kann sich um einen elementaren oder strukturierten Datentyp (außer ARRAY und STRUCT) oder um VOID handeln. Wollen Sie Systemattribute für den Rücksprungwert (RET_VAL) verwenden, dann müssen Sie die Systemattribute für Parameter vor den Doppelpunkt der Datendeklaration eintragen.
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten
[Bausteinkommentar]	Kommentare können hinter "/" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Variablendeklarationsteil	Deklaration von Eingangs-, Ausgangs- und Durchgangparametern sowie von temporären Variablen
"BEGIN"	Schlüsselwort zur Trennung der AWL-Anweisungen vom Variablendeklarationsteil
NETWORK	Beginn eines Netzwerks
[TITLE=]	Netzwerktitel (max. 64 Zeichen)
[Netzwerkkommentar]	Kommentare können hinter "/" eingegeben werden
Liste der AWL-Anweisungen	Bausteinanweisungen
"END_FUNCTION"	Schlüsselwort zum Beenden der Funktion

13.4.4 Formattabelle von DBs

Nachfolgende Tabelle zeigt in Kurzform das Format für Datenbausteine in einer AWL-Quelle:

Aufbau	Beschreibung
"DATA_BLOCK" db_nr oder db_name	db_nr ist die Bausteinnummer, z. B. DB 5; db_name ist der symbolische Name des Bausteins laut Symboltabelle
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten
[Bausteinkommentar]	Kommentare können hinter "/" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Deklarationsteil	Angabe von UDT oder FB, auf den sich der Baustein bezieht, als Bausteinnummer oder Name laut Symboltabelle oder Angabe des zusammengesetzten Datentyps
"BEGIN"	Schlüsselwort zum Trennen des Deklarationsteils von der Liste der Wertzuweisungen
[Zuweisung von Anfangswerten]	Variablen können bestimmte Anfangswerte zugewiesen werden. Dabei werden einzelnen Variablen Konstanten zugeordnet oder es wird auf andere Bausteine verwiesen.
"END_DATA_BLOCK"	Schlüsselwort zum Beenden des Datenbausteins

13.5 Erstellen von AWL-Quellen

13.5.1 Anlegen von AWL-Quellen

Die Quelle muss unter dem S7-Programm in einem Quell-Ordner angelegt werden. Sie können Quellen im SIMATIC Manager oder im Editor-Fenster anlegen.

Anlegen von Quellen im SIMATIC Manager

1. Öffnen Sie den entsprechenden Ordner "Quellen", in dem Sie darauf doppelklicken.
2. Wählen Sie zum Einfügen einer AWL-Quelle den Menübefehl **Einfügen > S7-Software > AWL-Quelle**.

Anlegen von Quellen im Editor-Fenster

1. Wählen Sie den Menübefehl **Datei > Neu**.
2. Wählen Sie im Dialogfeld den Quell-Ordner desselben S7-Programms, in dem sich auch das Anwenderprogramm mit den Bausteinen befindet.
3. Weisen Sie der anzulegenden Quelle einen Namen zu.
4. Bestätigen Sie mit "OK".

Die Quelle wird unter dem von Ihnen vergebenen Namen angelegt und in einem Arbeitsfenster zur weiteren Bearbeitung angezeigt.

13.5.2 Bearbeiten von S7-Quellen

In welcher Programmiersprache und mit welchem Editor die jeweilige Quelle bearbeitet wird, ist in den Objekteigenschaften der Quelle festgelegt. Damit wird für die Bearbeitung immer der richtige Editor mit der entsprechenden Programmiersprache gestartet. Das STEP 7-Basispaket unterstützt die Programmierung in AWL-Quellen.

Andere Programmiersprachen stehen Ihnen jeweils als Optionspaket zur Verfügung. Nur wenn die jeweilige Optionsoftware auf Ihrem Rechner geladen ist, können Sie durch Doppelklick auf die Quelle den entsprechenden Editor starten.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie den entsprechenden Ordner "Quellen", in dem Sie darauf doppelklicken.
2. Starten Sie den für die Bearbeitung notwendigen Editor wie folgt:
 - Führen Sie einen Doppelklick auf die jeweilige Quelle in der rechten Fensterhälfte aus.
 - Markieren Sie die entsprechende Quelle in der rechten Fensterhälfte und rufen Sie den Menübefehl **Bearbeiten > Objekt öffnen** auf.

13.5.3 Layout des Quelltextes festlegen

Um die Lesbarkeit des Textes in Quelldateien zu erhöhen, wählen Sie den Menübefehl **Extras > Einstellungen** und legen im Register "Quelltext" Format, Schriftstil und Farbe der unterschiedlichen Bestandteile des Quelltextes fest.

Sie können z. B. angeben, dass Zeilennummer angezeigt und Schlüsselwörter in Großbuchstaben angezeigt werden sollen.

13.5.4 Einfügen von Bausteinvorlagen in AWL-Quellen

Für das Programmieren in AWL-Quellen werden Ihnen Bausteinvorlagen für OBs, FBs, FCs, DBs, Instanz-DBs, DBs aus UDTs und UDTs angeboten. Die Bausteinvorlagen erleichtern Ihnen die Eingabe und Einhaltung von Syntax und Struktur.

Gehen Sie folgendermaßen vor:

1. Aktivieren Sie das Arbeitsfenster der Quelle, in die Sie eine Bausteinvorlage einfügen möchten.
2. Positionieren Sie den Cursor an die Stelle, hinter der die Bausteinvorlage eingefügt werden soll.
3. Wählen Sie den entsprechenden Menübefehl **Einfügen > Bausteinvorlage > OB/FB/FC/DB/IDB/DB aus UDT/UDT**.

Die Bausteinvorlage wird hinter der Cursorposition eingefügt.

13.5.5 Einfügen des Inhalts anderer AWL-Quellen

Sie können in Ihre AWL-Quelle den Inhalt anderer Quellen einfügen.

Dazu gehen Sie folgendermaßen vor:

1. Aktivieren Sie das Arbeitsfenster der Quelle, in die Sie den Inhalt einer anderen Quelle einfügen möchten.
2. Positionieren Sie den Cursor an die Stelle, hinter der der Inhalt der Quelle eingefügt werden soll.
3. Wählen Sie den Menübefehl **Einfügen > Objekt > Datei**.
4. Wählen Sie im nachfolgenden Dialogfeld die gewünschte Quelle aus.

Der Inhalt der ausgewählten Quelle wird hinter der Cursorposition eingefügt. Zeilenumbrüche bleiben erhalten.

13.5.6 Einfügen des Quellcodes vorhandener Bausteine in AWL-Quellen

Sie können in Ihre AWL-Quelle den Quellcode von Bausteinen einfügen, die in KOP, FUP oder AWL erstellt wurden. Dieses ist möglich für OBs, FBs, FCs, DBs und UDTs.

Dazu gehen Sie folgendermaßen vor:

1. Aktivieren Sie das Arbeitsfenster der Quelle, in die Sie einen Baustein einfügen möchten.
2. Positionieren Sie den Cursor an die Stelle, hinter der der Text des Bausteins eingefügt werden soll.
3. Wählen Sie den Menübefehl **Einfügen > Objekt > Baustein**.
4. Wählen Sie im nachfolgenden Dialogfeld den gewünschten Baustein aus.

Aus dem Baustein wird implizit eine entsprechende Quelle generiert. Der Inhalt der Quelle wird hinter der Cursorposition eingefügt.

13.5.7 Einfügen von externen Quellen

Sie können Ihre Quellen mit beliebigen ASCII-Editoren erstellen und bearbeiten, anschließend in ein Projekt importieren und in einzelne Bausteine übersetzen. Dazu müssen die Quellen in den Ordner "Quellen" des S7-Programms importiert werden, in dessen S7-Anwenderprogramm die aus der Übersetzung entstehenden Bausteine abgelegt werden sollen.

Gehen Sie folgendermaßen vor:

1. Markieren Sie den Ordner "Quellen" des S7-Programms, in den die externen Quellen importiert werden sollen.
2. Wählen Sie den Menübefehl **Einfügen > Externe Quelle**.
3. Geben Sie im anschließenden Dialogfeld die zu importierende Quelle an.

Der Dateiname der zu importierenden Quelle muss eine gültige Dateierweiterung besitzen. Aus der Dateierweiterung wird der Typ der Quelle in STEP 7 ermittelt. So entsteht z. B. aus einer Datei mit der Erweiterung **.awl** beim Importieren eine AWL-Quelle. Gültige Dateierweiterungen sind im Dialogfeld unter "Dateityp" ersichtlich.

Hinweis

Mit dem Menübefehl **Einfügen > Externe Quelle** können Sie auch Quellen einfügen, die Sie mit STEP 7 Version 1 erzeugt haben.

13.5.8 Generieren von AWL-Quellen aus Bausteinen

Sie haben die Möglichkeit, von bestehenden Bausteinen eine AWL-Quelle zu erstellen, die Sie mit jedem beliebigen Texteditor bearbeiten können. Die generierte Quelle wird im Quell-Ordner des S7-Programms erzeugt.

Gehen Sie folgendermaßen vor:

1. Wählen Sie im Programmeditor den Menübefehl **Datei > Quelle** generieren.
2. Wählen Sie im Dialogfeld den Quell-Ordner, in dem Sie die neue Quelle anlegen möchten.
3. Weisen Sie der Quelle in dem Textfeld einen Namen zu.
4. Wählen Sie im Dialogfeld "Auswahl STEP 7-Baustein" den oder die Bausteine aus, die Sie in die zuvor angegebene Quelle generieren möchten. Die ausgewählten Bausteine werden im rechten Listenfeld angezeigt.
5. Bestätigen Sie mit "OK".

Aus den ausgewählten Bausteinen wird eine zusammenhängende AWL-Quelle generiert und Ihnen zur weiteren Bearbeitung in einem Arbeitsfenster angezeigt.

13.5.9 Importieren von Quellen

Um eine Quelle aus einem beliebigen Verzeichnis in ein Projekt zu importieren, gehen Sie so vor:

1. Markieren Sie im SIMATIC Manager den Quell-Ordner, in den die Quelle aufgenommen werden soll.
2. Wählen Sie den Menübefehl **Einfügen > Externe Quelle**.
3. Wählen Sie im angezeigten Dialogfeld das Verzeichnis und die zu importierende Quelle aus.
4. Betätigen Sie die Schaltfläche "Öffnen".

13.5.10 Exportieren von Quellen

Um eine Quelle aus einem Projekt in ein beliebiges Zielverzeichnis zu exportieren, gehen Sie so vor:

1. Markieren Sie die Quelle im Ordner "Quellen".
2. Wählen Sie im SIMATIC Manager den Menübefehl **Bearbeiten > Quelle** exportieren.
3. Geben Sie im angezeigten Dialogfeld das Zielverzeichnis und den Dateinamen an.
4. Betätigen Sie die Schaltfläche "Speichern".

Hinweis

Besitzt der Objektname keine Dateierweiterung, dann wird eine aus dem Typ abgeleitete Dateierweiterung an den Dateinamen angefügt. Beispielsweise wird eine AWL-Quelle "**prog**" in die Datei "**prog.awl**" exportiert.

Besitzt der Objektname bereits eine gültige Dateierweiterung, so wird diese beibehalten und keine weitere angefügt. Beispielsweise wird eine AWL-Quelle "**prog.awl**" in die Datei "**prog.awl**" exportiert.

Besitzt ein Objektname eine ungültige Dateierweiterung (d. h. ein Punkt ist im Namen enthalten), so wird keine Dateierweiterung angehängt.

Gültige Dateierweiterungen sind im Dialogfeld "Quelle exportieren" unter "Dateityp" ersichtlich.

13.6 Speichern, Übersetzen von AWL-Quellen und Konsistenzprüfung

13.6.1 Speichern von AWL-Quellen

Sie können eine AWL-Quelle jederzeit im aktuellen Zustand abspeichern. Das Programm wird dabei nicht übersetzt, es findet keine Syntaxprüfung statt, d. h. Fehler werden mit abgespeichert.

Syntaxfehler werden erst beim Übersetzen der Quelle, bzw. bei einer Konsistenzprüfung festgestellt und gemeldet.

Quelle unter gleichem Namen speichern

1. Aktivieren Sie das Arbeitsfenster der zu speichernden Quelle.
2. Wählen Sie den Menübefehl **Datei > Speichern**.

Quelle unter anderem Namen/in anderem Projekt speichern

1. Aktivieren Sie das Arbeitsfenster der zu speichernden Quelle.
2. Wählen Sie den Menübefehl **Datei > Speichern unter**.
3. Wählen Sie im nachfolgenden Dialogfeld den Quell-Ordner an, dem die Quelle zugeordnet sein soll und geben Sie deren neuen Namen an.

13.6.2 Prüfen der Konsistenz in AWL-Quellen

Mit dem Menübefehl **Datei > Konsistenz prüfen** können Sie sich eventuelle Syntaxfehler in der AWL-Quelle anzeigen lassen. Dabei werden im Gegensatz zur Übersetzung keine Bausteine generiert.

Nach Abschluss der Konsistenzprüfung wird ein Dialogfeld aufgeblendet, das Ihnen die Gesamtzahl der gefundenen Fehler mitteilt.

Sind Fehler vorhanden, werden diese im unteren Teil des Arbeitsfensters einzeln mit Zeilenangabe aufgelistet. Diese müssen vor dem Übersetzen der Quelle behoben sein, damit alle Bausteine erzeugt werden.

13.6.3 Fehlersuche in AWL-Quellen

Das Arbeitsfenster für Quellen ist zweigeteilt. In der unteren Hälfte werden folgende Fehler aufgelistet:

- Fehler, die nach Anstoß eines Übersetzungsvorgangs mit dem Menübefehl **Datei > Übersetzen** festgestellt wurden.
- Fehler, die nach Anstoß der Konsistenzprüfung mit dem Menübefehl **Datei > Konsistenz prüfen** festgestellt wurden.

Um die Stelle eines Fehlers in der Quelle zu finden, positionieren Sie den Cursor auf die entsprechende Fehlermeldung im Register "Fehler" des Ausgabenfensters. Automatisch wird die entsprechende Textzeile im Anweisungsteil markiert. Zusätzlich befindet sich die Fehlermeldung in der Statuszeile.

13.6.4 Übersetzen von AWL-Quellen

Voraussetzungen

Um das in einer Quelle erstellte Programm in Bausteine übersetzen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Es können nur Quellen übersetzt werden, die im Ordner "Quellen" unter einem S7-Programm gespeichert sind.
- Parallel zum Ordner "Quellen" muss unter dem S7-Programm ein Ordner "Bausteine" liegen, in den die bei der Übersetzung entstehenden Bausteine gespeichert werden. Die in der Quelle programmierten Bausteine werden nur dann erzeugt, wenn die Quelle fehlerfrei übersetzt worden ist. Sind mehrere Bausteine in einer Quelle, werden nur die fehlerfreien erzeugt. Anschließend können Sie diese Bausteine einzeln öffnen, bearbeiten, in die CPU laden und testen.

Vorgehensweise im Editor

1. Öffnen Sie die Quelle, die Sie übersetzen wollen. Die Quelle muss im Quell-Ordner des S7-Programms liegen, in dessen S7-Anwenderprogramm die übersetzten Bausteine abgelegt werden sollen.
2. Wählen Sie den Menübefehl **Datei > Übersetzen**.
3. Sie erhalten das Dialogfeld "Übersetzungsprotokoll", das die Anzahl der übersetzten Zeilen und der gefundenen Syntaxfehler anzeigt.

Die in der Datei angegebenen Bausteine werden nur dann erzeugt, wenn die Quelle fehlerfrei übersetzt worden ist. Sind mehrere Bausteine in einer Quelle, werden nur die fehlerfreien erzeugt. Warnungen verhindern nicht das Erzeugen von Bausteinen.

Bei der Übersetzung erkannte Syntaxfehler werden in der unteren Hälfte des Arbeitsfensters dargestellt und müssen behoben werden, damit auch diese Bausteine erzeugt werden können.

Vorgehensweise im SIMATIC Manager

1. Öffnen Sie den entsprechenden Ordner "Quellen", in dem Sie darauf doppelklicken.
2. Wählen Sie eine oder mehrere Quelle(n) aus, die Sie übersetzen wollen. Sie können den Übersetzungslauf nicht für einen geschlossenen Ordner "Quellen" aufrufen, um damit alle enthaltenen Quellen zu übersetzen.
3. Wählen Sie den Menübefehl **Datei > Übersetzen**, um die Übersetzung zu starten. Dabei wird je nach angewählter Quelle der richtige Compiler aufgerufen. Die erfolgreich übersetzten Bausteine werden danach im Ordner "Bausteine" unter dem S7-Programm abgelegt.
Bei der Übersetzung erkannte Syntaxfehler werden in einem Dialogfeld dargestellt und müssen behoben werden, damit auch diese Bausteine erzeugt werden können.

13.7 Beispiele zu AWL-Quellen

13.7.1 Beispiele für Variablendeklarationen in AWL-Quellen

Variablen elementaren Datentyps

```
// Kommentare werden durch zwei Schrägstriche vom Deklarationsteil getrennt.
VAR_INPUT // Schlüsselwort Eingangsvariable
    in1 : INT; // Variablenname und Typ werden durch ":" getrennt
    in3 : DWORD; // Jede Variablendeklaration wird durch ein Semikolon beendet.
    in2 : INT := 10; // Optionale Festlegung des Anfangswertes in der Deklaration
END_VAR // Abschluss der Deklaration von Variablen des gleichen Deklarationstyps
VAR_OUTPUT // Schlüsselwort Ausgangsvariable
    out1 : WORD;
END_VAR // Schlüsselwort Temporäre Variable
VAR_TEMP
    temp1 : INT;
END_VAR
```

Variablen vom Datentyp Feld

```
VAR_INPUT // Eingangsvariable
    feld1 : ARRAY [1..20] of INT; // feld1 ist ein eindimensionales Feld
    feld2 : ARRAY [1..20, 1..40] of DWORD; // feld2 ist ein zweidimensionales Feld
END_VAR
```

Variablen vom Datentyp Struktur

```
VAR_OUT // Ausgangsvariable
AUSGANG1: STRUCT // AUSGANG1 ist vom Datentyp STRUCT
    var1 : BOOL; // Element 1 der Struktur
    var2 : DWORD; // Element 2 der Struktur
    END_STRUCT; // Ende der Struktur
END_VAR
```

13.7.2 Beispiel für OBs in AWL-Quellen

```

ORGANIZATION_BLOCK OB 1
TITLE = Beispiel fuer OB 1 mit unterschiedlichen Bausteinaufrufen
//Die 3 dargestellten Netzwerke zeigen Bausteinaufrufe
//mit und ohne Parameter

{S7_pdiag := 'true'} //Systemattribut fuer Bausteine
AUTHOR      :      Siemens
FAMILY      :      Beispiel
NAME        :      Test_OB
VERSION     :      1.1
VAR_TEMP
Zwischenwert : INT; // Zwischenspeicher
END_VAR

BEGIN

NETWORK
TITLE = Aufruf einer Funktion mit Uebergabe von Parametern
// Einzeiliger Parameteruebergabe
CALL FC1 (param1 :=E0.0,param2 :=E0.1);

NETWORK
TITLE = Aufruf eines Funktionsbausteins mit
//Uebergabe von Parametern
// Mehrzeilige Parameterangabe
CALL Ampelsteuerung, DB 6 ( // Name des FBs, Instanz-Datenbaustein
dur_g_p      := S5T#10S, // Zuweisung aktueller Werte an Parameter

del_r_p      := S5T#30S,
starter      := TRUE,
t_dur_y_car  := T 2,
t_dur_g_ped  := T 3,
t_delay_y_car := T 4,
t_dur_r_car  := T 5,
t_next_red_car := T 6,
r_car        := "re_main", // Anfuehrungszeichen kennzeichnen
y_car        := "ye_main", // Namen der Symboltabelle
g_car        := "gr_main",
r_ped        := "re_int",
g_ped        := "gr_int");

NETWORK
TITLE = Aufruf eines Funktionsbausteins mit
//Uebergabe von Parametern
// Einzeilige Parameterangabe
CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1);

END_ORGANIZATION_BLOCK

```

13.7.3 Beispiel für FCs in AWL-Quellen

```
FUNCTION FC 1: VOID
// Nur wegen Call!!
VAR_INPUT
  param1 : bool;
  param2 : bool;
END_VAR
begin
end_function

FUNCTION FC2 : INT
TITLE = Stueckzahlerhoehung
// Solange der uebergebene Wert < 1000 ist, erhoehrt diese
//Funktion den uebergebenen Wert. Liegt die Stueckzahl
//ueber 1000, so wird ueber den Rueckgabewert der
//Funktion (RET_VAL) "-1" zurueckgeliefert.

AUTHOR      :      Siemens
FAMILY      :      Durchs
NAME        :      STUECK
VERSION     :      1.0

VAR_IN_OUT
STUECKZAHL : INT;           // aktuell gefertigte Stueckzahl
END_VAR

BEGIN

NETWORK
TITLE = Erhoehung Stueckzahl um 1
// Solange die aktuelle Stueckzahl unter 1000 liegt,
// kann sie um 1 erhoehrt werden
L STUECKZAHL; L 1000;           // Beispiel fuer mehrere
>I; SPB ERR;                   // Anweisungen in einer Zeile.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;
END_FUNCTION
```

```
FUNCTION FC3 {S7_m_c := 'true'} : INT
TITLE = Stueckzahlerhoehung
// Solange der uebergebene Wert < 1000 ist, erhoehrt diese
//Funktion den uebergebenen Wert. Liegt die Stueckzahl
//ueber 1000, so wird ueber den Rueckgabewert der
//Funktion (RET_VAL) "-1" zurueckgeliefert.
//
//RET_VAL besitzt hier ein Systemattribut fuer Parameter

AUTHOR      :      SIEMENS
FAMILY      :      DURCH
NAME        :      STUECK
VERSION     :      1.0

VAR_IN_OUT
STUECKZAHL {S7_visible := 'true'}: INT;    // aktuell gefertigte Stueckzahl
//Systemattribute fuer Parameter
END_VAR

BEGIN

NETWORK
TITLE = Erhoehung Stueckzahl um 1
// Solange die aktuelle Stueckzahl unter 1000 liegt,
// kann sie um 1 erhoehrt werden
L STUECKZAHL; L 1000;                // Beispiel fuer mehrere
>I; SPB ERR;                          // Anweisungen in einer Zeile.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;

END_FUNCTION
```

13.7.4 Beispiel für FBs in AWL-Quellen

```
FUNCTION_BLOCK FB 6
TITLE = Ampelschaltung einfach
// Steuerung einer Ampelanlage fuer einen Fussgaengeruebergang
//an der Hauptstrasse

{S7_pdiag := 'true'} //Systemattribut fuer Bausteine
AUTHOR      :      SIEMENS
FAMILY      :      Ampel
NAME        :      Ampel01
VERSION     :      1.3

VAR_INPUT

starter      :      BOOL      :=      FALSE; // Ueberquerungswunsch Fußgaenger
t_dur_y_car  :      TIMER;     // Zeitdauer Gruen Fussgaenger
t_next_r_car :      TIMER;     // Zeitdauer zwischen Rot fuer Auto
t_dur_r_car  :      TIMER;
anz          {S7_server := 'alarm_archiv'; S7_a_type := 'alarm_8'} :DWORD;
// Anzahl Autos
// anz besitzt Systemattribute fuer Parameter

END_VAR
VAR_OUTPUT

g_car        :      BOOL      :=      FALSE; // GRUEN fuer Autos

END_VAR
VAR
condition    :      BOOL      :=      FALSE; // Vormerkung Rot fuer Autos
END_VAR

          UN   #condition; // keine Vormerkung Rot Strassenverkehr
          =   #g_car;      // GRUEN fuer Strassenverkehr

NETWORK
TITLE = Dauer fuer Gelbphase Autos
          // weiteres Programm zur Realisierung
          // der Ampelschaltung

END_FUNCTION_BLOCK
```



```
FUNCTION_BLOCK FB 10
```

```
VAR_INPUT
```

```
  para1 : bool;
```

```
  para2: bool;
```

```
end_var
```

```
begin
```

```
end_function_block
```

```
data_block db 10
```

```
fb10
```

```
begin
```

```
end_data_block
```

```
data_block db 6
```

```
fb6
```

```
begin
```

```
end_data_block
```

13.7.5 Beispiele für DBs in AWL-Quellen

Datenbaustein:

```
DATA_BLOCK DB 10
TITLE = DB Beispiel 10
STRUCT
    aa : BOOL;      // Variable aa vom Typ BOOL
    bb : INT; // Variable bb vom Typ INT
    cc : WORD;
END_STRUCT;
BEGIN // Zuweisung von Anfangswerten
    aa := TRUE;
    bb := 1500;
END_DATA_BLOCK
```

DB mit zugeordnetem anwenderdefiniertem Datentyp:

```
DATA_BLOCK DB 20
TITLE = DB (UDT) Beispiel
UDT 20 // Angabe des zugeordneten UDTs
BEGIN
    start := TRUE; // Zuweisung von Anfangswerten
    soll := 10;
END_DATA_BLOCK
```

Hinweis

Der verwendete UDT muss in der Quelle vor dem Datenbaustein stehen.

DB mit zugeordnetem Funktionsbaustein:

```
DATA_BLOCK DB 30
TITLE = DB (FB) Beispiel
FB 30          // Angabe des zugeordneten FBs
BEGIN
    start := TRUE;    // Zuweisung von Anfangswerten
    soll := 10;
END_DATA_BLOCK
```

Hinweis

Der zugeordnete FB muss in der Quelle vor dem Datenbaustein stehen.

13.7.6 Beispiel für UDTs in AWL-Quellen

```
TYPE UDT 20
STRUCT
    start : BOOL;          //Variable vom Typ BOOL
    soll : INT;            //Variable vom Typ INT
    wert : WORD;           //Variable vom Typ WORD
END_STRUCT;
END_TYPE
```


14 Anzeigen von Referenzdaten

14.1 Übersicht der möglichen Referenzdaten

Sie können Referenzdaten erzeugen und auswerten, um sich das Testen und Ändern Ihres Anwenderprogramms zu erleichtern. Referenzdaten werden beispielsweise benutzt als

- Übersicht über ein gesamtes Anwenderprogramm,
- Grundlage für Änderungen und Tests,
- Ergänzung der Programmdokumentation.

Die folgende Tabelle gibt Auskunft darüber, welche Informationen Sie den einzelnen Ansichten entnehmen können:

Ansicht	Anwendung
Querverweisliste	Überblick über die Verwendung von Operanden der Speicherbereiche E, A, M, P, T, Z und DB-, FB-, FC-, SFB- und SFC-Aufrufen innerhalb des Anwenderprogramms. Mit dem Menübefehl Ansicht > Querverweise für Operand können Sie sich alle Querverweise einschließlich der überlappenden Zugriffe auf den markierten Operanden anzeigen lassen.
Belegungsplan	Der Überblick, welche Bits der Operanden der Speicherbereiche E, A, und M bzw. welche Zeiten und Zähler innerhalb des Anwenderprogramms bereits belegt sind, ist eine wichtige Grundlage zur Fehlersuche oder für Änderungen im Anwenderprogramm.
Programmstruktur	Aufrufhierarchie der Bausteine innerhalb eines Anwenderprogramms und Überblick über die verwendeten Bausteine und deren Abhängigkeiten.
Nicht verwendete Symbole	Überblick über alle Symbole, die in der Symboltabelle definiert sind, aber in den Teilen des Anwenderprogramms nicht verwendet werden, für die Referenzdaten vorliegen.
Operanden ohne Symbol	Überblick über alle Absolutadressen (Absolutoperanden und Bausteine), die in den Teilen des Anwenderprogramms verwendet werden und für die Referenzdaten vorliegen, für die aber in der Symboltabelle kein Symbol definiert ist.

Die Referenzdaten des ausgewählten Anwenderprogramms umfassen die in der Tabelle angegebenen Listen. Sie können sich mehrere Listen zu einem Anwenderprogramm oder zu unterschiedlichen Programmen erzeugen und anzeigen lassen.

Gleichzeitige Anzeige mehrerer Ansichten

Das Anzeigen von Listen in zusätzlichen Fenstern ermöglicht Ihnen zum Beispiel:

- eine Gegenüberstellung von gleichen Listen zu unterschiedlichen S7-Anwenderprogrammen.
- verschiedene Ansichten einer Liste, z. B. einer Querverweisliste, unterschiedlich optimiert, nebeneinander auf dem Bildschirm zu haben. Sie können sich z. B. in der einen Querverweisliste nur die Eingänge, in der anderen nur die Ausgänge eines S7-Anwenderprogramms anzeigen lassen.
- mehrere Listen zu einem S7-Anwenderprogramm gleichzeitig zu öffnen, z. B. Programmstruktur und Querverweisliste.

14.1.1 Querverweisliste

Die Querverweisliste bietet einen Überblick über die Verwendung von Operanden innerhalb des S7-Anwenderprogramms.

Mit der Anzeige der Querverweisliste erhalten Sie eine Liste der im S7-Anwenderprogramm verwendeten Operanden der Speicherbereiche Eingang (E), Ausgang (A), Merker (M), Zeiten (T), Zähler (Z), Funktionsbaustein (FB), Funktion (FC), Systemfunktionsbaustein (SFB), Systemfunktion (SFC), Peripherie (P) und Datenbaustein (DB), ihrer Adressen (absolute Adresse, Name) und ihrer Verwendung. Sie wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des Anwenderprogramms an, zu dem die Querverweisliste gehört.

Jede Zeile im Fenster entspricht einem Querverweislisteneintrag. Eine Suchfunktion erleichtert Ihnen das gezielte Auffinden bestimmter Operanden und Symbole.

Die Querverweisliste ist die voreingestellte Ansicht beim Anzeigen von Referenzdaten. Sie können diese Voreinstellung ändern.

Aufbau

Ein Querverweislisteneintrag besteht aus folgenden Spalten:

Spalte	Inhalt/Bedeutung
Operand (Symbol)	Adresse des Operanden
Baustein	Angabe des Bausteins, in dem der Operand verwendet wird
Art	Angabe, ob es sich um einen lesenden (R) und/oder schreibenden (W) Zugriff auf den Operanden handelt
Sprache	sprachabhängige Informationen, hängen von der Erstellungssprache des Bausteins ab
Verwendungsstelle	Doppelklicken Sie dieses Feld, um zur Verwendungsstelle des referenzierten Operanden zu springen.

Die Spalten Baustein, Art, Sprache und Verwendungsstelle werden nur angezeigt, wenn die entsprechenden Eigenschaften für die Querverweisliste angewählt worden sind. Diese sprachabhängigen Informationen variieren je nach Erstellungssprache des Bausteins.

Die Spaltenbreite können Sie in der auf dem Bildschirm angezeigten Querverweisliste über Mausbedienung an ihre Anforderungen anpassen.

Sortierung

Voreingestellt in der Querverweisliste ist die Sortierung nach Speicherbereichen. Wenn Sie mit der Maus auf eine Spaltenüberschrift klicken, wird nach den Einträgen dieser Spalte sortiert.

Beispiel für den Aufbau der Querverweisliste

Operand (Symbol)	Baustein Symbol)	Art	Sprache	Verwendungsstelle
E 1.0 (Motor ein)	OB 2	R	AWL	NW 2 Anw 33 /O
M1.2 (MerkerBit)	FC 2	R	KOP	NW 33
Z2 (Zähler2)	FB2		FUP	NW2

14.1.2 Programmstruktur

Die Programmstruktur beschreibt die Aufrufhierarchie der Bausteine innerhalb eines S7-Anwenderprogramms. Außerdem erhalten Sie einen Überblick über die verwendeten Bausteine, deren Abhängigkeiten und deren Lokaldatenbedarf.

Mit dem Menübefehl **Ansicht > Filtern** im Fenster "S7-Referenzdaten anzeigen" öffnen Sie einen Registerdialog. Im Register "Programmstruktur" können Sie festlegen, wie die Programmstruktur dargestellt wird.

Sie haben die Wahl zwischen

- Aufrufstruktur und
- Abhängigkeitsstruktur.

Symbole in der Programmstruktur

Symbol Bedeutung

Baustein regulär aufgerufen (CALL FB10)

Baustein unbedingt aufgerufen (UC FB10)

Baustein bedingt aufgerufen (CC FB10)

Datenbaustein

Rekursion

Rekursion und bedingt aufgerufen

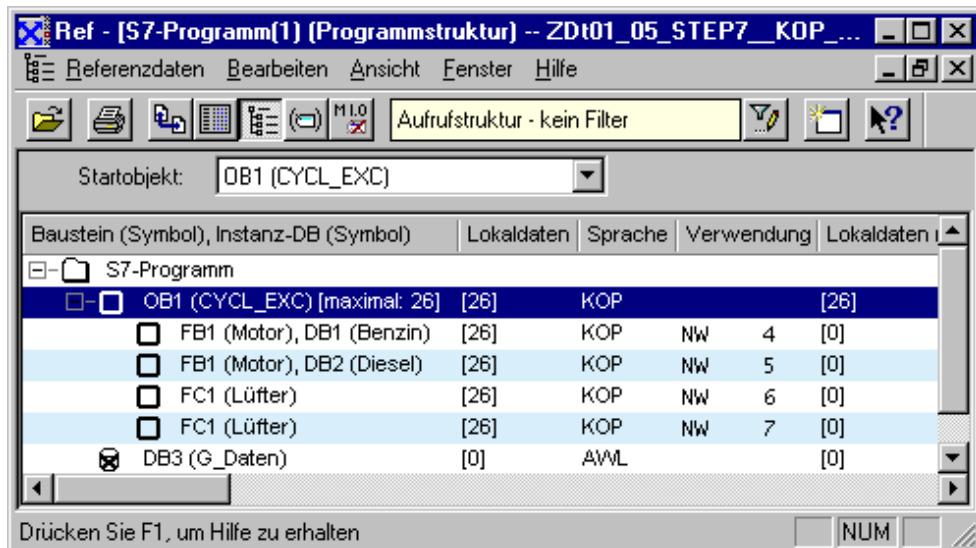
Rekursion und unbedingt aufgerufen

Baustein nicht aufgerufen

- Rekursionen im Aufruf werden erkannt und in der Aufrufstruktur grafisch gekennzeichnet.
- Rekursionen innerhalb der Aufrufhierarchie werden durch unterschiedliche Symbole dargestellt.
- Regulär aufgerufene Bausteine (CALL), bedingt aufgerufene Bausteine (CC) oder unbedingt aufgerufene Bausteine (UC) werden durch unterschiedliche Symbole gekennzeichnet.
- Nicht aufgerufene Bausteine werden am unteren Ende der Baumstruktur angezeigt und mit einem schwarzen Kreuz markiert. Die Aufrufstruktur eines nicht aufgerufenen Bausteins wird nicht weiter aufgeschlüsselt.

Anzeige als Aufrufstruktur

Es wird die gesamte Aufrufhierarchie dargestellt.



Soll die Programmstruktur für alle Organisationsbausteine (OB) erstellt werden und der OB1 ist nicht im S7-Anwenderprogramm, oder wurde ein Startbaustein angegeben, der nicht im Programm vorhanden ist, werden Sie automatisch zur Vergabe eines anderen Bausteins für die Wurzel der Programmstruktur aufgefordert.

Die Anzeige von Mehrfachaufrufen von Bausteinen kann über Optionseinstellungen sowohl für die Aufrufstruktur als auch für die Abhängigkeitsstruktur ausgeschaltet werden.

Anzeige des maximalen Lokaldatenbedarfs in der Aufrufstruktur

Um Ihnen einen schnellen Überblick über den Lokaldatenbedarf der OBs im angezeigten Anwenderprogramm zu geben, können Sie sich in der Baumdarstellung folgendes anzeigen lassen:

- den Maximalbedarf der Lokaldaten pro OB
- den Lokaldatenbedarf pro Pfad

Die Anzeige lässt sich im Register "Programmstruktur" ein- und ausschalten.

Falls synchrone Fehler-OBs (OB 121, OB 122) vorhanden sind, wird bei Anzeige des maximalen Lokaldatenbedarfs nach dem Zahlenwert ein Pluszeichen und danach der zusätzliche Bedarf für die synchronen Fehler-OBs angegeben.

Abhängigkeitsstruktur

Die Abhängigkeitsstruktur zeigt für jeden Baustein im Projekt die Abhängigkeiten zu anderen Bausteinen an. Ganz links steht der Baustein und darunter eingerückt die Bausteine, die diesen Baustein aufrufen oder verwenden.

Anzeige gelöschter Bausteine

Die Zeilen zu gelöschten Bausteinen werden rot hervorgehoben.

14.1.3 Belegungsplan

Der Belegungsplan zeigt Ihnen, welche Operanden innerhalb des Anwenderprogramms bereits belegt sind. Diese Anzeige ist eine wichtige Grundlage für Fehlersuche oder Änderungen im Anwenderprogramm.

Mit der Anzeige des Belegungsplans erhalten Sie eine Übersicht, welches Bit in welchem Byte der Speicherbereiche Eingang (E), Ausgang (A), und Merker (M) sowie welche Zeiten (T) und Zähler (Z) verwendet sind.

Der Belegungsplan wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des S7-Anwenderprogramms an, zu dem der Belegungsplan gehört.

Übersicht E/A/M

Jede Zeile beinhaltet ein Byte des Speicherbereichs, in dem die acht Bits je nach Zugriff gekennzeichnet werden. Zusätzlich wird noch angegeben, ob der Zugriff durch einen Byte-, Wort- oder Doppelwortzugriff erfolgt.

Kennzeichen in der Übersicht E/A/M

weißer Hintergrund	der Operand ist nicht angesprochen und somit noch nicht belegt
X	der Operand ist direkt verwendet
blauer Hintegrund	der Operand ist indirekt bearbeitet (Byte-, Wort- oder Doppelwortzugriff)

Spalten in der Übersicht E/A/M

Spalte	Inhalt/Bedeutung
7	Bitnummer des entsprechenden Bytes
6	
5	
4	
3	
2	
1	
0	
B	das Byte ist durch einen Bytezugriff belegt
W	das Byte ist durch einen Wortzugriff belegt
D	das Byte ist durch einen Doppelwortzugriff belegt

Beispiel

Folgendes Beispiel zeigt den typischen Aufbau eines Belegungsplans für Eingänge, Ausgänge und Merker (E/A/M).

△	7	6	5	4	3	2	1	0	B	W	D
EB0	X	X	X	X	X	X	X				
EB1		X	X	X		X	X	X			
AB4						X	X	X			
AB5		X	X	X		X	X	X			
MB1											
MB2											
MB3											
MB4											
MB5											

In der ersten Zeile ist die Belegung vom Eingangsbyte EB 0 aufgeführt. Eingänge des Operanden EB 0 werden direkt verwendet (Bitzugriff). In den Spalten "1", "2", "3", "4", "5", und "6" steht jeweils ein "X" als Kennzeichen für den Bitzugriff.

Es erfolgt aber auch ein Wortzugriff auf die Merkerbytes 1 und 2, 2 und 3 bzw. 4 und 5. Daher steht in der Spalte "W" jeweils ein "Balken" und die Zellen sind noch zusätzlich hellblau hinterlegt. Die schwarze Spitze der Balken zeigt den Beginn des Wortzugriffs an.

Übersicht T/Z

In jeder Zeile werden 10 Zeiten bzw. Zähler dargestellt.

Beispiel

	0	1	2	3	4	5	6	7	8	9
T 00-09	.	T1	.	.	.		T6	.	.	.
T 10-19	.	.	T12	T17	.	T19
T 20-29	T24
Z 00-09	.	.	Z2	Z7	.	.
Z 10-19	Z19
Z 20-29
Z 30-39	Z34

In diesem Beispiel sind die Zeiten T1, T6, T12, T17, T19, T24 und die Zähler Z2, Z7, Z19, Z34 belegt.

Diese Listen sind alphabetisch sortiert. Die Einträge können Sie durch Anklicken der Spaltenüberschrift anordnen.

14.1.4 Nicht verwendete Symbole

Sie erhalten einen Überblick über alle Symbole mit folgender Eigenschaft:

- Die Symbole sind in der Symboltabelle definiert.
- Die Symbole werden aber innerhalb der Anwenderprogrammteile, für die Referenzdaten vorliegen, nicht verwendet.

Sie werden in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des Anwenderprogramms an, zu dem die Liste gehört.

Jede angezeigte Zeile im Fenster entspricht einem Listeneintrag. Eine Zeile besteht aus Operand, Symbol, Datentyp und Kommentar.

Spalte	Inhalt/Bedeutung
Operand	absolute Adresse des Operanden
Datentyp	Datentyp des Operanden
Operandenkommentar	Kommentar des Operanden aus der Symboltabelle

Beispiel einer Liste nicht verwendeter Symbole

Symbol	Operand	Datentyp	Operandenkommentar
MS1	E103.6	BOOL	Motorschutzschalter 1
MS2	E120.5	BOOL	Motorschutzschalter 2
MS3	E121.3	BOOL	Motorschutzschalter 3

Die Einträge können Sie durch Anklicken der Spaltenüberschrift anordnen.

Ebenso können Sie in der Liste aufgeführte, nicht mehr benötigte Symbole auswählen und diese über die Funktion "Symbole löschen" aus der Symboltabelle entfernen.

14.1.5 Operanden ohne Symbol

Mit der Anzeige der Liste der Operanden ohne Symbol erhalten Sie eine Liste der Elemente, die im S7-Anwenderprogramm verwendet werden, in der Symboltabelle aber nicht definiert sind. Sie wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des Anwenderprogramms an, zu dem die Liste gehört.

Eine Zeile besteht aus dem Operanden und der Angabe, wie oft er verwendet worden ist. Die Einträge sind nach Operanden geordnet.

Beispiel:

Operand	Anzahl
A 2.5	4
E 23.6	3
M 34.1	20

Sie können die in der Liste angezeigten Operanden ohne Symbol anwählen und diesen über die Funktion "Symbole bearbeiten" Symbolnamen zuordnen.

14.1.6 Anzeige sprachabhängiger Informationen zu KOP, FUP, AWL

Die sprachabhängigen Informationen zu KOP, FUP und AWL werden in der Querverweisliste und in der Programmstruktur angezeigt. Sie bestehen aus der Bausteinsprache und Details.

In der Ansicht "Programmstruktur" werden die sprachabhängigen Informationen nur dann angezeigt, wenn bei den Filtereinstellungen im Register "Programmstruktur" die Darstellung "Aufrufstruktur" mit den entsprechenden Optionen gewählt wurde.

In der Ansicht "Querverweise" können die sprachabhängigen Informationen über **Ansicht > Filtern** ein- oder ausgeblendet werden.

- Aktivieren Sie die Kontrollkästchen "Bausteinsprache" und "Details" im Register "Querverweise" des Dialogfeldes "Filtern", um die sprachabhängigen Informationen einzublenden.

Die sprachabhängigen Informationen variieren je nach Erstellungssprache des Bausteins und werden mit Kürzeln dargestellt.

Sprache	Netzwerk	Anweisung	Operation
AWL	NW	Anw	/
KOP	NW		
FUP	NW		

NW, Anw geben an, in welchem Netzwerk, in welcher Anweisung der Operand benutzt (Querverweise) bzw. der Baustein aufgerufen wird (Programmstruktur).

Anzeige sprachabhängiger Informationen zu optionalen Programmiersprachen

Die Hilfe zu sprachabhängigen Informationen ist aufrufbar, wenn das entsprechende Optionspaket installiert ist.

14.2 Arbeiten mit Referenzdaten

14.2.1 Möglichkeiten zum Anzeigen von Referenzdaten

Zum Anzeigen von Referenzdaten bestehen folgende Möglichkeiten:

Anzeigen aus dem SIMATIC Manager

1. Selektieren Sie im Projektfenster in der Projektansicht offline den Ordner "Bausteine".
2. Wählen Sie den Menübefehl **Extras > Referenzdaten > Anzeigen**.

Anzeigen aus dem Editor-Fenster

1. Öffnen Sie einen Baustein im Ordner "Bausteine".
2. Wählen Sie im Fenster des entsprechenden Spracheditors den Menübefehl **Extras > Referenzdaten**.

Der Dialog "Einstellungen" wird eingeblendet. Hier können Sie die Ansicht auswählen, die zuerst angezeigt werden soll. Es ist die Ansicht voreingestellt, die in der Applikation zum Anzeigen von Referenzdaten zuletzt geschlossen wurde. Der Dialog kann für künftige Aufrufe unterdrückt werden.

Wenn die Referenzdaten unvollständig sind, wird ein Dialogfeld angezeigt, von dem aus Sie eine Aktualisierung der Referenzdaten anstoßen können.

Anzeigen direkt aus dem kompilierten Baustein

Die Referenzdaten zu einem kompilierten Baustein können Sie direkt aus dem Spracheditor heraus anzeigen und sich damit einen aktuellen Überblick über Ihr Anwenderprogramm verschaffen.

14.2.2 Anzeigen von Listen in zusätzlichen Arbeitsfenstern

Mit dem Menübefehl **Fenster > Neues Fenster** können Sie weitere Arbeitsfenster öffnen und sich darin andere Sichten von bereits angezeigten Referenzdaten (z. B. Liste der nicht verwendeten Symbole) anzeigen lassen.

Ein Arbeitsfenster für noch nicht angezeigte Referenzdaten öffnen Sie mit dem Menübefehl **Referenzdaten > Öffnen**.

Sie können auf eine andere Ansicht der Referenzdaten umschalten, indem Sie den Menübefehl **Ansicht** oder die zugehörige Schaltfläche in der Funktionsleiste wählen:

Referenzdaten-Ansicht	Menübefehl zur Anzeige dieser Referenzdaten-Ansicht
Operanden ohne Symbol	Ansicht > Operanden ohne Symbol
Nicht verwendete Symbole	Ansicht > Nicht verwendete Symbole
Belegung	Ansicht > Belegung
Programmstruktur	Ansicht > Programmstruktur
Querverweise	Ansicht > Querverweise

14.2.3 Erzeugen und Anzeigen von Referenzdaten

Erzeugen von Referenzdaten

1. Markieren Sie im SIMATIC Manager den Bausteinordner, für den Referenzdaten generiert werden sollen.
2. Wählen Sie im SIMATIC Manager den Menübefehl **Extras > Referenzdaten > Generieren**.

Vor dem Generieren von Referenzdaten wird zunächst geprüft, ob Referenzdaten vorliegen und ob eventuell vorliegende Referenzdaten aktuell sind.

- Falls keine Referenzdaten vorliegen, werden sie generiert.
- Falls keine aktuellen Referenzdaten vorliegen, können Sie in einem Dialogfeld entscheiden, ob die Referenzdaten aktualisiert oder komplett neu generiert werden sollen.

Anzeigen von Referenzdaten

Mit dem Menübefehl **Extras > Referenzdaten > Anzeigen** können Sie sich die Referenzdaten anzeigen lassen.

Vor dem Anzeigen von Referenzdaten wird zunächst geprüft, ob Referenzdaten vorliegen und ob eventuell vorliegende Referenzdaten aktuell sind.

- Falls keine Referenzdaten vorliegen, werden sie erzeugt.
- Falls unvollständige Referenzdaten vorliegen, wird ein Dialogfeld mit einem Hinweis auf die Inkonsistenz der Referenzdaten angezeigt. Sie können dann entscheiden, ob und in welchem Umfang die Referenzdaten aktualisiert werden sollen. Dabei haben Sie folgende Auswahlmöglichkeiten:

Wahl	Bedeutung
nur für geänderte Bausteine	Die Referenzdaten zu geänderten und neuen Bausteine werden aktualisiert; Information zu gelöschten Bausteinen wird aus den Referenzdaten entfernt.
für alle Bausteine	Die Referenzdaten für alle Bausteine werden komplett neu erstellt.
nicht aktualisieren	Die Referenzdaten werden nicht aktualisiert.

Zur Aktualisierung der Referenzdaten werden die Bausteine neu übersetzt. Dabei wird für jeden Baustein der passende Compiler aufgerufen. Mit dem Menübefehl **Ansicht > Aktualisieren** können Sie bereits angezeigte Referenzdaten im aktiven Fenster aktualisieren.

14.2.4 Schnelles Positionieren auf Verwendungsstellen im Programm

Sie können Referenzdaten benutzen, um beim Programmieren auf die Verwendungsstellen eines Operanden zu positionieren. Dazu müssen aktuelle Referenzdaten vorliegen. Der Aufruf der Applikation zum Anzeigen von Referenzdaten ist nicht erforderlich.

Prinzipielle Vorgehensweise

1. Wählen Sie im SIMATIC Manager den Menübefehl **Extras > Referenzdaten > Generieren**, um aktuelle Referenzdaten zu erzeugen. Dieser Schritt ist nur erforderlich, wenn Referenzdaten nicht vorliegen oder veraltet sind.
2. Markieren Sie in einem geöffneten Baustein den Operanden.
3. Wählen Sie den Menübefehl **Bearbeiten > Gehe zu > Verwendungsstelle**. Nun wird ein Dialogfeld angezeigt, das eine Liste mit den Verwendungsstellen des Operanden im Programm enthält.
4. Wählen Sie die Option "Überlappender Zugriff auf Speicherbereiche", wenn zusätzlich die Verwendungsstellen der Operanden angezeigt werden sollen, deren Adressen oder Adressbereich die Adresse bzw. den Adressbereich des aufgerufenen Operanden überlappen. Die Tabelle wird um die Spalte "Operand" erweitert.
5. Markieren Sie eine Verwendungsstelle in der Liste und klicken Sie auf die Schaltfläche "Gehe zu".

Wenn beim Aufruf des Dialogs die Referenzdaten nicht aktuell sind, werden Sie mit einer Meldung darauf hingewiesen. Sie können dann die Referenzdaten aktualisieren lassen.

Liste der Verwendungsstellen

Die Liste der Verwendungsstellen im Dialogfeld enthält folgende Angaben:

- Baustein, in dem der Operand verwendet wird
- Bausteinsymbol, falls vorhanden
- Details, d. h. von der Erstsprache des Bausteins/der Quelle (SCL) abhängige Information zur Verwendungsstelle und ggf. der Operation
- sprachabhängige Informationen
- Zugriffsart auf den Operanden: Lesend (R), schreibend (W), lesend und schreibend (RW), nicht ermittelbar (?)
- Bausteinsprache

Sie können die Anzeige der Verwendungsstellen filtern, und sich beispielsweise nur die schreibenden Zugriffe auf einen Operanden anzeigen lassen. Ausführliche Informationen zu den Eingabemöglichkeiten und Anzeigen finden Sie in der Online-Hilfe zu diesem Dialogfeld.

Hinweis

Referenzdaten sind nur offline vorhanden. Diese Funktion arbeitet daher stets mit den Querverweisen der Bausteine offline, auch wenn Sie die Funktion in einem Baustein online aufrufen.

14.2.5 Beispiel zum Arbeiten mit Verwendungsstellen

Sie möchten feststellen, an welchen Stellen der Ausgang A1.0 (direkt/indirekt) gesetzt wird. Als Beispiel dient der folgende AWL-Code im Baustein OB1:

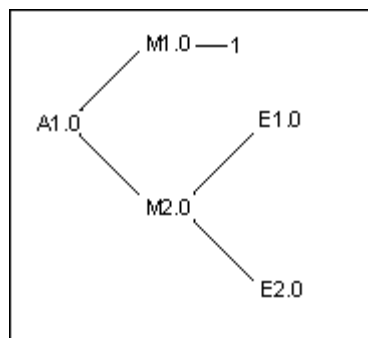
```
Netzwerk1: .....  
U A 1.0 // in diesem Beispiel  
= A 1.1 // nicht relevant
```

```
Netzwerk 2:  
U M1.0  
U M2.0  
= A 1.0 // Zuweisung
```

```
Netzwerk3:  
//nur Kommentarzeile  
SET  
= M1.0 // Zuweisung
```

```
Netzwerk 4:  
U E 1.0  
U E 2.0  
= M2.0 // Zuweisung
```

Für A1.0 ergibt sich daraus der folgende Zuweisungsbaum:



Dann gehen Sie so vor:

1. Im KOP/AWL/FUP-Editor im OB 1 auf A1.0 (NW 1, Anw 1) positionieren.
2. Über **Bearbeiten > Gehe zu > Verwendungsstelle** oder mit der rechten Maustaste "Gehe zur Verwendungsstelle" aufrufen.
Im Dialogfeld werden unter anderem alle Zuweisungen auf A1.0 angezeigt:
OB1 Cycle Execution NW 2 Anw 3 /= W AWL
OB1 Cycle Execution NW 1 Anw 1 /U R AWL
3. Mittels "Gehe zu" aus dem Dialogfeld zu "NW 2 Anw 3" in den Editor springen:
Netzwerk 2:
U M1.0
U M2.0
= A 1.0
4. Sowohl die Zuweisungen an M1.0 als auch an M2.0 müssen nun kontrolliert werden. Also im KOP/AWL/FUP-Editor zuerst auf M1.0 positionieren.
5. Über **Bearbeiten > Gehe zu > Verwendungsstelle** oder mit rechter Maustaste "Gehe zur Verwendungsstelle" aufrufen. Im Dialogfeld werden unter anderem alle Zuweisungen auf M1.0 angezeigt:
OB 1 Cycle Execution NW 3 Anw 2 /= W AWL
OB 1 Cycle Execution NW 2 Anw 1 /U R AWL
6. Mittels "Gehe zu" zu "NW 3 Anw 2" in den KOP/AWL/FUP-Editor springen.
7. Im KOP/AWL/FUP-Editor in Netzwerk 3 wird festgestellt, dass die Belegung von M1.0 uninteressant ist (da immer TRUE) und stattdessen die Belegung von M2.0 untersucht werden müsste.

Im STEP 7-Versionen vor V5 musste jetzt die gesamte Kette von Zuweisungen nochmal ganz von vorne durchlaufen werden. Die Schaltflächen ">>" und "<<" vereinfachen das weitere Vorgehen:

8. Entweder das noch geöffnete Dialogfeld "Gehe zu Verwendungsstelle" nach vorne holen oder aus der aktuellen Position im KOP/AWL/FUP-Editor "Gehe zur Verwendungsstelle" aufrufen.
9. Die Schaltfläche "<<" einmal oder zweimal betätigen, bis alle Verwendungsstellen von A 1.0 angezeigt werden, wobei die letzte Aussprungstelle "NW 2 Anw 3" markiert ist.
10. Mittels "Gehe zu" (wie in Punkt 3) aus dem Verwendungsstellendialog zu "NW 2 Anw 3" in den Editor zu springen:
Netzwerk 2:
U M1.0
U M2.0
= A 1.0
11. In Punkt 4ff. wurde die Zuweisung an M1.0 kontrolliert. Jetzt müssen alle (direkten/indirekten) Zuweisungen auf M2.0 kontrolliert werden. Also im Editor auf M2.0 positionieren und "Gehe zur Verwendungsstelle" aufrufen: Es werden u.a. alle Zuweisungen auf M2.0 angezeigt:
OB1 Cycle Execution NW 4 Anw 3 /= W AWL
OB1 Cycle Execution NW 2 Anw 2 /U R AWL

12. Mittels "Gehe zu" zu "NW 4 Anw 3" in den KOP/AWL/FUP-Editor springen:

Netzwerk 4:

U E 1.0

U E 2.0

= M2.0

13. Jetzt müssen die Zuweisungen an E1.0 und E2.0 überprüft werden. Dies wird in diesem Beispiel nicht mehr erläutert, da die weitere Vorgehensweise sich von der bisherigen (Punkt 4ff.) nicht unterscheidet

Durch abwechselndes Springen zwischen KOP/AWL/FUP-Editor und dem Verwendungsstellen-Dialog können sie so die relevanten Stellen ihres Programms ermitteln und untersuchen.

15 Bausteinkonsistenz prüfen und Zeitstempel als Bausteineigenschaft

15.1 Bausteinkonsistenz prüfen

Einführung

Müssen während oder nach der Programmerstellung die Schnittstellen bzw. der Code einzelner Bausteine angepasst oder ergänzt werden, so kann dies zu Zeitstempelkonflikten führen. Zeitstempelkonflikte können wiederum zu Bausteininkonsistenzen zwischen aufrufenden und aufgerufenen Bausteinen bzw. Referenzbausteinen und damit zu einem hohen Korrekturaufwand führen.

Die Funktion "Bausteinkonsistenz prüfen" nimmt Ihnen einen großen Teil dieser Korrekturarbeit ab. Die Funktion "Bausteinkonsistenz prüfen" bereinigt selbständig einen großen Teil aller Zeitstempelkonflikte und Bausteininkonsistenzen. Bei Bausteinen, deren Bausteininkonsistenzen nicht automatisch bereinigt werden konnten, führt Sie die Funktion an die zu ändernden Positionen im dazugehörigen Editor. Dort nehmen Sie die notwendigen Änderungen vor. Schritt für Schritt werden alle Bausteininkonsistenzen bereinigt und die Bausteine übersetzt.

Voraussetzungen

Die Bausteinkonsistenz-Prüfung ist nur bei Projekten möglich, die ab STEP 7 V5.0, Servicepack 3 erstellt wurden. Bei älteren Projekten müssen Sie daher beim Starten der Bausteinkonsistenz-Prüfung zunächst alles übersetzen (Menübefehl **Programm > Alles übersetzen**).

Bei Bausteinen, die mit einem Optionspaket erstellt wurden, muss das Optionspaket für eine Konsistenzprüfung installiert sein.

Starten der Bausteinkonsistenz-Prüfung

Mit dem Start der Bausteinkonsistenz-Prüfung werden die Zeitstempel der Bausteinschnittstellen geprüft und Bausteine, bei denen es zu Bausteinkonsistenzen führen kann, werden in der Baumansicht (Aufrufbaum: Referenzen / Abhängigkeitsbaum) markiert.

1. Markieren Sie im Projektfenster des SIMATIC Managers den gewünschten Bausteinordner und starten Sie die Bausteinkonsistenz-Prüfung mit dem Menübefehl **Bearbeiten > Bausteinkonsistenz prüfen**.
2. Wählen Sie in "Bausteinkonsistenz prüfen" den Menübefehl **Programm > Übersetzen**.
STEP 7 erkennt automatisch die Erstsprache für die betreffenden Bausteine und ruft den entsprechenden Editor auf. Soweit wie möglich, werden die Zeitstempelkonflikte und Bausteinkonsistenzen automatisch bereinigt und die Bausteine übersetzt. Konnte der Zeitstempelkonflikt bzw. die Inkonsistenz in einem Baustein nicht automatisch bereinigt werden, so erfolgt eine Fehlermeldung im Ausgabefenster (weitere Vorgehensweise siehe Schritt 3). Dieser Ablauf wird automatisch für alle Bausteine der Baumansicht wiederholt.
3. Konnten mit dem Übersetzungslauf nicht alle Bausteinkonsistenzen automatisch bereinigt werden, so werden die betreffenden Bausteine im Ausgabefenster als Fehlermeldung markiert. Positionieren Sie mit der Maus den entsprechenden Fehlereintrag und wählen Sie mit der rechten Maustaste im Kontextmenübefehl **Fehler anzeigen**. Der entsprechende Editor wird geöffnet und an die zu ändernden Positionen gesprungen. Beheben Sie alle Bausteinkonsistenzen und schließen und speichern Sie den Baustein. Wiederholen Sie den Ablauf für alle als Fehler markierten Bausteine.
4. Starten Sie Schritt 2 und 3 erneut. Wiederholen Sie diesen Vorgang solange, bis im Ausgabefenster keine Fehler mehr angezeigt werden.

15.2 Zeitstempel als Bausteineigenschaft und Zeitstempelkonflikte

Bausteine enthalten einen Code-Zeitstempel und einen Schnittstellen-Zeitstempel. Diese Zeitstempel werden im Dialogfeld zu den Baustein-Eigenschaften angezeigt. Anhand von Zeitstempeln wird die Konsistenz von STEP 7-Programmen überwacht.

STEP 7 zeigt einen Zeitstempelkonflikt an, wenn beim Vergleich von Zeitstempeln ein Regelverstoß erkannt wird. Folgende Regelverstöße können auftreten:

- Ein aufgerufener Baustein ist jünger als der aufrufende Baustein (CALL).
- Ein referenzierter Baustein ist jünger als der Baustein, der ihn verwendet.
- Beispiele zum zweiten Punkt:
- Ein UDT ist jünger als der Baustein, der ihn verwendet, z. B. ein DB oder ein anderer UDT, oder ein FC, FB, OB, der den UDT in der Variablendeklarationstabelle verwendet.
- Ein FB ist jünger als der zugehörige Instanz-DB.
- In FB1 ist ein FB2 als Multiinstanz definiert und FB2 ist jünger als FB1.

Hinweis

Auch wenn die Relation der Schnittstellen-Zeitstempel korrekt ist, können Inkonsistenzen vorkommen:

- Die Definition der Schnittstelle des referenzierten Bausteins stimmt nicht mit der an seiner Verwendungsstelle verwendeten Schnittstelle überein.

Solche Inkonsistenzen werden Schnittstellenkonflikte genannt. Sie können z. B. durch Kopieren von Bausteinen verschiedener Programme entstehen oder durch den Übersetzungslauf einer ASCII-Quelle, bei dem nur eine Teilmenge der Bausteine eines Gesamtprogramms erzeugt wird.

15.3 Zeitstempel in Codebausteinen

Code-Zeitstempel

Hier wird der Erstellungszeitpunkt des Bausteins eingetragen. Der Zeitstempel wird aktualisiert

- bei Änderung des Programmcodes
- bei Änderungen der Schnittstellenbeschreibung
- bei Änderung des Kommentars
- bei der Ersterstellung und der Übersetzung einer ASCII-Quelle
- bei Änderungen der bas00242 (Dialog: Eigenschaften)

Schnittstellen-Zeitstempel

Der Zeitstempel wird aktualisiert bei

- Änderungen der Schnittstellenbeschreibung (Änderung von Datentypen oder Initialwerten, neue Parameter)
- der Ersterstellung und der Übersetzung einer ASCII-Quelle, falls sich die Schnittstelle strukturell ändert.

Der Zeitstempel wird nicht aktualisiert bei:

- Änderung von Symbolen
- Änderung von Kommentaren in der Variablendeklaration
- Änderungen im TEMP-Bereich.

Regel für Bausteinaufrufe

- Der Schnittstellen-Zeitstempel des aufgerufenen Bausteins muss älter sein als der Code-Zeitstempel des aufrufenden Bausteins.
- Ändern Sie die Schnittstelle eines Bausteins nur, wenn kein Baustein geöffnet ist, der diesen aufruft. Denn wenn Sie die aufrufenden Bausteine später als den geänderten speichern, so erkennen sie diese Inkonsistenz nicht am Zeitstempel.

Vorgehen bei Zeitstempelkonflikt

Ein Zeitstempelkonflikt wird beim Öffnen des Aufrufers angezeigt. Nach Änderung einer FC- oder FB-Schnittstelle werden alle Aufrufe dieses Bausteins in rufenden Bausteinen in expandierter Form dargestellt.

Wird die Schnittstelle eines Bausteins geändert, so müssen alle Bausteine angepasst werden, die diesen Baustein aufrufen.

Nach Änderung einer FB-Schnittstelle müssen die vorhandenen Multiinstanzdefinitionen und Instanz-Datenbausteine aktualisiert werden.

15.4 Zeitstempel bei globalen Datenbausteinen

Code-Zeitstempel

Der Zeitstempel wird aktualisiert

- bei der Ersterstellung,
- bei der Übersetzung einer ASCII-Quelle,
- bei Änderungen in der Deklarationssicht oder in der Datensicht des Bausteins.

Schnittstellen-Zeitstempel

Der Zeitstempel wird aktualisiert

- bei Änderungen der Schnittstellenbeschreibung in der Deklarationssicht (Änderung von Datentypen oder Initialwerten, neue Parameter)

15.5 Zeitstempel bei Instanz-Datenbausteinen

Ein Instanz-Datenbaustein speichert die Formalparameter und statischen Daten von Funktionsbausteinen.

Code-Zeitstempel

Hier wird der Erstellungszeitpunkt der Instanz-Datenbausteine eingetragen. Der Zeitstempel wird aktualisiert, wenn Sie Aktualwerte in der Datensicht des Instanz-Datenbausteins eintragen. Änderungen der Struktur eines Instanz-Datenbausteins durch den Benutzer sind nicht möglich; da die Struktur vom zugehörigen Funktionsbaustein (FB) oder Systemfunktionsbaustein (SFB) abgeleitet ist.

Schnittstellen-Zeitstempel

Beim Erzeugen eines Instanz-Datenbausteins wird der Schnittstellen-Zeitstempel des zugehörigen FB oder SFB eingetragen.

Regeln für konfliktfreies Öffnen

Die Schnittstellen-Zeitstempel des FB/SFB und des zugeordneten Instanz-Datenbausteins müssen übereinstimmen.

Vorgehen bei Zeitstempelkonflikt

Wenn Sie die Schnittstelle eines FB ändern, wird der Schnittstellen-Zeitstempel des FB aktualisiert. Beim Öffnen eines zugehörigen Instanz-Datenbausteins wird ein Zeitstempelkonflikt angezeigt, da die Zeitstempel des Instanz-Datenbausteins und des FB nicht mehr übereinstimmen. Im Deklarationsteil des DB wird die Schnittstelle mit vom Compiler generierten Symbolen (Pseudosymbolik) dargestellt. Der Instanz-Datenbaustein ist nur noch zur Ansicht nutzbar.

Um solche Zeitstempelkonflikt zu beheben, müssen Sie den zu einem geänderten FB gehörigen Instanz-DB neu erzeugen.

15.6 Zeitstempel bei UDTs und von UDTs abgeleiteten DBs

Anwenderdefinierte Datentypen (UDT) können z. B. zur Erzeugung mehrerer Datenbausteine mit der gleichen Struktur genutzt werden.

Code-Zeitstempel

Der Code-Zeitstempel wird aktualisiert bei jeder Änderung.

Schnittstellen-Zeitstempel

Der Schnittstellen-Zeitstempel wird bei Änderungen der Schnittstellenbeschreibung (Änderung von Datentypen oder Initialwerten, neue Parameter) aktualisiert.

Der Schnittstellen-Zeitstempel eines UDT wird auch bei Übersetzung der ASCII-Quelle aktualisiert.

Regeln für konfliktfreies Öffnen

- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muss älter sein als der Schnittstellen-Zeitstempel in Codebausteinen, in denen dieser Datentyp verwendet wird.
- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muss identisch sein mit dem Zeitstempel eines aus einem UDT abgeleiteten DB.
- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muss jünger sein als der Zeitstempel eines unterlagerten UDT.

Vorgehen bei Zeitstempelkonflikt

Wenn Sie eine UDT-Definition ändern, die in einem DB, FC, FB oder einer anderen UDT-Definition verwendet wird, zeigt STEP 7 beim Öffnen eines solchen Bausteins ein Zeitstempelkonflikt an.

Die UDT-Komponente wird als Struktur aufgefächert dargestellt. Alle Variablennamen werden durch vom System vorgegebene Werte überschrieben.

15.7 Korrigieren der Schnittstellen in einem FC, FB oder UDT

Wenn Sie die Schnittstelle in einem FB, FC oder UDT korrigieren müssen, so gehen Sie wie folgt vor, um Zeitstempelkonflikte zu vermeiden:

1. Generieren Sie eine AWL-Quelle mit dem zu ändernden Baustein und allen direkt oder indirekt referenzierenden Bausteinen.
2. Speichern Sie die Änderungen in der erzeugten Quelle.
3. Übersetzen Sie die angepasste Quelle zurück in die Bausteine.

Nun können Sie die Schnittstellenänderung speichern/laden.

15.8 Vermeiden von Fehlern beim Aufrufen von Bausteinen

STEP 7 überschreibt Daten in DB-Register

STEP 7 ändert die Register der S7-300/S7-400-CPU bei verschiedenen Operationen. Die Inhalte der DB- und DI-Register beispielsweise werden getauscht, wenn Sie einen FB aufrufen. Dadurch kann der Instanz-DB des aufgerufenen FB geöffnet werden, ohne die Adresse des vorherigen Instanz-DB zu verlieren.

Wenn Sie mit der absoluten Adressierung arbeiten, können Fehler beim Zugreifen auf Daten auftreten, die in den Registern gespeichert sind: in einigen Fällen werden die Adressen in dem Register AR1 (Adressregister 1) und in dem DB-Register überschrieben. Dadurch kann es sein, dass Sie falsche Adressen lesen oder in falsche Adressen schreiben.



Gefahr

Gefahr von Sachschäden und Personenschäden bei der Verwendung von:

1. CALL FC, CALL FB, CALL Multiinstanz
2. vollqualifizierte DB-Zugriffe (z. B. DB20.DBW10)
3. Zugriffe auf Variablen eines zusammengesetzten Datentyps

Es kann dabei geschehen, dass die Inhalte von DB-Register (DB und DI), Adressregister (AR1, AR2) und Akkus (AKKU1, AKKU2) verändert werden.

Ebenso kann beim FB-CALL/FC-CALL das Verknüpfungsergebnis VKE nicht als zusätzlicher (impliziter) Parameter verwendet werden.

Wenn Sie die oben genannten Programmiermöglichkeiten nutzen, müssen Sie selbst für eine Wiederherstellung der Inhalte Sorge tragen, da es sonst zu einem Fehlverhalten kommen kann.

Speichern von korrekten Daten

Kritisch wird es mit den Inhalten des DB-Registers, wenn Sie im abgekürzten Format der absoluten Adressen auf Daten zugreifen. Gehen Sie beispielsweise davon aus, dass DB 20 geöffnet ist (und seine Nummer im DB-Register gespeichert ist), können Sie DBX0.2 angeben, um auf die Daten zuzugreifen, die in Bit 2 von Byte 0 des DB gespeichert sind, dessen Adresse in dem DB-Register abgelegt ist (also DB 20). Enthält das DB-Register allerdings eine andere DB-Nummer, greifen Sie auf die falschen Daten zu.

Fehler beim Zugreifen auf Daten des DB-Registers können Sie vermeiden, wenn Sie die folgende Methode zum Adressieren von Daten verwenden:

- Verwenden Sie symbolische Adressierung
- Verwenden Sie die vollständige absolute Adresse (z. B. *DB20.DBX0.2*)

Bei diesen beiden Adressierungsmethoden öffnet STEP 7 automatisch den richtigen DB. Wenn Sie das Register AR1 für die indirekte Adressierung verwenden, müssen Sie immer die richtige Adresse in AR1 laden.

Situationen, in denen Register modifiziert werden

Die Manipulation der Adressregister zur indirekten Adressierung ist ausschließlich in AWL relevant. Die anderen Sprachen unterstützen keinen indirekten Zugriff auf die Adressregister.

Die Anpassung des DB-Registers durch den Compiler muss bei allen Programmiersprachen beachtet werden, um eine korrekte Parameterübergabe bei Bausteinaufrufen sicherzustellen.

In den folgenden Situationen werden die Inhalte des Adressregisters AR1 und des DB-Registers des aufrufenden Bausteins überschrieben:

Situation	Erläuterung
Bei Aktualparametern aus einem DB	<ul style="list-style-type: none"> Nachdem Sie einem Baustein einen Aktualparameter zugeordnet haben, der in einem DB gespeichert ist (z. B. DB20.DBX0.2), öffnet STEP 7 diesen DB (DB 20) und passt dabei den Inhalt des DB-Registers an. Das Programm arbeitet im Anschluss an den Bausteinaufruf dann mit dem angepassten DB.
Bei Aufruf von Bausteinen in Zusammenhang mit höheren Datentypen	<ul style="list-style-type: none"> Nach einem Bausteinaufruf aus einem FC, der eine Komponente eines Formalparameters eines höheren Datentyps (String, Array, Struct oder UDT) an den aufgerufenen Baustein übergibt, wird der Inhalt von AR1 und des DB-Registers des aufrufenden Bausteins modifiziert. Dasselbe gilt bei Aufruf aus einem FB, wenn der Parameter im VAR_IN_OUT Bereich des Aufrufers liegt.
Bei Zugriff auf Komponenten höheren Datentyps	<ul style="list-style-type: none"> Beim Zugriff eines FB auf eine Komponente eines Formalparameters höheren Datentyps im VAR_IN_OUT-Bereich (String, Array, Struct oder UDT) verwendet STEP 7 das Adressregister AR1 und das DB-Register. Dadurch werden die Inhalte der beiden Register modifiziert. Beim Zugriff eines FC auf eine Komponente eines Formalparameters höheren Datentyps (String, Array, Struct oder UDT) verwendet STEP 7 das Adressregister AR1 und das DB-Register. Dadurch werden die Inhalte der beiden Register modifiziert.

Hinweis

- Beim Aufruf eines FB aus einem Baustein mit Bausteinversion 1 wird der Aktualparameter für den ersten booleschen IN- bzw. IN_OUT-Parameter nicht korrekt übergeben, wenn der Befehl vor dem Call nicht VKE begrenzend ist. In diesem Fall wird er mit dem anstehenden VKE verknüpft.
- Beim Aufruf eines FB (Single oder Multiinstanz) wird das Adressregister AR2 beschrieben.
- Wird innerhalb eines FB das AR2 geändert, z. B. durch die Operationen UC, CC oder CALL (FC/SFC aufrufen ohne Parameter), ist die ordnungsgemäße Bearbeitung dieses FB nicht mehr gewährleistet.
- Wird einem ANY-Parameter nicht die vollständige absolute DB-Adresse übergeben, enthält der ANY-Pointer nicht die DB-Nummer des aufgeschlagenen DB, sondern immer die Nummer 0.

16 Projektieren von Meldungen

16.1 Grundlagen des Meldekonzeptes

Meldungen ermöglichen Ihnen, Fehler bei der Prozessbearbeitung in den Automatisierungssystemen schnell zu erkennen, genau zu lokalisieren und zu beheben. Stillstandszeiten einer Anlage können so wesentlich verkürzt werden.

Bevor Meldungen ausgegeben werden können, müssen sie projiziert werden.

Mit STEP 7 können Sie ereignisabhängige Meldungen mit zugeordneten Meldungstexten und Meldungsattributen anlegen, bearbeiten, übersetzen und auf Anzeigeräten ausgeben.

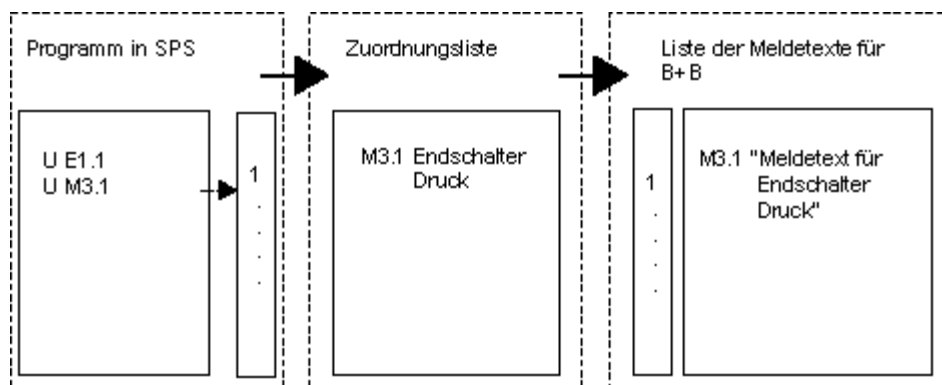
16.1.1 Welche Meldeverfahren gibt es?

Für das Erzeugen von Meldungen gibt es unterschiedliche Verfahren.

Bitmeldeverfahren

Das Bitmeldeverfahren erfordert 3 Arbeitsschritte durch den Programmierer:

- Er erstellt am PG das Anwenderprogramm und setzt das gewünschte Bit.
- Er erstellt mit einem beliebigen Texteditor eine Zuordnungsliste, in der dem Meldebit ein Meldetext zugewiesen wird (z. B. M 3.1 = Endschalter Druck).
- Er erstellt am Bediensystem auf der Basis der Zuordnungsliste die Liste der Meldetexte.

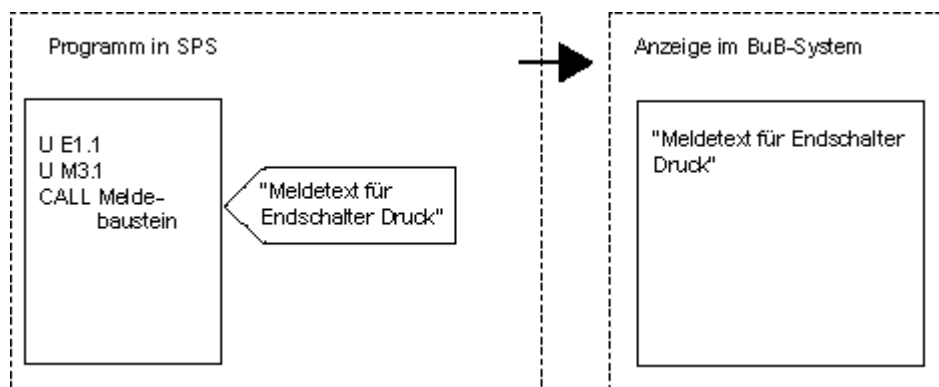


Das Bedien- und Beobachtungssystem fragt zyklisch das Automatisierungssystem ab, ob sich das Meldebit geändert hat oder nicht. Meldet das AS eine Änderung zurück, wird die entsprechende Meldung ausgegeben. Die Meldung erhält den Zeitstempel des Bedien- und Beobachtungssystems.

Meldenummernverfahren

Das Meldenummernverfahren erfordert nur einen Arbeitsschritt durch den Programmierer:

- Er erstellt am PG das Anwenderprogramm, setzt das gewünschte Bit und ordnet unmittelbar beim Programmieren dem Bit den gewünschten Meldetext zu.



Eine zyklische Abfrage des Automatisierungssystems findet nicht statt. Sobald das AS eine Änderung meldet, wird die entsprechende Meldenummer an das Bedien- und Beobachtungssystem weitergereicht und der zugehörige Meldetext ausgegeben. Die Meldung hat den Zeitstempel des AS und ist daher genauer zuzuordnen als beim Bitmeldeverfahren.

16.1.2 Auswahl des Meldeverfahrens

Übersicht

Die folgende Tabelle zeigt die Eigenschaften und Randbedingungen der unterschiedlichen Meldeverfahren.

Meldenummernverfahren	Bitmeldeverfahren
<ul style="list-style-type: none"> Meldungen werden in einer für PG und Bediensystem gemeinsamen Datenbasis verwaltet Die Busbelastung ist gering (AS meldet aktiv) Meldungen erhalten Zeitstempel des Automatisierungssystems 	<ul style="list-style-type: none"> Es gibt keine gemeinsame Datenbasis für PG und Bediensystem Die Busbelastung ist hoch (Bediensystem pollt) Meldungen erhalten Zeitstempel des Bediensystems

Das Meldenummernverfahren kennt die folgenden drei Arten von Meldungen:

Bausteinbezogene Meldungen	Symbolbezogene Meldungen	Anwenderdefinierte Diagnosemeldungen
<ul style="list-style-type: none"> programmsynchron Anzeige über ProTool (nur ALARM_S) und WinCC Möglich bei S7-300/400 Programmierung über Meldebausteine: <ul style="list-style-type: none"> ALARM ALARM_8 ALARM_8P NOTIFY NOTIFY_8P ALARM_S(Q) AR_SEND ALARM_D(Q) Übertragung in das Bediensystem <ul style="list-style-type: none"> für WinCC über AS-OS-Engineering für ProTool über ProTool-Funktionen 	<ul style="list-style-type: none"> programmasynchron Anzeige über WinCC nur möglich bei S7-400 Projektierung über die Symboltabelle Übertragung in das AS über Systemdatenbausteine (SDBs) Übertragung in das Bediensystem über AS-OS-Engineering 	<ul style="list-style-type: none"> programmsynchron Anzeige im Diagnosepuffer auf dem PG möglich bei S7-300/400 Programmierung über Meldebaustein (Systemfunktion) <ul style="list-style-type: none"> WR_USMSG keine Übertragung in das Bediensystem

STEP 7 unterstützt nur das komfortablere Meldenummernverfahren, das im weiteren Verlauf ausführlich beschrieben wird. Das Bitmeldeverfahren wird in den HMI-Geräten projektiert und ist dort beschrieben.

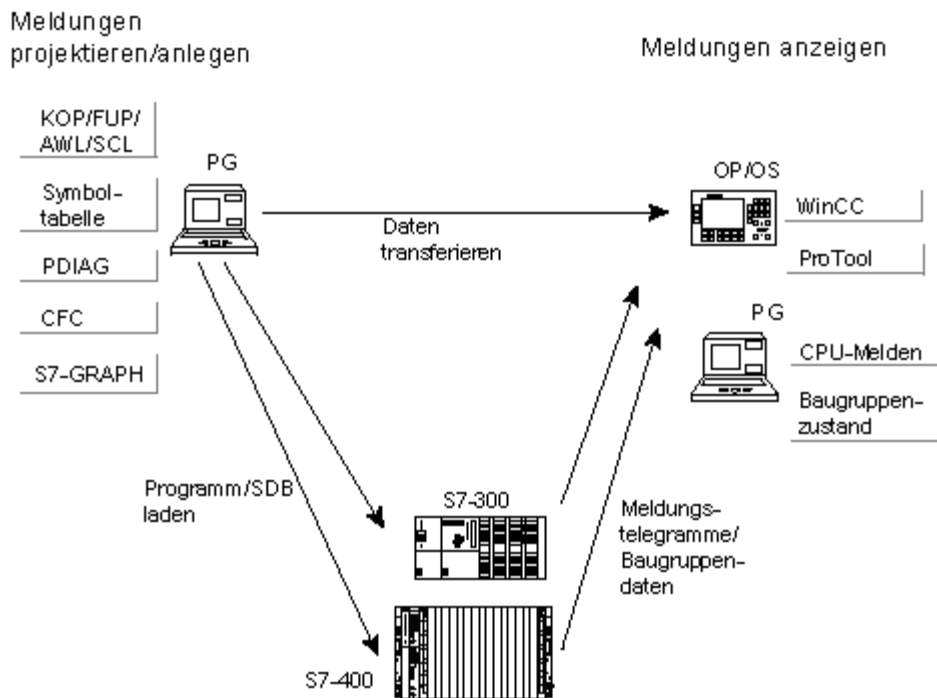
Beispiele für das Meldenummernverfahren

Meldeverfahren	Anwendungsbereich
Bausteinbezogene Meldungen	Zum Melden von programmsynchronen Ereignissen, z. B. soll angezeigt werden, dass ein Regler einen Grenzwert erreicht hat
Symbolbezogene Meldungen	Zum Melden von Ereignissen, die programmunabhängig sind, z. B. soll eine Schaltereinstellung überwacht werden
Anwenderdefinierte Meldungen	Zum Melden von Diagnoseereignissen im Diagnosepuffer, bei jedem Aufruf der SFC

16.1.3 SIMATIC-Komponenten

Übersicht

Das folgende Bild zeigt in einer Übersicht, welche SIMATIC-Komponenten beim Projektieren und Anzeigen von Meldungen beteiligt sind.



16.1.4 Bestandteile einer Meldung

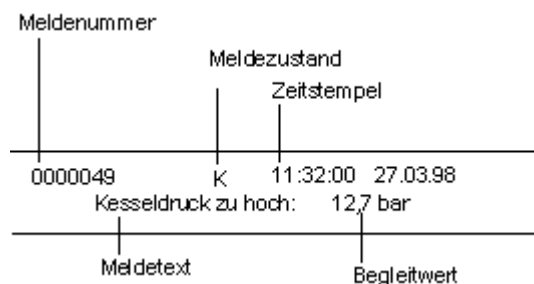
Wie eine Meldung angezeigt wird, ist abhängig vom Meldeverfahren, vom verwendeten Meldebaustein und vom Anzeigegerät.

Mögliche Bestandteile sind in der folgenden Tabelle aufgelistet:

Bestandteil	Beschreibung
Zeitstempel	Wird beim Auftreten des Meldeereignisses im Automatisierungssystem erzeugt
Meldezustand	Möglich sind: gekommen, gegangen, gegangen ohne Quittierung, gegangen mit Quittierung
Begleitwert	Einigen Meldungen kann ein Prozesswert beigefügt werden, der vom verwendeten Meldebaustein ausgewertet werden kann
Abbild	Bei einem Systemabsturz können die angefallenen Meldungen im Nachhinein an der OS angezeigt werden.
Meldenummer	Projektweit bzw. CPU-weit eindeutige Nummer, die vom System vergeben wird und eine Meldung identifiziert.
Meldetexte	Werden vom Anwender projektiert

Beispiel

Das folgende Beispiel zeigt eine Störmeldung an einem Operator Panel.



16.1.5 Welche Meldebausteine gibt es?

Sie haben die Auswahl zwischen den folgenden Meldebausteinen, in denen bereits eine Meldefunktion programmiert ist:

- SFB 33: "ALARM"
- SFB 34: "ALARM_8"
- SFB 35 "ALARM_8P"
- SFB 36 "NOTIFY"
- SFC 18: "ALARM_S" und SFC 17: "ALARM_SQ"
- SFB 37: "AR_SEND" (zum Senden von Archiven; keine Projektierung von Meldetexten und Meldeattributen möglich)
- SFB 31: "NOTIFY_8P"
- SFC 107: "ALARM_DQ"
- SFC 108: "ALARM_D"

Nähere Informationen finden Sie in der Referenzhilfe zu Bausteinen.

Wann setzen Sie welchen Meldebaustein ein?

Die folgende Tabelle hilft Ihnen bei der Entscheidung, welchen Meldebaustein Sie für Ihren Anwendungsfall auswählen. Die Auswahl des Meldebausteins ist abhängig von:

- der Anzahl der im Baustein zur Verfügung stehenden Kanäle und damit die Anzahl der Signale, die je Bausteinaufruf überwacht werden,
- der Möglichkeit, Meldungen zu quittieren,
- der Möglichkeit, Begleitwerte mitzugeben,
- den Anzeigegeräten, die eingesetzt werden sollen,
- dem Mengengerüst Ihrer CPU.

Meldebaustein	Kanäle	Quittierung	Begleitwerte	Anzeige WinCC	Anzeige PRO-TOOL	Anzeige CPU Melden/S7-Status	AS	Besonderheiten
ALARM SFB 33	1	möglich	bis zu 10	ja	nein	nein	S7-400	Sendet bei kommender oder gehender Flanke je eine Meldung
ALARM_8 SFB 34	8	möglich	nein	ja	nein	nein	S7-400	Sendet bei kommender oder gehender Flanke eines oder mehrerer Signale eine Meldung
ALARM_8P SFB 35	8	möglich	bis zu 10	ja	nein	nein	S7-400	wie ALARM_8
NOTIFY SFB 36	1	nein	bis zu 10	ja	nein	nein	S7-400	wie ALARM
NOTIFY_8P SFB 31	8	nein	bis zu 10	ja	nein	nein	S7-400	wie NOTIFY
AR_SEND SFB 37	1	-	-	ja	nein	nein	S7-400	dient zum Senden eines Archivs; keine Projektierung von Meldetexten und Meldeattributen möglich
ALARM_SQ SFC 17	1	möglich	1	ja	ja*	ja	S7-300/400	Nicht ein Flankenwechsel, sondern jeder SFC-Aufruf erzeugt eine Meldung
ALARM_S SFC 18	1	nein	1	ja	ja*	ja	S7-300/400	wie ALARM_SQ
ALARM_DQ SFC 107	1	möglich	1	ja	ja*	ja	S7-300/400	wie ALARM_SQ
ALARM_D SFC 108	1	nein	1	ja	ja*	ja	S7-300/400	wie ALARM_SQ
* je nach OP-Typ								

16.1.6 Formalparameter, Systemattribute und Meldebausteine

Formalparameter als Meldenummerneingang

Für jede Meldung oder Gruppe von Meldungen benötigen Sie in Ihrem Programm einen Formalparameter, den Sie als IN-Parameter in der Variablenübersicht Ihres Programms angeben. Der Formalparameter wird als Meldenummerneingang benutzt und bildet die Basis für eine Meldung.

Versorgung der Formalparameter mit Systemattributen

Als Voraussetzung für den Einstieg in die Meldungsprojektierung müssen Sie die Formalparameter mit Systemattributen versorgt haben:

1. Sie haben folgende Systemattribute für Parameter angefügt: "S7_server" und "S7_a_type"
2. Den Systemattributen haben Sie Werte zugewiesen, die den Meldebausteinen entsprechen, die Sie in Ihrem Programmcode aufgerufen haben. Der Wert für "S7_server" ist immer "alarm_archiv", der Wert für "S7_a_type" entspricht dem aufgerufenen Meldebaustein.

Systemattribute und zugehörige Meldebausteine

Als Objekte des Meldeservers werden nicht die Meldebausteine selbst angezeigt, sondern die zugehörigen Werte der Systemattribute "S7_a_type". Diese Werte haben die gleichen Bezeichnungen wie die Meldebausteine, die als SFB oder SFC vorhanden sind (Ausnahme: "alarm_s").

S7_a_type	Meldebaustein	Bezeichnung	Eigenschaften
alarm_8	ALARM_8	SFB 34	8 Kanäle, Quittierung möglich, keine Begleitwerte
alarm_8p	ALARM_8P	SFB 35	8 Kanäle, Quittierung möglich, bis zu 10 Begleitwerte pro Kanal
notify	NOTIFY	SFB 36	1 Kanal, keine Quittierung, bis zu 10 Begleitwerte
alarm	ALARM	SFB 33	1 Kanal, Quittierung möglich, bis zu 10 Begleitwerte
alarm_s	ALARM_S	SFC 18	1 Kanal, keine Quittierung, bis zu 1 Begleitwert
alarm_s	ALARM_SQ	SFC 17	1 Kanal, Quittierung möglich, bis zu 1 Begleitwert
ar_send	AR_SEND	SFB 37	dient zum Senden eines Archivs
notify_8p	NOTIFY_8P	SFB 31	8 Kanäle, keine Quittierung, bis zu 10 Begleitwerte
alarm_s	ALARM_DQ	SFC 107	1 Kanal, Quittierung möglich, bis zu 1 Begleitwert
alarm_s	ALARM_D	SFC 108	1 Kanal, keine Quittierung möglich, bis zu 1 Begleitwert

Nähere Informationen finden Sie in der Referenzhilfe zu Systemattributen.

Die Systemattribute werden automatisch zugewiesen, wenn die Meldebausteine, die Sie in Ihrem Programm verwenden, SFBs oder FBs mit entsprechenden Systemattributen sind und als Multiinstanzen aufgerufen werden.

16.1.7 Meldungstyp und Meldungen

Mit der Meldungsprojektierung können Sie in verschiedenen Arbeitsgängen entweder einen Meldungstyp oder Meldungen erstellen. Dies hängt von dem meldefähigen Baustein ab, über den Sie in die Meldungsprojektierung einsteigen.

Der meldefähige Baustein kann ein FB oder ein Instanz-DB sein

- Bei einem FB können Sie einen Meldungstyp erstellen, der als Vorlage für Meldungen dient. Alle Eingaben, die Sie für den Meldungstyp machen, werden automatisch in die Meldungen übernommen. Wenn Sie dem FB einen Instanz-DB zuordnen, werden nach der Vorlage des Meldungstyps automatisch Meldungen zum Instanz-DB generiert und Meldungsnummern zugeordnet.
- Bei einem Instanz-DB können Sie die Meldungen, die aus dem Meldungstyp generiert wurden, instanzspezifisch modifizieren.

Der sichtbare Unterschied ist, dass bei den Meldungen Meldungsnummern zugeordnet wurden, bei einem Meldungstyp dagegen noch nicht.

Verriegelung der Daten beim Meldungstyp

Mit der Meldungsprojektierung geben Sie Texte und Attribute zu ereignisabhängigen Meldungen ein. Dabei können Sie z. B. festlegen, wie die Meldungen auf bestimmten Anzeigegeräten angezeigt werden. Zur Erleichterung der Erstellung der Meldungen können Sie mit Vorlagen, den Meldungstypen, arbeiten.

- Wenn Sie Daten (Attribute und Texte) für den Meldungstyp eingeben, können Sie festlegen, ob sie verriegelt werden sollen oder nicht. Bei verriegelten Attributen wird ein Schlüsselsymbol neben das Eingabefeld gesetzt bzw. es wird ein Häkchen in der Spalte "Verriegelt" eingetragen. Verriegelte Texte erhalten ein Häkchen in der Spalte "Verriegelt".
- Im Meldungstyp verriegelte Daten können Sie in den instanzspezifischen Meldungen nicht mehr ändern. Die Daten werden nur angezeigt.
- Wenn Sie dennoch Änderungen benötigen, müssen Sie wieder zum Meldungstyp zurück gehen, die Verriegelung wieder aufheben, und die Änderungen dort durchführen. Die Änderungen gelten allerdings nicht für Instanzen, die vor der Änderung generiert wurden.

Ändern der Daten beim Meldungstyp

Ob sich die Änderungen von Daten beim Meldungstyp auf die Instanzen auswirken, hängt davon ab, ob Sie beim Erstellen des Projekts die Meldenummern projektweit oder CPU-weit vergeben haben.

- Projektweite Vergabe der Meldenummern: Wenn Sie nachträglich Daten beim Meldungstyp ändern, die auch für die Instanzen gelten sollen, so müssen Sie diese Änderungen ebenfalls an den Instanzen eingeben.

- CPU-weite Vergabe der Meldenummern: Wenn Sie nachträglich Daten beim Meldungstyp ändern, so werden diese Änderungen automatisch an den Instanzen übernommen.
Ausnahmen: Sie haben diese Daten zuvor an der Instanz geändert bzw. nachträglich am Meldungstyp Daten verriegelt oder entriegelt. Wenn Sie einen FB und einen Instanz-DB aus einem Projekt mit projektweiter Vergabe der Meldenummern in ein Projekt mit CPU-weiter Vergabe der Meldenummern kopieren, so müssen Sie die Änderung der Daten, die Sie zuvor am Meldungstyp durchgeführt haben, ebenfalls an der Instanz nachführen.

Achtung

- Wenn Sie Instanzen in ein anderes Programm kopieren, ohne dass Sie den Meldungstyp mitkopieren, so kann es vorkommen, dass die Instanz nicht vollständig angezeigt wird. Kopieren Sie in diesem Fall den Meldungstyp mit in das neue Programm.
 - Werden Texte und Attribute an einer Instanz grün dargestellt, so bedeutet dies folgendes: Diese Texte und Attribute sind noch so, wie sie am Typ projiziert wurden. Sie wurden zwischenzeitlich an der Instanz nicht verändert.
-

16.1.8 Erzeugen einer AWL-Quelle von meldenden Bausteinen

Wenn Sie eine AWL-Quelle von meldenden Bausteinen erzeugen, werden auch die Projektierinformationen in der Quelle mit abgelegt.

Die Information wird in einem Pseudokommentar abgelegt, der mit "\$ALARM_SERVER" beginnt und mit "*" endet.

Achtung

Wenn Sie einen Baustein symbolisch referenzieren, achten Sie darauf, dass vor dem Übersetzen der Quelle die Symboltabelle nicht verändert werden darf.

Sind mehrere Bausteine in der Quelle enthalten, werden mehrere Pseudokommentar-Blöcke zu einem Kommentar-Block zusammengefasst. Einzelne Bausteine mit Meldungsattributen dürfen nicht aus der AWL-Quelle gelöscht werden.

16.1.9 Vergabe von Meldenummern

Sie können entscheiden, ob Sie die Meldenummern projektweit oder CPU-weit vergeben möchten. Die CPU-weite Vergabe der Meldenummern hat den Vorteil, dass Sie ein Programm kopieren können, ohne dass sich die Meldenummern ändern und neu übersetzt werden müssen. Wenn Sie CPU-weite Meldenummern gewählt haben und diese in einem HMI-Gerät anzeigen lassen möchten, ist dies nur mit den Applikationen "WinCC V6.0" und/oder "ProTool V6.0" möglich. Sollten Sie mit Versionen von diesen Applikationen kleiner als V6.0 arbeiten, müssen Sie projektweite Meldenummern auswählen.

16.1.10 Unterschiede zwischen projektweiter und CPU-weiter Vergabe von Meldenummern

In der nachfolgenden Tabelle sind die Unterschiede zwischen der projektweiten und CPU-weiten Vergabe von Meldenummern aufgelistet:

Projektweit	CPU-weit
Einige Meldungsattribute und -texte sind abhängig vom eingesetzten HMI-Gerät und müssen anzeigespezifisch projektiert werden.	Die zugeordneten Attribute und Texte sind unabhängig vom eingesetzten HMI-Gerät, d. h. Sie müssen kein Anzeigegerät mehr einfügen und für dieses Gerät eine anzeigespezifische Meldung projektieren.
Beim Kopieren von Programmen muss eine Neuübersetzung stattfinden.	Programme können innerhalb eines Projekts und projektübergreifend kopiert werden. Beim Kopieren einzelner Bausteine ist jedoch eine Neuübersetzung erforderlich.
Wenn Sie nachträglich Daten (Texte und Attribute) beim Meldungstyp ändern, so werden müssen Sie diese Änderungen an den Instanzen nachziehen.	Wenn Sie nachträglich Daten (Texte und Attribute) beim Meldungstyp ändern, so werden diese Änderungen automatisch an den Instanzen übernommen (Ausnahme: Sie haben diese Daten zuvor an der Instanz geändert).
Sie können Texte nur einzellig eingeben.	Sie können Texte auch mehrzeilig eingeben.

16.1.11 Möglichkeiten zum Abändern der Meldenummernvergabe eines Projekts

Die Art der Meldenummernvergabe kann im SIMATIC Manager im Register "Meldenummern" (Menübefehl **Extras > Einstellungen**) für zukünftige neue Projekte und Bibliotheken voreingestellt werden. In diesem Register legen Sie fest, ob Meldenummern CPU-weit eindeutig oder projektweit eindeutig vergeben werden sollen. Sie können auch "Einstellung immer abfragen" wählen, um eine Zuordnung zu einem späteren Zeitpunkt zu treffen.

War zum Erstellzeitpunkt eines Projekts oder einer Bibliothek die Voreinstellung "CPU-weit eindeutig" oder "projektweit eindeutig" aktiv, so kann die Art der Meldenummernvergabe in diesem Projekt bzw. in dieser Bibliothek nicht mehr geändert werden.

Falls die Meldenummernvergabe "projektweit eindeutig" eingestellt wurde und Sie wünschen eine "CPU-weit eindeutige" Vergabe, so gehen Sie wie folgt vor:

1. Markieren Sie im SIMATIC Manager das entsprechende Projekt bzw. die Bibliothek.
2. Wählen Sie den Menübefehl **Datei > Speichern unter**.
3. Aktivieren Sie im Folgedialog das Kontrollkästchen "Mit Reorganisation" und geben einen neuen Namen an.
4. Starten Sie den Vorgang "Speichern unter" mit der Schaltfläche "OK".
5. In einem der folgenden Dialoge können Sie die "CPU-weit eindeutige" Meldenummernvergabe festlegen.

Mit dem Menübefehl **Datei > Löschen** können Sie das ursprüngliche Projekt bzw. die ursprüngliche Bibliothek löschen.

16.2 Projektweites Projektieren von Meldungen

16.2.1 Projektweite Vergabe von Meldenummern

Die Meldungen sind durch eine projektweit eindeutige Nummer identifiziert. Zu diesem Zweck wird den einzelnen S7-Programmen je ein Nummernbereich innerhalb des zur Verfügung stehenden Gesamtbereichs (1 - 2097151) zugeordnet. Kopiert man ein Programm, so muss im Konfliktfall - wenn im Zielbereich bereits die gleichen Meldenummern vergeben sind - dem neuen Programm ein anderer Nummernbereich zugeordnet werden. STEP 7 öffnet in einem solchen Fall automatisch das Dialogfeld, in dem Sie den neuen Nummernbereich angeben können.

Außerdem können Sie, wenn im Programm noch keine Meldungen projiziert sind, mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldenummern** den Nummernbereich für ein S7-Programm festlegen oder ändern.

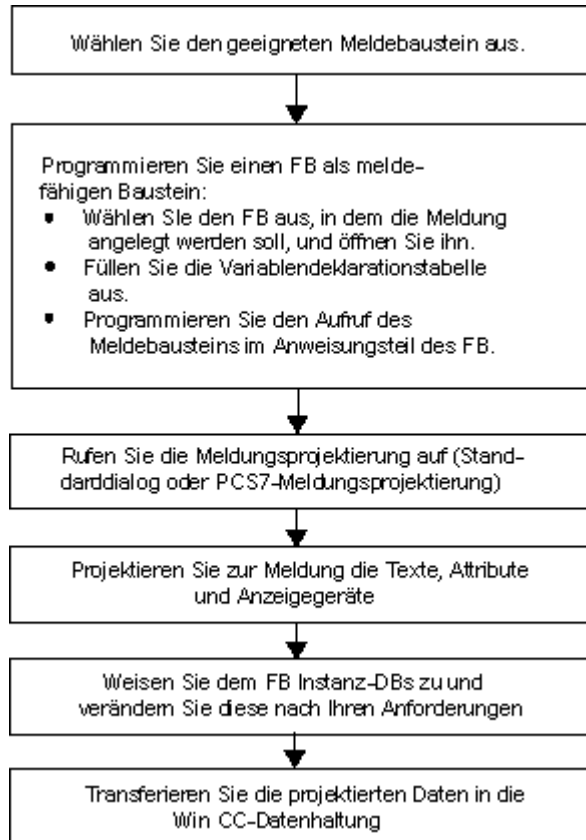
Die Meldenummernbereiche werden defaultmäßig in 20.000er Schritten vergeben.

16.2.2 Bausteinbezogene Meldungen zuordnen und bearbeiten

Bausteinbezogene Meldungen werden einem Baustein (Instanz-DB) zugeordnet. Für das Erstellen einer bausteinbezogenen Meldung können Sie Systemfunktionsbausteine (SFBs) und Systemfunktionen (SFCs) als Meldebausteine verwenden.

16.2.2.1 Bausteinbezogene Meldungen anlegen (projektweit)

Prinzipielle Vorgehensweise



Meldefähigen Baustein (FB) programmieren

1. Wählen Sie im SIMATIC Manager den Funktionsbaustein (FB) aus, für den Sie eine bausteinbezogene Meldung anlegen wollen und öffnen Sie diesen Baustein durch Doppelklick.

Ergebnis: Der angewählte Baustein wird im Fenster "KOP/AWL/FUP" geöffnet und angezeigt.

2. Füllen Sie die Variablendeklarationstabelle aus. Für jeden Meldebaustein, der im FB aufgerufen wird, müssen Sie im aufrufenden FB Variablen deklarieren.

Tragen Sie dazu in der Variablenübersicht folgende Variablen ein:

- beim Parameter "IN" einen symbolischen Namen für den Meldebausteineingang, z. B. "Meld01" (für Meldeeingang 01) und den Datentyp (muss "DWORD" ohne Anfangswert sein).
- beim Parameter "STAT" einen symbolischen Namen für den aufzurufenden Meldebaustein, z. B. "alarm" und den entsprechenden Datentyp, hier also "SFB33".

3. Fügen Sie im Anweisungsteil des FBs den Aufruf für den ausgewählten Meldebaustein ein, hier also "CALL alarm", und schließen Sie Ihre Eingabe mit der Taste RETURN ab.

Ergebnis: Im Anweisungsteil des FBs werden die Eingangsvariablen des aufgerufenen Meldebausteins, hier also des SFB 33, angezeigt.

4. Weisen Sie der Variable "EV_ID" den von Ihnen in Schritt 2 vergebenen symbolischen Namen für den Meldebausteineingang, hier also "Meld01", zu. Nun werden die Systemattribute für die Meldung vom Typ "alarm" übernommen.

Ergebnis: In der Spalte "Name" beim Parameter "IN" erscheint, falls diese nicht selektiert ist, ein "Fähnchen". Damit ist der ausgewählte Baustein meldefähig. Die benötigten Systemattribute (z. B. S7_server und S7_a_type) und die entsprechenden Werte werden automatisch zugewiesen (bei einigen SFCs müssen Sie die Systemattribute für den Parameter "IN" selbst zuweisen (Menübefehl **Bearbeiten > Objekteigenschaften**, Register "Attribute"))).

Achtung: Wenn Sie statt eines SFBs einen FB mit Multiinstanzen aufrufen, in dem auch Meldungen projektiert sind, müssen Sie im aufrufenden Baustein ebenfalls die Meldungen des FBs mit Multiinstanzen projektieren.

5. Wiederholen Sie die Schritte 2. bis 4. für alle Aufrufe von Meldebausteinen in diesem FB.
6. Speichern Sie den Baustein mit dem Menübefehl **Datei > Speichern**.
7. Schließen Sie auch das Fenster "KOP/AWL/FUP".

Meldungsprojektierung aufrufen

- Selektieren Sie den gewünschten Meldebaustein und wählen Sie im SIMATIC Manager den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung**.

Ergebnis: Das Dialogfeld für die STEP 7-Meldungsprojektierung (Standarddialog) wird geöffnet. Was Sie tun müssen, um die PCS 7-Meldungsprojektierung aufzurufen, lesen Sie unter PCS 7-Meldungsprojektierung.

Meldungstyp bearbeiten

1. Selektieren Sie den gewünschten Meldebaustein, öffnen Sie die Meldungsprojektierung und tragen Sie in den Registern "Text" und "Attribute" den gewünschten Meldungstext ein bzw. wählen Sie die gewünschten Meldungsattribute aus.
Falls Sie einen mehrkanaligen Meldebaustein (z. B. "ALARM_8") selektiert haben, können Sie jeder Subnummer eigene Texte und zum Teil eigene Attribute zuweisen.
2. Weisen Sie dem Meldungstyp die gewünschten Anzeigeräte zu, indem Sie auf die Schaltfläche "Neues Gerät" klicken und im nachfolgenden Dialogfeld "Anzeigerät einfügen" die gewünschten Anzeigeräte auswählen.

Tragen Sie dazu in den folgenden Registerblättern die gewünschten Texte und Attribute für die Anzeigergeräte ein. Verlassen Sie das Dialogfeld mit "OK".

Hinweis

Zum Bearbeiten der anzeigergerätespezifischen Texte und Attribute beachten Sie bitte die mit Ihrem Anzeigergerät mitgelieferte Dokumentation.

Instanz-DBs anlegen

1. Nachdem Sie einen Meldungstyp angelegt haben, können Sie diesem nun Instanz-Datenbausteine (DBs) zuweisen und die Meldungen für diese DBs instanzspezifisch bearbeiten.
Öffnen Sie dazu im SIMATIC Manager den Baustein, der Ihren zuvor projektierten FB aufrufen soll, z. B. "OB1", indem Sie darauf doppelklicken. Tragen Sie im geöffneten Anweisungsteil des OBs den Aufruf ("CALL"), sowie den Namen und die Nummer des aufzurufenden FBs und des DBs ein, den Sie dem FB als Instanz zuweisen wollen. Schließen Sie Ihre Eingabe ab mit der Taste RETURN.

Beispiel: Tragen Sie "CALL FB1, DB1" ein. Falls der DB1 noch nicht existiert, so bestätigen Sie die Frage, ob der Instanz-DB generiert werden soll, mit "Ja".

Ergebnis: Der Instanz-DB wird angelegt. Im Anweisungsteil des OBs werden die Eingangsvariablen des zugehörigen FBs, hier z. B. "Meld01" und die vom System vergebene Meldungsnummer, hier die "1", angezeigt.

2. Speichern Sie den OB mit dem Menübefehl **Datei > Speichern** und schließen Sie das Fenster "KOP/AWL/FUP".

Meldungen bearbeiten

1. Selektieren Sie im SIMATIC Manager den erzeugten Instanz-DB, z. B. "DB1" und rufen Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung** die Meldungsprojektierung auf.

Ergebnis: Das Dialogfeld "Meldungsprojektierung" wird geöffnet und der selektierte Instanz-DB mit der vom System zugewiesenen Meldungsnummer wird angezeigt.

2. Tragen Sie die gewünschten Änderungen für den jeweiligen Instanz-DB in den verschiedenen Registerblättern ein und fügen Sie weitere Anzeigergeräte hinzu, falls gewünscht. Verlassen Sie die Funktion mit "OK".

Ergebnis: Damit ist die Meldungsprojektierung für den angewählten Instanz-DB abgeschlossen.

Projektierungsdaten transferieren

- Transferieren Sie die projektierten Daten in die Datenhaltung von WinCC (über die AS-OS-Verbindungsprojektierung) bzw. in die Datenhaltung von ProTool.

16.2.2.2 Bausteinbezogene Meldungen bearbeiten (projektweit)

1. Selektieren Sie im SIMATIC Manager den gewünschten Baustein und wählen Sie den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung**.
2. Klicken Sie in der Verzeichnisstruktur auf einen Meldebausteineingang oder auf eine seiner Subnummern (falls vorhanden).

Ergebnis: Der Registerteil für eine allgemeine Meldung wird angezeigt.

3. Geben Sie im Register "Text" und im Register "Attribute" die Texte und Attribute ein.

Ergebnis: Sie haben eine Standardmeldung erstellt, die auf allen Anzeigegeräten angezeigt werden kann.

4. Fügen Sie über die Schaltfläche "Neues Gerät" ein neues Anzeigegerät vom Typ "ProTool" (Opx) oder "WinCC" ein. Es werden nur Anzeigegeräte zur Auswahl angeboten, auf denen die projektierten Meldungen angezeigt werden können.

Ergebnis: Das neue Gerät wird eingefügt und selektiert, der dazugehörige Registerteil wird angezeigt.

5. Geben Sie in den anzeigespezifischen Registern "Texte" und "Attribute" für die anzeigespezifische Meldung ein.

Ergebnis: Sie haben eine Meldungsvariante erstellt, die nur für die Meldung an dem ausgewählten Anzeigegerät benutzt wird.

Wenn Sie weitere Meldungsvarianten für bereits vorhandene Anzeigegeräte bearbeiten wollen:

- Selektieren und öffnen Sie den Meldebaustein in der Detailansicht durch Doppelklick.

Ergebnis: Das erste Anzeigegerät wird automatisch ausgewählt, und Sie können die anzeigespezifische Meldungsvariante dazu bearbeiten.

16.2.2.3 PCS 7-Meldungsprojektierung (projektweit)

Für das Bearbeiten von Meldungstypen und Meldungen, die auf WinCC-Anzeigegeräten ausgegeben werden sollen, bietet STEP 7 mit der PCS 7-Meldungsprojektierung eine komfortable Möglichkeit

- die Projektierung von Anzeigegeräten zu vereinfachen (werden automatisch angelegt)
- die Eingabe von Attributen und Texten für Meldungen zu vereinfachen
- die Einheitlichkeit von Meldungen zu gewährleisten.

Aufrufen der PCS 7-Meldungsprojektierung

1. Selektieren Sie im SIMATIC Manager den Baustein (FB oder DB), dessen Meldetexte Sie bearbeiten wollen und rufen Sie mit dem Menübefehl **Bearbeiten > Objekteigenschaften** das Dialogfeld für die Eingabe der Systemattribute auf.
2. Tragen Sie in die angezeigte Tabelle das Systemattribut "S7_alarm_ui" und den Wert: "1" ein (der Wert "0" schaltet die PCS 7-Meldungsprojektierung aus). Eigenschaften-Parameter können in KOP/FUP/AWL vergeben werden. Anschließend generierte und dem entsprechenden FB zugeordnete DBs übernehmen die Einstellungen und können über die eigenen Attributeinstellungen unabhängig vom Meldungstyp (FB) umgeschaltet werden.

Hinweis

Beim Eintragen der Systemattribute wird eine Syntaxprüfung durchgeführt und die fehlerhaften Eingaben werden rot gekennzeichnet.

1. Verlassen Sie das Dialogfeld mit "OK".
2. Wählen Sie den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung**.

Ergebnis: Das Dialogfeld für die PCS 7-Meldungsprojektierung wird geöffnet.

Bearbeiten von Meldungstypen

1. Selektieren Sie im SIMATIC Manager den FB, dessen Meldetexte Sie bearbeiten wollen und rufen Sie die PCS 7-Meldungsprojektierung auf.
Ergebnis: Im Dialogfeld wird für jeden Meldebaustein, für den Sie im FB eine Variable deklariert haben, ein Registerblatt angezeigt.
2. Füllen Sie die Textfelder für die Meldungsbestandteile "Herkunft", "OS-Bereich" und "Batch-Kennung" aus.
3. Geben Sie für alle Ereignisse der verwendeten Meldebausteine die Meldeklasse und den Ereignistext ein und legen Sie fest, ob jedes Ereignis einzeln quittiert werden muss.
4. Klicken Sie für die Meldungsbestandteile, die für alle Instanzen gelten und dort nicht verändert werden sollen, das Kontrollkästchen "Verriegelt" an.

Bearbeiten von Meldungen

1. Selektieren Sie im SIMATIC Manager den Instanz-DB, dessen Meldetexte Sie bearbeiten wollen und rufen Sie die PCS 7-Meldungsprojektierung auf.
2. Verändern Sie die nicht verriegelten instanzspezifischen Meldungsbestandteile.

16.2.3 Symbolbezogene Meldungen zuordnen und bearbeiten

16.2.3.1 Symbolbezogene Meldungen zuordnen und bearbeiten (projektweit)

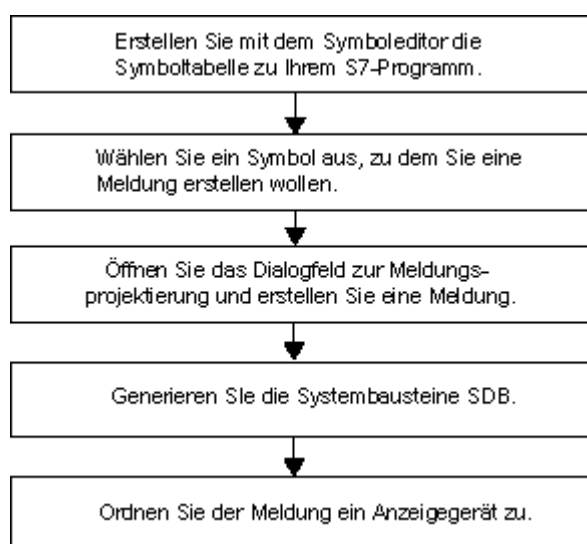
Symbolbezogenen Meldungen (SCAN) werden direkt einem Signal in der Symboltabelle zugeordnet. Zulässige Signale sind alle booleschen Operanden: also Eingänge (E), Ausgänge (A) und Merker (M). Diesen Signalen können Sie in der Meldungsprojektierung verschiedene Attribute, Meldungstexte und bis zu 10 Begleitwerte zuordnen. Die Auswahl von Signalen aus der Symboltabelle wird Ihnen dadurch erleichtert, dass Sie Filter einstellen können.

Mit einer symbolbezogenen Meldung können Sie in einem vorgegebenen Zeitraster ein Signal abtasten, um zu erfassen, ob ein Signalwechsel vorliegt.

Hinweis

Das Zeitraster ist abhängig von der verwendeten CPU.

Prinzipielle Vorgehensweise



Während des Ablaufs werden asynchron zu Ihrem Programm die Signale überprüft, für die Sie Meldungen projiziert haben. Die Überprüfungen finden in den projizierten Zeitrastern statt. Die Meldungen werden auf den zugeordneten Anzeigergeräten angezeigt.

Achtung

Wenn Sie symbolbezogene Meldungen zuordnen oder bearbeiten wollen und im gleichen Arbeitsgang vorher Symbole zwischen zwei Symboltabellen kopiert haben, so müssen Sie die Symboltabelle, in der Sie nicht mehr arbeiten möchten, zuerst schließen. Ihre Projektierungen für die Meldungen können sonst nicht gespeichert werden. Unter Umständen gehen die letzten Einträge im Projektierungsdialog für Meldungen verloren.

16.2.4 Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten

Mit dieser Funktion können Sie einen Anwendereintrag in den Diagnosepuffer schreiben und eine entsprechende Meldung senden, die Sie in der Meldungsprojektierung erstellen. Anwenderdefinierte Diagnosemeldungen werden durch die System-Funktion SFC 52 (WR_USMSG; Fehlerklasse A oder B) realisiert, die als Meldebaustein genutzt wird. Den Aufruf für die SFC 52 müssen Sie in Ihr Anwenderprogramm einfügen und ihn mit der Ereignis-ID versorgen.

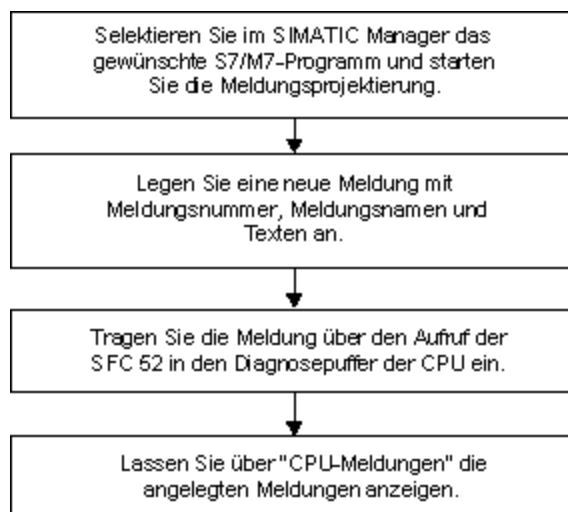
Voraussetzungen

Bevor Sie eine anwenderdefinierte Diagnosemeldung anlegen können, müssen Sie

- im SIMATIC Manager ein Projekt angelegt haben und
- im Projekt ein S7/M7-Programm erzeugt haben, dem Sie eine oder mehrere Meldungen zuordnen wollen.

Prinzipielle Vorgehensweise

Um eine anwenderdefinierte Diagnosemeldung anzulegen und anzeigen zu lassen, gehen Sie so vor:



16.3 CPU-weites Projektieren von Meldungen

16.3.1 CPU-weite Vergabe von Meldenummern

Die Meldungen sind durch eine CPU-weit eindeutige Nummer identifiziert. Zu diesem Zweck wird den einzelnen CPUs je ein Nummernbereich zugeordnet. Wenn Sie ein Programm kopieren, müssen Sie - im Gegensatz zur projektweiten Vergabe der Meldenummern - dem neuen Programm keinen neuen Nummernbereich zuordnen. Dadurch entfällt das Neuübersetzen von Programmen. Eine Ausnahme bildet das Kopieren einzelner Bausteine. Hier ist eine Neuübersetzung notwendig, um die geänderte Meldenummer in das Programm einzubinden.

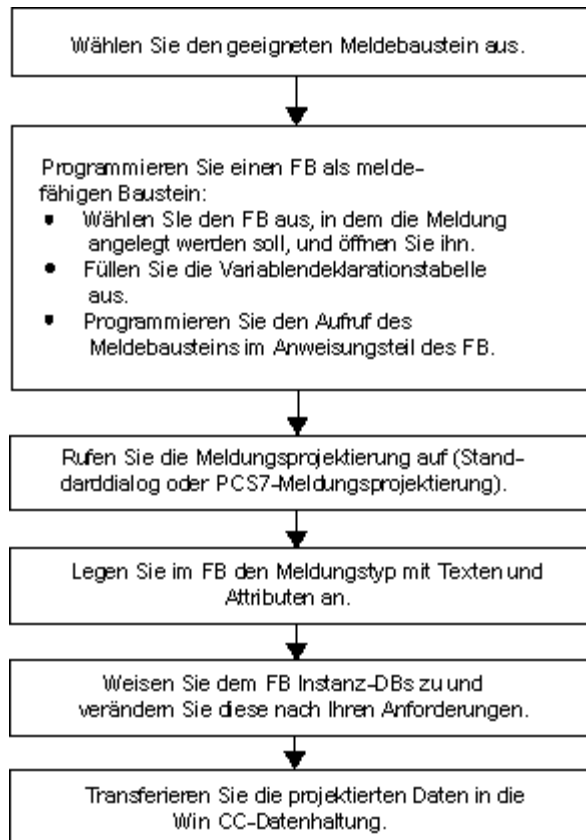
Voraussetzungen

- WinCC V6.0
- ProTool V6.0

16.3.2 Bausteinbezogene Meldungen zuordnen und bearbeiten

16.3.2.1 Bausteinbezogene Meldungen anlegen (CPU-weit)

Prinzipielle Vorgehensweise



Meldefähigen Baustein (FB) programmieren

1. Wählen Sie im SIMATIC Manager den Funktionsbaustein (FB) aus, für den Sie eine bausteinbezogene Meldung anlegen wollen und öffnen Sie diesen Baustein durch Doppelklick.

Ergebnis: Der angewählte Baustein wird im Fenster "KOP/AWL/FUP" geöffnet und angezeigt.

2. Füllen Sie die Variablendeklarationstabelle aus. Für jeden Meldebaustein, der im FB aufgerufen wird, müssen Sie im aufrufenden FB Variablen deklarieren.

Tragen Sie dazu in der Variablenübersicht folgende Variablen ein:

- beim Parameter "IN" einen symbolischen Namen für den Meldebausteineingang, z. B. "Meld01" (für Meldeeingang 01) und den Datentyp (muss "DWORD" ohne Anfangswert sein).

- beim Parameter "STAT" einen symbolischen Namen für den aufzurufenden Meldebaustein, z. B. "alarm" und den entsprechenden Datentyp, hier also "SFB33".
- 3. Fügen Sie im Anweisungsteil des FBs den Aufruf für den ausgewählten Meldebaustein ein, hier also "CALL alarm", und schließen Sie Ihre Eingabe mit der Taste RETURN ab.

Ergebnis: Im Anweisungsteil des FBs werden die Eingangsvariablen des aufgerufenen Meldebausteins, hier also des SFB 33, angezeigt.

- 4. Weisen Sie der Variable "EV_ID" den von Ihnen in Schritt 2 vergebenen symbolischen Namen für den Meldebausteineingang, hier also "Meld01", zu.

Ergebnis: In der Spalte "Name" beim Parameter "IN" erscheint, falls diese nicht selektiert ist, ein "Fähnchen". Damit ist der ausgewählte Baustein meldefähig. Die benötigten Systemattribute (z. B. S7_server und S7_a_type) und die entsprechenden Werte werden automatisch zugewiesen (bei einigen SFCs müssen Sie die Systemattribute für den Parameter "IN" selbst zuweisen (Menübefehl **Bearbeiten > Objekteigenschaften**, Register "Attribute")).

Achtung: Wenn Sie statt eines SFBs einen FB mit Multiinstanzen aufrufen, in dem auch Meldungen projiziert sind, müssen Sie im aufrufenden Baustein ebenfalls die Meldungen des FBs mit Multiinstanzen projektieren.

- 5. Wiederholen Sie die Schritte 2. bis 4. für alle Aufrufe von Meldebausteinen in diesem FB.
- 6. Speichern Sie den Baustein mit dem Menübefehl **Datei > Speichern**.
- 7. Schließen Sie das Fenster "KOP/AWL/FUP".

Meldungsprojektierung aufrufen

- Selektieren Sie den gewünschten Meldebaustein und wählen Sie im SIMATIC Manager den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung**.

Ergebnis: Das Dialogfeld für die STEP 7-Meldungsprojektierung wird geöffnet. Was Sie tun müssen, um die PCS7-Meldungsprojektierung aufzurufen, lesen Sie unter PCS7-Meldungsprojektierung (CPU-weit).

Meldungstyp bearbeiten

- Selektieren Sie den gewünschten Meldebaustein.
- Tragen Sie in den entsprechenden Spalten die gewünschten Texte ein bzw. wählen Sie die gewünschten Attribute aus.
Sie können auch im Dialogfeld "Meldungsprojektierung" auf die Schaltfläche "Erweitert" klicken und im Register "Standardtexte" den gewünschten Meldungstext und Zusatztext eintragen.
Falls Sie einen mehrkanaligen Meldebaustein (z. B. "ALARM_8") selektiert haben, können Sie jeder Subnummer eigene Texte und zum Teil eigene Attribute zuweisen.
- Wenn die Texte bzw. Attribute an der Instanz nicht änderbar sein sollen, verriegeln Sie diese im Meldungstyp.

Instanz-DBs anlegen

1. Nachdem Sie einen Meldungstyp angelegt haben, können Sie diesem nun Instanz-Datenbausteine (DBs) zuweisen und die Meldungen für diese DBs instanzspezifisch bearbeiten.
Öffnen Sie dazu im SIMATIC Manager den Baustein, der Ihren zuvor projektierten FB aufrufen soll, z. B. "OB1", indem Sie darauf doppelklicken. Tragen Sie im geöffneten Anweisungsteil des OBs den Aufruf ("CALL"), sowie den Namen und die Nummer des aufzurufenden FBs und des DBs ein, den Sie dem FB als Instanz zuweisen wollen. Schließen Sie Ihre Eingabe ab mit der Taste RETURN.

Beispiel: Tragen Sie "CALL FB1, DB1" ein. Falls der DB1 noch nicht existiert, so bestätigen Sie die Frage, ob der Instanz-DB generiert werden soll, mit "Ja".

Ergebnis: Der Instanz-DB wird angelegt. Im Anweisungsteil des OBs werden die Eingangsvariablen des zugehörigen FBs, hier z. B. "Meld01" und die vom System vergebene Meldungsnummer, hier die "1", angezeigt.

2. Speichern Sie den OB mit dem Menübefehl **Datei > Speichern** und schließen Sie das Fenster "KOP/AWL/FUP".

Meldungen bearbeiten

1. Selektieren Sie im SIMATIC Manager den erzeugten Instanz-DB, z. B. "DB1" und rufen Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung** die Meldungsprojektierung auf.

Ergebnis: Das Dialogfeld "Meldungsprojektierung" wird geöffnet und der selektierte Instanz-DB mit der vom System zugewiesenen Meldungsnummer wird angezeigt.

2. Tragen Sie die gewünschten Änderungen für den jeweiligen Instanz-DB ein. Verlassen Sie die Funktion mit "OK".

Ergebnis: Damit ist die Meldungsprojektierung für den angewählten Instanz-DB abgeschlossen.

Hinweis

Werden Texte und Attribute an einer Instanz grün dargestellt, so bedeutet dies folgendes: Diese Texte und Attribute sind noch so, wie sie am Typ projektiert wurden. Sie wurden zwischenzeitlich an der Instanz nicht verändert.

Projektierungsdaten transferieren

- Transferieren Sie die projektierten Daten in die Datenhaltung von WinCC (über die AS-OS-Verbindungsprojektierung) bzw. in die Datenhaltung von ProTool.

16.3.2.2 Bausteinbezogene Meldungen bearbeiten (CPU-weit)

1. Selektieren Sie einen Meldebaustein und rufen Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung** die Meldungsprojektierung auf.
2. Geben Sie in den Spalten "Standardtexte" und "Zusatztexte" die gewünschten Texte ein.
Sie können auch auf die Schaltfläche "Erweitert" klicken und in den aufgeblendeten Registern "Standardtexte" und "Zusatztexte" die gewünschten Texte mit Zeilenumbrüchen eingeben.

Ergebnis: Sie haben eine Standardmeldung erstellt.

Hinweis

Werden Texte und Attribute an einer Instanz grün dargestellt, so bedeutet dies folgendes: Diese Texte und Attribute sind noch so, wie sie am Typ projiziert wurden. Sie wurden zwischenzeitlich an der Instanz nicht verändert.

16.3.2.3 PCS 7-Meldungsprojektierung (CPU-weit)

Für das Bearbeiten von Meldungstypen und Meldungen, die auf WinCC-Anzeigegeräten (ab V6.0) ausgegeben werden sollen, bietet STEP 7 mit der PCS 7-Meldungsprojektierung eine komfortable Möglichkeit

- die Projektierung von Anzeigegeräten zu vereinfachen
- die Eingabe von Attributen und Texten für Meldungen zu vereinfachen
- die Einheitlichkeit von Meldungen zu gewährleisten.

Aufrufen der PCS 7-Meldungsprojektierung

1. Selektieren Sie im SIMATIC Manager den Baustein (FB oder DB), dessen Meldetexte Sie bearbeiten wollen und rufen Sie mit dem Menübefehl **Bearbeiten > Objekteigenschaften** das Dialogfeld für die Eingabe der Systemattribute auf.
2. Tragen Sie in die angezeigte Tabelle das Systemattribut "S7_alarm_ui" und den Wert: "1" ein (der Wert "0" schaltet die PCS 7-Meldungsprojektierung aus).
Eigenschaften-Parameter können in KOP/FUP/AWL vergeben werden.
Anschließend generierte und dem entsprechenden FB zugeordnete DBs übernehmen die Einstellungen und können über die eigenen Attributeinstellungen unabhängig vom Meldungstyp (FB) umgeschaltet werden.

Hinweis

Beim Eintragen der Systemattribute wird eine Syntaxprüfung durchgeführt und die fehlerhaften Eingaben werden rot gekennzeichnet.

3. Verlassen Sie das Dialogfeld mit "OK".
4. Wählen Sie den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung**.

Ergebnis: Das Dialogfeld für die PCS 7-Meldungsprojektierung wird geöffnet.

Bearbeiten von Meldungstypen

1. Selektieren Sie im SIMATIC Manager den FB, dessen Meldetexte Sie bearbeiten wollen und rufen Sie die PCS 7-Meldungsprojektierung auf.
2. Klicken Sie auf die Schaltfläche "Erweitert" und füllen Sie im Register "Meldetexte Baustein" die Textfelder für die Meldungsbestandteile "Herkunft", "OS-Bereich" und "Batch-Kennung" aus.
3. Geben Sie für alle Ereignisse der verwendeten Meldebausteine die Meldeklasse und den Ereignistext ein und legen Sie fest, ob jedes Ereignis einzeln quittiert werden muss.
4. Klicken Sie für die Meldungsbestandteile, die für alle Instanzen gelten und dort nicht verändert werden sollen, das Kontrollkästchen "Verriegelt" an.

Bearbeiten von Meldungen

1. Selektieren Sie im SIMATIC Manager den Instanz-DB, dessen Meldetexte Sie bearbeiten wollen und rufen Sie die PCS 7-Meldungsprojektierung auf.
2. Verändern Sie die nicht verriegelten instanzspezifischen Meldungsbestandteile.

16.3.3 Symbolbezogene Meldungen zuordnen und bearbeiten

16.3.3.1 Symbolbezogene Meldungen zuordnen und bearbeiten (CPU-weit)

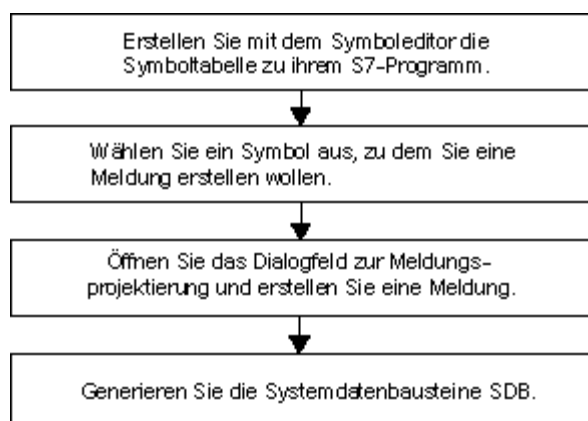
Symbolbezogenen Meldungen (SCAN) werden direkt einem Signal in der Symboltabelle zugeordnet. Zulässige Signale sind alle booleschen Operanden: also Eingänge (E), Ausgänge (A) und Merker (M). Diesen Signalen können Sie in der Meldungsprojektierung verschiedene Attribute, Meldungstexte und bis zu 10 Begleitwerte zuordnen. Die Auswahl von Signalen aus der Symboltabelle wird Ihnen dadurch erleichtert, dass Sie Filter einstellen können.

Mit einer symbolbezogenen Meldung können Sie in einem vorgegebenen Zeitraster ein Signal abtasten, um zu erfassen, ob ein Signalwechsel vorliegt.

Hinweis

Das Zeitraster ist abhängig von der verwendeten CPU.

Prinzipielle Vorgehensweise



Während des Ablaufs werden asynchron zu Ihrem Programm die Signale überprüft, für die Sie Meldungen projiziert haben. Die Überprüfungen finden in den projizierten Zeitrastern statt. Die Meldungen werden auf den Anzeigegeräten angezeigt.

Achtung

Wenn Sie symbolbezogene Meldungen zuordnen oder bearbeiten wollen und im gleichen Arbeitsgang vorher Symbole zwischen zwei Symboltabellen kopiert haben, so müssen Sie die Symboltabelle, in der Sie nicht mehr arbeiten möchten, zuerst schließen. Ihre Projektierungen für die Meldungen können sonst nicht gespeichert werden. Unter Umständen gehen die letzten Einträge im Projektierungsdialog für Meldungen verloren.

16.3.4 Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten

Mit dieser Funktion können Sie einen Anwendereintrag in den Diagnosepuffer schreiben und eine entsprechende Meldung senden, die Sie in der Meldungsprojektierung erstellen. Anwenderdefinierte Diagnosemeldungen werden durch die System-Funktion SFC 52 (WR_USMSG; Fehlerklasse A oder B) realisiert, die als Meldebaustein genutzt wird. Den Aufruf für die SFC 52 müssen Sie in Ihr Anwenderprogramm einfügen und ihn mit der Ereignis-ID versorgen.

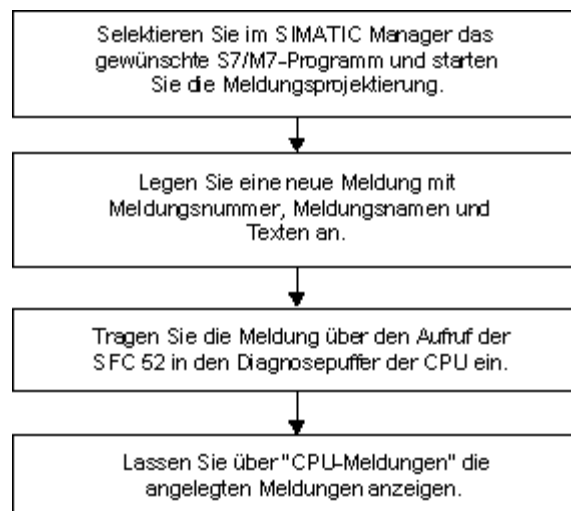
Voraussetzungen

Bevor Sie eine anwenderdefinierte Diagnosemeldung anlegen können, müssen Sie

- im SIMATIC Manager ein Projekt angelegt haben und
- im Projekt ein S7/M7-Programm erzeugt haben, dem Sie eine oder mehrere Meldungen zuordnen wollen.

Prinzipielle Vorgehensweise

Um eine anwenderdefinierte Diagnosemeldung anzulegen und anzeigen zu lassen, gehen Sie so vor:



16.4 Tipps zum Bearbeiten von Meldungen

16.4.1 Begleitwerte in Meldungen einfügen

Um bausteinbezogenen und symbolbezogenen Meldungen aktuelle Informationen, z. B. aus dem Prozess, mitzugeben, können Sie Begleitwerte an beliebigen Stellen eines Meldungstextes einfügen.

Dazu gehen Sie folgendermaßen vor:

1. Stellen Sie einen Block mit folgendem Aufbau zusammen:
@<Nr. des Begleitwertes><Elementtyp><Formatangabe>@.
2. Fügen Sie diesen Block an den Stellen im Meldungstext ein, an denen der Begleitwert angezeigt werden soll.

Elementtyp

Hiermit wird der Datentyp des Begleitwertes eindeutig projiziert:

Elementtyp	Datentyp
Y	BYTE
W	WORD
X	DWORD
I	Integer
D	Integer
B	BOOL
C	CHAR
R	REAL

Der Elementtyp macht nur den von der AS übertragenen Datentyp eindeutig. Er wird nicht als Casting Operator verwendet.

Formatangabe

Bestimmen Sie das Ausgabeformat für den Begleitwert auf dem Anzeigegerät. Die Formatangabe wird durch das Zeichen "%" eingeleitet. Für Meldungstexte gibt es folgende feste Formatangaben:

Formatangabe	Beschreibung
%[i]X	Hexadezimalzahl mit i Stellen
%[i]u	Dezimalzahl ohne Vorzeichen mit i Stellen
%[i]d	Dezimalzahl mit Vorzeichen mit i Stellen
%[i]b	Binärzahl mit i Stellen
%[i][.y]f	Festpunktzahl Vorzeichenbehafteter Wert der Form [-]ddd.dddd ddd: eine oder mehrere Ziffern mit y Stellen nach dem Dezimalpunkt und i Gesamtstellen
%[i]s	Zeichenkette (ANSI String) mit i Stellen Zeichen werden gedruckt bis zum ersten 0 Byte (00Hex).
%t#<Name der Textbibliothek>	Zugriff auf Textbibliothek.

Ist die Formatangabe zu klein, so wird der Wert trotzdem in voller Länge ausgegeben.

Ist die Formatangabe zu groß, so wird vor dem Wert eine passende Anzahl Leerzeichen ausgegeben.

Hinweis

Bitte beachten Sie, dass Sie "[i]" optional angeben können, wobei Sie die eckigen Klammern bei der Eingabe weglassen müssen.

Beispiele für Begleitwerte:

@1%6d@: Der Wert aus Begleitwert 1 wird als Dezimalzahl mit maximal 6 Stellen dargestellt.

@2R%6f@: Der Wert "5.4" zum Beispiel aus Begleitwert 2 wird als Festpunktzahl "5.4" dargestellt (drei führende Blanks).

@2R%2f@: Der Wert "5.4" zum Beispiel aus Begleitwert 2 wird als Festpunktzahl "5.4" dargestellt (bei zu kleiner Stellenzahl wird nicht abgeschnitten).

@1W%t#Textbib1@: Begleitwert 1 vom Datentyp WORD ist der Index, mit dem in der Textbibliothek Textbib1 der einzusetzende Text referenziert wird.

Hinweis

Bei S7-PDIAG muss als Elementtyp für CHAR immer "C" und für REAL immer "R" angegeben werden. Für die anderen für S7-PDIAG gültigen Elementtypen BOOL, BYTE, WORD, INT, DWORD und DINT muss immer "X" angegeben werden.

Möchten Sie einem der ALARM_S-Bausteine mehr als einen Begleitwert übergeben, können Sie einen Array mit maximal 12 Byte Länge übergeben. Dies können z. B. maximal 12 Byte oder Char, maximal 6 Word oder Int bzw. maximal 3 DWord, Real oder DInt sein.

16.4.2 Texte aus Textbibliotheken in Meldungen integrieren

In eine Meldung können Sie beliebig viele Texte aus maximal 4 verschiedenen Textbibliotheken integrieren. Die Texte können frei platziert werden, so ist auch die Verwendung in fremdsprachigen Meldungen gewährleistet.

Gehen Sie folgendermaßen vor:

1. Selektieren Sie im SIMATIC Manager die CPU oder ein der CPU unterlagertes Objekt und wählen Sie den Menübefehl **Extras > Textbibliotheken > System-Textbibliothek** bzw. **Extras > Textbibliotheken > Anwender-Textbibliothek**, um eine Textbibliothek zu öffnen.

Achtung

Sie können nur Texte aus Anwender-Textbibliotheken in Meldungen integrieren, wenn Sie die CPU-weite Vergabe der Meldenummern gewählt haben.

1. Ermitteln Sie den Index des Textes, den Sie integrieren möchten.
2. Geben Sie an der Stelle in der Meldung, an welcher der Text erscheinen soll, einen Platzhalter vom Format @[Index]%t#[Textbib]@ ein.

Hinweis

[Index] = z. B. 1W, wobei 1W der erste Begleitwert der Meldung vom Typ WORD ist.

Beispiel

Projektiertes Meldetext: Druck ist @2W%t#Textbib1@ gestiegen.

Textbibliothek mit dem Namen "Textbib1":

Index	deutsch	französisch
1734	zu hoch	trop haut

Der zweite übergebene Begleitwert ist mit dem Wert 1734 versorgt.

Folgender Meldetext wird angezeigt : Druck ist zu hoch gestiegen.

16.4.3 Begleitwerte löschen

Sie können Begleitwerte löschen, indem Sie im Meldungstext den String löschen, der einen Begleitwert repräsentiert.

Dazu gehen Sie folgendermaßen vor:

1. Suchen Sie den Block im Meldungstext, der dem Begleitwert, den Sie löschen wollen, entspricht.
Der Block beginnt mit dem Zeichen "@", enthält dann den Ortsbezeichner, an dem Sie den Begleitwert erkennen können, sowie eine Formatangabe, und endet mit dem Zeichen "@".
2. Löschen Sie den Block, den Sie gefunden haben, aus dem Meldungstext heraus.

16.5 Bedienerrelevante Texte übersetzen und bearbeiten

Texte, die während der Prozessbearbeitung auf Anzeigegeräten ausgegeben werden, werden üblicherweise in der Sprache eingegeben, in der die Automatisierungslösung programmiert wurde.

Häufig tritt jedoch der Fall ein, dass ein Bediener, der später auf die Meldungen reagieren soll, diese Sprache nicht beherrscht. Er benötigt die Texte in seiner Muttersprache. Nur so können einwandfreie Bearbeitung und schnelle Reaktion auf die ausgegebenen Meldungen gewährleistet werden.

STEP 7 bietet Ihnen die Möglichkeit, sämtliche bedienerrelevanten Texte in jede beliebige Sprache zu übersetzen. Einzige Voraussetzung ist, dass Sie die Sprache in Ihrem Projekt bereits installiert haben (Menübefehl im SIMATIC Manager: **Extras > Sprache für Anzeigegeräte**). Die Anzahl der angebotenen Sprachen wird bei der Installation von Windows festgelegt (Systemeigenschaft).

Sie können so sicherstellen, dass jeder, der zu einem späteren Zeitpunkt mit den ausgegebenen Meldungen konfrontiert wird, diese Texte in der von ihm gewünschten Sprache angezeigt bekommt. Auf diese Weise erhöhen Sie die Sicherheit im Prozess ganz wesentlich.

Bedienerrelevante Texte sind Anwendertexte und Textbibliotheken.

16.5.1 Anwendertexte übersetzen und bearbeiten

Anwendertexte können für ein ganzes Projekt, für S7-Programme, den Baustein-Ordner oder einzelne Bausteine und für die Symboltabelle angelegt werden, sofern Meldungen in diesen Objekten projiziert sind. Sie enthalten alle Texte und Meldungen, die z. B. auf Anzeigegeräten angezeigt werden können. Zu einem Projekt kann es mehrere Listen bedienerrelevanter Texte geben, die Sie in die erforderlichen Sprachen übersetzen können.

Die in einem Projekt verfügbaren Sprachen können Sie auswählen (Menübefehl: **Extras > Sprache für Anzeigegeräte**). Sie können auch nachträglich Sprachen hinzufügen oder löschen.

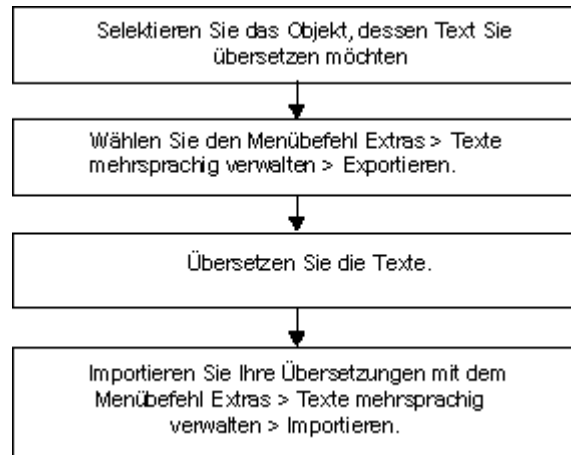
Exportieren und Importieren bedienerrelevanter Texte

Sie können in STEP 7 erzeugte bedienerrelevante Texte außerhalb von STEP 7 übersetzen oder bearbeiten. Dazu exportieren Sie die angezeigte Liste bedienerrelevanter Texte in Exportdateien, die Sie mit einem ASCII-Editor oder einem Tabellenbearbeitungswerkzeug, z. B. der Microsoft-Applikation EXCEL, bearbeiten können (Menübefehl **Extras > Texte mehrsprachig verwalten > Exportieren**). Nach dem Öffnen wird auf dem Bildschirm eine Tabelle angezeigt, deren Spalten jeweils eine Sprache darstellen. In der ersten Spalte wird immer die als Standardsprache eingestellte Sprache angezeigt. Nach dem Übersetzen der Texte importieren Sie diese wieder in STEP 7.

Bedienerrelevante Texte können nur in den Projektteil importiert werden, aus dem sie exportiert wurden.

Prinzipielle Vorgehensweise

Stellen Sie sicher, dass Sie im SIMATIC Manager mit dem Menübefehl **Extras > Sprache für Anzeigeräte** die Sprachen eingestellt haben, in die Sie die Anwendertexte übersetzen wollen.



Hinweis

Sie können Anwendertexte nur in der Applikation drucken, in der Sie die Texte übersetzen.

16.6 Textbibliotheken bearbeiten und übersetzen

16.6.1 Anwender-Textbibliotheken

Mit Hilfe einer Anwender-Textbibliothek können Texte oder Textteile dynamisch in Abhängigkeit von einem Begleitwert angezeigt werden. Der Begleitwert liefert dazu den Index in der Textbibliothek auf den aktuellen Text. An der Stelle, an der der dynamische Text eingeblendet werden soll, wird ein Platzhalter eingetragen.

Sie können zu einem Programm Anwender-Textbibliotheken anlegen, dort Texte eintragen und mit einem selbst gewählten Index versehen. Die Applikation prüft automatisch, ob der Index innerhalb der Anwender-Textbibliothek eindeutig ist. Eine Anwender-Textbibliothek kann von allen zur Verfügung stehenden Meldungen dieser CPU aus referenziert werden.

Ein Textbibliotheksordner kann beliebig viele Textbibliotheken enthalten. Damit ist es z. B. möglich, dasselbe Programm für verschiedene Steuerungsaufgaben zu verwenden und lediglich die Textbibliotheken den Erfordernissen anzupassen.

Achtung

Wird ein meldender Baustein, der eine Textbibliothek referenziert, in ein anderes Programm kopiert, so müssen Sie die zugehörigen Textbibliotheken ebenfalls kopieren, bzw. eine andere Textbibliothek mit gleichem Namen und Inhalt anlegen oder den Verweis im Meldetext ändern.

Beim Anlegen eines Texteintrages wird die Vergabe eines Indexes erzwungen. Defaultmäßig wird von der Applikation beim Anlegen einer neuen Zeile der nächste freie Index vorgeschlagen. Doppeldeutige Indizes innerhalb einer Textbibliothek sind nicht zulässig und werden von der Applikation abgewiesen.

16.6.2 Erstellen von Anwender-Textbibliotheken

Wenn Sie eine Anwender-Textbibliotheken erstellen möchten, gehen Sie folgendermaßen vor:

1. Selektieren Sie im SIMATIC Manager das Programm oder ein dem Programm unterlagertes Objekt, zu der bzw. dem Sie eine Anwender-Textbibliothek erstellen möchten und wählen Sie im SIMATIC Manager den Menübefehl **Einfügen > Textbibliothek > Textbibliotheksordner**.
Ergebnis: Der Ordner "Textbibliothek" wird angelegt.
2. Selektieren Sie nun den Ordner "Textbibliothek", wählen Sie den Menübefehl **Einfügen > Textbibliothek > Anwender-Textbibliothek** und geben Sie der Textbibliothek einen Namen.
3. Zum Öffnen der Textbibliothek wählen Sie den Menübefehl **Extras > Textbibliotheken > Anwender-Textbibliothek**.
4. Nun können Sie die gewünschten Texte eingeben.

Hinweis

Beim Anlegen eines Texteintrages wird die Vergabe eines Indexes erzwungen. Defaultmäßig wird von der Applikation beim Anlegen einer neuen Zeile der nächste freie Index vorgeschlagen. Doppeldeutige Indizes innerhalb einer Textbibliothek sind nicht zulässig und werden von der Applikation abgewiesen.

16.6.3 Bearbeiten von Anwender-Textbibliotheken

Wenn Sie bestehende Anwender-Textbibliotheken bearbeiten wollen, gehen Sie folgendermaßen vor:

1. Selektieren Sie im SIMATIC Manager das Programm oder ein dem Programm unterlagertes Objekt, dessen Textbibliothek Sie bearbeiten möchten, und wählen Sie im SIMATIC Manager den Menübefehl **Extras > Textbibliotheken > Anwender-Textbibliothek**.
2. Wählen Sie im Dialogfeld "Verfügbare Textbibliotheken" die Textbibliothek, die Sie öffnen möchten.
3. Bearbeiten Sie die angezeigten Texte. Ihnen werden hierzu verschiedene Bearbeitungsfunktionen angeboten (z. B. Suchen und Ersetzen). Sie können eigene Texte eingeben. Der Index für die einzelnen Texte wird automatisch erstellt, kann aber jederzeit geändert werden. Wenn Sie einen bereits vergebenen Index eintragen, so wird dieser Wert rot angezeigt. Zum Einfügen einer neuen Zeile wählen Sie entweder den Menübefehl **Einfügen > Neue Zeile** oder klicken Sie auf das entsprechende Symbol in der Symbolleiste.
4. Drucken Sie sich die Texte aus, wenn Sie sie auf Papier benötigen.
5. Nach Abschluss aller Arbeiten speichern Sie die Anwender-Textbibliothek.
6. Beenden Sie die Applikation, wenn Sie alle gewünschten Texte bearbeitet haben.

Achtung

Wird ein meldender Baustein, der eine Textbibliothek referenziert, in ein anderes Programm kopiert, so müssen Sie die zugehörigen Textbibliotheken ebenfalls kopieren, bzw. eine andere Textbibliothek mit gleichem Namen anlegen oder den Verweis im Meldetext ändern.

Wenn Sie den Namen einer bereits bestehenden Textbibliothek ändern, werden die auf diese Textbibliothek referenzierten Begleitwerte in bereits projektierten Meldungen ungültig!

16.6.4 System-Textbibliotheken

System-Textbibliotheken werden z. B. in "Systemfehler melden" beim Generieren von Bausteinen automatisch erzeugt. Sie können System-Textbibliotheken nicht selbst erstellen, sondern nur vorhandene Textbibliotheken bearbeiten.

Eine Textbibliothek kann von allen zur Verfügung stehenden Meldungen dieser CPU aus referenziert werden.

16.6.5 Textbibliotheken übersetzen

System-Textbibliotheken und Anwender-Textbibliotheken stellen eine Liste von Texten zur Verfügung, die in Meldungen integriert, dynamisch zur Laufzeit aktualisiert und am PG oder anderen Anzeigegeräten angezeigt werden können.

Die Texte in System-Textbibliotheken werden von STEP 7 oder STEP 7-Optionspaketen bereitgestellt. Zu einer CPU kann es mehrere Textbibliotheken geben, die Sie in die erforderlichen Sprachen übersetzen können.

Die in einem Projekt verfügbaren Sprachen können Sie im SIMATIC Manager auswählen (Menübefehl: **Extras > Sprache für Anzeigegeräte**). Sie können auch nachträglich Sprachen hinzufügen oder löschen.

Wenn Sie eine Textbibliothek übersetzen möchten (Menübefehl **Extras > Texte mehrsprachig verwalten > Exportieren**), wird eine Exportdatei erstellt, die Sie z. B. mit Microsoft EXCEL bearbeiten können. Nach dem Öffnen wird auf dem Bildschirm eine Tabelle angezeigt, deren Spalten jeweils eine Sprache darstellen.

Achtung

Eine Exportdatei im CSV-Format darf nicht über Doppelklick auf die Datei geöffnet werden. Öffnen Sie diese Datei über den Menübefehl **Datei > Öffnen** von Microsoft EXCEL.

Hinweis

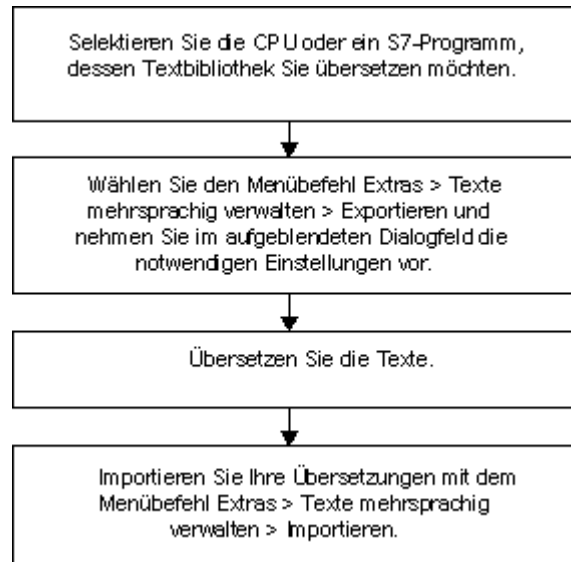
Sie können Anwendertexte nur in der Applikation drucken, in der Sie die Texte übersetzen.

Beispiel für eine Exportdatei

deutsch	französisch
ausgefallen	ne répond pas
gestört	défaillant
Parametrierfehler	erreur de paramétrage

Prinzipielle Vorgehensweise

Stellen Sie sicher, dass Sie im SIMATIC Manager mit dem Menübefehl **Extras > Sprache für Anzeigegeräte** die Sprachen eingestellt haben, in die Sie eine Textbibliothek übersetzen wollen.



16.7 Meldungsprojektierungsdaten zum Zielsystem transferieren

Einleitung

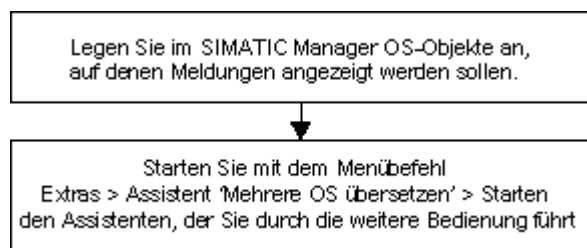
Mit dem Transferprogramm AS-OS-Engineering transferieren Sie die erstellten Meldungsprojektierungsdaten in den Datenbestand von WinCC.

Voraussetzungen

Bevor Sie mit dem Transfer beginnen, müssen folgende Voraussetzungen erfüllt sein:

- Sie haben das Programm AS-OS-Engineering installiert.
- Sie haben die Projektierungsdaten für das Anlegen von Meldungen erzeugt.

Prinzipielle Vorgehensweise



16.8 CPU-Meldungen und anwenderdefinierte Diagnosemeldungen anzeigen

Mit der Funktion "CPU-Meldungen" (Menübefehl **Zielsystem > CPU-Meldungen**) können asynchrone Meldungen von Diagnoseereignissen und anwenderdefinierte Diagnosemeldungen sowie Meldungen von ALARM_S-Bausteinen (SFC 18 und SFC 108 zur Erzeugung stets quittierter bausteinbezogener Meldungen sowie SFC 17 und SFC 107 zur Erzeugung quittierbarer bausteinbezogener Meldungen) ausgegeben werden.

Außerdem können Sie von CPU-Meldungen aus mit dem Menübefehl **Bearbeiten > Meldung > Anwenderdefinierte Diagnose** die Meldungsprojektierung starten und anwenderdefinierte Diagnosemeldungen anlegen.

Anzeigemöglichkeiten

Mit der Funktion "CPU-Meldungen" können Sie entscheiden, ob und wie Meldungen für ausgewählte CPUs angezeigt werden:

- **"In der Taskleiste hervorheben"**: Sobald eine Meldung empfangen wird, und sich das Fenster nicht im Vordergrund befindet, wird "CPU Melden" in der Windows-Taskleiste hervorgehoben.
- **"Im Hintergrund lassen"**: Das Empfangen der CPU-Meldungen läuft im Hintergrund. Das Fenster bleibt bei neuen Meldungen im Hintergrund und kann bei Bedarf in den Vordergrund geholt werden.
- **"Meldung ignorieren"**: Neue CPU-Meldungen werden **nicht** angezeigt und im Gegensatz zu den ersten beiden Fällen **nicht** archiviert.

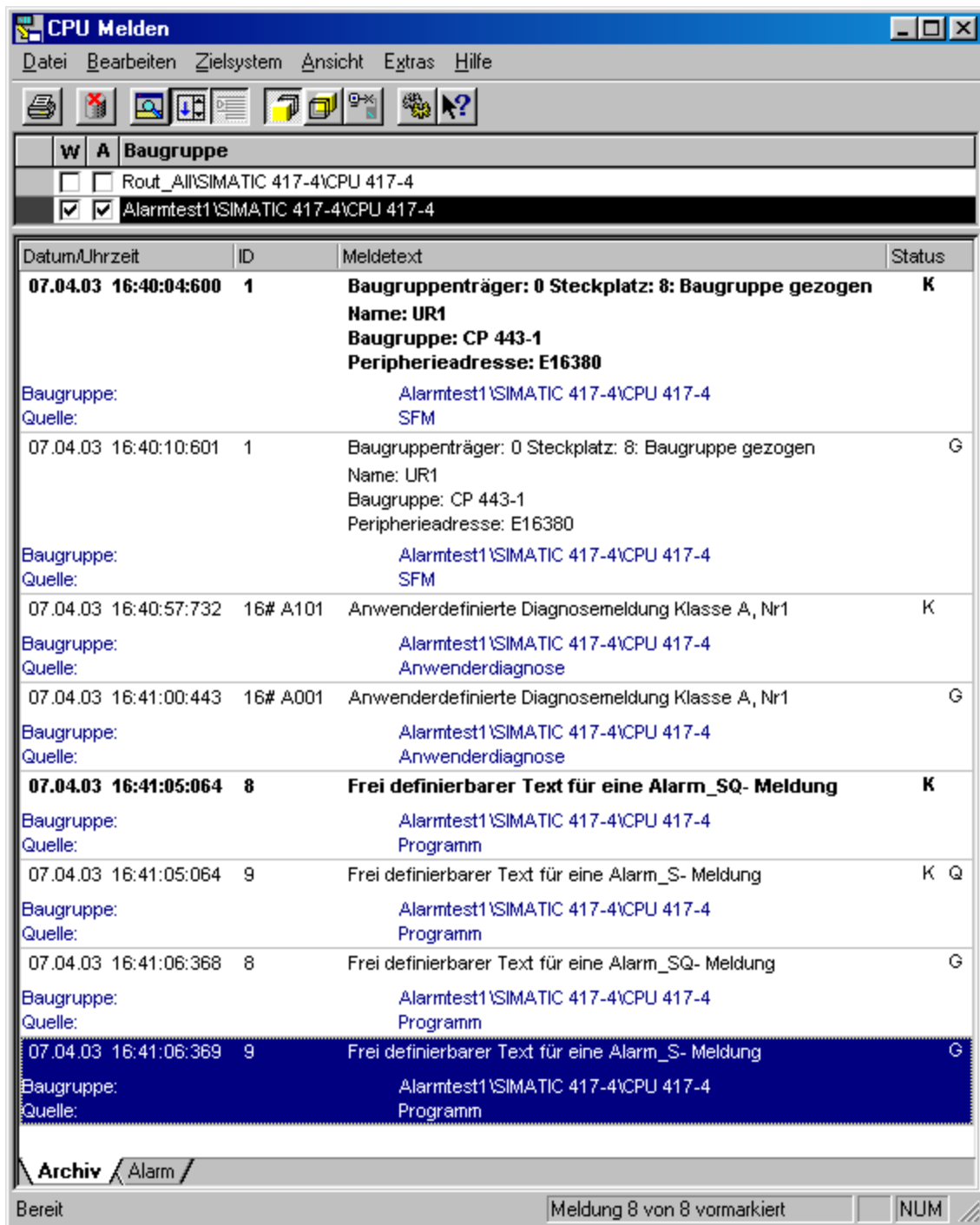
Im Fenster "CPU Melden" können Sie zwischen den Registern "Archiv" und "Alarm" wählen. Sie können in beiden Registern über den Menübefehl **Ansicht > Infotext anzeigen** wählen, ob die Meldungen mit oder ohne Infotext angezeigt werden.

Register "Archiv"

Hier werden ankommende Meldungen nach Ihrem zeitlichen Erscheinen angezeigt und archiviert. Die Archivgröße (zwischen 40 und 3000 CPU-Meldungen) können Sie über den Menübefehl **Extras > Einstellungen** im Dialogfeld "Einstellungen - CPU Melden" festlegen. Ist die eingestellte Archivgröße überschritten, wird jeweils die älteste vorhandene Meldung gelöscht.

Quittierbare Meldungen (ALARM_SQ und ALARM_DQ) werden in fetter Schrift angezeigt und können mit dem Menübefehl **Bearbeiten > CPU-Meldung quittieren** quittiert werden.

Einige Beispiele sind im folgenden Bild dargestellt:



Register "Alarm"

Noch nicht gegangene bzw. noch nicht quittierte Meldungen von ALARM_S-Bausteinen werden zusätzlich im Register "Alarm" mit Status angezeigt.

Sie können über den Menübefehl **Ansicht > Meldungen mehrzeilig** wählen, ob die Meldungen einzeilig oder mehrzeilig angezeigt werden. Außerdem können Sie die Spalten nach Ihren Vorstellungen sortieren.

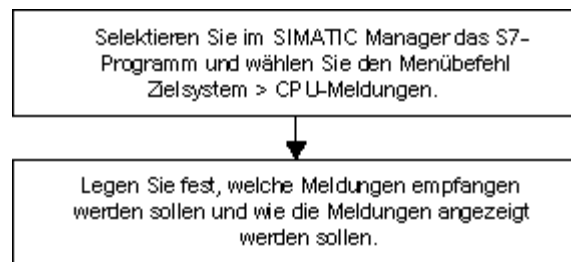
Meldungen von ALARM_S-Bausteinen aktualisieren

Beim Aktualisieren werden alle nicht gegangenen bzw. nicht quittierten Meldungen erneut in das Archiv eingetragen. Aktualisiert werden die Meldungen

- beim Wiederanlauf der Baugruppe, auf die sich die Meldungen beziehen (nicht bei Neustart)
- durch Anklicken des Feldes "A" für Meldungen von ALARM_S-Bausteinen in der Baugruppenliste.

Prinzipielle Vorgehensweise

Um CPU-Meldungen für ausgewählte Baugruppen zu konfigurieren,



16.8.1 CPU-Meldungen konfigurieren

Um CPU-Meldungen für ausgewählte Baugruppen zu konfigurieren, gehen Sie folgendermaßen vor:

1. Starten Sie im SIMATIC Manager die Applikation CPU Melden über ein Online-Projekt. Markieren Sie dazu online ein S7-Programm und rufen Sie mit dem Menübefehl **Zielsystem > CPU Meldungen** die Applikation CPU Melden für die ausgewählte CPU auf.

Ergebnis: Es erscheint das Applikationsfenster "CPU Melden", das die angemeldete CPU auflistet.

2. Melden Sie bei Bedarf weitere CPUs an, indem Sie Schritt 1 für weitere Programme oder Schnittstellen wiederholen.
3. Klicken Sie die Kontrollkästchen vor den Listeneinträgen an und legen Sie fest, welche Meldungen für die Baugruppe empfangen werden sollen:

A: aktiviert Meldungen von ALARM_S-Bausteinen (SFC 18 und SFC 108 zur Erzeugung stets quittierter bausteinbezogener Meldungen sowie SFC 17 und SFC 107 zur Erzeugung quittierbarer bausteinbezogener Meldungen), z. B. Prozessdiagnosemeldungen von S7-PDIAG; S7-GRAPH oder Systemfehler melden.

W: aktiviert Diagnoseereignisse.

4. Legen Sie die Archivgröße fest.

Ergebnis: Sobald die o.g. Meldungen auftreten, werden Sie in der von Ihnen festgelegten Form in das Meldungsarchiv geschrieben und angezeigt.

Hinweis

In der Liste der angemeldeten Baugruppen im Applikationsfenster "CPU Melden" sind die CPUs eingetragen, für die Sie den Menübefehl **Zielsystem > CPU-Meldungen** aufgerufen haben. Die Einträge in der Liste bleiben erhalten, bis sie im oben genannten Dialogfeld gelöscht werden.

16.8.2 Anzeigen von gespeicherten CPU-Meldungen

CPU-Meldungen werden immer im Archiv aufgezeichnet, sofern Sie nicht den Menübefehl **Ansicht > Meldung ignorieren** gewählt haben. Es werden immer alle archivierten Meldungen angezeigt.

16.9 'Melden von Systemfehlern' projektieren

Einführung

S7-Komponenten und "DP-Normslaves" (Slaves, deren Eigenschaften durch ihre GSD-Datei bestimmt werden) können bei Eintritt eines Systemfehlers Aufrufe von Organisationsbausteinen auslösen.

Beispiel: Bei Drahtbruch kann eine diagnosefähige Baugruppe Diagnosealarm auslösen (OB 82).

Die S7-Komponenten stellen Informationen zum aufgetretenen Systemfehler zur Verfügung. Die Starterereignisinformationen, d. h. die Lokaldaten des zugeordneten OB (enthalten u.a. den Datensatz 0), geben allgemeine Auskunft über Ort (z. B. logische Adresse der Baugruppe) und Art (z. B. Kanalfehler oder Pufferungsausfall) des Fehlers.

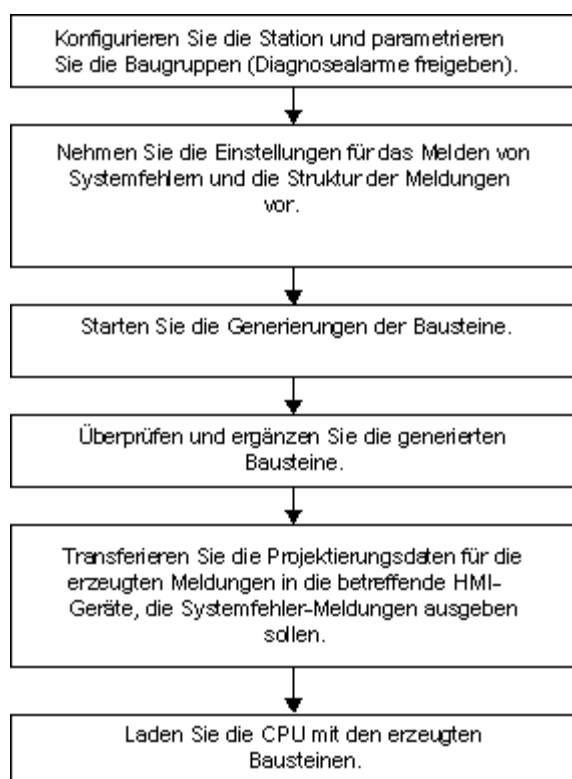
Darüber hinaus kann über zusätzliche Diagnoseinformationen (Lesen von Datensatz 1 mit SFC51 bzw. Lesen des Diagnosetelegramms von DP-Normslaves mit SFC 13) der Fehler genauer spezifiziert werden: z. B. Kanal 0 oder 1, Drahtbruch oder Messbereichsüberschreitung.

STEP 7 bietet mit der Funktion "Systemfehler melden" eine komfortable Möglichkeit, die von der Komponente zur Verfügung gestellten Diagnoseinformationen in Form von Meldungen anzuzeigen.

Die hierfür notwendigen Bausteine und Meldetexte werden von STEP 7 automatisch erzeugt. Vom Anwender müssen die erzeugten Bausteine lediglich in die CPU geladen und die Texte in angeschlossene HMI-Geräte transferiert werden.

Eine genaue Übersicht über die unterstützten Diagnose-Informationen bei verschiedenen Slaves finden Sie unter Abschnitt Unterstützte Komponenten und Funktionsumfang.

Prinzipielle Vorgehensweise



Die Meldungen werden über den Standardmeldeweg ALARM_S/SQ an "CPU-Meldungen" am PG oder an die angeschlossenen HMI-Geräte gesendet.

16.9.1 Unterstützte Komponenten und Funktionsumfang

Die Komponenten der S7-300-Stationen, S7-400-Stationen, DP-Slaves und WinAC werden von "Systemfehler melden" unterstützt, sofern sie Funktionen wie Diagnosealarm, Ziehen/Stecken-Alarm und kanalspezifische Diagnose unterstützen.

Folgende Komponenten werden von "Systemfehler Melden" **nicht** unterstützt:

- M7-, C7 und PROFIBUS-DP-Konfigurationen über DP-Masteranschaltung (CP 342-5 DP) in S7-300 Stationen

Im Falle eines Wiederanlaufes ist noch zusätzlich zu beachten, dass es zu fehlenden Alarmmeldungen kommen kann. Dieses Verhalten beruht darauf, dass der Meldequittierspeicher der CPU bei einem Wiederanlauf nicht gelöscht wird, aber "Systemfehler melden" die internen Daten zurücksetzt.

In den beiden unteren Tabellen finden Sie alle Diagnoseblöcke von den verschiedenen PROFIBUS-Slaves, die von "Systemfehler melden" unterstützt werden:

Diagnose-block	Kennungsbez. (fehlerhafter Steckplatz)	Kanalbez. (Kanalfehler) 1)	Modulstatus (Modulfehler, falsches/kein Modul)	Gerätebezeichnung
Headerkennung 2)	0x01	0x10	0x00 Typ 0x82	0x00 + 1 Byte Diagnoseinfo
ET 200 S	Meldung: "Diagnose liegt vor"	Klartextmeldung	Klartextmeldung	-
ET 200 M	wird nicht ausgewertet	wird nicht ausgewertet	wird nicht ausgewertet	-
ET 200 X	Meldung: "Diagnose liegt vor"	-	-	-
ET 200 X DESINA	Meldung: "Diagnose liegt vor"	Klartextmeldung	Klartextmeldung	-
ET 200 L	wird nicht ausgewertet	-	-	-
ET 200 B Digital	-	-	-	Meldung: "Baugruppe gestört"
ET 200 B Analog	-	-	-	-
ET 200 C Digital	-	-	-	-
ET 200 C Analog	Meldung: "Diagnose liegt vor"	-	-	Meldung: "Baugruppe gestört"
ET 200 U	Meldung: "Diagnose liegt vor"	-	-	Meldung: "Baugruppe gestört"
ET 200 iS	Meldung: "Diagnose liegt vor"	Klartextmeldung	Klartextmeldung	-
ET 200 eco	-	-	-	Klartextmeldung
DP AS-i Link	Meldung: "Diagnose liegt vor"	-	Klartextmeldung	-

Diagnoseblock	DS0/DS1 1)	sonst. Ausprägung
Headerkennung 2)	0x00 Typ 0x01	0x00 sonst. Typ
ET 200 S	-	-
ET 200 M	Klartextmeldung	wird nicht ausgewertet
ET 200 X	-	-
ET 200 X DESINA	Klartextmeldung	-
ET 200 L	Klartextmeldung	-
ET 200 B Digital	-	-
ET 200 B Analog	Klartextmeldung	-
ET 200 C Digital	-	-
ET 200 C Analog	Klartextmeldung	-
ET 200 iS	Klartextmeldung	-
ET 200 eco	-	-
DP AS-i Link	Meldung: "Modulfehler"	-

In den nachfolgenden Tabellen finden Sie alle Diagnoseblöcke von den verschiedenen PROFINET-Slaves, die von "Systemfehler melden" unterstützt werden:

Diagnose-block	Kennungsbez. (fehlerhafter Steckplatz)	Kanalbez. (Kanalfehler) 1)	Modulstatus (Modulfehler, falsches/kein Modul)	Gerätebezeichnung
Header- kennung 2)	0x01	0x10	0x00 Typ 0x82	0x00 + 1 Byte Diagnoseinfo
ET 200 S	Meldung: "Diagnose liegt vor"	Klartextmeldung	Klartextmeldung	-
SCALANCE Switches	-	-	Klartextmeldung	-

Diagnoseblock	DS0/DS1 1)	sonst. Ausprägung
Headerkennung 2)	0x00 Typ 0x01	0x00 sonst. Typ
ET 200 S	-	-
SCALANCE Switches	Klartextmeldung	wird nicht ausgewertet

- 1) DS0: Standard-Diagnose, z. B. Baugruppenstörung, externe Hilfsspannung oder Frontstecker fehlt, Umfang 4 Byte, in den Lokaldaten des OB 82 enthalten.
DS1: Kanalfehler, für jeden Kanaltyp unterschiedlich definiert, lesbar im Anwenderprogramm über SFC 51.
Die Texte kommen aus der S7-HW-Diagnose.
- 2) Headerkennung: Kennung im Diagnosetelegramm, die unterschiedliche Diagnoseteile identifiziert.

Das Diagnosetelegramm (auch Norm Slave Telegramm genannt) ist aus den oben genannten Diagnoseblöcken aufgebaut und kann über SFC 13 im Anwenderprogramm gelesen werden.

In STEP 7 wird das Diagnosetelegramm über den Aufruf des Baugruppenzustands im Online-Fenster "HW Konfig" (Hardware diagnostizieren) im Register "DP-Slave Diagnose" unter "Hexdarstellung" angezeigt.

Diagnose-Repeater: Die Meldungen des Diagnose-Repeaters werden als Klartext ausgegeben. Die Texte werden aus der GSD-Datei ausgelesen.

PROFINET

- Bei PROFINET IO wird die Kanaldiagnose als Klartextmeldung ausgegeben.
- ET 200S: Slaves mit gepackten Adressen werden unterstützt.
- Bei PROFINET iO-Geräten wird die herstellereigene Diagnose unterstützt.

16.9.2 Einstellungen für das Melden von Systemfehlern

Sie haben mehrere Möglichkeiten, den Dialog für die Einstellungen aufzurufen:

- Selektieren Sie in HW Konfig die CPU, für welche Sie das Melden von Systemfehlern projektieren möchten. Wählen Sie dann den Menübefehl **Extras > Systemfehler melden**.
- Wenn Sie bereits Bausteine für das Melden von Systemfehlern erzeugt haben, dann können Sie den Dialog durch Doppelklicken auf einen erzeugten Baustein (FB, DB) aufrufen.
- Wählen Sie die Option zum automatischen Aufruf bei Speichern und Übersetzen der Konfiguration im Eigenschaftsdialog der Station.

Die Option zum automatischen Aufruf bei Speichern und Übersetzen erreichen Sie wie folgt:

1. Markieren Sie im SIMATIC Manager die betreffende Station.
2. Wählen Sie den Menübefehl Bearbeiten > Objekteigenschaften.
3. Wählen Sie das Register "Einstellungen".

Hinweis

Sie können das Register "Einstellungen" des Eigenschaftsdialogs auch in HW Konfig über den Menübefehl **Station > Eigenschaften** aufrufen.

Im Dialog geben Sie unter anderem ein:

- Welcher FB und welcher zugeordnete Instanz-DB erzeugt werden soll.
- Ob Referenzdaten erzeugt werden sollen.
- Ob Warnungen während der Generierung von Systemfehler melden stets angezeigt werden sollen.
- Ob der Dialog erscheinen soll, wenn Systemfehler melden automatisch nach dem Speichern und Übersetzen der Konfiguration aufgerufen wird (siehe Einstellung oben).
- Erzeugen von Fehler-OBs: Ob im S7-Programm noch nicht vorhandene Fehler-OBs erzeugt werden sollen oder nicht und in welchen OBs 'Systemfehler melden' aufgerufen werden soll.
- Das CPU-Verhalten im Fehlerfall: Sie können einstellen, welche Fehlerklassen nach ihrem Auftreten die CPU in Stop gehen lassen sollen.
- Das Erscheinungsbild der Meldungen (Aufbau und Reihenfolge der möglichen Textanteile).
- Ob Meldungen quittierbar sein sollen.
- Welche Parameter in der Schnittstelle des Anwender-Bausteins enthalten sein sollen.

Detaillierte Informationen finden Sie in der Hilfe zum aufgerufenen Dialog.

16.9.3 Bausteine für die Meldung von Systemfehlern generieren

Nachdem Sie die Einstellungen für das Melden von Systemfehlern vorgenommen haben, können Sie die notwendigen Bausteine (FB und DB, je nach Einstellung auch noch nicht vorhandenen OBs) erzeugen. Klicken Sie dazu im Dialogfeld "Systemfehler melden" die Schaltfläche "Generieren" an.

Folgende Bausteine werden erzeugt:

- Diagnose-FB (voreingestellt: FB 49),
- Instanz-DB für den Diagnose-FB (voreingestellt: DB 49),
- Fehler-OBs (falls Sie diese Option im Register "OB Konfiguration" angewählt haben),
- Optionaler Anwender-Baustein, der von dem Diagnose-FB aufgerufen wird.

16.9.4 Erzeugter FB, DB

Der erzeugte FB wertet die Lokaldaten des Fehler-OBs aus, liest evtl. zusätzliche Diagnoseinformationen der Fehler auslösenden S7-Komponente und erzeugt automatisch die entsprechende Meldung.

Er hat folgende Eigenschaften:

- Erstellungssprache SFM (Systemfehler Melden) (gilt ebenfalls für den erzeugten Instanz-DB)
- Know-How-protected (gilt ebenfalls für den erzeugten Instanz-DB)
- Verzögert während der Laufzeit ankommende Alarmer
- Ruft durch Doppelklick den Dialog zum Einstellen der Funktion "Melden von Systemfehlern" auf (gilt ebenfalls für den erzeugten Instanz-DB)

Anwender-Baustein

Da der Diagnose-FB Know-how-geschützt ist, können Sie ihn nicht editieren. Der FB stellt aber eine Schnittstelle für das Anwenderprogramm zur Verfügung, so dass Sie Zugriff auf z. B. den Fehlerstatus oder die Meldungsnummer haben.

Der Baustein für die Auswertung im Anwenderprogramm (im Register "Anwender-Baustein" des Dialogs einstellbar) wird im erzeugten FB mit den ausgewählten Parametern aufgerufen. Folgende Parameter stehen zur Auswahl:

Name	Datentyp	Kommentar
EV_C	BOOL	//Meldung kommend (TRUE) oder gehend (FALSE)
EV_ID	DWORD	//generierte Meldungsnummer
IO_Flag	BYTE	//Eingabebaugruppe: B#16#54 Ausgabebaugruppe: B#16#55
logAdr	WORD	//logische Adresse
TextlistId	WORD	//ID der Textbibliothek (Standard- Textbibliothek = 1)
ErrorNo	WORD	//generierte Fehlernummer
Channel_Error	BOOL	//Kanalfehler (TRUE)
ChannelNo	WORD	//Kanalnummer
ErrClass	WORD	//Fehlerklasse
HErrClass	WORD	//Fehlerklasse von H-Systemen

Falls der Anwender-FB noch nicht existiert, wird er von SFM mit den ausgewählten Parametern angelegt.

Die erzeugten Fehlertexte für Standardfehler entsprechen folgender Zuordnung:

Fehlernummer (dezimal)		betroffener Fehler-OB	Fehlercode im OB	
von	bis		von	bis
1	86	OB 72	B#16#1	B#16#56
162	163	OB 70	B#16#A2	B#16#A3
193	194	OB 72	B#16#C1	B#16#C2
224		OB 73	B#16#E0	
289	307	OB 81	B#16#21	B#16#33
513	540	OB 82		
865	900	OB 83	B#16#61	B#16#84
1729	1763	OB 86	B#16#C1	B#16#C8

Die Fehlernummern größer 12288 entsprechen Kanalfehlern. Wenn Sie sich die Fehlernummer in der Hexadezimaldarstellung anschauen, können Sie den Kanaltyp errechnen und das Fehlerbit erkennen. Die genaue Beschreibung müssen Sie aber der jeweiligen Baugruppenhilfe oder der Kanalhilfe entnehmen.

Beispiel:

12288 = W#16#3000 -> Highbyte 0x30 - 0x10 = Kanaltyp 0x20 (CP-Schnittstelle);
Lowbyte 0x00, also Fehlerbit 0

32774 = W#16#8006 -> Highbyte 0x80 - 0x10 = Kanaltyp 0x70 (Digitaleingabe);
Lowbyte 0x06, also Fehlerbit 6

16.9.5 Anlegen von fremdsprachigen Meldetexten in 'Systemfehler melden'

Sie können die in "Systemfehler melden" projektierten Meldungen in den Sprachen, in denen Sie STEP 7 installiert haben, anzeigen lassen.

Gehen Sie dazu folgendermaßen vor:

1. Wählen Sie im SIMATIC Manager den Menübefehl **Extras > Sprache für Anzeigegeräte...** und fügen Sie im aufgeblendeten Dialogfeld die gewünschte Sprache zu ihrem Projekt hinzu.
2. Bestätigen Sie Ihre Einstellungen mit OK.
3. Wählen Sie in HW Konfig den Menübefehl **Extras > Systemfehler melden...** und klicken Sie im aufgeblendeten Dialogfeld auf die Schaltfläche "Generieren".

Ergebnis: Die Meldetexte werden in allen Sprachen, in denen Sie STEP 7 installiert haben, erzeugt, jedoch nur in der Sprache angezeigt, die Sie im Dialogfeld "Sprache hinzufügen, entfernen, Standardsprache setzen" durch Klicken auf die Schaltfläche "Als Standard" definiert haben.

Beispiel

Sie haben STEP 7 in deutsch, englisch und französisch installiert und diese Sprachen sind in Ihrem Projekt definiert. Generieren Sie nun die Meldetexte wie oben beschrieben. Um die Meldetexte in der jeweiligen Sprache anzuzeigen, definieren Sie die gewünschte Sprache im Dialogfeld "Sprache hinzufügen, entfernen, Standardsprache setzen" als Standardsprache.

17 Bedienen und Beobachten von Variablen

17.1 Projektieren von bedien- und beobachtbaren Variablen

Übersicht

STEP 7 bietet Ihnen eine komfortable Methode, um variable Größen Ihres Prozesses bzw. Ihres Automatisierungssystems mit WinCC bedien- und beobachtbar zu machen.

Der Vorteil gegenüber bisherigen Methoden liegt für Sie darin, dass Sie das Projektieren der Daten nicht mehr für jede Operator Station (OS) einzeln, sondern nur einmalig mit STEP 7 durchführen müssen. Die beim Projektieren mit STEP 7 erzeugten Daten können Sie mit Hilfe des Transferprogramms AS-OS-Engineering (Bestandteil des SW-Paketes "Process Control System PCS7") in die Datenhaltung von WinCC übertragen, wobei die Konsistenz und Verträglichkeit der Daten mit dem Anzeigesystem überprüft wird. WinCC verwendet die Daten in Bildbausteinen und Grafikobjekten.

Mit STEP 7 können Sie BuB-Attribute für folgende Variablen projektieren bzw. ändern:

- Ein-, Ausgangs- und Durchgangparameter von Funktionsbausteinen
- Merker und E/A-Signale
- Parameter von CFC-Bausteinen in CFC-Plänen

Prinzipielle Vorgehensweise

Die Vorgehensweise bei der Projektierung von bedien- und beobachtbaren Variablen ist abhängig von der gewählten Programmier- bzw. Projektiersprache und der Art der Variablen, die Sie bedienen und beobachten möchten. Die prinzipielle Vorgehensweise umfasst aber stets die folgenden Schritte:

1. Weisen Sie den Parametern eines Funktionsbausteins oder den Symbolen in einer Symboltabelle Systemattribute für das Bedienen und Beobachten zu.

Bei CFC entfällt dieser Schritt, da Sie die Bausteine schon vorbereitet aus einer Bibliothek entnehmen.

2. Versetzen Sie die Variablen, die Sie bedienen und beobachten wollen, in einem Editierdialog mit den benötigten BuB-Attributen (S7_m_c). Im Dialogfeld "Bedienen und Beobachten" (Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Bedienen und Beobachten**) können Sie WinCC-Attribute wie Grenzwerte, Ersatzwerte und Protokollierungseigenschaften, usw. verändern.
3. Transferieren Sie die mit STEP 7 erzeugten Projektierungsdaten mit Hilfe des AS-OS-Engineering in Ihr Anzeigesystem (WinCC).

Namenskonventionen

Damit die Projektierungsdaten für WinCC gespeichert und transferiert werden können, werden sie unter einem vom STEP 7 automatisch vergebenen, eindeutigen Namen abgelegt. Die Namen der bedien- und beobachtbaren Variablen, der CFC-Pläne und der S7-Programme bilden Bestandteile dieses Namens und unterliegen aus diesem Grund bestimmten Konventionen:

- Die Namen der S7-Programme in einem S7-Projekt müssen eindeutig sein (unterschiedliche Stationen dürfen keine S7-Programme mit gleichem Namen enthalten).
- Die Namen der Variablen, S7-Programme und CFC-Pläne dürfen keine Unterstriche, Leerzeichen oder die folgenden Sonderzeichen: ['] [.] [%] [-] [/] [*] [+] enthalten.

17.2 Projektieren von BuB-Attributen mit AWL, KOP und FUP

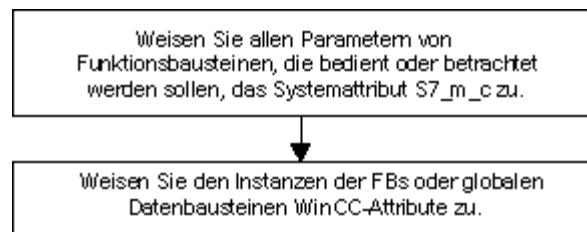
Einleitung

Mit der hier beschriebenen Vorgehensweise können Sie Parameter von Funktionsbausteinen bedien- und beobachtbar machen und zugeordneten Instanz-DBs oder Global-Datenbausteinen in Ihrem Anwenderprogramm die erforderlichen BuB-Attribute zuweisen.

Voraussetzung

Sie haben ein STEP 7-Projekt, ein S7-Programm und einen FB angelegt.

Prinzipielle Vorgehensweise



17.3 Projektieren von BuB-Attributen über die Symboltabelle

Einleitung

Unabhängig von der eingesetzten Programmiersprache projektieren Sie mit der hier beschriebenen Vorgehensweise folgende Variablen:

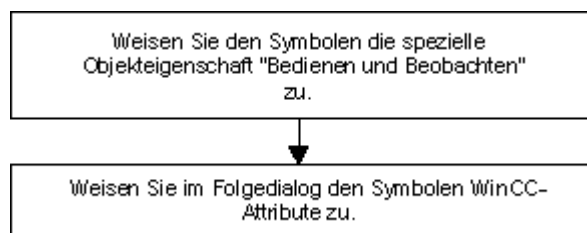
- Merker
- E/A-Signale

Voraussetzung

Bevor Sie beginnen, müssen folgende Voraussetzungen erfüllt sein:

- Sie haben im SIMATIC Manager ein Projekt angelegt.
- In diesem Projekt muss ein S7-Programm mit einer Symboltabelle vorhanden sein.
- Die Symboltabelle muss geöffnet sein.

Prinzipielle Vorgehensweise



17.4 Abändern von Attributen für Bedienen und Beobachten mit CFC

Einleitung

Mit CFC erstellen Sie Ihr Anwenderprogramm, indem Sie Bausteine, die bereits bedien- und beobachtbar sind, aus einer Bibliothek auswählen und in einem Plan platzieren und verschalten.

Voraussetzung

Sie haben in einem STEP 7-Projekt ein S7-Programm eingefügt, einen CFC-Plan angelegt und darin Bausteine platziert.

Prinzipielle Vorgehensweise

Bearbeiten Sie die Objekteigenschaften der Bausteine.

Hinweis

Verwenden Sie selbsterstellte Bausteine, denen Sie in der Erstsprache das Systemattribut S7_m_c zugewiesen haben, können Sie diese Bausteine bedien- und beobachtbar machen, indem Sie im Dialogfeld "Bedienen und Beobachten" (Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Bedienen und Beobachten**) das Optionsfeld "Bedienen und Beobachten" aktivieren.

17.5 Transferieren der Projektierungsdaten zum BuB-Zielsystem

Einleitung

Mit dem Transferprogramm AS-OS-Engineering transferieren Sie die erstellten Projektierungsdaten für das Bedienen und Beobachten in den Datenbestand von WinCC.

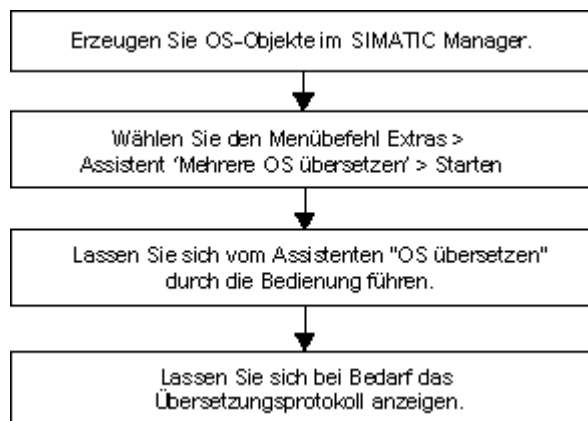
Voraussetzung

Bevor Sie mit dem Transfer beginnen, müssen folgende Voraussetzungen erfüllt sein:

- Sie haben das Programm AS-OS-Engineering installiert.
- Sie haben die Projektierungsdaten für das Bedienen und Beobachten erzeugt.

Prinzipielle Vorgehensweise

Um die Projektierungsdaten für das Bedienen und Beobachten in den Datenbestand von WinCC zu transferieren, gehen Sie folgendermaßen vor:



18 Aufbau der Online-Verbindung und CPU-Einstellung

18.1 Aufbau von Online-Verbindungen

Eine Online-Verbindung zwischen Erstellsystem und Zielsystem ist für das Laden von S7-Anwenderprogrammen/Bausteinen, das Zurückladen von Bausteinen aus dem S7-Zielsystem in das Erstellsystem und für folgende weitere Tätigkeiten erforderlich:

- Testen von Anwenderprogrammen
- Anzeigen und Ändern des Betriebszustands der CPU
- Anzeigen und Einstellen von Uhrzeit und Datum der CPU
- Anzeigen des Baugruppenzustands
- Baustein on-/offline vergleichen
- Hardware diagnostizieren

Um eine Online-Verbindung herstellen zu können, müssen Erstellsystem und Zielsystem über eine geeignete Schnittstelle miteinander verbunden sein (z. B. mehrpunktfähige Schnittstelle (MPI)). Sie können dann über das Online-Fenster des Projekts oder über das Fenster "Erreichbare Teilnehmer" auf das Zielsystem zugreifen.

18.1.1 Online-Verbindung aufbauen über das Fenster "Erreichbare Teilnehmer"

Diese Art bietet die Möglichkeit des schnellen Zugriffs, z. B. für Servicezwecke. Sie können auf alle im Netz erreichbaren programmierbaren Baugruppen zugreifen. Wählen Sie diese Methode, wenn keine Projektdaten zu den Zielsystemen auf ihrem PG vorhanden sind.

Das Fenster "Erreichbare Teilnehmer" lässt sich mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** aufrufen. Unter dem Objekt "Erreichbare Teilnehmer" werden alle im Netz erreichbaren Teilnehmer mit ihrer Adresse angezeigt.

Es können auch Teilnehmer angezeigt werden, die nicht mit STEP 7 programmierbar sind (z. B. Programmiergeräte oder Operator-Panels).

Folgende Zusatzinformationen können zusätzlich in Klammern angezeigt werden:

- (direkt): Dieser Teilnehmer ist direkt an das Erstellsystem (PG oder PC) angeschlossen.
- (passiv): Programmieren und Status/Steuern über PROFIBUS-DP ist bei diesem Teilnehmer nicht möglich
- (wartet): Mit diesem Teilnehmer kann nicht kommuniziert werden, weil seine Projektierung nicht zu den übrigen Einstellungen am Netz passt.

Direkt angeschlossenen Teilnehmer ermitteln

Die Zusatzinformation "direkt" wird bei PROFINET-Teilnehmern nicht unterstützt. Um den direkt angeschlossenen Teilnehmer dennoch identifizieren zu können, verwenden Sie den Menübefehl **Zielsystem > Diagnose/Einstellung > Teilnehmer-Blinktest**.

Im dann erscheinenden Dialog können Sie die Blinkdauer einstellen und den Blinktest starten. Der direkt angeschlossene Teilnehmer gibt sich durch eine blinkende FORCE-LED zu erkennen.

Der Blinktest ist nicht durchführbar, wenn die Funktion FORCEN aktiv ist.

18.1.2 Online-Verbindung aufbauen über das Online-Fenster des Projekts

Wählen Sie diese Methode, wenn Sie das Zielsystem in einem Projekt auf Ihrem PG/PC projiziert haben. Das Online-Fenster lässt sich im SIMATIC Manager mit dem Menübefehl **Ansicht > Online** aufrufen. Es veranschaulicht die Projektdaten auf dem Zielsystem (im Gegensatz zum Offline-Fenster, das die Projektdaten auf dem PG/PC veranschaulicht). Im Online-Fenster werden sowohl für das S7-Programm als auch für das M7-Programm die Daten angezeigt, die auf dem Zielsystem liegen.

Diese Sicht auf das Projekt benutzen Sie für Zugriffe auf das Zielsystem. So sind gewisse Funktionen im Menü "Zielsystem" des SIMATIC Managers zwar im Online-Fenster, nicht aber im Offline-Fenster aufrufbar.

Man unterscheidet:

- **Zugriff mit konfigurierter Hardware**
Damit können Sie nur auf die offline projizierten Baugruppen zugreifen. Auf welche der online erreichbaren Baugruppen zugegriffen wird, liegt durch die MPI-Adresse fest, die beim Konfigurieren der programmierbaren Baugruppe eingestellt wurde.
- **Zugriff ohne konfigurierte Hardware**
Dafür ist ein vorhandenes S7-Programm bzw. M7-Programm Voraussetzung, das unabhängig von der Hardware erstellt wurde (d. h. direkt unter dem Projekt liegt). Auf welche der online erreichbaren Baugruppen zugegriffen wird, wird hier durch Angabe der entsprechenden MPI-Adresse in den Objekteigenschaften des S7-/M7-Programms festgelegt.

Der Zugriff über das Online-Fenster kombiniert die Daten auf dem Zielsystem mit den zugehörigen Daten auf dem Erstellsystem. Wenn Sie zum Beispiel einen S7-Baustein unter einem Projekt online öffnen, setzt sich die Anzeige wie folgt zusammen:

- Codeteil des Bausteins aus der CPU des S7-Zielsystems und
- Kommentare und Symbolik aus der Datenhaltung des Erstellsystems (sofern offline vorhanden). Wenn Sie Bausteine ohne vorhandene Projektstruktur direkt in der angeschlossenen CPU öffnen, werden diese angezeigt, wie sie in der CPU sind, d. h. ohne Symbolik und Kommentare.

18.1.3 Online-Zugriff auf Zielsysteme im Multiprojekt

Projektübergreifende Zugriffe mit zugeordnetem PG/PC

Die Funktion "PG/PC zuordnen" für die Objekte "PG/PC" und "SIMATIC PC-Station" steht Ihnen auch im Multiprojekt zur Verfügung.

Die Zielbaugruppe für den Online-Zugriff können Sie aus einem beliebigen Projekt des Multiprojekts auswählen. Die Vorgehensweise ist identisch wie beim Arbeiten mit nur einem Projekt.

Voraussetzungen

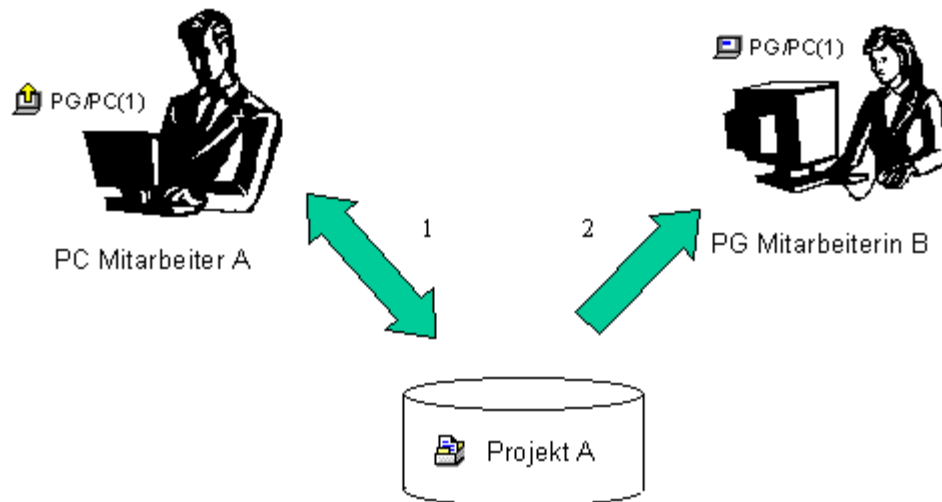
- Für PGs/PCs bzw. für PC-Stationen, mit denen online auf Zielsysteme zugegriffen werden soll, muss die Zuordnung in einem beliebigen Projekt innerhalb des Multiprojekts vorhanden sein.
Hinweis: Das zugeordnete PG/PC bzw. die zugeordnete PC-Station ist, wenn das betreffende Projekt geöffnet ist, gelb hervorgehoben.
Eine PG/PC-Zuordnung ist nur dann sichtbar, wenn die Zuordnung für das projekt-öffnende PG korrekt ist.
- Projektübergreifende Subnetze sind zusammengeführt.
- Alle Projekte des Multiprojekts sind übersetzt und die Projektierungsinformation ist in die beteiligten Stationen geladen, so dass z. B. Routing-Informationen für alle beteiligten Baugruppen für den Aufbau der Verbindung zwischen PG/PC und Zielbaugruppe Verfügung stehen.
- Die Zielbaugruppe ist über Netze erreichbar.

Mögliche Probleme bei verteilten Projekten

Wenn die Zuständigkeiten für Projekte wechseln und ein Projekt von einem PG/PC geöffnet wird, auf dem es nicht erstellt wurde, dann ist die PG/PC-Zuordnung nicht sichtbar.

Gleichwohl hat das projektierte Objekt PG/PC noch die Eigenschaft "zugeordnet" - aber mit dem "falschen" PG/PC.

In diesem Fall müssen Sie zunächst die bestehende Zuordnung aufheben und anschließend das Objekt PG/PC erneut zuordnen. Danach ist ein Online-Zugriff auf erreichbare Baugruppen innerhalb des Multiprojekts problemlos möglich.



1. Projekt A mit zugeordnetem PG/PC im Netz speichern
2. Dasselbe Projekt A mit einem anderen Rechner öffnen

Tipp für das Arbeiten in verteilten Projekten

Wenn mehrere Mitarbeiter mit PGs online auf Zielsysteme zugreifen, dann ist es sinnvoll, für jedes dieser PGs jeweils ein Objekt "PG/PC" bzw. "SIMATIC PC-Station" im Multiprojekt zu projektieren und anschließend für jedes PG eine Zuordnung einzurichten.

Je nachdem, von welchem PG aus das Projekt geöffnet ist, wird nur das dem öffnenden PG zugeordnete Objekt mit einem gelben Pfeil im SIMATIC Manager markiert.

18.1.4 Passwortschutz für Zugriff auf Zielsysteme

Mit Hilfe des Passwortschutzes können Sie

- das Anwenderprogramm in der CPU mit seinen Daten vor ungewollten Änderungen schützen (Schreibschutz)
- das Know-How, das in Ihrem Anwenderprogramm enthalten ist, bewahren (Leseschutz)
- Online-Funktionen, die den Prozess stören würden, unterbinden

Sie können eine Baugruppe bzw. MMC-Inhalte (z. B. für eine CPU 31xC) nur dann mit einem Passwort schützen, wenn die Baugruppe diese Funktionalität unterstützt.

Wenn Sie eine Baugruppe bzw. MMC-Inhalte mit einem Passwort schützen möchten, müssen Sie die Schutzstufe und das Passwort im Rahmen der Baugruppenparametrierung definieren und anschließend die geänderte Parametrierung auf die Baugruppe laden.

Wenn für die Ausführung einer Online-Funktion oder den Zugriff auf Inhalte einer MMC die Eingabe eines Passworts erforderlich ist, wird das Dialogfeld "Paßwort eingeben" angezeigt. Durch die Eingabe des korrekten Passworts erhalten Sie die Zugangsberechtigung zu Baugruppen, für die im Rahmen der Parametrierung eine besondere Schutzstufe festgelegt wurde. Sie können dann Online-Verbindungen zur geschützten Baugruppe aufbauen und die zur Schutzstufe gehörigen Online-Funktionen ausführen.

Mit dem Menübefehl **Zielsystem > Zugangsberechtigung > Einrichten** können Sie das Dialogfeld zur Eingabe des Passworts direkt aufrufen. Damit können Sie - z. B. zu Beginn einer Sitzung - das Passwort einmalig eingeben und werden dann bei späteren Online-Zugriffen nicht mehr abgefragt. Die Passwordeingabe wirkt entweder bis zum Beenden des SIMATIC Managers oder bis zum Aufheben der Passwordeingabe mit dem Menübefehl **Zielsystem > Zugangsberechtigung > Aufheben**.

CPU-Parameter	Bemerkungen
Testbetrieb/ Prozessbetrieb (nicht für S7-400 bzw. CPU 318-2)	<p>Einstellbar Im Register "Schutz".</p> <p>Im Prozessbetrieb werden Testfunktionen wie Programmstatus oder Status/Steuern Variable so eingeschränkt, dass die eingestellte zulässige Zykluszeiterhöhung nicht überschritten wird. So sind z. B. beim Programmstatus keine Aufrufbedingungen erlaubt und die Statusanzeige einer programmierten Schleife wird an der Rücksprungstelle abgebrochen.</p> <p>Das Testen mit Haltepunkten und schrittweise Programmausführung können im Prozessbetrieb nicht ausgeführt werden.</p> <p>Im Testbetrieb sind alle Testfunktionen über PG/PC ohne Einschränkungen nutzbar, die auch größere Verlängerungen der Zykluszeit bewirken können.</p>
Schutzstufe	<p>Einstellbar im Register "Schutz". Schreibende bzw. lesende und schreibende Zugriffe auf die CPU können von der Kenntnis eines Passwortes abhängig gemacht werden. Das Passwort wird in diesem Register parametrierung.</p>

18.1.5 Hinweis zur Aktualisierung des Fensterinhalts

Beachten Sie:

- Änderungen im Online-Fenster eines Projekts aufgrund von Benutzeraktionen (z. B. Laden oder Löschen von Bausteinen) werden nicht automatisch in einem eventuell geöffneten Fenster "Erreichbare Teilnehmer" nachgeführt.
- Entsprechende Änderungen im Fenster "Erreichbare Teilnehmer" werden nicht automatisch in einem eventuell geöffneten Online-Fenster eines Projekts nachgeführt.

Um eine aktuelle Anzeige in einem parallel geöffneten Fenster zu erhalten, muss dieses Fenster ebenfalls aktualisiert werden (über Menübefehl oder Funktionstaste F5).

18.2 Anzeigen und Ändern des Betriebszustands

Mit dieser Funktion können Sie beispielsweise die CPU nach Korrektur eines Fehlers wieder in den Betriebszustand "RUN" setzen.

Anzeigen des Betriebszustands

1. Öffnen Sie Ihr Projekt und markieren Sie ein S7-/M7-Programm oder öffnen Sie mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** das Fenster "Erreichbare Teilnehmer" und markieren Sie einen Teilnehmer ("MPI=...").
2. Wählen Sie den Menübefehl **Zielsystem > Diagnose/Einstellung > Betriebszustand**.

Dieses Dialogfeld zeigt den aktuellen und den letzten Betriebszustand sowie die aktuelle Stellung des Schüsselschalters an der Baugruppe an. Für Baugruppen, bei denen die aktuelle Schüsselschalterstellung nicht ausgelesen werden kann, wird der Text "Undefiniert" ausgegeben.

Ändern des Betriebszustands

Über die Schaltflächen können Sie den Betriebszustand der CPU-Baugruppe ändern. Es sind nur die Schaltflächen aktiv, die im aktuellen Betriebszustand angewählt werden können.

18.3 Anzeigen und Einstellen von Uhrzeit und Datum

18.3.1 CPU-Uhren mit Zeitzone-Einstellung und Sommer-/Winterzeit

Ab STEP 7 V5.1, Servicepack 2 können Sie bei neuen CPUs (ab Firmware-Version 3) zusätzlich zu Uhrzeit/Datum auch folgende Einstellungen vornehmen bzw. auswerten:

- Sommer-/Winterzeit
- Korrekturfaktor zur Darstellung von Zeitzonen

Darstellung von Zeitzonen

Anlagenweit gibt es eine einzige, unterbrechungsfrei durchlaufende Uhrzeit, die Baugruppenzeit.

Lokal im Automatisierungssystem kann eine zusätzliche, von der Baugruppenzeit unterschiedliche Lokalzeit berechnet und vom Anwenderprogramm verwendet werden. Die Lokalzeit wird nicht direkt eingegeben, sondern errechnet sich aus der Baugruppenzeit zuzüglich bzw. abzüglich eines Zeitunterschiedes gegenüber der Baugruppenzeit.

Sommer/Winterzeit

Beim Stellen der Uhrzeit und des Datums können Sie auch Sommer- oder Winterzeit einstellen. Wenn von Sommer- auf Winterzeit umgestellt wird, z. B. per Anwenderprogramm, dann wird das nur beim Zeitunterschied gegenüber der Baugruppenzeit berücksichtigt. Sie können die Umschaltung über einen Baustein vornehmen, der Ihnen über das Internet zur Verfügung gestellt wird.

Lesen und Stellen der Uhrzeit und des Uhrzeitstatus

Sommer-/Winterzeitkennung sowie Zeitunterschied gegenüber der Baugruppenzeit sind im Uhrzeitstatus enthalten.

Folgende Möglichkeiten haben Sie zum Lesen bzw. Stellen von Uhrzeit und Uhrzeitstatus:

Mit STEP 7 (online)

- Über Menübefehl **Zielsystem > Diagnose/Einstellung > Uhrzeit stellen** (Lesen und Stellen)
- Über Dialogfeld "Baugruppenzustand", Register "Zeitsystem" (nur Lesen)

Im Anwenderprogramm

- SFC 100 "SET_CLKS" (Lesen und Stellen)
- SFC 51 "RDSYSST" mit SZL 132, Index 8 (nur Lesen)

Zeitstempel im Diagnosepuffer, in Meldungen, und OB-Startinformationen

Die Zeitstempel werden mit der Baugruppenzeit erzeugt.

Uhrzeitalarme

Wenn durch die Umstellung von Winterzeit auf Sommerzeit Uhrzeitalarme wegen des "Zeitsprungs" nicht ausgelöst wurden, dann wird der OB 80 aufgerufen.

Bei der Umstellung von Sommerzeit auf Winterzeit bleibt bei Uhrzeitalarmen mit minütlicher und stündlicher Periodizität die Periodizität erhalten.

Uhrzeitsynchronisation

Eine CPU, die als Uhrzeit-Master parametrier ist (z. B. im CPU-Register "Diagnose/Uhr"), synchronisiert andere Uhren immer mit der Baugruppenzeit und ihrem aktuellen Uhrzeitstatus.

18.4 Aktualisieren der Firmware

18.4.1 Online-Aktualisierung der Firmware von Baugruppen und Modulen

Ab STEP 7 V5.1, Servicepack 3 können Sie Baugruppen bzw. Module einer Station auf eine einheitliche Weise online aktualisieren. Die Vorgehensweise ist im folgenden beschrieben.

Konzept

Für die Aktualisierung der Firmware einer Baugruppe wie z. B. eine CPU, ein CP oder eine IM erhalten Sie Dateien (*.UPD) mit der aktuellen Firmware.

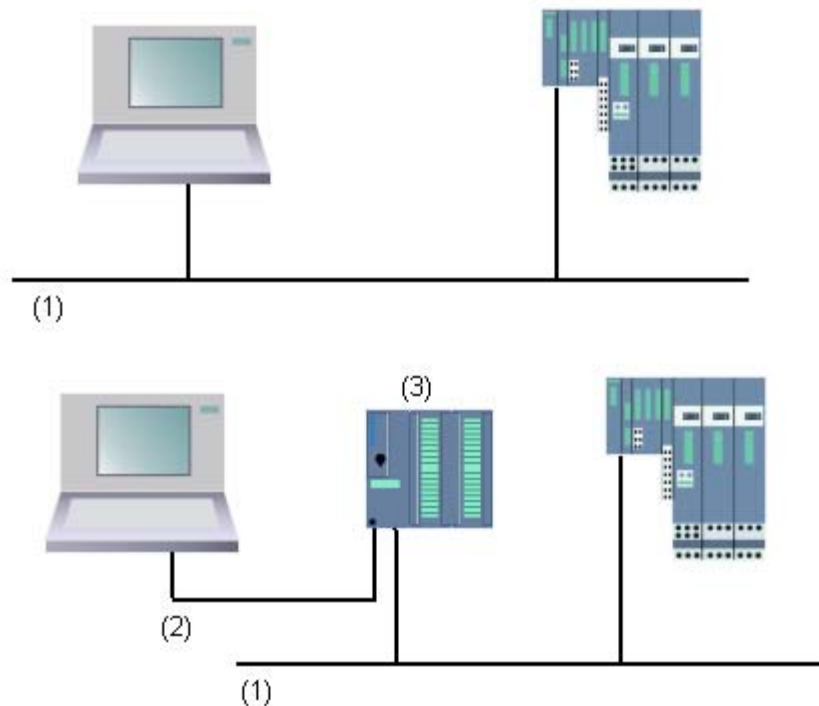
Eine dieser Dateien wählen Sie aus und laden sie in die Baugruppe (Menü Zielsystem).

Voraussetzungen

Die Baugruppe in der Station, deren Firmware aktualisiert werden soll, muss online erreichbar sein, z. B. wenn das PG am selben MPI, PROFIBUS oder Ethernet angeschlossen ist wie die Baugruppe, deren Firmware aktualisiert werden soll. Ein Aktualisieren der Firmware ist auch möglich, wenn das PG an der MPI-Schnittstelle der DP-Master-CPU angeschlossen ist und die Baugruppe, deren Firmware aktualisiert werden soll, am PROFIBUS der DP-Schnittstelle bzw. am Ethernet der PN-Schnittstelle. Die CPU muss dazu S7-Routing zwischen MPI- und DP-Schnittstelle bzw. zwischen MPI- und PN-Schnittstelle unterstützen.

Die Baugruppe selbst muss Firmware-Update unterstützen.

Die Dateien mit den aktuellen Firmware-Versionen müssen im Dateisystem Ihres PG/PC zur Verfügung stehen. In einem Ordner dürfen sich nur Dateien für **einen** Firmwarestand befinden.



(1) PROFIBUS- oder Ethernet-Subnetz

(2) MPI-Subnetz

(3) CPU mit MPI- und DP-Schnittstelle oder PN-Schnittstelle (mit S7-Routing)

Vorgehensweise in HW Konfig

1. Öffnen Sie die Station mit der zu aktualisierenden Baugruppe.
2. Markieren Sie die Baugruppe
Bei PROFIBUS-DP-Anschaltungen wie z. B. einer IM 151-1HF (151-1BA...) markieren Sie das Symbol für den DP-Slave, hier also die entsprechende ET 200S.
3. Wählen Sie den Menübefehl **Zielsystem > Firmware aktualisieren**.
Der Menübefehl ist nur dann aktivierbar, wenn die markierte Baugruppe bzw. der markierte DP-Slave die Funktion "Firmware aktualisieren" unterstützt.
4. Im aufgeblendeten Dialog "Firmware aktualisieren" wählen Sie über die Schaltfläche "Durchsuchen" den Pfad zu den Firmware-Update-Dateien (*.UPD).
5. Wenn Sie eine Datei ausgewählt haben, erscheint in den unteren Feldern des Dialogs "Firmware aktualisieren" die Information, für welche Baugruppen die Datei geeignet ist und ab welcher Firmware-Version.

6. Klicken Sie auf die Schaltfläche "Ausführen".
STEP 7 prüft, ob die ausgewählte Datei von der Baugruppe interpretiert werden kann und lädt bei positiver Prüfung die Datei in die Baugruppe. Falls dazu der Betriebszustand der CPU geändert werden muss, werden Sie über Dialoge zu diesen Aktionen aufgefordert.
Die Baugruppe führt danach selbständig den Firmware-Update durch.
Hinweis: Für den Firmware-Update z. B. zu einer CPU 317-2 PN/DP wird i. d. R. eine eigene Verbindung zur CPU aufgebaut. In diesem Fall ist der Vorgang unterbrechbar. Wenn keine Ressourcen für eine weitere Verbindung verfügbar sind, wird automatisch eine bestehende Verbindung genutzt. In diesem Fall ist der Vorgang nicht unterbrechbar; die Schaltfläche "Abbrechen" im Transferdialog ist gegraut und nicht bedienbar.
7. Prüfen Sie mit STEP 7 (Diagnosepuffer der CPU auslesen), ob die Baugruppe mit der neuen Firmware erfolgreich anläuft.

Vorgehensweise im SIMATIC Manager

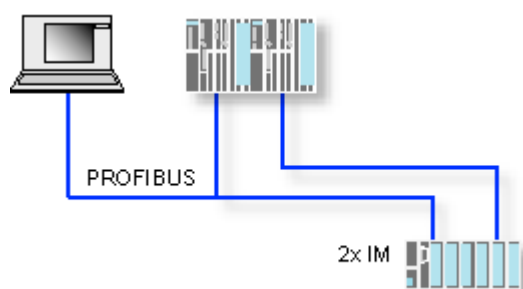
Die Funktion ist aktivierbar, wenn das Fenster "Erreichbare Teilnehmer" geöffnet ist. Die Vorgehensweise entspricht der in HW Konfig, der Menübefehl heißt hier ebenfalls **Zielsystem > Firmware aktualisieren**. Allerdings prüft STEP 7 erst zum Zeitpunkt der Ausführung, ob die Baugruppe die Funktion unterstützt.

Firmware-Update für Baugruppen im redundanten Betrieb

Ab STEP 7 V5.4 wird das Firmware-Update für Baugruppen im redundanten Betrieb unterstützt, z. B. für die IM 153-2BA00 mit aktivem Rückwandbus an einer H-Station. Für die redundanten IMs können Sie einen Firmware-Update in einem Arbeitsgang durchführen; die redundante IM wird automatisch mit dem neuen Firmware-Stand versorgt.

Voraussetzung: Das PG ist am selben PROFIBUS angeschlossen wie eine der IMs und Sie führen den Firmware-Update über "Erreichbare Teilnehmer" im SIMATIC Manager aus.

Prinzip



Konsequenzen beim Aktualisieren der Firmware im laufenden Betrieb

Sie haben die Möglichkeit, im Dialog zum Aktualisieren der Firmware durch eine Option zu bestimmen, dass die neue Firmware sofort nach dem Aktualisieren aktiviert werden soll.

Wenn diese Option gewählt ist, führt die Station einen Neuanlauf wie nach NETZ AUS/NETZ EIN durch. Eine CPU kann dabei z. B. im Betriebszustand STOP verbleiben oder der Ablauf des Anwenderprogramms beeinträchtigt sein. Für diese Einschränkungen im Betrieb müssen Sie Vorkehrungen treffen.

Bei einem Neuanlauf fallen z. B. sämtliche Module der Station aus - inklusive eventuell vorhandener F-Peripherie.

Die F-Peripherie meldet bei NETZ AUS der Anschaltung einen Kommunikationsfehler und schaltet sicherheitsgerichtet ab - sie wird passiviert. Diese Passivierung wird nicht durch den Neustart der Anschaltung aufgehoben. Sie müssen die Module einzeln depassivieren. Das hat jedoch zur Folge, dass sicherheitsgerichtete Anwendungen nicht laufen.

19 Laden

19.1 Laden aus dem PG in das Zielsystem

19.1.1 Voraussetzungen für das Laden

Voraussetzungen zum Laden in das Zielsystem

- Es besteht eine Verbindung zwischen Ihrem PG und der CPU des Zielsystems (z. B. über die MPI-Schnittstelle).
- Der Zugriff auf das Zielsystem ist möglich.
- Beim Laden von Bausteinen ins Zielsystem muss unter den Objekteigenschaften des Projekts für "Verwendung" der Eintrag "STEP 7" angewählt sein.
- Das zu ladende Programm ist fehlerfrei übersetzt.
- Die CPU muss in einem Betriebszustand sein, bei dem das Laden zulässig ist (STOP oder RUN-P).
Beachten Sie jedoch beim Laden im RUN-P-Zustand, dass das Programm bausteinweise übertragen wird. Überschreiben Sie dabei ein altes CPU-Programm, können Konflikte auftreten, z. B. wenn sich Bausteinparameter geändert haben. Die CPU geht dann während der Bearbeitung des Zyklus in den Betriebszustand STOP. Es wird daher empfohlen, die CPU vor dem Laden in den Betriebszustand "STOP" zu schalten.
- Falls Sie den Baustein offline geöffnet haben und laden wollen, muss der CPU-Baugruppe ein Online-Anwenderprogramm im SIMATIC Manager zugeordnet sein.
- Vor dem Laden des Anwenderprogramms, sollten Sie die CPU urlöschen, um sicherzustellen, dass sich keine "alten" Bausteine auf der CPU befinden.

Betriebszustand STOP

Setzen Sie den Betriebszustand von RUN auf STOP, bevor Sie

- das gesamte oder Teile des Anwenderprogramm auf die CPU laden,
- ein Urlöschen der CPU veranlassen,
- den Anwenderspeicher komprimieren.

Neustart (Warmstart) (Übergang in den Betriebszustand RUN)

Wenn Sie vom Zustand "STOP" aus einen Neustart (Warmstart) veranlassen, wird das Programm neu gestartet und zunächst im Betriebszustand "ANLAUF" das Anlauf-Programm (in Baustein OB 100) abgearbeitet. Ist der Anlauf erfolgreich, wechselt die CPU in den Betriebszustand RUN. Ein Neustart (Warmstart) ist erforderlich nach:

- Urlöschen der CPU,
- Laden des Anwenderprogramms im STOP-Zustand.

19.1.2 Unterschied zwischen Speichern und Laden von Bausteinen

Grundsätzlich ist zwischen Speichern und Laden von Bausteinen zu unterscheiden:

	Speichern	Laden
Menübefehle	Datei > Speichern Datei > Speichern unter...	Zielsystem > Laden
Funktion	Der aktuelle Stand des Bausteins aus dem Editor wird auf der Festplatte des PGs gespeichert.	Der aktuelle Stand des Bausteins aus dem Editor wird nur in die CPU geladen.
Syntaxprüfung	Eine Syntaxprüfung wird ausgeführt. Gegebenenfalls werden Ihnen Fehler in Dialogfeldern gemeldet. Die Fehlerursache und die Fehlerstellen werden Ihnen angezeigt. Diese Fehler müssen vor dem Speichern oder Laden des Bausteins behoben sein. Bei fehlerfreier Syntax wird der Baustein anschließend in Maschinencode übersetzt und gespeichert bzw. geladen.	Eine Syntaxprüfung wird ausgeführt. Gegebenenfalls werden Ihnen Fehler in Dialogfeldern gemeldet. Die Fehlerursache und die Fehlerstellen werden Ihnen angezeigt. Diese Fehler müssen vor dem Speichern oder Laden des Bausteins behoben sein. Bei fehlerfreier Syntax wird der Baustein anschließend in Maschinencode übersetzt und gespeichert bzw. geladen.

Die Tabelle gilt unabhängig davon, ob Sie den Baustein online oder offline geöffnet haben.

Hinweis zu Bausteinänderungen - erst speichern dann laden

Um neu erstellte Bausteine bzw. Änderungen im Anweisungsteil von Codebausteinen, in Deklarationstabellen oder bei Datenwerten in Datenbausteinen zu übernehmen, müssen Sie den entsprechenden Baustein abspeichern. Änderungen, die Sie im Editor durchführen und mit dem Menübefehl **Zielsystem > Laden** in die CPU transferieren, z. B. zum Testen kleiner Änderungen, sollten auf jeden Fall vor dem Beenden des Editors auch auf der Festplatte des PG gespeichert werden. Andernfalls haben Sie anschließend unterschiedliche Stände Ihres Anwenderprogramms in der CPU und im PG. Allgemein ist zu empfehlen, immer zuerst Änderungen zu speichern und anschließend zu laden.

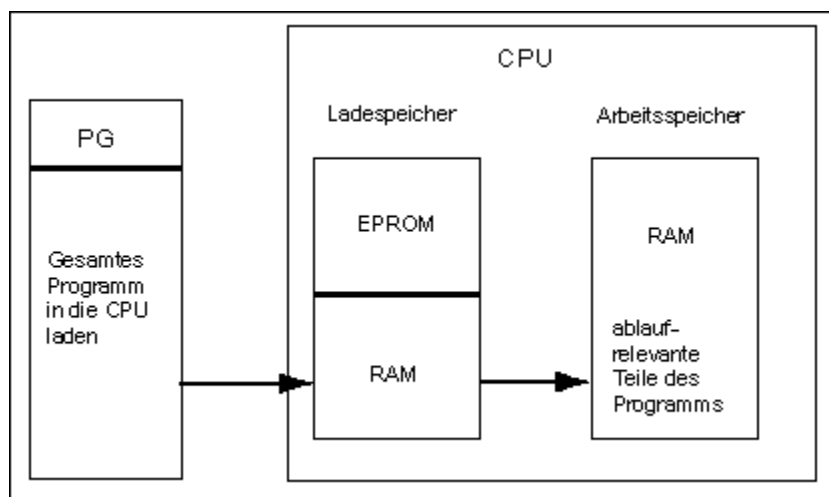
19.1.3 Lade- und Arbeitsspeicher in der CPU

Nach Abschluss der Konfiguration, Parametrierung und Programmerstellung sowie Aufbau der Online-Verbindung können Sie komplette Anwenderprogramme oder einzelne Bausteine auf ein Zielsystem übertragen. Für den Test einzelner Bausteine müssen Sie zumindest einen OB sowie die darin aufgerufenen FBs und FCs und die benutzten DBs laden. Um die nach dem Konfigurieren der Hardware, dem Projektieren von Netzen oder dem Erstellen einer Verbindungstabelle entstehenden Systemdaten auf das Zielsystem zu übertragen, laden Sie das Objekt "Systemdatenbausteine".

Anwenderprogramme laden Sie mit Hilfe des SIMATIC Managers in ein Zielsystem, z. B. in der Endphase des Programmtests oder um das fertige Anwenderprogramm zum Ablauf zu bringen.

Zusammenspiel von Lade- und Arbeitsspeicher der CPU

Das gesamte Anwenderprogramm wird in den Ladespeicher geladen, die ablaufrelevanten Teile des Programms auch in den Arbeitsspeicher.



Ladespeicher der CPU

- Der Ladespeicher dient zur Aufnahme des Anwenderprogramms ohne Symboltabelle und Kommentare (diese bleiben im Speicherbereich des PG).
- Bausteine, die als nicht ablaufrelevant gekennzeichnet sind, werden ausschließlich in den Ladespeicher aufgenommen.
- Beim Ladespeicher kann es sich zielsystemabhängig um RAM-, ROM- oder EPROM-Speicher handeln.
- Bei S7-300 kann der Ladespeicher außer einem integrierten RAM- auch einen integrierten EEPROM-Anteil haben (z. B. CPU312 IFM und CPU314 IFM).
- Bei S7-400 ist der Einsatz einer Memory Card (RAM oder EEPROM) zur Erweiterung des Ladespeichers unerlässlich.

Arbeitsspeicher der CPU

Der Arbeitsspeicher (integriertes RAM) dient zur Aufnahme der für den Programmablauf relevanten Teile des Anwenderprogramms.

Mögliche Vorgehensweisen beim Laden

Über die Ladefunktion können Sie das Anwenderprogramm oder ladbare Objekte (z. B. Bausteine) in das Zielsystem laden. Ist ein Baustein schon im RAM der CPU vorhanden, so werden Sie beim Laden gefragt, ob der Baustein überschrieben werden soll oder nicht.

- Die ladbaren Objekte können Sie im Projektfenster markieren und vom SIMATIC Manager aus laden (Menübefehl: **Zielsystem > Laden**).
- Beim Programmieren von Bausteinen und beim Konfigurieren von Hardware und Netzen können Sie das gerade bearbeitete Objekt direkt über das Menü in dem zu Ihrer Tätigkeit gehörigen Hauptfenster laden (Menübefehl: **Zielsystem > Laden**).
- Eine weitere Möglichkeit ist, ein Online-Fenster mit der Sicht auf das Zielsystem zu öffnen (z. B. über **Ansicht > Online** oder **Zielsystem > Erreichbare Teilnehmer anzeigen**) und das zu ladende Objekt in das Online-Fenster zu kopieren.

Umgekehrt können Sie sich über die Ladefunktion aktuelle Inhalte von Bausteinen aus dem RAM-Ladespeicher der CPU in Ihr PG laden.

19.1.4 Lademöglichkeiten abhängig vom Ladespeicher

Aus der Aufteilung des Ladespeichers in einen RAM- und einen EEPROM-Bereich ergeben sich Konsequenzen für die Möglichkeiten beim Laden Ihres Anwenderprogramms bzw. von Bausteinen Ihres Anwenderprogramms. Um die Daten in die CPU zu laden, bestehen folgende Möglichkeiten:

Ladespeicher	Lademöglichkeiten	Kommunikationsart zwischen Erstell- und Zielsystem
RAM	Laden und Löschen von einzelnen Bausteinen	Online-Verbindung PG - Zielsystem
	Laden und Löschen eines gesamten Anwenderprogramms	Online-Verbindung PG - Zielsystem
	Nachladen von einzelnen Bausteinen	Online-Verbindung PG - Zielsystem
EPROM integriert (nur bei S7-300) oder steckbar	Laden von gesamten Anwenderprogrammen	Online-Verbindung PG - Zielsystem
EPROM steckbar	Laden von gesamten Anwenderprogrammen	Externes Laden des EPROMS und Stecken der Memory Card oder über Online-Verbindung auf das im Zielsystem gesteckte EPROM.

Laden in das RAM über Online-Verbindung

Im Zielsystem haben Sie bei Spannungsausfall keine Datensicherheit, falls das RAM ungepuffert ist. Die Daten im RAM gehen in diesem Fall verloren.

Speichern auf EPROM-Memory Card

Bausteine bzw. das Anwenderprogramm werden auf eine EPROM-Memory Card gespeichert, die dann in einen Schacht der CPU gesteckt wird.

Memory Cards sind portable Datenträger. Sie werden auf dem Erstellsystem beschrieben und dann in den dafür vorgesehenen Schacht der CPU eingesetzt.

Die gespeicherten Daten bleiben bei Spannungsausfall und beim Umröscheln der CPU erhalten. Der Inhalt des EPROMs wird nach Umröscheln der CPU und nach Netzausfall mit ungepuffertem RAM-Speicher bei Spannungswiederkehr wieder in den RAM-Bereich des Speichers der CPU kopiert.

Speichern im integrierten EPROM

Für die CPU 312 gibt es noch die Möglichkeit, den Inhalt des RAM in das integrierte EPROM zu speichern. Die Daten im integrierten EPROM bleiben bei Spannungsausfall remanent. Der Inhalt des integrierten EPROMs wird nach Netzausfall mit ungepuffertem RAM-Speicher bei Spannungswiederkehr und nach Umröscheln der CPU wieder in den RAM-Bereich des Speichers der CPU kopiert.

19.1.5 Programm in S7-CPU laden

19.1.5.1 Laden mit Projektverwaltung

1. Markieren Sie im Projektfenster das zu ladende Anwenderprogramm bzw. die zu ladenden Bausteine.
2. Laden Sie die markierten Objekte in das Zielsystem, indem Sie den Menübefehl **Zielsystem > Laden** wählen.

Alternative Vorgehensweise (Drag&Drop)

1. Öffnen Sie je ein Offline-Fenster und ein Online-Fenster ihres Projekts.
2. Markieren Sie dann die zu ladenden Objekte im Offline-Fenster und ziehen Sie diese in das Online-Fenster.

19.1.5.2 Laden ohne Projektverwaltung

1. Öffnen Sie das Fenster "Erreichbare Teilnehmer" mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** oder durch Klicken auf das entsprechende Symbol in der Funktionsleiste.
2. Doppelklicken Sie im Fenster "Erreichbare Teilnehmer" auf den betreffenden Teilnehmer ("MPI=..."), um den Ordner "Bausteine" zu sehen.
3. Öffnen Sie die Bibliothek bzw. das Projekt, woraus Sie das Anwenderprogramm bzw. Bausteine in das Zielsystem laden wollen. Dazu wählen Sie den Menübefehl **Datei > Öffnen**.
4. Markieren Sie im geöffneten Fenster des Projekts bzw. der Bibliothek die zu ladenden Objekte.
5. Laden Sie die Objekte auf das Zielsystem, indem Sie diese per Drag&Drop oder mit Menübefehlen in das Fenster "Erreichbare Teilnehmer" auf den Ordner "Bausteine" kopieren.

19.1.5.3 Nachladen von Bausteinen in das Zielsystem

Sie können Bausteine, die im Ladespeicher (RAM) bzw. Arbeitsspeicher der CPU des S7-Zielsystems vorhanden sind, mit einer neuen Version überschreiben (nachladen). Dabei wird der bis dahin vorhandene Stand überschrieben.

Die Vorgehensweise beim Nachladen entspricht der des Ladens von S7-Bausteinen. Es erfolgt lediglich eine Abfrage, ob der vorhandene Baustein überschrieben werden soll.

Ein im EPROM gespeicherter Baustein kann nicht gelöscht werden, wird aber bei der Nachladefunktion für ungültig erklärt. Der ihn ersetzende Baustein wird in das RAM geladen. Dabei entstehen im Lade- oder Arbeitsspeicher Lücken. Falls diese Lücken dazu führen, dass keine neuen Bausteine mehr ladbar sind, sollten Sie die Speicher komprimieren.

Hinweis

Im Fall eines Netzausfalls mit ungepuffertem RAM-Speicher und Spannungswiederkehr oder beim Umrücken der CPU werden wieder die "alten" Bausteine aus dem EPROM gültig und geladen!

19.1.5.4 Speichern geladener Bausteine auf integriertem EPROM

Bei CPU-Baugruppen, die ein integriertes EPROM besitzen (z. B. CPU 312), können Sie Bausteine vom RAM in das integrierte EPROM speichern, um die Daten bei Spannungsausfall oder Umlöschen nicht zu verlieren.

1. Lassen Sie sich mit dem Menübefehl **Ansicht > Online** ein Fenster mit der Online-Ansicht zu einem geöffneten Projekt anzeigen oder lassen Sie sich das Fenster "Erreichbare Teilnehmer" anzeigen, indem Sie auf die Schaltfläche "Erreichbare Teilnehmer" in der Funktionsleiste klicken oder den Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** wählen.
2. Wählen Sie das S7- bzw. M7-Programm im "Online-Fenster" des Projekts oder den Teilnehmer im Fenster "Erreichbare Teilnehmer" aus.
3. Markieren Sie den zu speichernden Ordner "Bausteine" der CPU:
 - im Online-Fenster des Projekts, wenn Sie mit Projektverwaltung arbeiten.
 - im Fenster "Erreichbare Teilnehmer", wenn Sie ohne Projektverwaltung arbeiten.
4. Wählen Sie den Menübefehl **Zielsystem > RAM nach ROM kopieren**.

19.1.5.5 Laden über EPROM-Memory Cards

Voraussetzungen

Für Zugriffe auf EPROM-Memory Cards im Erstellsystem, die für ein S7-Zielsystem bestimmt sind, benötigen Sie die entsprechenden EPROM-Treiber. Für Zugriffe auf EPROM-Memory Cards, die für ein M7-Zielsystem bestimmt sind, muss das Flash File System installiert sein (nur bei PG 720/740/760, Field PG und Power PG möglich). EPROM-Treiber und Flash File System werden bei der Installation des Basispakets von STEP 7 als Option angeboten. Wenn Sie einen PC einsetzen, so ist zum Speichern auf EPROM-Memory Cards zusätzlich ein externer Prommer erforderlich.

Sie können die Treiber auch nachträglich installieren. Rufen Sie dazu das entsprechende Dialogfeld über die Startleiste (**Start > Simatic > STEP 7 > Memory-Card parametrieren**) oder über die Systemsteuerung (Doppelklick auf das Symbol "Memory Card parametrieren") auf.

Speichern auf der Memory Card

Um Bausteine oder Anwenderprogramme in einer Memory Card zu speichern, gehen Sie wie folgt vor:

1. Stecken Sie die Memory Card in den Schacht Ihres Erstellsystems.
2. Öffnen Sie das Fenster "S7-Memory Card" wie folgt:
 - Klicken Sie auf das Symbol für "S7-Memory Card" in der Funktionsleiste. Aktivieren Sie ggf. die Anzeige dieser Leiste mit dem Menübefehl **Ansicht > Funktionsleiste**.
 - Wählen Sie alternativ den Menübefehl **Datei > S7-Memory Card > Öffnen**.
3. Öffnen bzw. aktivieren Sie ein Fenster, in dem Sie sich die zu speichernden Bausteine anzeigen lassen. Möglich sind:
 - Projektfenster, Ansicht "ONLINE"
 - Projektfenster, Ansicht "offline"
 - Bibliotheksfenster
 - Fenster "Erreichbare Teilnehmer"
4. Markieren Sie den Ordner "Bausteine" bzw. die Bausteine und kopieren Sie diese in das Fenster "S7-Memory Card".
5. Ist ein Baustein bereits auf der Memory Card vorhanden, wird eine Fehlermeldung ausgegeben. Löschen Sie in diesem Fall den Inhalt der Memory Card und wiederholen Sie die Schritte ab 2.

19.2 Mehrere Objekte aus dem PG übersetzen und laden

19.2.1 Voraussetzungen und Hinweise zum Laden

Laden von Bausteinordnern

Bei Bausteinordnern können nur Codebausteine geladen werden. Andere Objekte des Bausteinordners, wie Systemdaten (SDBs) etc. können an dieser Stelle nicht geladen werden. SDBs werden über das Objekt "Hardware" geladen.

Hinweis

Für PCS 7-Projekte ist das Laden von Bausteinen aus dem Dialog "Objekte übersetzen und Laden" genau so wenig möglich wie aus dem SIMATIC Manager. Für PCS 7-Projekte gilt, dass Zielsysteme nur über CFC geladen werden dürfen, um eine korrekte Lade-Reihenfolge sicherzustellen und damit ein STOP der CPU zu vermeiden.

Ob es sich bei dem Projekt um ein PCS 7-Projekt handelt, sehen Sie in den Objekteigenschaften des Projekts.

Laden von F-Anteilen fehlersicherer Steuerungen

Aus Sicherheitsgründen kann das Laden von geänderten F-Anteilen fehlersicherer Steuerungen nur nach Eingabe eines Passworts erfolgen. In "Objekte übersetzen und laden" wird deshalb der Ladevorgang mit einem Fehlerhinweis abgebrochen. Laden Sie die entsprechenden Programmteile mit dem Optionspaket ins Zielsystem.

Laden der Hardware

Das Laden der Hardware (d. h. Laden der Offline-SDBs) über "Objekte übersetzen und laden" läuft nur dann unterbrechungsfrei für alle markierten Objekte, wenn keine Fehlermeldungen oder Abfragen ausgelöst werden. Der folgende Abschnitt gibt Informationen zur Vermeidung von Fehlermeldungen und Abfragen.

Voraussetzungen für das Laden der Hardware

- CPUs müssen im Betriebszustand STOP sein.
- Online-Verbindungen zu den CPUs müssen möglich sein; passwortgeschützte CPUs erfordern legitimierte Verbindung bzw. Eingabe des Passwortes über die Schaltfläche "Bearbeiten" bei markierter CPU bzw. bei markiertem Bausteinordner vor dem Starten von "Objekte übersetzen und laden".

- Die Schnittstelle des Zielsystems, über die geladen wird, darf beim Laden nicht wesentlich umparametriert werden:
 - Die Adresse der Schnittstelle darf nicht geändert werden.
 - Eine Änderung der Netzeinstellungen kann dazu führen, dass nicht alle Baugruppen erreicht werden können.
- Bei H-CPU's können Sie vor dem Starten von "Objekte übersetzen und laden" die zu ladende CPU (H-CPU 0 oder H-CPU 1) wählen (Objekt "CPU" markieren und auf Schaltfläche "Bearbeiten" klicken).
- Folgende CPU-Parameter dürfen sich nicht geändert haben:
 - Die maximale Größe für Lokaldaten und Kommunikationsressourcen der CPU (Register "Speicher")
 - Passwortschutz für F-CPU (Register "Schutz")
- Für jede konfigurierte Baugruppe müssen folgende Bedingungen erfüllt sein:
 - Bestellnummer der konfigurierten Baugruppe muss identisch sein mit der Bestellnummer der tatsächlich gesteckten Baugruppe.
 - Firmware-Version der konfigurierten Baugruppe darf nicht größer sein als die Firmware-Version der tatsächlich gesteckten Baugruppe.
 - Stationsname, Name der Baugruppe und Anlagenkennzeichen dürfen sich nicht geändert haben seit dem letzten Laden. Neuvergabe eines Anlagenkennzeichens ist erlaubt.

Hinweise zum Ablauf des Ladens

- Es werden alle Offline-SDBs geladen (d. h. neben der Hardware Konfiguration auch Verbindungs-SDBs und SDBs, die durch die Globaldaten-Projektierung entstanden sind).
- Das Laden wird nur ausgeführt, wenn zuvor beim Übersetzen kein Fehler aufgetreten ist.
- Hinweise während des Ladens werden unterdrückt; bei einem CPU-Speicherengpass wird z. B. ohne Rückfrage komprimiert.
- Nach dem Laden bleiben die geladenen Baugruppen im STOP-Zustand (ausgenommen Baugruppen, die ohne Rückfrage automatisch gestoppt und wieder gestartet werden).

Tipp

Wenn nach dem Laden die Meldung erscheint, dass das Laden des Objekts mit Warnungen ausgeführt worden ist, dann sollten Sie unbedingt das Protokoll einsehen. Möglicherweise ist das Objekt nicht oder nicht vollständig geladen worden.

19.2.2 Objekte übersetzen und laden

Mit dem Dialog "Objekte übersetzen und laden" bereiten Sie die anwählbaren Objekte Ihres Projekts bzw. Multiprojekts zur Übertragung auf das Zielsystem vor und laden diese nach Wunsch auf das Zielsystem. Der Dialog ist auf Objekte in einer Station, eines Projekts oder eines Multiprojekts anwendbar.

Abhängig vom selektierten Objekt können bestimmte Informationen nicht angezeigt werden oder es sind für diese Objekte nicht sämtliche unten beschriebene Funktionen verfügbar. Insbesondere können Einschränkungen auftreten bei Objekten, die von Optionspaketen erzeugt werden.

Bei Bausteinen eines Bausteinordners bedeutet "übersetzen", dass die Konsistenz der Bausteine geprüft wird. Der Einfachheit halber wird die Konsistenzprüfung von Bausteinen nachfolgend als Übersetzungsvorgang beschrieben.

Gehen Sie wie folgt vor:

1. Wählen Sie im SIMATIC Manager das Objekt aus, das Sie übersetzen oder übersetzen und laden möchten. Folgende Objekte sind im SIMATIC Manager wählbar:
 - Multiprojekt
 - Projekt
 - Station
 - S7-Programm ohne Stationszuordnung
2. Wählen Sie im SIMATIC Manager den Menübefehl **Zielsystem > Objekte übersetzen und laden**.
3. Wählen Sie "Nur übersetzen", wenn Sie die Bausteine prüfen, jedoch nicht ins Zielsystem laden wollen. Verwenden Sie diese Option, wenn Sie keines der Objekte ins Zielsystem laden möchten.
4. Falls Sie verhindern möchten, dass wegen Übersetzungsfehlern die Stationen unvollständig geladen werden, dann aktivieren Sie das Kontrollkästchen "Kein Laden bei Übersetzungsfehler". In diesem Fall wird nichts geladen. Wenn das Kontrollkästchen nicht aktiviert ist, dann werden alle Objekte geladen, die ohne Fehler übersetzt wurden. Objekte, die einen Fehler beim Übersetzen verursacht haben, werden nicht geladen.
5. Falls Sie Verbindungen übersetzen und laden wollen, aktivieren Sie die entsprechenden Kontrollkästchen am Objekt "Verbindungen".
6. Insbesondere das Multiprojekt eignet sich als Einstiegsobjekt, da von diesem Objekt aus automatisch alle Verbindungspartner von projektübergreifenden Verbindungen mit geladen werden können.
7. Markieren Sie in den Spalten "Übersetzen" und "Laden" die Objekte, die Sie übersetzen bzw. laden wollen. Ein Häkchen symbolisiert die entsprechende Auswahl. Falls Sie in Schritt 3 "Nur übersetzen" angewählt haben, wurde die Spalte "Laden" ausgeblendet.
8. Starten Sie den Übersetzungsvorgang mit der Schaltfläche "Starten".
9. Folgen Sie den Anweisungen am Bildschirm.

Nach dem Übersetzungs- bzw. Ladevorgang wird automatisch das Gesamtprotokoll angezeigt. Sie können aber jederzeit das Gesamt- bzw. Einzelprotokoll öffnen:

- Klicken Sie auf die Schaltfläche "Gesamt", wenn Sie ein Protokoll des gesamten Vorgangs betrachten wollen.
- Klicken Sie auf die Schaltfläche "Einzelobjekt", wenn Sie ein Protokoll des in der Auswahlliste markierten Objekts betrachten wollen.

Besonderheiten beim Übersetzen und Laden von Verbindungen

Wenn Sie für bei einer Baugruppe das Objekt "Verbindungen" zum **Übersetzen** markieren, dann markiert STEP 7 automatisch die entsprechenden Objekte "Verbindungen" der Verbindungspartner. Auf diese Weise erzeugt STEP 7 immer konsistente Projektierungsdaten (Systemdatenbausteine). Die automatisch markierten Objekte können nicht direkt manuell abgewählt werden. Die Markierung wird aber automatisch entfernt, wenn das ursprünglich markierte Objekt "Verbindungen" wieder abgewählt wird.

Wenn Sie für eine Baugruppe das Objekt "Verbindungen" zum **Laden** wählen, dann markiert STEP 7 automatisch auch das Kontrollkästchen "Übersetzen". Zusätzlich markiert STEP 7 die Kontrollkästchen "Übersetzen" und "Laden" aller Verbindungspartner. Wenn ausschließlich Objekte vom Typ "Verbindungen" gewählt wurden, können Sie die Verbindungen auch im Betriebszustand RUN-P der CPU laden.

Einzelne Verbindungen können Sie mit Hilfe von NetPro laden.

Hardware übersetzen und laden: Auswirkungen auf Verbindungen

Wenn Sie das Objekt "Hardware" zum Übersetzen bzw. Laden markieren, werden automatisch die Objekte "Verbindungen" unterhalb der markierten Hardware zum Übersetzen bzw. Laden markiert. In diesem Fall werden aber die Verbindungsobjekte der Verbindungspartner **nicht** automatisch mitmarkiert!

19.3 Laden aus dem Zielsystem in das PG

Diese Funktion unterstützt Sie bei folgenden Tätigkeiten:

- Sichern von Informationen aus dem Zielsystem (z. B. für Service-Zwecke)
- Schnelles Konfigurieren und Bearbeiten einer Station, wenn die Hardware-Komponenten zum Beginn der Konfigurierungstätigkeit vorhanden sind.

Sichern von Informationen aus dem Zielsystem

Diese Maßnahme kann erforderlich sein, wenn z. B. die Offline-Projektdateien des in der CPU laufenden Standes nicht oder nur teilweise vorhanden sind. Sie können sich dann zumindest den online verfügbaren Teil der Projektdateien auf Ihr PG holen.

Schnelles Konfigurieren

Sie können sich die Eingabe des Stationsaufbaus erleichtern, indem Sie nach Aufbau der Hardware und Neustart (Warmstart) der Station die Konfigurationsdaten aus dem Zielsystem auf Ihr PG laden. Sie erhalten damit den Stationsaufbau mit den Tyangaben der einzelnen Baugruppen. Danach müssen nur noch die einzelnen Baugruppen genauer spezifiziert (Bestell-Nr.) und parametrisiert werden.

Folgende Information wird in das PG geladen:

- S7-300: Konfiguration für den Zentralbaugruppenträger und eventuell vorhandene Erweiterungsbaugruppenträger
- S7-400: Konfiguration des Zentralbaugruppenträgers mit einer CPU und Signalbaugruppen ohne Erweiterungsbaugruppenträger
- Konfigurationsdaten zur dezentralen Peripherie können nicht in das PG geladen werden.

Dieser Informationsumfang wird geladen, wenn noch keine Projektierinformation auf dem Zielsystem vorliegt z. B. bei urgelöschten Systemen. Ansonsten liefert das "Laden in das PG" weit bessere Ergebnisse.

Bei S7-300-Systemen ohne dezentrale Peripherie brauchen Sie somit nur noch die Baugruppen genau zu spezifizieren (Bestell-Nr.) und zu parametrisieren.

Hinweis

Beim Laden in das PG (ohne dass eine Offline-Konfiguration vorhanden ist) kann STEP 7 nicht alle Bestellnummern der Komponenten vollständig ermitteln.

Die "unvollständig" erscheinenden Bestellnummern können Sie beim Konfigurieren der Hardware mit dem Menübefehl **Extras > Baugruppe spezifizieren** eingeben. Auf diese Weise können Baugruppen, die STEP 7 nicht bekannt sind (d. h. nicht im Fenster "Hardware Katalog" erscheinen), parametrisiert werden; allerdings ohne dass Parameterregeln überwacht werden!

Einschränkungen beim Laden aus dem Zielsystem

Für die aus dem Zielsystem in das PG geladene Daten gelten folgende Einschränkungen:

- Bausteine enthalten keine symbolischen Namen für Parameter, Variablen und Marken.
- Bausteine enthalten keine Kommentare.
- Es wird das gesamte Programm mit allen Systemdaten ("AG-Abzug") auf das PG geladen. Dabei kann von den Systemdaten nur der Teil von "HW-Konfigurieren" wie gewohnt weiterbearbeitet werden.
- Die Daten zu den Themen "Globaldaten-Kommunikation (GD)", "Projektierung von symbolbezogenen Meldungen" können nicht weiterbearbeitet werden.
- Ein Forceauftrag wird nicht in das PG mitgeladen. Er muss über die Anzeige des Forceauftrags separat als VAT gespeichert werden.
- Kommentare in den Dialogen der Baugruppen werden nicht geladen.
- Die Namen der Baugruppen werden nur dann angezeigt, wenn dies beim Konfigurieren angewählt wurde (HW-Konfig: Extras>Einstellungen, Objektamen im Zielsystem speichern).

19.3.1 Station laden in PG

Mit dem Menübefehl **Zielsystem > Station laden in PG** können Sie die aktuelle Konfiguration und alle Bausteine aus dem zu wählenden Automatisierungssystem in das PG laden.

Dazu legt STEP 7 eine neue Station im aktuellen Projekt an, unter der die Konfiguration abgespeichert wird. Sie können den voreingestellten Namen der eingefügten Station (z. B. "SIMATIC 300-Station(1)") ändern. Die eingefügte Station wird sowohl in der Ansicht "online" als auch in der Ansicht "offline" angezeigt.

Der Menübefehl ist wählbar, wenn ein Projekt geöffnet ist. Die Markierung eines Objekts im Projektfenster oder die Ansicht (online oder offline) ist für den Menübefehl ohne Bedeutung.

Sie können diese Funktion benutzen, um sich das Konfigurieren zu erleichtern.

- Bei Zielsystemen S7-300 wird die Konfiguration für den vorhandenen Aufbau einschließlich Erweiterungsbaugruppenträger ohne dezentrale Peripherie (DP) geladen.
- Bei S7-400-Zielsystemen wird die Konfiguration des Baugruppenträgers ohne Erweiterungsbaugruppenträger und ohne dezentrale Peripherie geladen.

Bei S7-300-Systemen ohne dezentrale Peripherie brauchen Sie somit nur noch die Baugruppen genau zu spezifizieren (MLFB-Nr.) und zu parametrieren.

Einschränkungen bei Station laden in PG

Für die in das PG geladenen Daten gelten folgende Einschränkungen:

- Bausteine enthalten keine symbolischen Namen für Formalparameter, temporäre Variablen und Marken.
- Bausteine enthalten keine Kommentare.
- Es wird das gesamte Programm mit allen Systemdaten ("AG-Abzug") auf das PG geladen, davon können aber nicht alle Daten sinnvoll weiterbearbeitet werden.
- Die Daten zu den Themen "Globaldaten-Kommunikation (GD)", "Projektierung von symbolbezogenen Meldungen", "Netzprojektierung" können nicht weiterbearbeitet werden.
- Forceaufträge können nicht auf das PG und von dort wieder auf das Zielsystem geladen werden.

19.3.2 Zurückladen von Bausteinen aus S7-CPU

Sie können mit dem SIMATIC Manager S7-Bausteine aus der CPU auf die Festplatte des Erstellsystems zurückladen. Die Möglichkeit des Ladens von Bausteinen in das PG nutzen:

- eine Sicherung des aktuellen Anwenderprogramms, das in der CPU geladen ist. Diese Sicherung kann dann z. B. im Servicefall nach evtl. Umräumen der CPU vom Wartungspersonal übertragen werden.
- Sie können das Anwenderprogramm aus der CPU in das Erstellsystem laden und dort bearbeiten, z. B. im Servicefall zur Fehlersuche. Sie haben in diesem Fall keine Symbolik bzw. Kommentare zur Programmdokumentation zur Verfügung. Deswegen empfiehlt sich diese Vorgehensweise wirklich nur für den Servicefall.

19.3.3 Bearbeiten von geladenen Bausteinen im PG/PC

Die Möglichkeit zum Bearbeiten von Bausteinen im PG bietet Ihnen folgenden Nutzen:

- Sie können in der Testphase einen Baustein direkt in der CPU korrigieren und das Ergebnis dokumentieren.
- Sie können sich über die Ladefunktion aktuelle Inhalte von Bausteinen aus dem RAM-Ladespeicher der CPU in Ihr PG laden.

Hinweis

Zeitstempelkonflikt bei Arbeiten online und offline

Die anschließend beschriebenen Vorgehensweisen führen zu Zeitstempelkonflikten und sollten deshalb vermieden werden.

Zeitstempelkonflikte entstehen beim Online-Öffnen eines Bausteins, wenn

- online durchgeführte Änderungen nicht in das S7-Anwenderprogramm offline gespeichert wurden
- offline durchgeführte Änderungen nicht in die CPU geladen wurden.

Zeitstempelkonflikte entstehen beim Offline-Öffnen eines Bausteins, wenn

- ein Online-Baustein mit Zeitstempelkonflikt in das S7-Anwenderprogramm offline kopiert und anschließend der Baustein offline geöffnet wird.
-

Fallunterscheidung

Beim Laden von Bausteinen aus der CPU in das Programmiergerät müssen Sie zwei Fälle unterscheiden:

1. Das Anwenderprogramm, zu dem die Bausteine gehören, befindet sich im Programmiergerät.
2. Das Anwenderprogramm, zu dem die Bausteine gehören, befindet sich nicht im Programmiergerät.

Dieses hat zur Folge, dass Programmteile, die nicht in die CPU geladen werden können, nicht zur Verfügung stehen. Solche Programmteile sind:

- die Symboltabelle mit den symbolischen Namen für die Operanden sowie Kommentare
- Netzwerk-Kommentare eines FUP- oder KOP-Programms
- Zeilenkommentare eines AWL-Programms
- anwenderdefinierte Datentypen

19.3.3.1 Bearbeiten von geladenen Bausteinen, falls Anwenderprogramm im PG/PC vorliegt

Um Bausteine aus der CPU zu bearbeiten, gehen Sie so vor:

1. Öffnen Sie im SIMATIC Manager das Online-Fenster des Projekts.
2. Markieren Sie im Online-Fenster einen Ordner "Bausteine". Die Liste der geladenen Bausteine wird angezeigt.
3. Wählen Sie nun Bausteine aus, öffnen und bearbeiten Sie diese.
4. Wählen Sie den Menübefehl **Datei > Speichern**, um die Änderung offline auf dem PG zu speichern.
5. Wählen Sie den Menübefehl **Zielsystem > Laden**, um die geänderten Bausteine auf das Zielsystem zu laden.

19.3.3.2 Bearbeiten von geladenen Bausteinen, falls Anwenderprogramm nicht im PG/PC vorliegt

Um Bausteine aus der CPU zu bearbeiten, gehen Sie so vor:

1. Betätigen Sie die Schaltfläche "Erreichbare Teilnehmer" im SIMATIC Manager oder wählen Sie den Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen**.
2. Wählen Sie aus der aufgeblendeten Liste den Teilnehmer (Objekt "MPI=...") aus und öffnen den Ordner "Bausteine", um sich die Bausteine anzeigen zu lassen.
3. Nun können Sie Bausteine öffnen und nach Bedarf bearbeiten, überwachen oder kopieren.
4. Wählen Sie den Menübefehl **Datei > Speichern unter...** und tragen Sie im zugehörigen Dialogfeld den Pfad ein, der den gewünschten Speicherort auf dem PG bezeichnet.
5. Wählen Sie den Menübefehl **Zielsystem > Laden**, um die geänderten Bausteine auf das Zielsystem zu laden.

19.4 Löschen im Zielsystem

19.4.1 Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU

Vor dem Laden Ihres Anwenderprogramms in das S7-Zielsystem, sollten Sie die CPU urlöschen, um sicherzustellen, dass sich keine "alten" Bausteine mehr auf der CPU befinden.

Voraussetzung für das Urlöschen

Zum Urlöschen muss sich die CPU im Betriebszustand STOP befinden (Betriebsartenschalter auf STOP oder Betriebsartenschalter auf RUN-P und Ändern des Betriebszustands nach STOP über den Menübefehl **Zielsystem > Diagnose/Einstellung > Betriebszustand**).

Urlöschen von S7-CPUs

Beim Urlöschen von S7-CPUs wird folgendes durchgeführt:

- Die CPU wird zurückgesetzt.
- Alle Anwenderdaten werden gelöscht (Bausteine und SDBs mit Ausnahme der MPI-Parametrierung).
- Die CPU bricht alle bestehenden Verbindungen ab.
- Wenn Daten auf einem EPROM vorhanden sind (Memory Card oder integriertes EPROM), kopiert die CPU nach dem Urlöschen den EPROM-Inhalt wieder in den RAM-Bereich des Speichers.

Der Inhalt des Diagnosepuffers und die Parameter der MPI-Schnittstelle bleiben erhalten.

Urlöschen von M7-CPUs/-FMs

Beim Urlöschen von M7-CPUs/-FMs wird folgendes durchgeführt:

- Der ursprüngliche Zustand wird hergestellt.
- Die SDBs mit Ausnahme der MPI-Parametrierung werden gelöscht.
- Die CPU/FM bricht alle bestehenden Verbindungen ab. Anwenderprogramme bleiben erhalten und laufen nach dem Umschalten von STOP auf RUN weiter.

Mit der Funktion "Urlöschen" können Sie den ursprünglichen Zustand der M7-CPU/-FM nach schwerwiegenden Fehlern wiederherstellen, indem Sie die aktuellen Systemdatenbausteine (SDB) aus dem Arbeitsspeicher löschen und die im Festwertspeicher befindlichen SDBs neu laden. In manchen Fällen ist zusätzlich ein Neustart (Warmstart) des Betriebssystems erforderlich. Dafür müssen Sie den M7 mit dem Betriebsartenschalter urlöschen (in Position MRES). Ein Reset über den Betriebsartenschalter von SIMATIC M7-CPUs/FMs ist nur möglich, wenn dort das Betriebssystem RMOS32 verwendet wird.

19.4.2 Löschen von S7-Bausteinen auf dem Zielsystem

Das Löschen einzelner Bausteine in der CPU kann während der Testphase des CPU-Programms notwendig werden. Bausteine sind im Anwenderspeicher der CPU entweder im EPROM oder RAM gespeichert (abhängig von CPU und Ladevorgang).

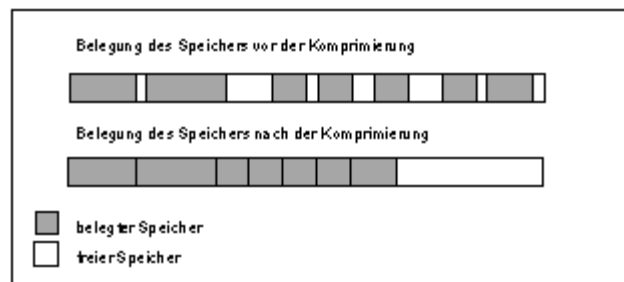
- Bausteine, die im RAM geladen sind, können Sie direkt löschen. Der belegte Speicherplatz wird im Ladespeicher und im Arbeitsspeicher freigegeben.
- Bausteine, die im integrierten EPROM gespeichert sind, werden immer nach dem Urlöschen der CPU in den RAM-Bereich kopiert. Die Kopien im RAM-Bereich können direkt gelöscht werden. Die gelöschten Bausteine werden dabei im EPROM bis zum nächsten Urlöschen oder Netzausfall mit ungepuffertem RAM-Speicher für ungültig markiert. Nach Urlöschen oder Netzausfall mit ungepuffertem RAM-Speicher werden die "gelöschten" Bausteine wieder vom EPROM in das RAM kopiert und aktiv. Bausteine im integrierten EPROM (z. B. bei CPU 312) werden durch Überschreiben mit dem neuen RAM-Inhalt gelöscht.
- EPROM-Memory Cards müssen im Erstellsystem gelöscht werden.

19.5 Komprimieren des Anwenderspeichers (RAM)

19.5.1 Entstehung von Lücken im Anwenderspeicher (RAM)

Durch mehrfaches Löschen und Nachladen von Bausteinen entstehen im Anwenderspeicher (Lade- und Arbeitsspeicher) Lücken, die den nutzbaren Speicherbereich verringern. Durch Komprimieren werden die vorhandenen Bausteine lückenlos im Anwenderspeicher angeordnet und es entsteht ein zusammenhängender freier Speicherbereich.

Nachfolgendes Bild zeigt schematisch, wie belegte Speicherblöcke durch die Funktion "Speicher komprimieren" verschoben werden.



Möglichst im Betriebszustand "STOP" komprimieren

Nur das Komprimieren im Betriebszustand "STOP" schließt alle Speicherlücken. Wenn Sie im Betriebszustand RUN-P (Stellung des Betriebsartenschalters) komprimieren, so werden die gerade bearbeiteten Bausteine nicht verschoben, da sie geöffnet sind. Im Betriebszustand RUN (Stellung des Betriebsartenschalters) kann die Funktion Komprimieren nicht durchgeführt werden (Schreibschutz!).

19.5.2 Komprimieren des Speicherinhalts einer S7-CPU

Möglichkeiten zum Komprimieren

Sie haben zwei Möglichkeiten zum Komprimieren des Anwenderspeichers:

- Tritt beim Laden ein Speicherplatzmangel auf dem Zielsystem auf, so wird ein Dialogfeld aufgeblendet, das Sie auf den Fehler hinweist. Durch Betätigen der entsprechenden Schaltfläche im Dialogfeld können Sie den Speicher komprimieren.
- Als vorbeugende Maßnahme können Sie sich die Speicherauslastung anzeigen lassen (Menübefehl **Zielsystem > Diagnose/Einstellung > Baugruppenzustand..../Register Speicher**) und ggf. die Komprimierung anstoßen.

Vorgehensweise

1. Markieren Sie das S7-Programm in der Ansicht online oder im Fenster "Erreichbare Teilnehmer".
2. Wählen Sie den Menübefehl **Zielsystem > Diagnose/Einstellung > Baugruppenzustand**.
3. Wählen Sie im folgenden Dialogfeld das Register "Speicher". In diesem Register gibt es eine Schaltfläche für das Komprimieren des Speichers, falls die CPU diese Funktion unterstützt.

20 Testen mit der Variablen-tabelle

20.1 Einführung zum Testen mit Variablen-tabellen

Variablen-tabellen bieten den Vorteil unterschiedliche Testumgebungen abspeichern zu können. So können Tests und Beobachtungen während einer Inbetriebsetzung oder zu Service- und Wartungszwecken mühelos reproduziert werden. Die Anzahl der abspeicherbaren Variablen-tabellen ist nicht begrenzt.

Beim Testen mit Variablen-tabellen stehen Ihnen folgende Funktionen zur Verfügung:

- **Beobachten von Variablen**
Damit lassen sich die aktuellen Werte einzelner Variablen eines Anwenderprogramms bzw. einer CPU am PG/PC anzeigen.
- **Steuern von Variablen**
Mit dieser Funktion können Sie einzelnen Variablen eines Anwenderprogramms bzw. einer CPU feste Werte zuweisen. Ein einmaliges und sofortiges Steuern ist auch beim Testen mit Programmstatus möglich.
- **PA freischalten und Steuerwerte aktivieren**
Diese beiden Funktionen geben Ihnen die Möglichkeit, einzelnen Peripherieausgängen einer CPU im Betriebszustand STOP feste Werte zuzuweisen.
- **Forcen von Variablen**
Verwenden Sie diese Funktion, um einzelnen Variablen eines Anwenderprogramms bzw. einer CPU mit einem Wert fest zu belegen, der vom Anwenderprogramm nicht überschrieben werden kann.

Sie können die Werte von folgenden Variablen vorgeben bzw. sich diese anzeigen lassen:

- Eingänge, Ausgänge, Merker, Zeiten und Zähler
- Inhalte von Datenbausteinen
- Peripherie

Die Variablen, die Sie anzeigen oder steuern wollen, fassen Sie in Variablen-tabellen zusammen.

Wann und wie oft die Variablen beobachtet bzw. mit Werten überschrieben werden, bestimmen Sie durch Definieren von Triggerpunkt und Triggerbedingung.

20.2 Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablen-tabellen

Um die Funktionen **Beobachten** und **Steuern** zu benutzen, gehen Sie folgendermaßen vor:

1. Erzeugen Sie eine neue Variablen-tabelle oder öffnen Sie eine bereits vorhandene.
2. Bearbeiten bzw. überprüfen Sie die Variablen-tabelle.
3. Stellen Sie eine Online-Verbindung zwischen der aktuellen Variablen-tabelle und der gewünschten CPU mit dem Menübefehl **Zielsystem > Verbindung herstellen zu >...** her.
4. Wählen Sie mit dem Menübefehl **Variable > Trigger** einen geeigneten Triggerpunkt aus und stellen Sie die Triggerbedingung ein.
5. Die Menübefehle **Variable > Beobachten** und **Variable > Steuern** schalten die jeweilige Funktion ein und wieder aus.
6. Speichern Sie die fertig erstellte Variablen-tabelle mit dem Menübefehl **Tabelle > Speichern** oder **Tabelle > Speichern unter**, damit Sie diese jederzeit wieder aufrufen können.

20.3 Bearbeiten und Speichern von Variablen-tabellen

20.3.1 Erstellen und Öffnen einer Variablen-tabelle

Bevor Sie Variablen beobachten oder steuern können, müssen Sie eine Variablen-tabelle (VAT) erstellen und die gewünschten Variablen eintragen. Um eine Variablen-tabelle zu erstellen, haben Sie folgende Möglichkeiten:

Im SIMATIC Manager

- Markieren Sie den Ordner "Bausteine" und wählen Sie mit dem Menübefehl **Einfügen > S7-Baustein > Variablen-tabelle**. Im Folgedialog können Sie die Tabelle benennen (Eingabefeld "Symbolischer Name"). Dieser Name wird im Projektfenster angezeigt. Die Variablen-tabelle können Sie durch Doppelklick auf das Objekt öffnen.
- Wählen Sie aus der Liste der erreichbaren Teilnehmer eine Verbindung oder in der Online-Ansicht ein S7-/M7-Programm aus. Mit dem Menübefehl **Zielsystem > Variable beobachten/steuern** erzeugen Sie eine unbenannte Variablen-tabelle.

In "Variable beobachten/steuern"

- Sie können mit dem Menübefehl **Tabelle > Neu** eine neue Variablen-tabelle erzeugen, die noch keinem S7/M7-Programm zugeordnet sind. Bereits vorhandene Tabellen öffnen Sie mit **Tabelle > Öffnen**.
- Sie können die entsprechenden Symbole in der Funktionsleiste zum Erzeugen oder Öffnen von Variablen-tabellen benutzen.

Eine einmal erstellte Variablen-tabelle können Sie speichern, ausdrucken und immer wieder zum Beobachten und Steuern verwenden.

20.3.2 Kopieren/Verschieben von Variablentabellen

Variablentabellen können Sie in Baustein-Ordner eines S7/M7-Programms kopieren oder verschieben.

Beim Kopieren und Verschieben von Variablentabellen müssen Sie folgendes beachten:

- In der Symboltabelle des Zielprogramms werden dort bereits vorhandene Symbole aktualisiert.
- Beim Verschieben einer Variablentabelle werden auch die entsprechenden Symbole aus der Symboltabelle des Quellprogramms in die Symboltabelle des Zielprogramms verschoben.
- Beim Löschen von Variablentabellen aus dem Bausteinordner werden auch die entsprechenden Symbole aus der Symboltabelle des S7/M7-Programms gelöscht.
- Ist im Zielprogramm bereits eine Variablentabelle mit demselben Namen vorhanden, erhalten Sie beim Kopieren die Möglichkeit die Variablentabelle umzubennen (defaultmäßig wird zum bestehenden Namen eine Nummer angehängt).

20.3.3 Speichern einer Variablentabelle

Gespeicherte Variablentabellen können Sie beim erneuten Test Ihres Programms wieder zum Beobachten und Steuern verwenden.

1. Speichern Sie die Variablentabelle mit dem Menübefehl **Tabelle > Speichern**.
2. Wurde die Variablentabelle neu erstellt, so müssen Sie nun einen Namen für die Variablentabelle angeben, z. B. "Programmtest_1".

Beim Speichern der Variablentabelle werden alle aktuellen Einstellungen sowie das Tabellenformat gespeichert, d. h. die Einstellungen im Menüpunkt Trigger werden gespeichert.

20.4 Eingeben von Variablen in Variablen-tabellen

20.4.1 Einfügen von Operanden oder Symbolen in eine Variablen-tabelle

Bestimmen Sie die Variablen, deren Werte Sie vorgeben oder beobachten wollen und tragen Sie diese in die Variablen-tabelle ein. Gehen Sie dabei von "außen" nach "innen" vor, d. h. wählen Sie zuerst die Eingänge und danach die Variablen, die von den Eingängen beeinflusst werden bzw. die die Ausgänge beeinflussen und zum Schluss die Ausgänge.

Wenn Sie z. B. das Eingangsbit 1.0, das Merkerwort 5 und das Ausgangsbyte 0 beobachten möchten, geben Sie in der Operandenspalte folgendes ein:

Beispiel:

E 1.0
MW 5
AB 0

Beispiel für eine ausgefüllte Variablen-tabelle

Nachfolgendes Bild zeigt eine Variablen-tabelle mit den eingblendeten Spalten: Operand, Symbol, Anzeigeformat, Statuswert und Steuerwert.

	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	//OB1 Netzwerk 1				
2	E 0.1	"Taster 1"	BOOL	true	
3	E 0.2	"Taster 2"	BOOL	true	
4	A 4.0	"Lampe Grün"	BOOL	false	
5	//OB1 Netzwerk 3				
6	E 0.5	"Automatik Ein"	BOOL	true	
7	E 0.6	"Hand Ein"	BOOL	true	
8	A 4.2	"Automatikbetrieb"	BOOL	true	true
9	//OB1 Aufuf FB1 für Benzimotor einschalten				
10	E 1.0	"BM_einschalten"	BOOL	false	
11	E 1.1	"BM_ausschalten"	BOOL	false	
12	E 1.2	"BM_Störung"	BOOL	false	
13	A 5.1	"BM_Soll_erreicht"	BOOL	false	
14	A 5.0	"BM_Ein"	BOOL		true
15	//OB1 Aufuf FB1 für Dieselmotor einschalten				
16	E 1.4	"DM_einschalten"	BOOL	false	
17	E 1.5	"DM_ausschalten"	BOOL		

MPI = 3 (direkt) Run

Hinweise zum Einfügen von Symbolen

- Die zu steuernde Variable geben Sie mit ihrem Operand oder als Symbol an. Operanden und Symbole können Sie sowohl in die Spalte "Operand" als auch in die Spalte "Symbol" eintragen. Der Eintrag wird automatisch in die passende Spalte geschrieben.
Falls in der Symbol-tabelle ein entsprechendes Symbol definiert ist, wird der Eintrag in der Symbol-spalte bzw. der Operandenspalte automatisch ergänzt.
- Sie können nur solche Symbole eintragen, die bereits in der Symbol-tabelle definiert sind.
- Ein Symbol muss genau so eingegeben werden, wie es in der Symbol-tabelle definiert ist.
- Symbolnamen, die Sonderzeichen enthalten, müssen in Hochkommas eingeschlossen werden (z. B. "Motor.Aus", "Motor+Aus", "Motor-Aus").
- Um neue Symbole in der Symbol-tabelle zu definieren, wählen Sie den Menübefehl **Extras > Symbol-tabelle**. Symbole können auch aus der Symbol-tabelle kopiert und in eine Variablen-tabelle eingefügt werden.

Syntaxprüfung

Wenn Sie Variablen in die Variablentabelle eintragen, so wird vor Verlassen der Zeile eine Syntaxprüfung durchgeführt. Fehlerhafte Einträge werden rot gekennzeichnet.

Wenn Sie den Cursor in eine rot markierte Zeile stellen, informiert Sie eine Kurzinformation die Ursache des Fehlers. Mit der Taste F1 erhalten Sie Hinweise zur Behebung.

Hinweis

Wenn Sie die Variablentabelle vorzugsweise mit der Tastatur (ohne Maus) bearbeiten wollen, sollten Sie die "Kurzinformationen bei Tastaturbedienung" eingeschaltet haben.

Ändern Sie gegebenenfalls die Einstellung in der Variablentabelle über den Menübefehl **Extras > Einstellungen** im Register "Allgemein".

Maximale Größe

Eine Variablentabelle kann maximal 255 Zeichen pro Zeile umfassen. Ein Umbruch in die nächste Zeile ist nicht möglich. Ihre Länge ist auf 1024 Zeilen festgelegt. Damit ist die maximale Größe erreicht.

20.4.2 Einfügen eines Bereichs zusammenhängender Operanden in eine Variablentabelle

1. Öffnen Sie eine Variablentabelle.
2. Positionieren Sie den Cursor in die Zeile, nach der Sie den Bereich zusammenhängender Operanden einfügen möchten.
3. Wählen Sie den Menübefehl **Einfügen > Bereich**. Das Dialogfeld "Bereich einfügen" wird angezeigt.
4. Tragen Sie in das Feld "Ab Operand" eine Operandenadresse als Startadresse ein.
5. Tragen Sie in das Feld "Anzahl" die Anzahl der einzufügenden Zeilen ein.
6. Wählen Sie aus der angezeigten Liste das gewünschte Anzeigeformat aus.
7. Klicken Sie auf "OK".

Der Bereich wird in die Variablentabelle eingefügt.

20.4.3 Einfügen von Steuerwerten

Steuerwert als Kommentar

Wenn Sie den "Steuerwert" einer Variablen unwirksam schalten wollen, so verwenden Sie den Menübefehl **Variable > Steuerwert als Kommentar**. Ein Kommentarzeichen "//" vor dem zu steuernden Wert einer Variablen zeigt deren Unwirksamkeit an. Anstelle des Menübefehlsaufrufs, kann dem "Steuerwert" auch das Kommentarzeichen "//" vorangestellt werden. Die Unwirksamkeit des "Steuerwertes", kann durch den erneuten Aufruf des Menübefehls **Variable > Steuerwert als Kommentar** oder der Entfernung des Kommentarzeichens wieder aufgehoben werden.

20.4.4 Obergrenzen für die Eingabe von Zeiten

Beachten Sie folgende Obergrenzen für die Eingabe von Zeiten:

W#16#3999 (Maximalwert im BCD-Format)

Beispiele:

Operand	Anzeigeformat	Eingabe	Anzeige Steuerwert	Erläuterung
T	SIMATIC_ZEIT	137	S5TIME#130MS	Konvertierung in Millisekunden
MW	SIMATIC_ZEIT	137	S5TIME#890MS	Darstellung im BCD-Format möglich
MW	HEX	137	W#16#0089	Darstellung im BCD-Format möglich
MW	HEX	157	W#16#009D	Darstellung im BCD-Format nicht möglich, deshalb ist das Anzeigeformat SIMATIC_ZEIT nicht wählbar

Hinweis

- Sie können Zeiten in Millisekunden-Schritten eingeben, aber der eingegebene Wert wird an ein Zeitraster angepasst. Die Größe des Zeitrasters ist abhängig von der Größe des eingegebenen Zeitwertes (137 ergibt 130 ms, die 7 ms wurden abgerundet).
- Die Steuerwerte von Operanden vom Datentyp WORD, z. B. EW 1, werden ins BCD-Format konvertiert. Nicht jedes Bitmuster ist jedoch eine gültige BCD-Zahl ! Wenn bei einem Operanden vom Datentyp WORD die Eingabe nicht als SIMATIC_ZEIT dargestellt werden kann, wird automatisch auf das voreingestellte Format (hier: HEX, siehe Anzeigeformat wählen, Voreinstellung Menü Ansicht) umgeschaltet, so dass der eingegebene Wert angezeigt werden kann.

BCD-Format für Variablen im Format SIMATIC_ZEIT

Werte von Variablen im Format SIMATIC_ZEIT werden im BCD-Format eingegeben.

Die 16 Bit haben folgende Bedeutung:

| 0 0 x x | h h h h | z z z z | e e e e |

Bit 15 und 14 sind immer Null.

Bit 13 und 12 (mit xx markiert) legen den Multiplikator für die Bits 0 bis 11 fest:

- 00 => Multiplikator 10 Millisekunden
- 01 => Multiplikator 100 Millisekunden
- 10 => Multiplikator 1 Sekunde
- 11 => Multiplikator 10 Sekunden

Bit 11 bis 8 Hunderter (hhhh)

Bit 7 bis 4 Zehner (zzzz)

Bit 3 bis 0 Einer (eeee)

20.4.5 Obergrenzen für die Eingabe von Zählern

Bitte beachten Sie folgende Obergrenzen für die Eingabe von Zählern:

Obergrenzen für Zähler: C#999
W#16#0999 (Maximalwert im BCD-Format)

Beispiele:

Operand	Anzeigeformat	Eingabe	Anzeige Steuerwert	Erläuterung
Z 1	ZÄHLER	137	C#137	Konvertierung
MW 4	ZÄHLER	137	C#89	Darstellung im BCD-Format möglich
MW 4	HEX	137	W#16#0089	Darstellung im BCD-Format möglich
MW 6	HEX	157	W#16#009D	Darstellung im BCD-Format nicht möglich, deshalb ist das Anzeigeformat ZÄHLER nicht wählbar

Hinweis

- Wenn Sie bei einem Zähler eine Dezimalzahl eingeben und den Wert nicht durch C# kennzeichnen, so wird dieser Wert automatisch in das BCD-Format umgerechnet (137 ergibt C#137).
- Die Steuerwerte von Operanden vom Datentyp WORD, z. B. EW 1, werden ins BCD-Format konvertiert. Nicht jedes Bitmuster ist jedoch eine gültige BCD-Zahl! Wenn bei einem Operanden vom Datentyp WORD die Eingabe nicht als ZÄHLER dargestellt werden kann, wird automatisch auf das voreingestellte Format (hier: HEX, siehe Anzeigeformat wählen, Voreinstellung (Menü Ansicht) umgeschaltet, so dass der eingegebene Wert angezeigt werden kann.

20.4.6 Einfügen von Kommentarzeilen

Kommentarzeilen werden durch das Kommentarzeichen "//" eingeleitet.

Wenn Sie eine oder mehrere Zeilen der Variablen­tabelle unwirksam schalten wollen, verwenden Sie den Menübefehl **Bearbeiten > Zeile nicht wirksam** oder das entsprechende Symbol in der Funktionsleiste:



20.4.7 Beispiele

20.4.7.1 Beispiel für die Eingabe von Operanden in Variablen­tabellen

Erlaubter Operand:	Datentyp:	Beispiel (Mnemonic deutsch):
Eingang Ausgang Merker	BOOL	E 1.0 A 1.7 M 10.1
Eingang Ausgang Merker	BYTE	EB 1 AB 10 MB 100
Eingang Ausgang Merker	WORD	EW 1 AW 10 MW 100
Eingang Ausgang Merker	DWORD	ED 1 AD 10 MD 100
Peripherie (Eingang Ausgang)	BYTE	PEB 0 PAB 1
Peripherie (Eingang Ausgang)	WORD	PEW 0 PAW 1
Peripherie (Eingang Ausgang)	DWORD	PED 0 PAD 1
Zeiten	TIMER	T 1
Zähler	COUNTER	Z 1
Datenbaustein	BOOL	DB1.DBX 1.0
Datenbaustein	BYTE	DB1.DBB 1
Datenbaustein	WORD	DB1.DBW 1
Datenbaustein	DWORD	DB1.DBD 1

Hinweis

Die Eingabe von "DB0. .." ist aufgrund interner Benutzung nicht erlaubt.

Im Fenster Forcewerte

- Beim Forcen von S7-300-Baugruppen sind nur Eingänge, Ausgänge und Peripherie (Ausgänge) erlaubt.
- Beim Forcen von S7-400-Baugruppen sind nur Eingänge, Ausgänge, Merker und Peripherie (Eingänge/Ausgänge) erlaubt.

20.4.7.2 Beispiel für die Eingabe eines zusammenhängenden Operandenbereichs

Öffnen Sie eine Variablen-tabelle und lassen Sie sich das Dialogfeld "Bereich einfügen" mit dem Menübefehl **Einfügen > Bereich** anzeigen.

Bei den Angaben im Dialogfeld werden folgende Zeilen für Merker in die Variablen-tabelle eingefügt:

Ab Operand: M 3.0

Anzahl: 10

Anzeigeformat: BIN

Operand	Anzeigeformat
M 3.0	BIN
M 3.1	BIN
M 3.2	BIN
M 3.3	BIN
M 3.4	BIN
M 3.5	BIN
M 3.6	BIN
M 3.7	BIN
M 4.0	BIN
M 4.1	BIN

Beachten Sie in diesem Beispiel, wie sich die Bezeichnung in der Spalte "Operand" nach dem achten Eintrag ändert.

20.4.7.3 Beispiele für die Eingabe von Steuer-/Forcewerten

Bitoperanden

Mögliche Bitoperanden	erlaubte Steuer-/Forcewerte
E1.0	true
M1.7	false
A10.7	0
DB1.DBX1.1	1
E1.1	2#0
M1.6	2#1

Byteoperanden

mögliche Byteoperanden	erlaubte Steuer-/Forcewerte
EB 1	2#00110011
MB 12	b#16#1F
MB 14	1F
AB 10	'a'
DB1.DBB 1	10
PAB 2	12

Wortoperanden

mögliche Wortoperanden	erlaubte Steuer-/Forcewerte
EW 1	2#0011001100110011
MW 12	w#16#ABCD
MW 14	ABCD
AW 10	b#(12,34)
DB1.DBW 1	'ab'
PAW 2	12345
MW 3	12345
MW 5	S5t#12s340ms
MW 7	0.3s oder 0,3s
MW 9	C#123
MW 11	d#1990-12-31

Doppelwortoperanden

mögliche Doppelwortoperanden	erlaubte Steuer-/Forcewerte
ED 1	2#0011001100110011001100110011
MD 0	1.23e4
MD 4	1.2
AD 10	dw#16#abcdef10
AD 12	ABCDEF10
DB1.DBD 1	b#(12,34,56,78)
PAD 2	'abcd'
MD 8	L# -12
MD 12	L#12
MD 16	123456789
MD 20	123456789
MD 24	T#12s345ms
MD 28	Tod#1:2:34.567
MD 32	p#e0.0

Zeit

mögliche Operanden vom Typ Zeit	erlaubte Steuer-/Forcewerte	Erläuterung
T 1	0	Zeitwert in Millisekunden (ms)
T 12	20	Zeitwert in Millisekunden (ms)
T 14	12345	Zeitwert in Millisekunden (ms)
T 16	s5t#12s340ms	Zeitwert 12s 340ms
T 18	1.3	Zeitwert 1s 300 ms
T 20	1.3s	Zeitwert 1s 300 ms

Das Steuern einer Zeit beeinflusst nur den Wert, nicht den Zustand. D. h. die Zeit T1 kann auf den Wert 0 gesteuert werden, das Verknüpfungsergebnis bei U T1 wird nicht verändert.

Die Zeichenfolgen "s5t", "s5time" können sowohl klein als auch groß geschrieben werden.

Zähler

Mögliche Operanden vom Typ Zähler	erlaubte Steuer-/Forcewerte
Z 1	0
Z 14	20
Z 16	c#123

Das Steuern eines Zählers beeinflusst nur den Wert, nicht den Zustand. D. h. der Zähler Z1 kann auf den Wert 0 gesteuert werden, das Verknüpfungsergebnis bei U Z1 wird nicht verändert.

20.5 Herstellen einer Verbindung zur CPU

Damit Sie die Variablen, die sie in Ihrer aktuellen Variablen-tabelle (VAT) zusammengefasst haben, beobachten oder steuern können, müssen Sie eine Verbindung zur entsprechenden CPU herstellen. Es ist möglich, jede Variablen-tabelle mit einer anderen CPU zu verbinden.

Anzeige der Online-Verbindung

Wenn eine Online-Verbindung besteht, so wird dies in der Titelzeile des Variablen-tabellenfensters mit dem Begriff "ONLINE" angezeigt. In der Statuszeile werden, abhängig von der CPU, die Betriebszustände "RUN", "STOP", "GETRENNT" oder "VERBUNDEN" abgebildet.

Herstellen der Online-Verbindung zur CPU

Besteht keine Online-Verbindung zur gewünschten CPU, definieren Sie diese mit dem Menübefehl **Zielsystem > Verbindung herstellen zu >...** zur gewünschten CPU, um die Variablen zu beobachten oder zu steuern.

Abbauen der Online-Verbindung zur CPU

Mit dem Menübefehl **Zielsystem > Verbindung abbauen** brechen Sie die Verbindung zwischen Variablen-tabelle und CPU wieder ab.

Hinweis

Wenn Sie eine unbenannte Variablen-tabelle mit dem Menübefehl **Tabelle > Neu** erzeugt haben, können Sie eine Verbindung zur letzten projektierten CPU herstellen, falls sie definiert ist.

20.6 Beobachten von Variablen

20.6.1 Einführung zum Beobachten von Variablen

Zum Beobachten von Variablen haben Sie folgende Möglichkeiten:

- Schalten Sie die Funktion "Beobachten" mit dem Menübefehl **Variable > Beobachten** ein. Die Werte der ausgewählten Variablen werden, abhängig vom eingestellten Triggerpunkt und der eingestellten Triggerbedingung, in der Variablen-tabelle angezeigt. Wenn Sie die Triggerbedingung "Permanent" eingestellt haben, schalten Sie die Funktion "Beobachten" mit dem Menübefehl **Variable > Beobachten** wieder aus.
- Aktualisieren Sie die Werte der ausgewählten Variablen einmalig und sofort mit dem Menübefehl **Variable > Statuswerte aktualisieren**. Die aktuellen Werte der ausgewählten Variablen werden in der Variablen-tabelle angezeigt.

Beobachten abbrechen mit ESC-Taste

Wenn Sie - bei laufendem "Beobachten" - die Taste ESC drücken, wird "Beobachten" ohne Rückfrage beendet.

20.6.2 Festlegen des Triggers zum Beobachten von Variablen

Sie können sich zum Beobachten die aktuellen Werte einzelner Variablen eines Anwenderprogramms an einem vorgegebenen Punkt im Ablauf des Programms (Triggerpunkt) am PG anzeigen lassen.

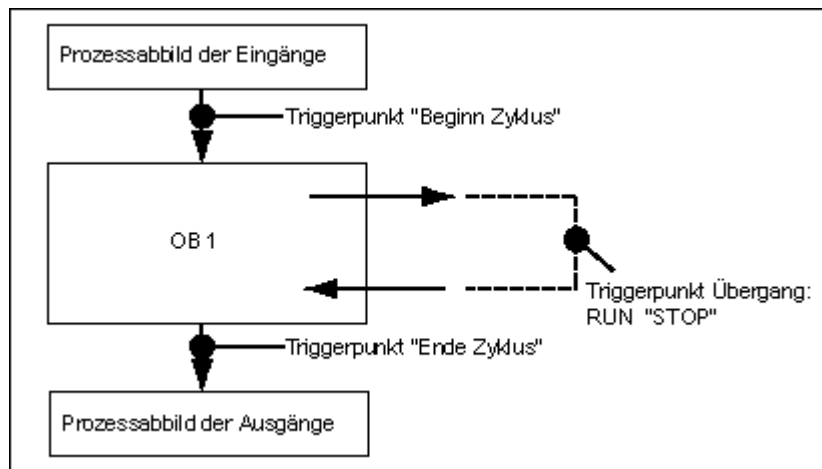
Mit der Wahl des Triggerpunktes bestimmen Sie den Zeitpunkt, an dem die Statuswerte von Variablen angezeigt werden.

Mit dem Menübefehl **Variable > Trigger** können Sie den Triggerpunkt und eine Triggerbedingung einstellen

Trigger	Einstellungsmöglichkeiten
Triggerpunkt	Beginn Zyklus Ende Zyklus Übergang von RUN nach STOP
Triggerbedingung	einmalig permanent

Triggerpunkt

Die Lage der Triggerpunkte veranschaulicht das folgende Bild.



Damit Ihnen der gesteuerte Wert in der Spalte "Statuswert" angezeigt wird, sollten Sie den Triggerpunkt für Beobachten auf "Zyklusbeginn" und den Triggerpunkt für Steuern auf "Zyklusende" einstellen!

Trigger sofort

Sie können die Werte ausgewählter Variablen mit dem Menübefehl **Variable > Statuswerte aktualisieren**. Dieser Auftrag wird einmalig und schnellstmöglich durchgeführt, ohne Bezug zu einer bestimmten Stelle im Anwenderprogramm. Diese Funktionen werden hauptsächlich im STOP-Zustand zum Steuern und Beobachten angewendet.

Triggerbedingung

Nachfolgende Tabelle zeigt, was die eingestellte Triggerbedingung für das Beobachten von Variablen bewirkt:

	Triggerbedingung Einmalig	Triggerbedingung Permanent
Beobachten von Variablen	<i>Einmaliges Aktualisieren abhängig vom Triggerpunkt</i>	<i>Beobachten mit definiertem Trigger</i> Beim Testen eines Bausteins können Sie die Weiterverarbeitung genau mitverfolgen.

20.7 Steuern von Variablen

20.7.1 Einführung zum Steuern von Variablen

Zum Steuern von Variablen haben Sie folgende Möglichkeiten:

- Schalten Sie die Funktion "Steuern" mit dem Menübefehl **Variable > Steuern** ein. Das Anwenderprogramm übernimmt die Steuerwerte für die ausgewählten Variablen aus der Variablen-tabelle, abhängig vom jeweils eingestellten Triggerpunkt und der eingestellten Triggerbedingung. Wenn Sie die Triggerbedingung "Permanent" eingestellt haben, schalten Sie die Funktion "Steuern" mit dem Menübefehl **Variable > Steuern** wieder aus.
- Aktualisieren Sie die Werte der ausgewählten Variablen einmalig und sofort mit dem Menübefehl **Variable > Steuerwert aktivieren**.

Weitere Möglichkeiten bieten die Funktionen Forcen und Peripherieausgänge (PA) freischalten.

Beachten Sie beim Steuern:

- Es werden nur Operanden gesteuert, die zu Beginn des Steuerns in der Variablen-tabelle sichtbar waren.
Wird der sichtbare Bereich der Variablen-tabelle nach Beginn des Steuerns verkleinert, werden eventuell Operanden gesteuert, welche nicht mehr sichtbar sind.
Wird der sichtbare Bereich der Variablen-tabelle vergrößert, gibt es eventuell Operanden, die nicht gesteuert werden.
- Steuern kann nicht rückgängig gemacht werden (z. B. mit **Bearbeiten > Rückgängig**).



Gefahr

Ein Verändern der Variablenwerte bei laufendem Anlagenbetrieb kann bei Funktionsstörungen oder Programmfehlern schwere Sach- und Personenschäden verursachen!

Vergewissern Sie sich, dass keine gefährlichen Zustände eintreten können, bevor Sie die Funktion "Steuern" ausführen!

Steuern abbrechen mit ESC-Taste

Wenn Sie - bei laufendem "Steuern" - die Taste ESC drücken, wird "Steuern" ohne Rückfrage beendet.

20.7.2 Festlegen des Triggers zum Steuern von Variablen

Sie können einzelnen Variablen eines Anwenderprogramms an einem vorgegebenen Punkt im Ablauf des Programms (Triggerpunkt) einmalig oder permanent feste Werte zuweisen.

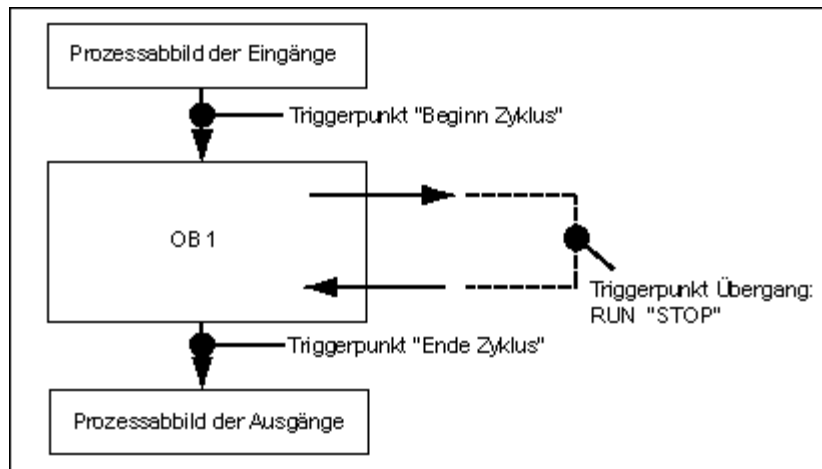
Mit der Wahl des Triggerpunktes bestimmen Sie den Zeitpunkt, an dem den Variablen die Steuerwerte zugewiesen werden.

Mit dem Menübefehl **Variable > Trigger** können Sie den Triggerpunkt und eine Triggerbedingung einstellen

Trigger	Einstellungsmöglichkeiten
Triggerpunkt	Beginn Zyklus Ende Zyklus Übergang von RUN nach STOP
Triggerbedingung	einmalig permanent

Triggerpunkt

Die Lage der Triggerpunkte veranschaulicht das folgende Bild.



Aus der Lage der Triggerpunkte geht hervor:

- das Steuern von Eingängen ist nur sinnvoll bei Triggerpunkt "Zyklusbeginn" (entspricht Beginn des Anwenderprogramms OB 1), da sonst das Prozessabbild der Eingänge nach dem Steuern wieder aktualisiert und damit überschrieben wird
- das Steuern von Ausgängen ist nur sinnvoll bei Triggerpunkt "Zyklusende" (entspricht Ende des Anwenderprogramms OB 1), da sonst das Prozessabbild der Ausgänge vom Anwenderprogramm überschrieben werden kann

Damit Ihnen der gesteuerte Wert in der Spalte "Statuswert" angezeigt wird, sollten Sie den Triggerpunkt für Beobachten auf "Zyklusbeginn" und den Triggerpunkt für Steuern auf "Zyklusende" einstellen!

Beim Steuern von Variablen gilt bezüglich der Triggerpunkte:

- Wenn Sie als Triggerbedingung "Einmalig" eingestellt haben, so erhalten Sie eine Meldung, falls die ausgewählten Variablen nicht zu steuern sind.
- Bei der Triggerbedingung "Permanent" erhalten Sie keine Rückmeldung.

Trigger sofort

Sie können die Werte ausgewählter Variablen mit dem Menübefehl **Variable > Steuerwerte aktivieren** steuern. Dieser Auftrag wird einmalig und schnellstmöglich durchgeführt, ohne Bezug zu einer bestimmten Stelle im Anwenderprogramm. Diese Funktion wird hauptsächlich im STOP-Zustand zum Steuern angewendet.

Triggerbedingung

Nachfolgende Tabelle zeigt, was die eingestellte Triggerbedingung für das Steuern von Variablen bewirkt:

	Triggerbedingung Einmalig	Triggerbedingung Permanent
Steuern von Variablen	<p><i>Einmaliges Aktivieren (Steuern von Variablen)</i></p> <p>Sie können Variablen abhängig vom Triggerpunkt einmalig Werte zuweisen.</p>	<p><i>Steuern mit definiertem Trigger</i></p> <p>Durch das Zuweisen von festen Werten können Sie für Ihr Anwenderprogramm bestimmte Situationen simulieren und somit die programmierten Funktionen testen.</p>

20.8 Forcen von Variablen

20.8.1 Sicherheitsvorkehrungen zum Forcen von Variablen



Personen- und Sachschaden verhüten!

Beachten Sie, dass beim Ausführen der Funktion "Forcen" eine falsche Handlung

- das Leben oder die Gesundheit von Personen gefährden oder
- Schäden an der Maschine oder an der gesamten Anlage verursachen kann.



Vorsicht

- Bevor Sie die Funktion "Forcen" starten, sollten Sie sicherstellen, dass niemand gleichzeitig auf derselben CPU diese Funktion ausführt.
- Ein Forceauftrag kann nur mit dem Menübefehl **Variable > Force löschen** gelöscht bzw. beendet werden. Das Schließen des Fensters Forcewerte oder Beenden der Applikation "Variable beobachten und steuern" löscht den Forceauftrag nicht.
- "Forcen" kann mit dem Menübefehl **Bearbeiten > Rückgängig** nicht rückgängig gemacht werden.
- Informieren Sie sich über die Unterschiede zwischen Forcen und Steuern von Variablen.
- Wenn eine CPU die Funktion Forcen nicht unterstützt, sind alle das Forcen betreffende Menübefehle im Menü Variable nicht anwählbar.

Wenn die Ausgabesperre mit dem Menübefehl **Variable > PA freischalten** aufgehoben ist, geben alle geforcten Ausgabebaugruppen ihren Forcewert aus.

20.8.2 Einführung zum Forcen von Variablen

Sie können einzelne Variablen eines Anwenderprogramms mit festen Werten vorbelegen, so dass sie auch vom Anwenderprogramm, das in der CPU abläuft, nicht verändert oder überschrieben werden können. Voraussetzung ist, dass die CPU diese Möglichkeit unterstützt (wie z. B. S7-400-CPUs). Durch das feste Vorbelegen von Variablen mit Werten können Sie für Ihr Anwenderprogramm bestimmte Situationen einstellen und damit die programmierten Funktionen testen.

Fenster "Forcewerte"

Die Menübefehle zum Forcen sind erst anwählbar, wenn das Fenster "Forcewerte" aktiv ist.

Zur Anzeige dieses Fensters wählen Sie den Menübefehl **Variable > Forcewerte anzeigen**.

Für eine CPU sollten Sie nur ein einziges Fenster "Forcewerte" öffnen. In diesem Fenster werden die Variablen mit den zugehörigen Forcewerten für den aktiven Forceauftrag angezeigt.

Beispiel eines Forcefensters

	Operand	Symbol	Anzeigeformat	Forcewert
1	F EB 0		HEX	B#16#10
2	F A 0.1		BOOL	true
3	F A 1.2		BOOL	true
4				

In der **Titelzeile** steht der Name der aktuellen Online-Verbindung.

In der **Statuszeile** steht der Zeitpunkt (Datum und Uhrzeit), zu dem der Forceauftrag aus der CPU gelesen wurde.

Wenn kein Forceauftrag aktiv ist, ist das Fenster leer.

Die unterschiedliche Art der **Anzeige von Variablen** im Fenster Forcewerte hat folgende Bedeutung:

Anzeige	Bedeutung
Anzeige fett	Variablen, die bereits fest mit einem Wert in der CPU vorbelegt sind.
Anzeige normal	Variablen, die gerade editiert werden.
Anzeige gegraut	Variablen einer Baugruppe, die nicht vorhanden/nicht gesteckt ist oder Variablen mit Adressierungsfehler, es wird eine Fehlermeldung eingeblendet.

Übernehmen von forcefähigen Operanden aus der Variablen-tabelle

Wenn Sie Variablen aus einer Variablen-tabelle in das Forcewerte-Fenster übernehmen möchten, selektieren Sie bitte die Tabelle und markieren die gewünschten Variablen. Öffnen Sie anschließend das Forcewerte-Fenster mit dem Menübefehl **Variable > Forcewerte** anzeigen. Die Variablen, die von der Baugruppe geforct werden können, werden im Forcewerte-Fenster übernommen.

Übernehmen des Forceauftrags aus der CPU oder Neuerstellung eines Forceauftrags

Wenn das Fenster "Forcewerte" geöffnet und aktiv ist, wird eine weitere Meldung eingeblendet:

- Wenn Sie bestätigen, werden Änderungen im Fenster "Forcewerte" mit dem in der CPU vorhandenen Forceauftrag überschrieben. Mit dem Menübefehl **Bearbeiten > Rückgängig** können Sie den vorherigen Fensterinhalt wiederherstellen.
- Wenn Sie abbrechen, bleibt der aktuelle Inhalt des Fensters "Forcewerte" erhalten.
Danach können Sie den Inhalt im Fenster "Forcewerte" als Variablen-tabelle mit dem Menübefehl **Tabelle > Speichern unter** sichern oder den Menübefehl **Variable > Forcen** anwählen: damit wird der aktuelle Inhalt des Fensters "Forcewerte" als neuer Forceauftrag in die CPU geschrieben.

Beobachten und Steuern von Variablen ist nur in der Variablen-tabelle möglich, nicht aber im Fenster "Forcewerte".

Löschen von Forcewerten

Mit dem Menübefehl **Variable > Forcewerte anzeigen** öffnen Sie das Fenster Forcewerte. Anschließend können Sie mit dem Menübefehl **Variable > Force löschen** die Forcewerte in der ausgewählten CPU löschen.

Speichern eines Forcewerte-Fensters

Den Inhalt eines Fensters "Forcewerte" können Sie in einer Variablen-tabelle sichern. Mit dem Menübefehl **Einfügen > Variablen-tabelle** können Sie den gesicherten Inhalt wieder in ein Forcewerte-Fenster einfügen.

Hinweise zur Symbolik im Fenster "Forcewerte"

Die Symbolik des letzten aktiven Fensters wird übernommen, außer wenn Sie "Variable beobachten und steuern" von einer anderen Applikation aufrufen, die keine Symbolik hat.

Wenn Sie keine symbolischen Namen eingeben können, ist die Spalte "Symbol" ausgeblendet. Der Menüeintrag **Extras > Symbol-tabelle** ist dann ebenfalls nicht anwählbar.

20.8.3 Unterschiede zwischen Forcen und Steuern von Variablen

In der folgenden Übersicht sind die Unterschiede zwischen Forcen und Steuern zusammengefasst:

Merkmal / Funktion	Forcen mit S7-400 (incl. CPU 318-2DP)	Forcen mit S7-300 (ohne CPU 318-2DP)	Steuern
Merker (M)	ja	-	ja
Zeiten und Zähler (T, Z)	-	-	ja
Datenbausteine (DB)	-	-	ja
Peripherie-Eingänge (PEB, PEW, PED)	ja	-	-
Peripherieausgänge (PAB, PAW, PAD)	ja	-	ja
Eingänge und Ausgänge (E, A)	ja	ja	ja
Anwenderprogramm kann die Steuer-/Forcewerte überschreiben	-	ja	ja
Ersetzen des Forcewertes unterbrechungsfrei wirksam	ja	ja	-
Nach Beenden der Applikation behalten die Variablen die Werte	ja	ja	-
Nach dem Auftrennen der Verbindung zur CPU behalten die Variablen die Werte	ja	ja	-
Adressierungsfehler erlaubt: z. B. EW1 Steuer-/Forcewert: 1 EW1 Steuer-/Forcewert: 0	-	-	der letzte wird wirksam
Trigger einstellen	immer Trigger sofort	immer Trigger sofort	Einmalig oder permanent
Funktion wirkt nur auf Variable im sichtbaren Bereich des aktiven Fensters	wirkt auf alle Forcewerte	wirkt auf alle Forcewerte	ja

Hinweis

- Beim "PA freischalten" werden die Forcewerte für geforcete Peripherieausgänge an den entsprechenden Ausgabebaugruppen wirksam, die Steuerwerte für permanent gesteuerte Peripherieausgänge jedoch nicht.
- Beim Forcen besitzt die Variable immer den geforcten Wert. Dieser Wert wird bei jedem Lesezugriff im Anwenderprogramm gelesen. Sämtliche Schreibzugriffe sind unwirksam.
- Beim permanenten Steuern werden Schreibzugriffe des Programms wirksam und bleiben bis zum nächsten Triggerpunkt wirksam.

21 Testen mit Programmstatus

Sie können Ihr Programm testen, indem Sie sich für jede Anweisung den Zustand des Programmstatus (VKE, Statusbit) oder den Inhalt der entsprechenden Register anzeigen lassen. Den Umfang der angezeigten Information legen Sie im Register "AWL" des Dialogfelds "Einstellungen" fest. Das Dialogfeld öffnen Sie mit dem Menübefehl **Extras > Einstellungen** im Fenster "KOP/AWL/FUP: Bausteine programmieren".



Warnung

Ein Test bei laufendem Anlagenbetrieb kann bei Funktionsstörungen oder Programmfehlern schwere Sach- und Personenschäden verursachen!

Vergewissern Sie sich, dass keine gefährlichen Zustände eintreten können, bevor Sie diese Funktion ausführen!

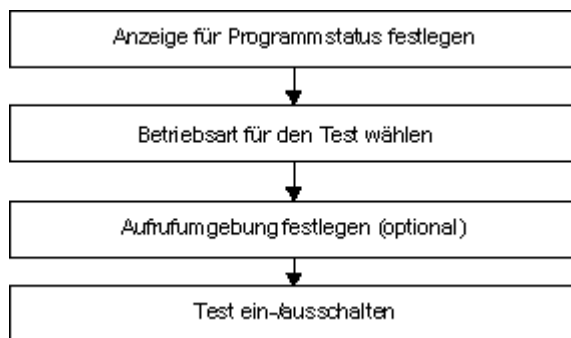
Voraussetzungen

Um den Programmstatus anzeigen lassen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Sie müssen den Baustein fehlerfrei gespeichert und danach auf die CPU geladen haben.
- Die CPU ist in Betrieb, das Anwenderprogramm läuft.

Prinzipielle Vorgehensweise für die Beobachtung des Programmstatus

Es ist dringend zu empfehlen, nicht sofort das gesamte Programm aufzurufen und zu testen, sondern die Bausteine nach und nach aufzurufen und zu testen. Dabei beginnen Sie mit den unterlagerten Codebausteinen, d. h. Sie testen die Bausteine in der letzten Schachtelungstiefe der Aufrufhierarchie, z. B. indem Sie diese im OB 1 aufrufen und über Beobachten und Steuern von Variablen dem Baustein die zu testende Umgebung schaffen.



Zum Setzen von Haltepunkten und zur Programmausführung in Einzelschritten muss der Betriebsmodus Testbetrieb eingestellt sein (siehe Menübefehl **Test > Betrieb**). Im Prozessbetrieb sind diese Testfunktionen nicht möglich.

21.1 Anzeigen im Programmstatus

Die Anzeige des **Programmstatus** wird zyklisch aktualisiert. Sie beginnt mit dem markierten Netzwerk.

Voreingestellte Kennzeichnungen in KOP und FUP

- Status erfüllt: grüne durchgezogene Linien
- Status nicht erfüllt: blaue gestrichelte Linien
- Status unbekannt: schwarze durchgezogene Linien

Diese Voreinstellung von Linienart und Farbe ist änderbar über den Menübefehl **Extras > Einstellungen** / Register KOP/FUP.

Status von Elementen

- Der Status eines Kontakts ist
 - erfüllt, wenn der Operand den Wert "1" hat,
 - nicht erfüllt, wenn der Operand den Wert "0" hat,
 - unbekannt, wenn der Wert des Operanden unbekannt ist.
- Der Status von Elementen mit Freigabeausgang (ENO) entspricht dem Status eines Kontakts mit dem Wert des ENO-Ausgangs als Operanden.
- Der Status von Elementen mit Q-Ausgang entspricht dem Status eines Kontakts mit dem Wert des Operanden.
- Der Status bei CALLs ist erfüllt, wenn nach dem Aufruf das BIE-Bit gesetzt ist.
- Der Status einer Sprungoperation ist erfüllt, wenn der Sprung ausgeführt wird, d. h. wenn die Sprungbedingung erfüllt ist.
- Elemente mit Freigabeausgang (ENO) werden schwarz dargestellt, wenn der Freigabeausgang nicht beschaltet ist.

Status von Linien

- Linien sind schwarz, wenn sie nicht durchlaufen wurden oder ihr Status unbekannt ist.
- Der Status von Linien, die an der Stromschiene beginnen, ist immer erfüllt ("1").
- Der Status von Linien am Beginn von Parallelzweigen ist immer erfüllt ("1").
- Der Status der Linie nach einem Element ist erfüllt, wenn sowohl der Status der Linie vor dem Element als auch der Status des Elements erfüllt ist.
- Der Status der Linie nach NOT ist erfüllt, wenn der Status der Linie vor NOT nicht erfüllt ist (und umgekehrt).
- Der Status der Linie **nach** der Zusammenführung mehrerer Linien ist erfüllt, wenn
 - sowohl der Status von mindestens einer Linie **vor** der Zusammenführung erfüllt ist,
 - als auch der Status der Linie vor der Verzweigung erfüllt ist.

Status von Parametern

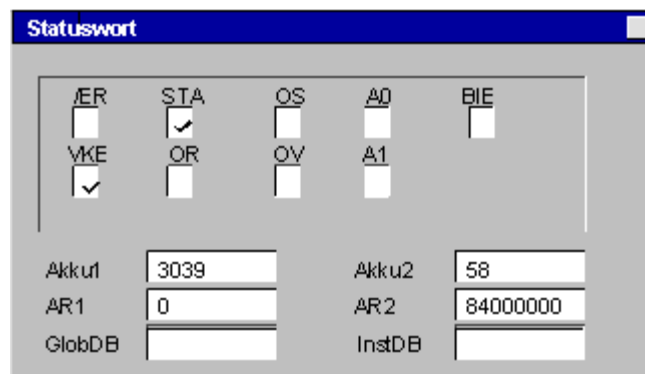
- Die Werte von Parametern **in fetter Schrift** sind aktuell.
- Die Werte von Parametern in magerer Schrift stammen aus einem vorhergehenden Zyklus; die Programmstelle wurde im aktuellen Zyklus nicht durchlaufen.

21.2 Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte

Beim Testen im Einzelschrittmodus können Sie

- Programme Anweisung für Anweisung (= Einzelschritt) bearbeiten
- Haltepunkte setzen

Die Funktion "Testen im Einzelschrittmodus" ist nicht bei allen Automatisierungssystemen realisiert (vgl. Dokumentation des entsprechenden Automatisierungssystems):



Voraussetzungen

- Der Betriebsmodus Testbetrieb muss eingestellt sein. Im Prozessbetrieb ist ein Testen im Einzelschrittmodus nicht möglich (siehe Menübefehl **Test > Betrieb**).
- Das Testen im Einzelschrittmodus ist nur in AWL möglich. Bei Bausteinen in KOP oder FUP müssen Sie über den Menübefehl **Ansicht > AWL** die Ansicht ändern.
- Der Baustein darf nicht geschützt sein.
- Der Baustein muss online geöffnet sein.
- Der geöffnete Baustein darf nicht im Editor verändert worden sein.

Anzahl der Haltepunkte

Die Anzahl der Haltepunkte ist variabel und ist abhängig von:

- Anzahl bereits gesetzter Haltepunkte
- Anzahl laufender Variablenstati
- Anzahl laufender Programmstati

Bitte entnehmen Sie der Dokumentation zu Ihrem Automatisierungssystem, ob es das Testen im Einzelschrittmodus unterstützt.

Die Menübefehle, mit denen Sie Haltepunkte setzen, aktivieren oder löschen können, finden Sie im Menü "Test". Außerdem haben Sie die Möglichkeit, diese Menübefehle über Symbole in der Haltepunktleiste auszuwählen. Um die Haltepunktleiste anzuzeigen, wählen Sie den Menübefehl **Ansicht > Haltepunktleiste**.

Zulässige Testfunktionen

- Variable beobachten/steuern
- Baugruppenzustand
- Betriebszustand



Gefahr

Achtung gefährlicher Anlagenzustand bei Betriebszustand "HALT".

21.3 Wissenswertes zum Betriebszustand HALT

Läuft das Programm auf einen Haltepunkt, so geht das Automatisierungssystem in den Betriebszustand "HALT".

Anzeige der Leuchtdioden (LEDs) im Betriebszustand "HALT"

- LED RUN blinkt
- LED STOP leuchtet

Programmbearbeitung im Betriebszustand "HALT"

- Im Betriebszustand "HALT" erfolgt keine Bearbeitung des S7-Codes, d. h. alle Ablafebeneen werden nicht weiterbearbeitet.
- Alle Zeiten werden eingefroren:
 - keine Bearbeitung der Zeitzellen
 - Anhalten aller Überwachungszeiten
 - Anhalten der Grundtakte der zeitgesteuerten Ebenen
- Echtzeituhr läuft
- Aus Sicherheitsgründen wird im Betriebszustand "HALT" immer die Ausgänge abgeschaltet (vgl. Ausgabebaugruppen "output disable")

Verhalten bei Netzausfall im Betriebszustand "HALT"

- Gepufferte Automatisierungssysteme gehen bei Netzausfall während des Betriebszustandes "HALT" und anschließender Netzwiederkehr in den Betriebszustand "STOP" und bleiben dort. Die CPU führt keinen automatischen Anlauf aus. Ausgehend vom STOP können Sie das weitere Vorgehen bestimmen (z. B. Haltepunkte setzen/rücksetzen, manuellen Anlauf durchführen).
- Nicht gepufferte Automatisierungssysteme haben kein "Gedächtnis" und führen deshalb bei Netzwiederkehr einen automatischen Neustart (Warmstart) aus, unabhängig vom vorherigem Betriebszustand.

21.4 Programmstatus von Datenbausteinen

Ab STEP 7 Version 5 ist es möglich, einen Datenbaustein in der Datensicht online zu beobachten. Die Anzeige kann sowohl von einem Online- als auch von einem Offline-Datenbaustein aktiviert werden. In beiden Fällen wird der Inhalt des Online-Datenbausteins aus dem Zielsystem dargestellt.

Der Datenbaustein darf vor dem Start des Programmstatus nicht geändert werden. Besteht ein struktureller Unterschied (Deklaration) zwischen dem Online- und dem Offline-Datenbaustein, so kann direkt per Rückfrage der Offline-Datenbaustein in das Zielsystem geladen werden.

Der Datenbaustein muss sich in der Ansicht "Datensicht" befinden, damit die Onlinewerte in der Spalte "Aktualwert" dargestellt werden können. Aktualisiert wird dabei nur der am Bildschirm sichtbare Teil des Datenbausteins. Während der Status aktiv ist kann nicht in die Deklarationssicht umgeschaltet werden.

Während der Aktualisierung ist der grüne Laufbalken in der Statuszeile des Datenbausteins sichtbar und der Betriebszustand wird eingeblendet.

Die Werte werden im Format des jeweiligen Datentyps ausgegeben, eine Änderung des Formats ist nicht möglich.

Nach der Beendigung des Programmstatus wird in der Aktualwertspalte wieder der vor dem Programmstatus gültige Inhalt dargestellt. Eine Übernahme der aktualisierten Online-Werte in den Offline-Datenbaustein ist nicht möglich.

Aktualisierung von Datentypen

Alle elementaren Datentypen werden sowohl in einem globalen DB als auch in allen Deklarationen (in/out/inout/stat) eines Instanzdatenbausteins aktualisiert.

Einige Datentypen können nicht aktualisiert werden. Bei aktiviertem Programmstatus werden diese Felder mit nicht aktualisierten Daten in der Spalte "Aktualwert" mit grauem Hintergrund dargestellt.

- Die zusammengesetzten Datentypen DATE_AND_TIME und STRING werden nicht aktualisiert.
- In den zusammengesetzten Datentypen ARRAY, STRUCT, UDT, FB, SFB werden nur diejenigen Elemente aktualisiert, die elementare Datentypen sind.
- In der INOUT-Deklaration eines Instanzdatenbausteins wird nur der Pointer auf den zusammengesetzten Datentyp dargestellt aber nicht dessen Elemente. Der Pointer wird nicht aktualisiert.
- Parametertypen werden nicht aktualisiert.

21.5 Festlegen der Anzeigen für den Programmstatus

Sie haben die Möglichkeit die Anzeige des Programmstatus in einem AWL-, FUP- oder KOP-Baustein selbst festzulegen.

Gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Extras > Einstellungen**.
2. Wählen Sie im Dialogfeld "Einstellungen" das Register "AWL" oder das Register "KOP/FUP".
3. Wählen Sie die gewünschten Optionen für den Test aus. Folgende Statusfelder können Sie sich anzeigen lassen

Bei Ankreuzen von ...	wird angezeigt:
Statusbit	Statusbit, also Bit 2 des Statusworts
Verknüpfungsergebnis	Bit 1 des Statusworts; gibt das Ergebnis einer Verknüpfung oder eines arithmetischen Vergleichs an.
Standardstatus	Inhalt von Akku 1.
Adressregister 1/2	Inhalt des entsprechenden Adressregisters bei registerindirekter Adressierung (bereichsintern oder -übergreifend)
Akku2	Inhalt von Akku 2
DB-Register 1/2	Inhalt des Datenbausteinregisters, des ersten bzw. zweiten geöffneten Datenbausteins.
Indirekt	Indirekte Speicherreferenz; Angabe des Pointers (Adresse), nicht des Adressinhalts; nur bei speicherindirekter Adressierung, nicht bei registerindirekter Adressierung möglich. Inhalt eines Zeitworts oder Zählerworts, wenn entsprechende Operationen in der Anweisung vorkommen
Statuswort	alle Statusbits des Statuswortes

21.6 Festlegen der Betriebsart für den Test

Vorgehensweise

1. Lassen Sie sich die eingestellte Testumgebung über den Menübefehl **Test > Betrieb** anzeigen.
2. Wählen Sie die gewünschte Betriebsart. Zur Verfügung stehen Testbetrieb und Prozessbetrieb.

Betriebsmodus	Erläuterung
Testbetrieb	Alle Testfunktionen sind ohne Einschränkung nutzbar. Größere Verlängerungen der CPU-Zykluszeit können auftreten, da z. B. der Status von Anweisungen in programmierten Schleifen bei jedem Durchlauf ermittelt wird.
Prozessbetrieb	Die Testfunktion Programmstatus wird eingeschränkt, um eine möglichst geringe Belastung der Zykluszeit zu gewährleisten. <ul style="list-style-type: none">• So sind z. B. keine Aufrufbedingungen erlaubt.• Die Statusanzeige einer programmierten Schleife wird an der Rücksprungstelle abgebrochen.• Die Testfunktionen "HALT" und schrittweise Programmausführung können nicht ausgeführt werden.

Hinweis

Falls der Betriebsmodus im Rahmen der CPU-Parametrierung festgelegt wurde, können Sie den Betriebsmodus nur durch Umparametrieren ändern. Ansonsten lässt er sich im angezeigten Dialogfeld umstellen.

22 Testen mit Simulationsprogramm (Optionspaket)

22.1 Testen mit dem Simulationsprogramm S7-PLCSIM (Optionspaket)

Mit dem Optionspaket Simulation können Sie Ihr Programm auf einem simulierten Automatisierungssystem, das auf Ihrem Computer bzw. Ihrem Programmiergerät (z. B. Power PG) existiert, ablaufen lassen und testen. Da die Simulation vollständig in der STEP 7-Software realisiert wird, benötigen Sie keine S7-Hardware (CPU oder Signalbaugruppen). Mit der simulierten S7-CPU können Sie Programme für S7-300 und S7-400 CPUs testen und Fehler beheben.

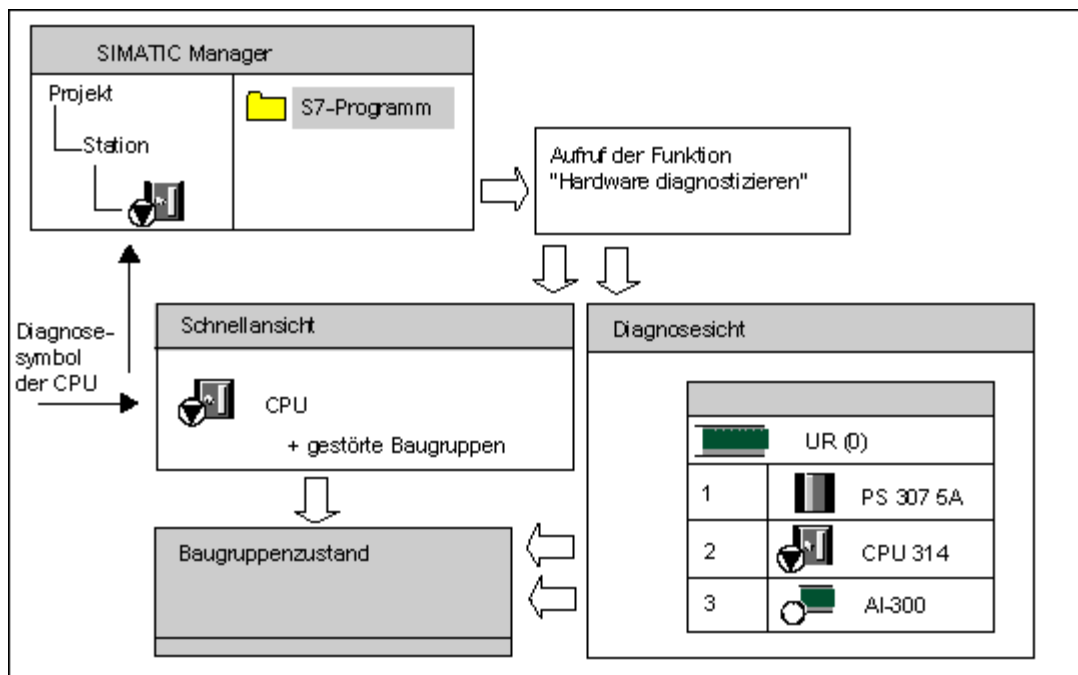
Die Applikation bietet Ihnen eine einfache Bedienoberfläche zum Überwachen und Ändern der verschiedenen Parameter, die in Ihrem Programm verwendet werden (z. B. zum Ein- und Ausschalten von Eingängen). Sie können außerdem die verschiedenen Applikationen der STEP 7-Software einsetzen, während Ihr Programm von der simulierten CPU bearbeitet wird. So können Sie beispielsweise mit der Variablen-tabelle Variablen überwachen und ändern.

23 Diagnose

23.1 Diagnose der Hardware und Fehlersuche

Das Vorliegen von Diagnoseinformation für eine Baugruppe können Sie anhand von Diagnosesymbolen erkennen. Diagnosesymbole zeigen den Status der zugehörigen Baugruppe und bei CPUs auch den Betriebszustand an.

Diagnosesymbole werden im Projektfenster in der Ansicht "Online" sowie nach Aufruf der Funktion "Hardware diagnostizieren" in der Schnellansicht (Voreinstellung) bzw. der Diagnosesicht angezeigt. Ausführliche Diagnoseinformationen werden im "Baugruppenzustand" angezeigt, den Sie durch Doppelklick auf ein Diagnosesymbol in der Schnellansicht bzw. Diagnosesicht aufrufen können.



Vorgehensweise zum Eingrenzen von Störungen

1. Öffnen Sie das Online-Fenster des Projekts mit dem Menübefehl **Ansicht > Online**.
2. Öffnen Sie alle Stationen, so dass die darin konfigurierten programmierbaren Baugruppen angezeigt werden.
3. Stellen Sie fest, bei welcher CPU ein Diagnosesymbol angezeigt wird, das auf einen Fehler oder eine Störung hinweist. Die Hilfeseite mit der Erklärung der Diagnosesymbole öffnen Sie mit der Taste F1.
4. Markieren Sie die Station, die Sie untersuchen möchten.
5. Wählen Sie den Menübefehl **Zielsystem > Diagnose/Einstellung > Baugruppenzustand**, um den Baugruppenzustand der CPU dieser Station anzuzeigen.
6. Wählen Sie den Menübefehl **Zielsystem > Diagnose/Einstellung > Hardware diagnostizieren**, um die "Schnellansicht" mit der CPU und den **gestörten** Baugruppen dieser Station anzuzeigen. Die Anzeige der Schnellansicht ist voreingestellt (Menübefehl **Extras > Einstellungen**, Register "Ansicht").
7. Markieren Sie in der Schnellansicht eine gestörte Baugruppe.
8. Klicken Sie auf die Schaltfläche "Baugruppenzustand", um die Informationen für diese Baugruppe zu erhalten.
9. Klicken Sie in der Schnellansicht die Schaltfläche "Station online öffnen", um die Diagnosesicht anzuzeigen. Die Diagnosesicht enthält **alle** Baugruppen der Station in der Steckplatzanordnung.
10. Doppelklicken Sie auf eine Baugruppe in der Diagnosesicht, um ihren Baugruppenzustand anzuzeigen. So erhalten Sie Informationen auch für diejenigen Baugruppen, die nicht gestört sind und deshalb in der Schnellansicht nicht angezeigt werden.

Sie müssen nicht unbedingt alle Schritte durchführen, sondern können abbrechen, sobald Sie die benötigte Diagnoseinformation erhalten haben.




23.2 Diagnosesymbole in der Online-Sicht

Diagnosesymbole werden im Online-Fenster des Projekts sowie im Fenster der Hardware-Konfiguration mit der Online-Ansicht von Konfigurationstabellen angezeigt.






Diagnosesymbole erleichtern Ihnen die Fehlersuche im Störfall. Sie können mit einem Blick auf ein Baugruppensymbol erkennen, ob dazu Diagnoseinformation vorliegt. Bei störungsfreiem Betrieb werden die Symbole der Baugruppentypen ohne zusätzliche Diagnosesymbole dargestellt.

Wenn zu einer Baugruppe Diagnoseinformation vorliegt, so wird zusätzlich zum Baugruppensymbol ein Diagnosesymbol angezeigt, oder das Baugruppensymbol wird kontrastvermindert dargestellt.


Diagnosesymbole für Baugruppen (Beispiel FM / CPU)

Symbol	Bedeutung
	Soll-Ist-Abweichung von der Projektierung: Die projektierte Baugruppe ist nicht vorhanden oder ein anderer Baugruppentyp ist gesteckt.
	Fehler: Baugruppe ist gestört. Mögliche Ursachen: Erkennung eines Diagnosealarms, eines Peripheriezugriffsfehlers oder einer Fehler-LED.
	Es ist keine Diagnose möglich, weil keine Online-Verbindung besteht oder die CPU keine Diagnoseinformation zur Baugruppe liefert (z. B. Stromversorgung, Submodule).

Diagnosesymbole für Betriebszustände (Beispiel CPU)

Symbol	Betriebszustand
	ANLAUF
	STOP
	STOP ausgelöst durch Betriebszustand "STOP" einer anderen CPU im Multicomputing-Betrieb
	RUN
	HALT

Diagnosesymbol für das Forcen

Symbol	Betriebszustand
	<p>Auf dieser Baugruppe wird ein Forcen von Variablen durchgeführt, d. h. Variable im Anwenderprogramm der Baugruppe sind mit festen Werten vorbelegt, die vom Programm nicht geändert werden können.</p> <p>Die Kennzeichnung für Forcen kann auch im Zusammenhang mit anderen Symbolen auftreten (hier mit dem Symbol für den Betriebszustand "RUN").</p>

Aktualisieren der Anzeige von Diagnosesymbolen

Das betreffende Fenster muss aktiviert sein.

- Drücken Sie die Funktionstaste F5 oder
- wählen Sie im Fenster den Menübefehl **Ansicht > Aktualisieren**.

23.3 Hardware diagnostizieren: Schnellansicht

23.3.1 Aufruf der Schnellansicht

Die Schnellansicht bietet Ihnen einen schnellen Einstieg in "Hardware diagnostizieren" mit reduzierter Information im Vergleich zu den ausführlicheren Anzeigen in der Diagnosesicht von HWKonfig. Die Schnellansicht wird gemäß Voreinstellung beim Aufruf der Funktion "Hardware diagnostizieren" angezeigt.

Anzeigen der Schnellansicht

Die Funktion rufen Sie aus dem SIMATIC Manager mit dem Menübefehl **Zielsystem > Diagnose/Einstellung > Hardware diagnostizieren** auf.

Der Menübefehl kann wie folgt benutzt werden:

- im Online-Fenster des Projekts, wenn eine Baugruppe oder ein S7-/M7-Programm markiert ist.
- im Fenster "Erreichbare Teilnehmer", wenn ein Teilnehmer ("MPI=...") markiert ist und dieser Eintrag zu einer CPU-Baugruppe gehört.

Aus den aufgeblendeten Konfigurationstabellen können Sie dann Baugruppen auswählen, deren Baugruppenzustand angezeigt werden soll.

23.3.2 Auskunftsfunktionen der Schnellansicht

In der Schnellansicht werden folgende Informationen angezeigt:

- Daten zur Online-Verbindung zur CPU
- Diagnosesymbol zur CPU
- Diagnosesymbole zu Baugruppen, für die die CPU eine Störung erkannt hat (z. B. Diagnosealarm, Peripheriezugriffsfehler).
- Baugruppentyp und Adresse der Baugruppe (Baugruppenträger, Steckplatz, DP Mastersystem mit Stationsnummer)

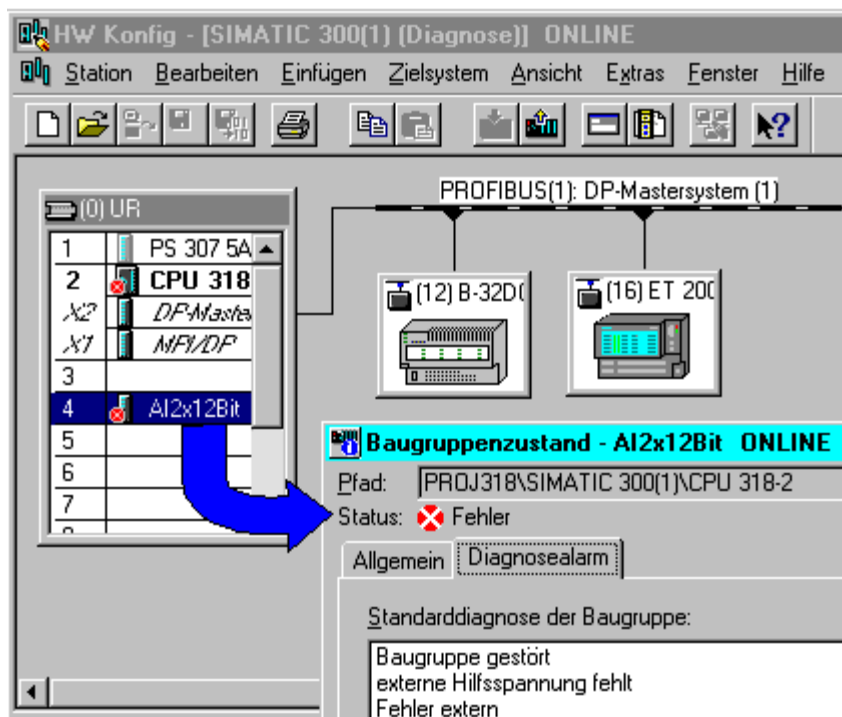
Weitere Diagnosemöglichkeiten in der Schnellansicht

- **Anzeige des Baugruppenzustands**
Mit der Schaltfläche "Baugruppenzustand" rufen Sie dieses Dialogfeld auf. Dort erhalten Sie je nach Diagnosefähigkeit der Baugruppe detaillierte Diagnoseinformationen zur markierten Baugruppe. Insbesondere können Sie sich über den Baugruppenzustand der CPU die Einträge im Diagnosepuffer anzeigen lassen.
- **Anzeige der Diagnosesicht**
Mit der Schaltfläche "Station online öffnen" rufen Sie das Dialogfeld auf, das im Gegensatz zur Schnellansicht eine grafische Übersicht über die gesamte Station sowie Projektierinformationen enthält. Es wird auf die Baugruppe positioniert, die in der Liste "CPU / Gestörte Baugruppen" markiert ist.

23.4 Hardware diagnostizieren: Diagnosesicht

23.4.1 Aufruf der Diagnosesicht von HW Konfig

Über diesen Weg können Sie sich den Registerdialog "Baugruppenzustand" für alle Baugruppen im Baugruppenträger anzeigen lassen. Die Diagnosesicht (Konfigurationstabelle) zeigt den tatsächlichen Aufbau einer Station auf Ebene der Baugruppenträger und DP-Stationen mit Ihren Baugruppen.



Hinweis

- Ist die Konfigurationstabelle bereits offline geöffnet, so können Sie über den Menübefehl **Station > Online öffnen** ebenfalls die Online-Ansicht der Konfigurationstabellen erhalten.
- Je nach Diagnosefähigkeit der Baugruppe wird eine unterschiedliche Anzahl von Registern im Registerdialog "Baugruppenzustand" angezeigt.
- Im Fenster "Erreichbare Teilnehmer" sind immer nur die Baugruppen mit eigener Teilnehmeradresse (Ethernet-, MPI- oder PROFIBUS-Adresse) sichtbar.

Aufruf im SIMATIC Manager aus der Ansicht ONLINE eines Projekts

1. Stellen Sie in der Projektansicht des SIMATIC Managers mit dem Menübefehl **Ansicht > Online** eine Online-Verbindung zum Zielsystem her.
2. Wählen Sie eine Station aus und öffnen Sie diese durch Doppelklick.
3. Öffnen Sie anschließend das darin enthaltene Objekt "Hardware". Die Diagnosesicht wird geöffnet.

Nun können Sie eine Baugruppe markieren und deren Baugruppenzustand mit dem Menübefehl **Zielsystem > Diagnose/Einstellung > Baugruppenzustand** aufrufen.

Aufruf im SIMATIC Manager aus der Ansicht Offline eines Projekts

Gehen Sie folgendermaßen vor:

1. Wählen Sie aus der Projektansicht des SIMATIC Managers eine Station aus und öffnen Sie diese durch Doppelklick.
2. Öffnen Sie anschließend das darin enthaltene Objekt "Hardware". Die Konfigurationstabelle wird geöffnet.
3. Wählen Sie den Menübefehl **Station > Online öffnen**.
4. Die Diagnosesicht von HW Konfig wird geöffnet mit der Stationskonfiguration, wie sie aus den Baugruppen (z. B. CPU) ermittelt wurde. Die Zustände der Baugruppen werden durch Symbole angezeigt. Die Bedeutung der verschiedenen Symbole ersehen Sie aus der Online-Hilfe. Falls Baugruppen gestört sind oder konfigurierte Baugruppen fehlen, werden sie in einem separaten Dialogfeld aufgelistet. Von diesem Dialogfeld aus können Sie unmittelbar zu einer der angezeigten Baugruppen navigieren (Schaltfläche "Gehe zu").
5. Doppelklicken Sie auf das Symbol der Baugruppe, deren Baugruppenzustand Sie interessiert. Ein Dialogfeld mit Registern (abhängig vom Baugruppentyp) ermöglicht Ihnen eine detaillierte Analyse des Baugruppenzustands.

Aufruf im SIMATIC Manager aus dem Fenster "Erreichbare Teilnehmer"

Gehen Sie folgendermaßen vor:

1. Öffnen Sie im SIMATIC Manager mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** das Fenster "Erreichbare Teilnehmer".
2. Wählen Sie im Fenster "Erreichbare Teilnehmer" einen Teilnehmer aus.
3. Wählen Sie anschließend den Menübefehl **Zielsystem > Diagnose/Einstellung > Hardware diagnostizieren**.

Hinweis

Im Fenster "Erreichbare Teilnehmer" sind immer nur die Baugruppen mit eigener Teilnehmeradresse (Ethernet-, MPI- oder PROFIBUS-Adresse) sichtbar.

23.4.2 Auskunftsfunktionen der Diagnosesicht

In der Diagnosesicht wird im Gegensatz zur Schnellansicht die gesamte online erreichbare Stationskonfiguration angezeigt. Diese umfasst:

- Konfiguration der Baugruppenträger
- Diagnosesymbole zu **allen** konfigurierten Baugruppen
Daraus ist der Status der jeweiligen Baugruppe und bei CPU-Baugruppen zusätzlich der Betriebszustand ablesbar.
- Baugruppentyp, Bestellnummer und Adressangaben, Kommentare zur Konfigurierung

Weitere Diagnosemöglichkeiten in der Diagnosesicht

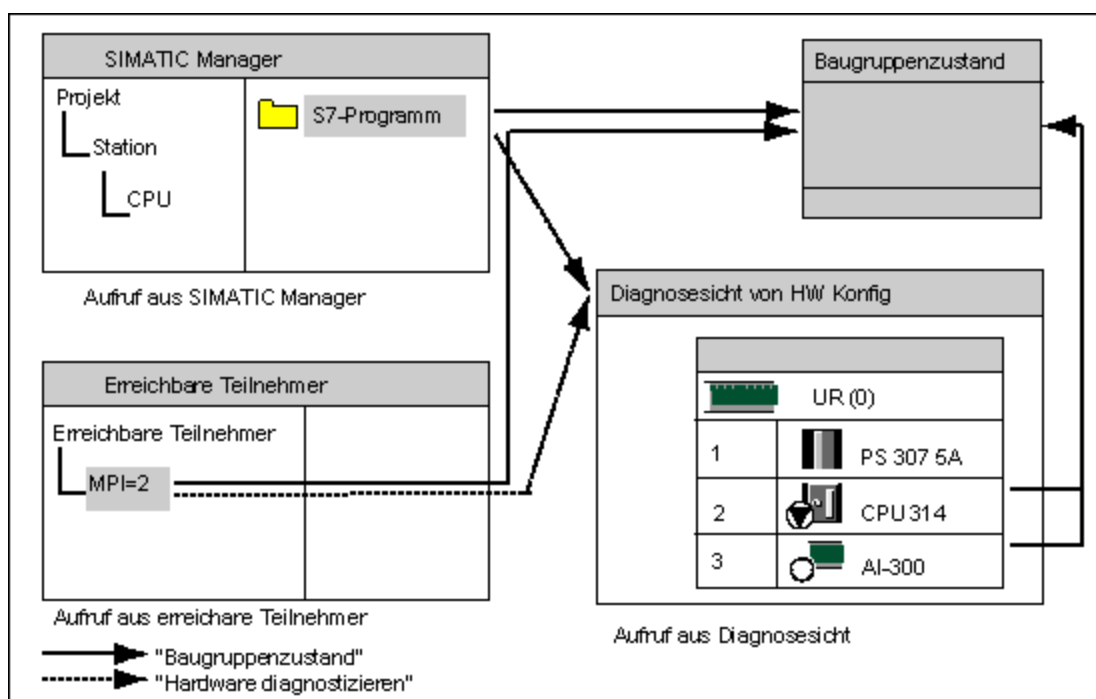
Durch Doppelklick auf eine Baugruppe können Sie sich den Baugruppenzustand zu dieser Baugruppe anzeigen lassen.

23.5 Baugruppenzustand

23.5.1 Aufrufmöglichkeiten des Baugruppenzustands

Den Registerdialog "Baugruppenzustand" können Sie sich von unterschiedlichen Ausgangspunkten aus anzeigen lassen. Die folgenden Vorgehensweisen sind exemplarisch und geben häufige Anwendungsfälle wieder:

- Aufruf im SIMATIC Manager aus einem Fenster mit der Projektansicht "Online" oder "Offline"
- Aufruf im SIMATIC Manager aus einem Fenster "Erreichbare Teilnehmer"
- Aufruf aus der Diagnosesicht von HW Konfig



Um den Zustand einer **Baugruppe mit eigener Teilnehmeradresse** abrufen zu können, benötigen Sie eine Online-Verbindung zum Zielsystem. Diese stellen Sie über die Online-Ansicht eines Projekts oder über das Fenster "Erreichbare Teilnehmer" her.

23.5.2 Auskunftsfunktionen des Baugruppenzustands

Die Auskunftsfunktionen stehen auf der gleichnamigen Registerseite des Dialogfelds "Baugruppenzustand" zur Verfügung. Im konkreten Anwendungsfall werden nur die Registerseiten angezeigt, die für die ausgewählte Baugruppe sinnvoll sind.

Auskunftsfunktion	Information	Einsatz
Allgemein	Identifikationsdaten der ausgewählten Baugruppe, z. B. Typ, Bestellnummer, Ausgabestand, Status, Steckplatz im Baugruppenträger.	Die Online-Auskunft aus der gesteckten Baugruppe kann mit den Daten der projektierten Baugruppe verglichen werden.
Diagnosepuffer	Übersicht über Ereignisse im Diagnosepuffer sowie detaillierte Information zum ausgewählten Ereignis.	Zum Auswerten der STOP-Ursache einer CPU und zum Auswerten der vorangegangenen Ereignisse auf der ausgewählten Baugruppe. Fehler im System können durch den Diagnosepuffer auch nach längerer Zeit noch ausgewertet werden, um die Ursache für einen STOP festzustellen oder um das Auftreten einzelner Diagnoseereignisse zurückzuverfolgen und zuordnen zu können.
Diagnosealarm	Diagnosedaten der ausgewählten Baugruppe.	Zur Ermittlung der Ursache einer Baugruppenstörung.
DP-Slave Diagnose	Diagnosedaten des ausgewählten DP-Slave (nach EN 50170).	Zur Ermittlung der Ursache eines Fehlers eines DP-Slaves.
Speicher	Speicherausbau, Aktuelle Auslastung von Arbeitsspeicher, Ladespeicher und Remanenzspeicher Daten der gewählten CPU oder M7-FM.	Vor dem Übertragen neuer oder erweiterter Bausteine auf eine CPU um zu prüfen, ob genügend Ladespeicher in dieser CPU/FM zur Verfügung steht sowie zum Komprimieren des Speicherinhalts.
Zykluszeit	Dauer des längsten, kürzesten und letzten Zyklus der gewählten CPU oder M7-FM.	Zur Kontrolle der parametrisierten Mindestzykluszeit sowie der maximalen und aktuellen Zykluszeit.
Zeitsystem	Aktuelle Uhrzeit, Betriebsstunden und Informationen zur Uhrensynchronisation (Synchronisationsintervalle).	Zur Anzeige von Uhrzeit und Datum einer Baugruppe und zur Kontrolle der Uhrzeitsynchronisation.
Leistungsdaten	Operandenbereiche und die verfügbaren Bausteine der angewählten (CPU-/FM-) Baugruppe.	Vor und während der Erstellung eines Anwenderprogramms und zur Überprüfung, ob die CPU geeignete Voraussetzungen zum Ablauf eines Anwenderprogramms besitzt, z. B. bezüglich Größe des Prozessabbilds.
	Anzeige aller Bausteinarten, die im Funktionsumfang der angewählten Baugruppe verfügbar sind. Liste der OBs, SFBs und SFCs, die in dieser Baugruppe eingesetzt werden können.	Zur Prüfung, welche Standard-Bausteine Ihr Anwenderprogramm enthalten bzw. aufrufen darf, um auf der angewählten CPU ablaufen zu können.

Auskunftsfunktion	Information	Einsatz
Kommunikation	Baudraten, die Verbindungsübersicht, die Kommunikationsbelastung sowie die maximale Größe der Telegramme auf dem K-Bus der angewählten Baugruppe.	Um festzustellen, wieviel und welche Verbindungen der CPU bzw. M7-FM möglich bzw. belegt sind.
Stacks	Register Stacks : kann nur im Betriebszustand STOP oder HALT aufgeblättert werden. Der B-Stack der angewählten Baugruppe wird angezeigt. Sie können außerdem U-Stack, L-Stack und Klammer-Stack auslesen und an die Fehlerstelle des unterbrechenden Bausteins springen.	Zur Feststellung der Ursache eines Übergangs in STOP und zur Korrektur eines Bausteins.

Angezeigte Zusatzinformation

Für jede Registerseite werden folgende Informationen dargestellt:

- Online-Pfad zur ausgewählten Baugruppe
- Betriebszustand der zugehörigen CPU (z. B. RUN, STOP)
- Status der ausgewählten Baugruppe (z. B. Fehler, OK)
- Betriebszustand der ausgewählten Baugruppe (z. B. RUN, STOP), soweit diese über einen eigenen Betriebszustand verfügt (z. B. CP342-5).

Der Betriebszustand der CPU selbst und der Status der ausgewählten Baugruppe können nicht angezeigt werden, wenn der Baugruppenzustand einer Nicht-CPU-Baugruppe aus dem Fenster "Erreichbare Teilnehmer" gestartet wurde.

Gleichzeitig mehrere Baugruppen anzeigen

Sie können den Baugruppenzustand verschiedener Baugruppen abrufen und gleichzeitig darstellen. Dazu müssen Sie in den entsprechenden Baugruppenkontext wechseln, eine andere Baugruppe auswählen und den Baugruppenzustand dazu aufrufen. Es wird Ihnen dann ein weiterer Registerdialog angezeigt. Zu einer Baugruppe kann jedoch nur ein Registerdialog geöffnet werden.

Aktualisieren der Anzeigen im Baugruppenzustand

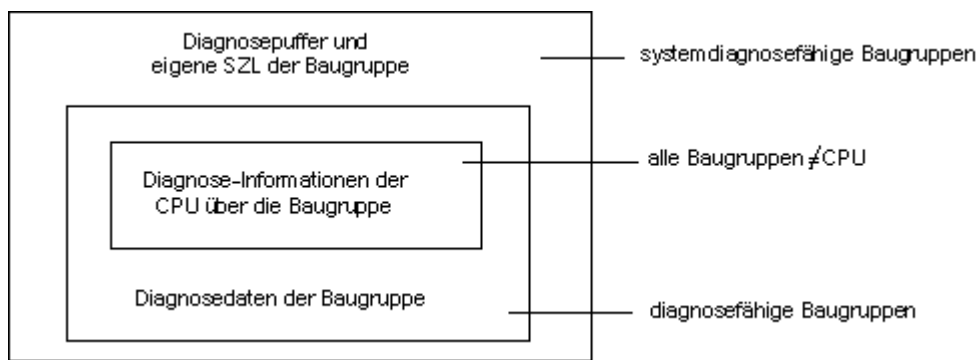
Bei jedem Wechsel auf eine andere Registerseite des Dialogfelds "Baugruppenzustand" werden die Daten neu von der Baugruppe gelesen. Während der Anzeige einer Seite wird der Inhalt jedoch nicht automatisch aktualisiert. Durch Klicken auf die Schaltfläche "Aktualisieren" werden die Daten erneut von der Baugruppe gelesen, ohne die Registerseite zu wechseln .

23.5.3 Baugruppentyp-abhängiger Umfang der Auskunftsfunktionen im Baugruppenzustand

Der Umfang der Informationen, die ausgewertet und angezeigt werden können, ist abhängig von

- der angewählten Baugruppe und
- aus welcher Ansicht Sie den Baugruppenzustand abfragen.
Voller Informationsumfang ist beim Aufruf aus der Online-Ansicht der Konfigurationstabellen oder aus dem Fenster des Projekts vorhanden.
Eingeschränkter Informationsumfang ist beim Aufruf aus dem Fenster "Erreichbare Teilnehmer" möglich.

Je nach Umfang der Informationen werden die Baugruppen eingeteilt in systemdiagnosefähige, diagnosefähige oder nicht diagnosefähige Baugruppen. Diesen Zusammenhang veranschaulicht die folgende Grafik:



- Systemdiagnosefähig sind z. B. die Baugruppen FM351 und FM354.
- Diagnosefähig sind die meisten analogen SM-Baugruppen.
- Nicht diagnosefähig sind die meisten digitalen SM-Baugruppen.

Angezeigte Registerseiten

Die Tabelle zeigt, welche Registerseiten bei den einzelnen Baugruppentypen im Dialogfeld "Baugruppenzustand" vorhanden sind.

Registerseite	CPU oder M7-FM	Systemdiagnosefähige Baugruppe	Diagnosefähige Baugruppe	Nicht diagnosefähige Baugruppe	DP-Slave
Allgemein	ja	ja	ja	ja	ja
Diagnosepuffer	ja	ja	-	-	-
Diagnosealarm	-	ja	ja	-	ja
Speicher	ja	-	-	-	-
Zykluszeit	ja	-	-	-	-
Zeitsystem	ja	-	-	-	-
Leistungsdaten	ja	-	-	-	-
Stacks	ja	-	-	-	-
Kommunikation	ja	-	-	-	-
DP-Slave Diagnose	-	-	-	-	ja
H-Zustand ¹⁾	ja	-	-	-	-

¹⁾ nur bei CPUs in H-Systemen

Neben der Information auf den Registerseiten wird der Betriebszustand bei Baugruppen mit Betriebszustand angezeigt. Beim Aufruf aus den Konfigurationstabellen online wird der Status angegeben, den diese Baugruppe aus Sicht der CPU besitzt (z. B. OK, Fehler, Baugruppe nicht vorhanden).

23.5.4 Baugruppenzustand von PA-Feldgeräten und DP-Slaves hinter Y-Link anzeigen

Ab STEP 7 V5.1, Servicepack 3 ist auch ein Auswerten des Baugruppenzustands von DP-Slaves und PA-Feldgeräten "hinter" einem DP/PA-Link (IM 157) möglich.

Das betrifft folgende Konfigurationen:

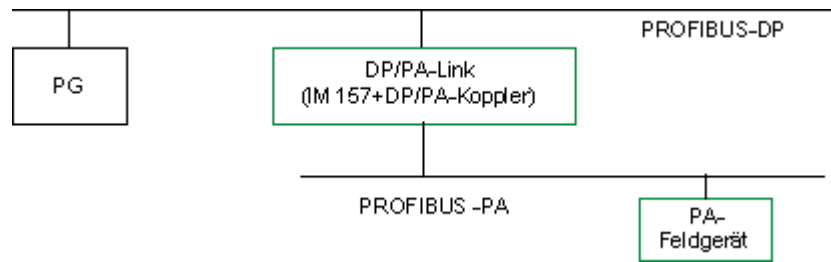
- IM 157 mit DP/PA-Kopplern zur Anbindung des PROFIBUS-PA
- IM 157 als redundant aufbaubare Anschaltungsbaugruppe zur Anbindung eines nicht redundanten PROFIBUS-DP ("Y-Link")

Bei diesen Konfigurationen ist das PG am selben PROFIBUS-Subnetz angeschlossen wie auch das DP/PA-Link.

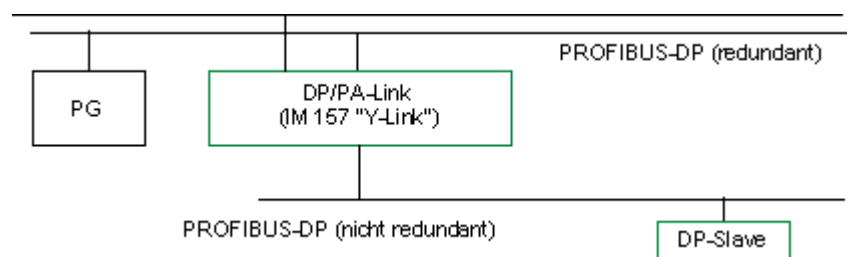
Daneben gibt es noch eine Konfigurationsmöglichkeit, bei der das PG am Industrial Ethernet angeschlossen ist und eine S7-400-Station auf das PROFIBUS-Subnetz routet.

Voraussetzungen dafür sind im Bild unten beschrieben.

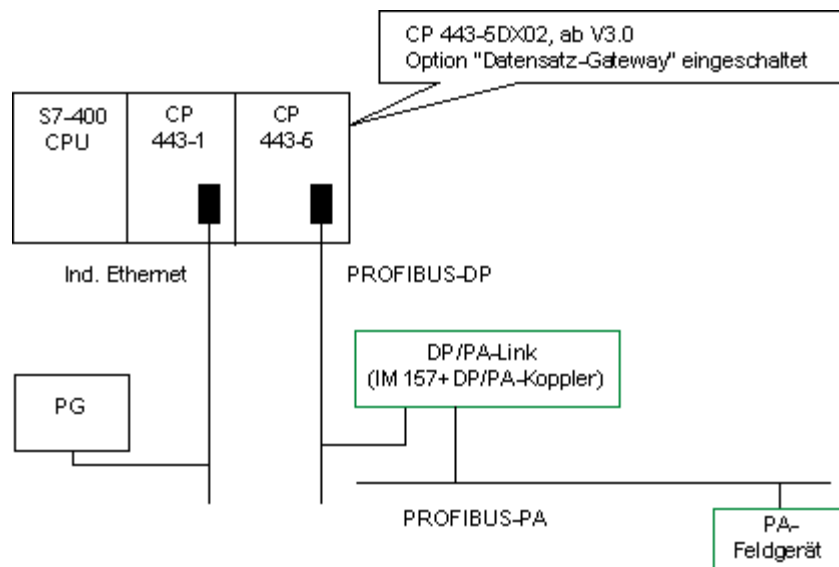
IM 157 mit DP/PA-Kopplern zur Anbindung des PROFIBUS-PA



IM 157 als Y-Link



PG am Industrial Ethernet



23.6 Diagnose im Betriebszustand STOP

23.6.1 Prinzipielles Vorgehen zum Ermitteln einer STOP-Ursache

Um zu ermitteln, warum die CPU in den Betriebszustand "STOP" gewechselt hat, gehen Sie wie folgt vor:

1. Markieren Sie die CPU, die in den Betriebszustand STOP gegangen ist.
2. Wählen Sie den Menübefehl **Zielsystem > Diagnose/Einstellung > Baugruppenzustand**.
3. Wählen Sie das Register "Diagnosepuffer".
4. Anhand der letzten Einträge können Sie die STOP-Ursache ermitteln.

Bei Programmierfehler:

1. Der Eintrag "STOP durch nicht geladenen Programmierfehler-OB" bedeutet beispielsweise, dass die CPU einen Programmierfehler erkannt und danach versucht hat, den (nicht vorhandenen) OB zur Behandlung des Programmierfehlers zu starten. Auf den eigentlichen Programmierfehler weist der vorhergehende Eintrag hin.
2. Markieren Sie die Meldung zum Programmierfehler.
3. Betätigen Sie die Schaltfläche "Baustein öffnen".
4. Wählen Sie die Registerseite "Stacks".

23.6.2 Stack-Inhalte im Betriebszustand STOP

Durch Auswertung des Diagnosepuffers und der Stack-Inhalte können Sie die Ursache für die Störung im Ablauf eines Anwenderprogramms ermitteln.

Wenn die CPU z. B. durch einen Programmierfehler oder durch den Stoppbefehl in den Betriebszustand "STOP" gewechselt hat, wird im Register "Stacks" des Baugruppenzustands der Baustein-Stack angezeigt. Weitere Stack-Inhalte können Sie über die Schaltflächen "U-Stack", "L-Stack" und "Klammer-Stack" anzeigen. Die Stack-Inhalte geben Ihnen Informationen darüber, welcher Befehl in welchem Baustein zu dem STOP bei der CPU geführt hat.

B-Stack-Inhalt

Im B-Stack werden alle Bausteine aufgeführt, die bis zum Übergang in den STOP-Zustand aufgerufen wurden und noch nicht zu Ende bearbeitet worden sind.

U-Stack-Inhalt

Durch Klicken auf die Schaltfläche "U-Stack" werden Ihnen Daten zur Unterbrechungsstelle angezeigt. Der Unterbrechungsstack (U-Stack) enthält Daten bzw. Zustände, die zum Zeitpunkt der Unterbrechung gültig waren, z. B.

- Akku-Inhalte und Registerinhalte
- Offene DBs und deren Größe
- Inhalt des Statusworts
- Prioritätsklasse
- Unterbrochenen Baustein
- Baustein, in dem der Programmablauf nach der Unterbrechung fortgesetzt würde.

L-Stack-Inhalt

Für jeden im B-Stack aufgelisteten Baustein können Sie sich durch Selektieren des Bausteins und Klicken auf die Schaltfläche "L-Stack" die zugehörigen Lokaldaten anzeigen lassen.

Der Lokaldaten-Stack (L-Stack) enthält die Lokaldatenwerte der Bausteine, mit denen das Anwenderprogramm bis zum Zeitpunkt der Unterbrechung gearbeitet hat.

Für die Interpretation und Auswertung der angezeigten Lokaldaten sind sehr gute Systemkenntnisse erforderlich. Der vordere Teil der angezeigten Daten entspricht den temporären Variablen des Bausteins.

Klammerstack-Inhalt

Wenn Sie die Schaltfläche "Klammer-Stack" betätigen, wird der Inhalt des Klammer-Stacks an der Unterbrechungsstelle dargestellt.

Der Klammer-Stack ist ein Speicherbereich, den die Verknüpfungsoperationen **U**(, **UN**(, **O**(, **ON**(, **X**(und **XN**(verwenden.

Die Schaltfläche ist nur aktiv, wenn zum Zeitpunkt der Unterbrechung noch offene Klammerausdrücke existierten.

23.7 Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern

Das Register "Zykluszeit" des Baugruppenzustands gibt Ihnen Auskunft über die Zykluszeiten des Anwenderprogramms.

Wenn die Dauer des längsten Zyklus dicht an der Überwachungszeit liegt, besteht die Gefahr, dass Schwankungen in der Zykluszeit zu einem Zeitfehler führen. Dies können Sie vermeiden, indem Sie die maximale Zykluszeit des Anwenderprogramms vergrößern.

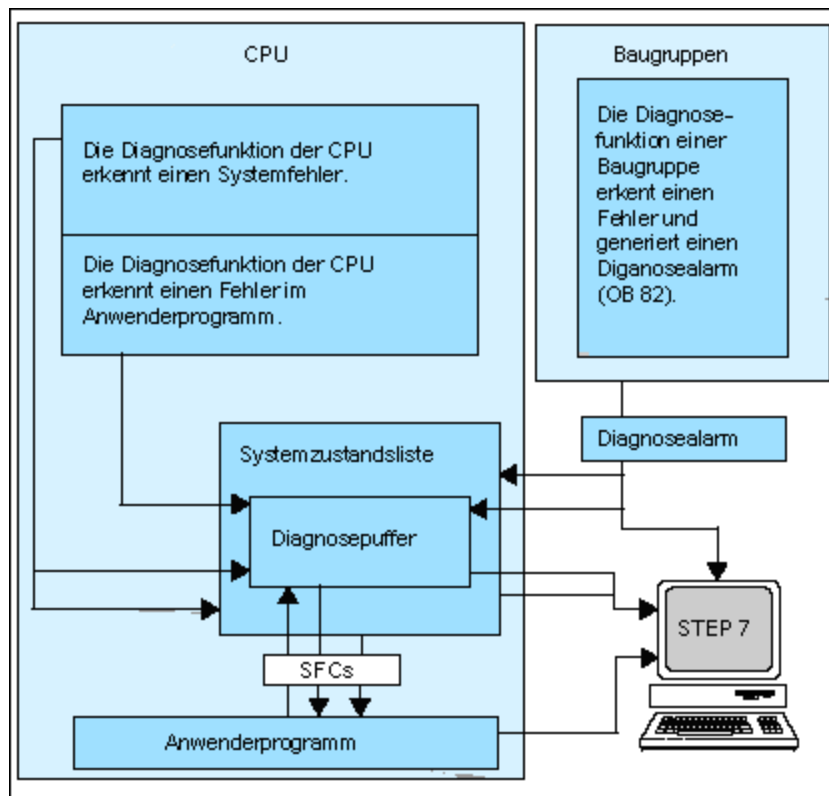
Wenn die Zyklusdauer die parametrisierte Mindestzykluszeit unterschreitet, wird der Zyklus automatisch von der CPU/FM auf die parametrisierte Mindestzykluszeit verlängert. Im Falle einer CPU wird während dieser Verlängerung der Hintergrund-OB (OB 90) bearbeitet, falls er geladen ist.

Einstellen der Zykluszeit

Die maximale und minimale Zykluszeit können Sie bei der Konfigurierung der Hardware festlegen. Dazu markieren Sie in der Offline-Ansicht der Konfigurationstabelle die CPU/FM aus und wählen über die rechte Maustaste den Menübefehl **Objekteigenschaften**, um ihre Eigenschaften zu definieren. Im Register "Zyklus/Taktmerker" können Sie die Werte einstellen.

23.8 Übermittlung von Diagnoseinformationen

Das folgende Bild zeigt die Übermittlung von Diagnoseinformationen bei SIMATIC S7.



Auslesen der Diagnoseinformation

Sie können die Diagnoseeinträge im Anwenderprogramm mit der SFC 51 RDSYSST auslesen oder sich die Diagnosemeldungen mit STEP 7 im Klartext anzeigen lassen.

Sie geben Auskunft darüber:

- wo und wann der Fehler aufgetreten ist
- zu welcher Art von Diagnoseereignissen der Eintrag gehört (anwenderdefiniertes Diagnoseereignis, synchroner/asynchroner Fehler, Betriebszustandswechsel).

Erzeugung von Leittechniksammelmeldungen

Die CPU trägt die Ereignisse der Standarddiagnose und der erweiterten Diagnose in den Diagnosepuffer ein. Darüber hinaus erzeugt sie bei den Standard-Diagnoseereignissen eine Leittechniksammelmeldung, falls die folgenden Voraussetzungen erfüllt sind:

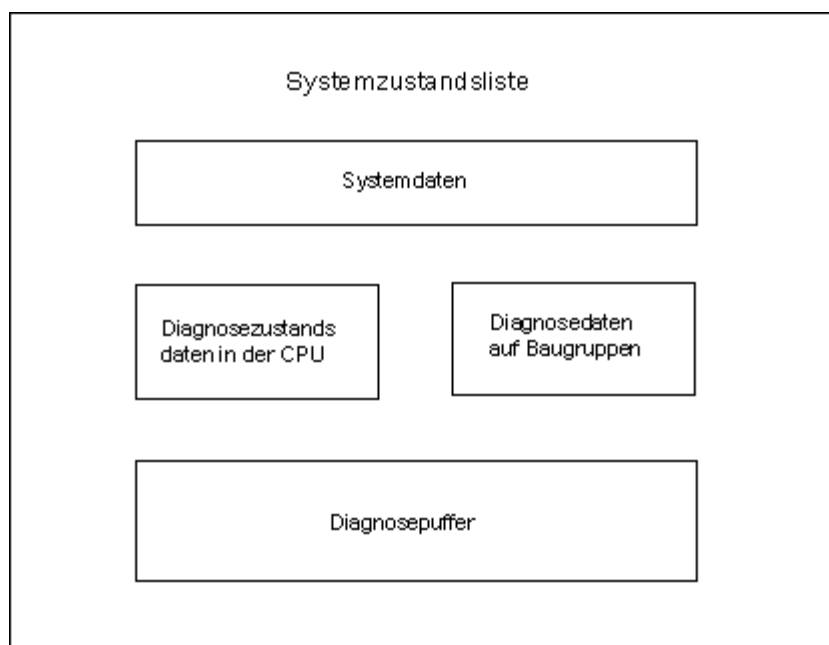
- Sie haben mit STEP 7 vorgegeben, dass Leittechniksammelmeldungen erzeugt werden sollen.
- Mindestens ein Anzeigegerät hat sich bei der CPU für Leittechniksammelmeldungen angemeldet.
- Eine Leittechniksammelmeldung wird erst dann gebildet, wenn noch keine Leittechniksammelmeldung der zugehörigen Klasse (es gibt sieben Klassen) vorliegt.
- Pro Klasse kann eine Leittechniksammelmeldung gebildet werden.

23.8.1 Systemzustandsliste SZL

Die Systemzustandsliste SZL beschreibt den aktuellen Zustand des Automatisierungssystems: sie gibt einen Überblick über den Ausbau, die aktuelle Parametrierung, die aktuellen Zustände und Abläufe in der CPU und den zugeordneten Baugruppen.

Die Daten der SZL können nur gelesen, nicht geändert werden. Sie ist eine virtuelle Liste, die nur auf Anforderung zusammengestellt wird.

Die Informationen, die Sie über die SZL ausgeben können, lassen sich in vier Bereiche aufteilen:



Systemzustandsliste lesen

Es gibt zwei Möglichkeiten, die Informationen der Systemzustandsliste SZL zu lesen:

- implizit über STEP 7-Menübefehle vom Programmiergerät aus (z. B. Speicherausbau, statische CPU-Daten, Diagnosepuffer, Zustandsanzeigen).
- explizit über die Systemfunktion SFC 51 RDSYSST vom Anwenderprogramm aus mittels Angabe der gewünschten Teillistennummer (siehe Baustein-Hilfen).

Systemdaten der SZL

Systemdaten sind feste oder parametrisierte Kenndaten einer CPU. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können.

Bereich	Information
Baugruppen-Identifikation	Bestellnummer, Typkennung und Ausgabestand der Baugruppe
CPU-Merkmale	Zeitsystem, Systemverhalten (z. B. Multicomputing) und Sprachbeschreibung der CPU
Speicherbereiche	Speicherausbau der Baugruppe (Größe des Arbeitsspeichers)
Systembereiche	Systemspeicher der Baugruppe (z. B. Anzahl der Merker, Zeiten, Zähler, Speichertyp)
Bausteintypen	Welche Bausteintypen (OB, DB, SDB, FC, FB) in Baugruppe vorhanden, wieviel Bausteine eines Typs maximal vorhanden und maximale Größe eines Bausteintyps
Alarm-Fehlerzuordnung	Zuordnung Alarme/Fehler zu OBs
Alarmstatus	Aktuelle Alarmbearbeitung/-generierung
Status der Prioritätsklassen	Welcher OB in Bearbeitung, welche Prioritätsklasse durch Parametrierung gesperrt
Betriebszustand und Betriebszustandsübergang	Welche Betriebszustände möglich, letzter Übergang, aktueller Betriebszustand

Diagnosezustandsdaten in der CPU

Diagnosezustandsdaten beschreiben den aktuellen Zustand der Komponenten, die durch die Systemdiagnose überwacht werden. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können.

Bereich	Information
Kommunikationszustandsdaten	Aktuell im System eingestellte Kommunikationsfunktionen
Diagnoseteilnehmer	An der CPU angemeldete diagnosefähige Baugruppen
Startinformationsliste des OB	Startinformationen zu den OBs der CPU
Startereignisliste	Startereignisse und Prioritätsklassen der OBs
Baugruppenzustandsinformation	Zustandsinformationen aller gesteckten, gestörten, prozessalarmerzeugenden, zugeordneten Baugruppen

Diagnosedaten auf Baugruppen

Außer der CPU gibt es noch andere diagnosefähige Baugruppen (SMs, CPs, FMs), deren Diagnosedaten in die Systemzustandsliste eingetragen werden. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können.

Bereich	Information
Baugruppendiagnoseinfo	Baugruppenanfangsadresse, interne/externe Fehler, Kanalfehler, Parameterfehler (4 Byte)
Baugruppendiagnosedaten	Alle Diagnosedaten einer bestimmten Baugruppe

23.8.2 Eigene Diagnosemeldungen senden

Sie können die standardmäßige Systemdiagnose von SIMATIC S7 außerdem erweitern, indem Sie über die Systemfunktion SFC 52 WR_USMSG

- anwenderdefinierte Diagnoseereignisse (z. B. Informationen zum Ablauf des Anwenderprogramms) in den Diagnosepuffer eintragen
- anwenderdefinierte Diagnosemeldungen an angemeldete Teilnehmer (Beobachtungsgeräte wie PG, OP, TD) senden.

Anwenderdefinierte Diagnoseereignisse

Die Diagnoseereignisse sind in Ereignisklassen von 1 bis F eingeteilt. Die anwenderdefinierten Diagnoseereignisse gehören zu den Ereignisklassen 8 bis B. Sie lassen sich in zwei Gruppen unterteilen:

- Die Ereignisklassen 8 und 9 umfassen Ereignisse mit festgelegter Nummer und vorgefertigtem Text, den Sie anhand der Nummer abrufen können.
- Die Ereignisklassen A und B umfassen Ereignisse mit frei wählbarer Nummer (A000 bis A0FF, B000 bis B0FF) und frei wählbarem Text.

Diagnosemeldungen an Teilnehmer senden

Zusätzlich zum Eintrag eines anwenderdefinierten Diagnoseereignisses in den Diagnosepuffer können Sie mit der SFC 52 WR_USMSG Ihre anwenderdefinierten Diagnosemeldungen auch noch an angemeldete Anzeigegeräte senden. Bei Aufruf der SFC 52 mit SEND = 1 wird diese Diagnosemeldung in den Sendepuffer geschrieben und automatisch an den oder die an der CPU angemeldeten Teilnehmer geschickt.

Ist ein Senden nicht möglich (z. B. weil kein Teilnehmer angemeldet wurde oder der Sendepuffer voll ist), erfolgt dennoch der Eintrag des anwenderdefinierten Diagnoseereignisses in den Diagnosepuffer.

Meldung mit Quittieranzeige erzeugen

Falls Sie ein anwenderdefiniertes Diagnoseereignis quittieren und diese Quittierung programmtechnisch erfassen wollen, gehen Sie wie folgt vor:

- Beim kommenden Ereignis beschreiben Sie eine Variable vom Typ BOOL mit 1, beim gehenden Ereignis beschreiben Sie diese mit 0.
- Diese Variable überwachen Sie mit Hilfe des SFB 33 ALARM.

23.8.3 Diagnosefunktionen

Die Systemdiagnose ist die Erkennung, Auswertung und Meldung von Fehlern, die innerhalb eines Automatisierungssystems auftreten. Dazu gibt es in jeder CPU und jeder systemdiagnosefähigen Baugruppe (z. B. FM354) einen Diagnosepuffer, in den nähere Informationen zu allen Diagnoseereignissen in der Reihenfolge ihres Auftretens eingetragen werden.

Diagnoseereignisse

Als Diagnoseereignisse werden Ihnen folgende Einträge angezeigt, z. B.:

- interne und externe Fehler auf einer Baugruppe
- Systemfehler in der CPU
- Betriebszustandsübergänge (z. B. von RUN nach STOP)
- Fehler im Anwenderprogramm
- Ziehen/Stecken von Baugruppen
- mit der Systemfunktion SFC 52 eingetragene Anwendermeldungen

Beim Umräumen von CPUs bleibt der Inhalt des Diagnosepuffers erhalten. Fehler im System können durch den Diagnosepuffer auch nach längerer Zeit noch ausgewertet werden, um die Ursache für einen STOP festzustellen oder um das Auftreten einzelner Diagnoseereignisse zurückzuverfolgen und zuordnen zu können.

Aufzeichnung der Diagnosedaten

Die Erfassung von Diagnosedaten durch die Systemdiagnose muss nicht programmiert werden, sie ist standardmäßig vorhanden und läuft automatisch ab. SIMATIC S7 bietet verschiedene bas00112. Einige sind in die CPU integriert, andere werden von den Baugruppen (SMs, CPs und FMs) zur Verfügung gestellt.

Anzeige von Fehlern

Baugruppeninterne und -externe Fehler werden über LED-Anzeigen auf der Frontplatten der jeweiligen Baugruppe angezeigt. Die LED-Anzeigen und ihre Auswertung ist beschrieben in den Handbüchern zur S7-Hardware. Bei S7-300 werden interne und externe Fehler zu einem Sammelfehler zusammengefasst.

Die CPU erkennt Systemfehler sowie Fehler im Anwenderprogramm und trägt Diagnoseereignisse in die Systemzustandsliste und den Diagnosepuffer ein. Diese Diagnosemeldungen können am PG ausgelesen werden.

Diagnosefähige Signal- und Funktionsbaugruppen erkennen interne und externe Baugruppenfehler und erzeugen einen Diagnosealarm, auf den Sie mit Hilfe eines Alarm-OBs reagieren können.

23.9 Maßnahmen im Programm zur Störungsbehandlung

Die CPU ruft bei Erkennung von Fehlern im Programmablauf (Synchronfehler) und von Fehlern im Automatisierungssystem (Asynchronfehler) den zu dem jeweiligen Fehler gehörenden Fehler-OB auf:

Aufgetretener Fehler	Fehler-OB
Peripherie-Redundanzfehler	OB 70
CPU-Redundanzfehler	OB 72
Zeitfehler	OB 80
Stromversorgungsfehler	OB 81
Diagnosealarm	OB 82
Ziehen/Stecken-Alarm	OB 83
CPU-Hardwarefehler	OB 84
Programmablauffehler	OB 85
Baugruppenträgerausfall oder Ausfall einer Station bei dezentraler Peripherie	OB 86
Kommunikationsfehler	OB 87
Programmierfehler	OB 121
Peripheriezugriffsfehler	OB 122

Ist der entsprechende OB nicht vorhanden, geht die CPU in den Betriebszustand "STOP" (Ausnahmen: OB 70, OB 72, OB 81, OB 87). Andernfalls besteht die Möglichkeit, in dem OB Anweisungen zu hinterlegen, wie auf diese Fehlersituation reagiert werden soll. Dadurch können evtl. die Auswirkungen des Fehlers gemindert oder aufgehoben werden.

Prinzipielle Vorgehensweise

Anlegen und Öffnen des OB

1. Rufen Sie den Baugruppenzustand Ihrer CPU auf.
2. Wählen Sie das Register "Leistungsdaten".
3. Entnehmen Sie der angezeigten Liste, ob der zu programmierende OB für diese CPU zulässig ist.
4. Fügen Sie den OB in den Ordner "Bausteine" Ihres Programms ein und öffnen Sie diesen.
5. Geben Sie das Programm zur Behandlung des Fehlers ein.
6. Laden Sie den OB auf das Zielsystem.

Programmieren der Maßnahmen zur Fehlerbehandlung

1. Lokaldaten des OB auswerten, um genauere Fehlerursache zu ermitteln.
Die Variablen OB8x_FLT_ID bzw. OB12x_SW_FLT der Lokaldaten enthalten den Fehlercode. Die Bedeutung ist im Referenzhandbuch System- und Standardfunktionen beschrieben.
2. In den Programmteil verzweigen, mit dem auf diesen Fehler reagiert wird.

Ein Beispiel zur Behandlung von Diagnosealarmen enthält die Referenzhilfe zu System- und Standardfunktionen unter dem Titel "Beispiel zur Baugruppendiagnose mit der SFC 51 (RDSYSST)".

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.1 Auswerten des Ausgangsparameters RET_VAL

Eine Systemfunktion zeigt über den Ausgangsparameter RET_VAL (Rückgabewert) an, ob die CPU die Funktion der SFC erfolgreich ausführen konnte oder nicht.

Fehlerinformationen im Rückgabewert

Der Rückgabewert ist vom Datentyp Ganzzahl (INT). Das Vorzeichen einer Ganzzahl gibt an, ob es sich um eine positive oder negative Ganzzahl handelt. Die Relation des Rückgabewerts zu dem Wert "0" zeigt an, ob während der Bearbeitung der Funktion ein Fehler aufgetreten ist (siehe auch nachfolgende Tabelle):

- Tritt während der Bearbeitung der Funktion ein Fehler auf, ist der Rückgabewert kleiner als Null. Das Vorzeichenbit der Ganzzahl ist "1".
- Wird die Funktion fehlerfrei bearbeitet, ist der Rückgabewert größer gleich Null. Das Vorzeichenbit der Ganzzahl ist "0".

Bearbeitung der SFC durch die CPU	Rückgabewert	Vorzeichen der Ganzzahl
fehlerhaft	kleiner als "0"	negativ (Vorzeichenbit ist "1")
fehlerfrei	größer oder gleich "0"	positiv (Vorzeichenbit ist "0")

Reagieren auf Fehlerinformationen

Tritt während der Bearbeitung einer SFC ein Fehler auf, dann stellt die SFC über den Rückgabewert RET_VAL einen Fehlercode zur Verfügung.

Man unterscheidet dabei zwischen:

- einem allgemeinen Fehlercode, den alle SFCs ausgeben können, und
- einem spezifischen Fehlercode, den eine SFC abhängig von seinen spezifischen Funktionen ausgeben kann.

Funktionswertübergabe

Einige SFCs benutzen den Ausgangsparameter RET_VAL auch zur Übergabe des Funktionswertes, z. B. übergibt die SFC 64 TIME_TCK mit RET_VAL die gelesene Systemzeit.

Ausführliche Informationen zum Parameter RET_VAL finden Sie in der Hilfe zu SFBs/SFCs.

23.9.2 Fehler-OBs als Reaktion auf die Erkennung eines Fehlers

Feststellbare Fehler

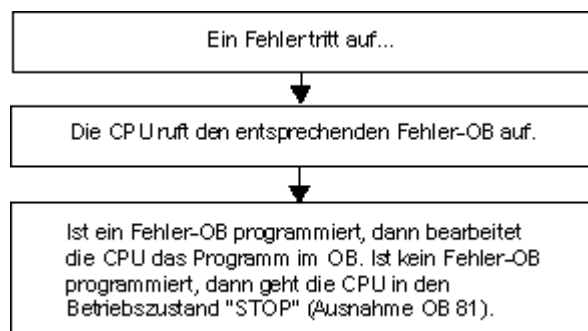
Das Systemprogramm kann folgende Fehler feststellen:

- fehlerhaftes Arbeiten der CPU
- Fehler in der Systemprogrammbearbeitung
- Fehler im Anwenderprogramm
- Fehler in der Peripherie

Je nach Fehlerart wird die CPU in den STOP-Zustand gesetzt oder ein Fehler-OB aufgerufen.

Programmieren von Reaktionen

Sie können Programme entwerfen, um auf die verschiedenen Arten von Fehlern zu reagieren und das weitere Verhalten der CPU zu bestimmen. Das Programm für einen bestimmten Fehler kann dann in einem Fehler-OB gespeichert werden. Wenn der Fehler-OB aufgerufen wird, wird das Programm bearbeitet.



Fehler-OBs

Es wird unterschieden zwischen synchronen und asynchronen Fehlern:

- Synchroner Fehler können einem MC7-Befehl zugeordnet werden (z. B. Ladebefehl auf gezogene Signalbaugruppe).
- Asynchrone Fehler können einer Prioritätsklasse oder dem gesamten Automatisierungssystem zugeordnet werden (z. B. Zykluszeitüberlauf).

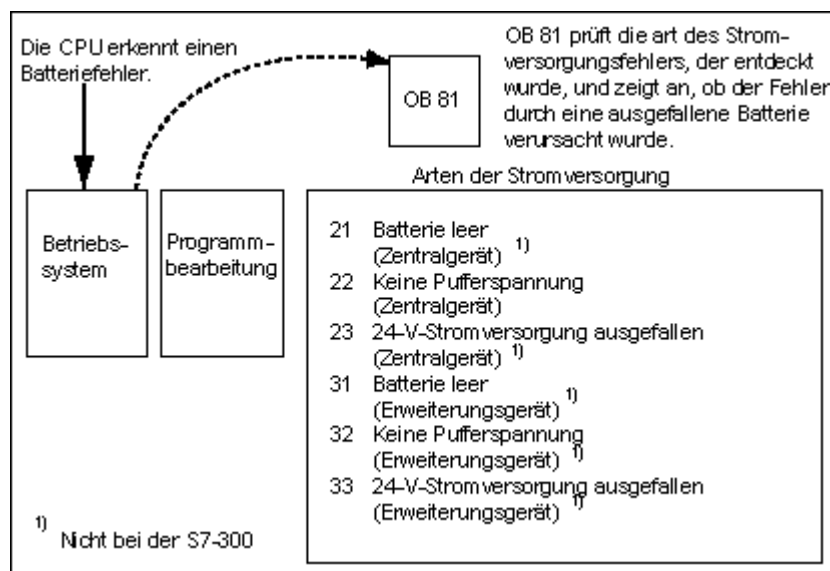
Nachfolgende Tabelle zeigt welche Fehlerarten prinzipiell auftreten können. Ob Ihre CPU die angegebenen OBs anbietet, entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" oder dem Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Aufbauen".

Fehlertyp	Fehlerart	OB	Priorität
Redundanz	Peripherie-Redundanzfehler (nur in H-CPU's)	OB 70	25
	CPU-Redundanzfehler (nur in H-CPU's)	OB 72	28
Asynchron	Zeitfehler	OB 80	26
	Stromversorgungsfehler	OB 81	(bzw. 28, wenn der Fehler-OB im
	Diagnosealarm	OB 82	Anlaufprogramm vorkommt)
	Ziehen/Stecken-Alarm	OB 83	
	CPU-Hardwarefehler	OB 84	
	Programmablauffehler	OB 85	
	Baugruppenträgerausfall	OB 86	
	Kommunikationsfehler	OB 87	
Synchron	Programmierfehler	OB 121	Priorität des fehlerverursachenden OB
	Zugriffsfehler	OB 122	

Beispiel zur Verwendung des Fehler-OB 81

Sie können mit den Lokaldaten (Startinformation) des Fehler-OB die Art des aufgetretenen Fehlers auswerten.

Wenn die CPU beispielsweise einen Batteriefehler erkennt, dann ruft das Betriebssystem OB 81 auf (siehe Bild).



Sie können ein Programm schreiben, das den Ereigniscode, der den Aufruf von OB 81 ausgelöst hat, auswertet. Sie können auch ein Programm schreiben, das eine Reaktion hervorruft, wie beispielsweise das Einschalten eines Ausgangs, der an eine Lampe der Operator-Station angeschlossen ist.

Lokaldaten des Fehler-OB 81

Die folgende Tabelle beschreibt die temporären (TEMP) Variablen, die in der Variablendeklarationstabelle des OB 81 deklariert sind.

In der Symboltabelle muss auch das Symbol *Batteriefehler* (BOOL) als Ausgang gekennzeichnet werden (z. B. A 4.0), so dass andere Teile des Programms auf diese Daten zugreifen können.

Dekl.	Name	Typ	Beschreibung
TEMP	OB81_EV_CLASS	BYTE	Fehlerklasse/Fehlerkennung 39xx
TEMP	OB81_FLT_ID	BYTE	Fehlercode: b#16#21 = Mindestens eine Pufferbatterie des Zentralgeräts leer ¹⁾ b#16#22 = Pufferspannung im Zentralgerät fehlt b#16#23 = Ausfall der 24-V-Stromversorgung im Zentralgerät ¹⁾ b#16#31 = Mindestens eine Pufferbatterie eines Erweiterungsgeräts leer ¹⁾ b#16#32 = Pufferspannung in einem Erweiterungsgerät fehlt ¹⁾ b#16#33 = Ausfall der 24-V-Stromversorgung eines Erweiterungsgeräts ¹⁾
TEMP	OB81_PRIORITY	BYTE	Prioritätsklasse = 26/28
TEMP	OB81_OB_NUMBR	BYTE	81 = OB 81
TEMP	OB81_RESERVED_1	BYTE	Reserviert
TEMP	OB81_RESERVED_2	BYTE	Reserviert
TEMP	OB81_MDL_ADDR	INT	Reserviert
TEMP	OB81_RESERVED_3	BYTE	Nur relevant für die Fehlercodes B#16#31, B#16#32, B#16#33
TEMP	OB81_RESERVED_4	BYTE	
TEMP	OB81_RESERVED_5	BYTE	
TEMP	OB81_RESERVED_6	BYTE	
TEMP	OB81_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB gestartet wurde

¹⁾ = Nicht bei der S7-300.

Beispielprogramm für den Fehler-OB 81

Anhand eines AWL-Beispielprogramms wird gezeigt, wie Sie den Fehlercode in OB 81 lesen können.

Das Programm ist wie folgt aufgebaut:

- Der Fehlercode in OB 81 (OB81_FLT_ID) wird gelesen und verglichen mit dem Wert des Ereignisses "Batterie leer" (B#16#3921).
- Entspricht der Fehlercode dem Code für "Batterie leer", dann springt das Programm zur Marke BFeh und schaltet den Ausgang *Batteriefehler* ein.
- Entspricht der Fehlercode nicht dem Code für "Batterie leer", dann vergleicht das Programm den Code mit dem Code für "Batterieausfall".
- Entspricht der Fehlercode dem Code für "Batterieausfall", dann springt das Programm zur Marke BFeh und schaltet den Ausgang *Batteriefehler* ein. Andernfalls wird der Baustein beendet.

AWL	Beschreibung
L	B#16#21 // Vergleiche Ereigniscode "Batterie leer" // (B#16#21) mit
L	#OB81_FLT_ID // dem Fehlercode für OB 81.
==I	// Wenn gleich (Batterie ist leer), // dann springe zu BF.
SPB BF	
L	B#16#22 // Vergleiche Ereigniscode "Batterieausfall" // (b#16#22) mit
==I	// dem Fehlercode für OB 81.
SPB BF	// Wenn gleich, dann springe zu BF.
BEA	// keine Meldung über Batteriefehler
BF: L	B#16#39 // Vergleiche Kennung für kommendes Ereignis
L	#OB81_EV_CLASS // mit dem Fehlercode für OB 81.
==I	// wenn Batterieausfall oder leere Batterie // entdeckt wird,
S	Batteriefehler // setze Batteriefehler // (Variable aus Symboltabelle)
L	B#16#38 // Vergleiche Kennung für gehendes Ereignis mit
==I	// dem Fehlercode für OB 81.
R	Batteriefehler // rücksetze Batteriefehler, wenn behoben.

Ausführliche Informationen zu OBs, SFBs und SFCs und Erläuterungen zu den Ereignis-IDs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.3 Einfügen von Ersatzwerten bei Fehlererkennung

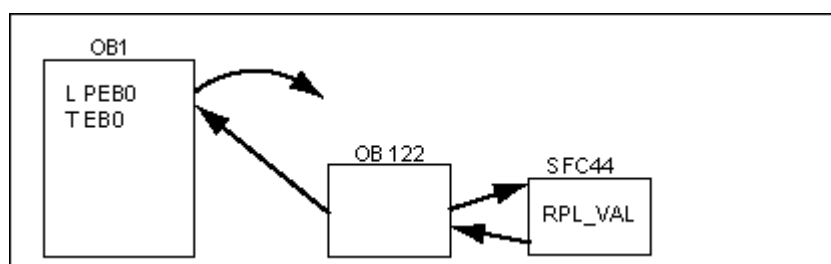
Für bestimmte Fehlerarten (z. B. Drahtbruch bei Eingangssignal) können Sie Ersatzwerte vorgeben für Werte, die aufgrund des Fehlers nicht verfügbar sind. Es gibt zwei Möglichkeiten Ersatzwerte vorzugeben:

- Sie können mit STEP 7 Ersatzwerte für parametrierbare Ausgabebaugruppen parametrieren. Nicht parametrierbare Ausgabebaugruppen haben den voreingestellten Ersatzwert 0.
- Sie können mit Hilfe der SFC 44 RPL_VAL in Fehler-OBs Ersatzwerte programmieren (nur für Eingabebaugruppen).

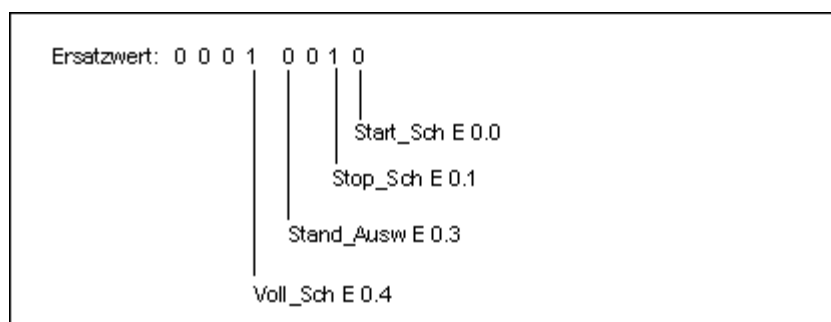
Bei allen Ladebefehlen, die zu synchronen Fehlern führen, können Sie im Fehler-OB einen Ersatzwert für den AKKU-Inhalt angeben.

Beispielprogramm zum Ersetzen eines Wertes

Im folgenden Beispielprogramm wird ein Ersatzwert in der SFC 44 RPL_VAL bereitgestellt. Nachfolgendes Bild zeigt, wie OB 122 aufgerufen werden kann, wenn die CPU erkennt, dass eine Eingabebaugruppe nicht reagiert.



In diesem Beispiel wird der Ersatzwert aus nachfolgendem Bild in das Programm eingegeben, damit das Programm mit sinnvollen Werten weiterlaufen kann.



Fällt eine Eingabebaugruppe aus, erzeugt die Bearbeitung der Anweisung L PEB0 einen synchronen Fehler und startet den OB 122. Standardmäßig liest der Ladebefehl den Wert 0 ein. Sie können jedoch mit der SFC 44 beliebige zum Prozess passende Ersatzwerte definieren. Die SFC ersetzt den AKKU-Inhalt durch den vorgegebenen Ersatzwert.

Das folgende Beispielprogramm könnte in OB 122 gespeichert sein. Die folgende Tabelle zeigt die temporären Variablen, die in diesem Fall in der Variablendeklarationstabelle des OB 122 deklariert werden müssen.

Dekl.	Name	Typ	Beschreibung
TEMP	OB122_EV_CLASS	BYTE	Fehlerklasse/Fehlerkennung 29xx
TEMP	OB122_SW_FLT	BYTE	Fehlercode: 16#42, 16#43, 16#44 ¹⁾ , 16#45 ¹⁾
TEMP	OB122_PRIORITY	BYTE	Prioritätsklasse = Priorität des OB, in dem der Fehler aufgetreten ist
TEMP	OB122_OB_NUMBR	BYTE	122 = OB 122
TEMP	OB122_BLK_TYPE	BYTE	Bausteintyp, in dem der Fehler aufgetreten ist
TEMP	OB122_MEM_AREA	BYTE	Speicherbereich und Zugriffsart
TEMP	OB122_MEM_ADDR	WORD	Adresse im Speicher, an der der Fehler aufgetreten ist
TEMP	OB122_BLK_NUM	WORD	Nummer des Bausteins, in dem der Fehler aufgetreten ist
TEMP	OB122_PRG_ADDR	WORD	Relativadresse des fehlerverursachenden Befehls
TEMP	OB122_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB gestartet wurde
TEMP	Fehler	INT	Speichert den Fehlercode von SFC44

¹⁾ Nicht bei der S7-300.

AWL	Beschreibung
L B#16#2942	Vergleiche den Ereigniscode von OB 122 mit dem Ereigniscode (B#16#2942) für die Quittierung eines Zeitfehlers beim Lesen von der Peripherie. Wenn gleich, springe zu "QFeh".
L #OB122_SW_FLT	
==I	Vergleiche den Ereigniscode von OB 122 mit dem Ereigniscode (B#16#2943) für einen Adressierungsfehler (Schreiben von einer Baugruppe, die nicht vorhanden ist). Wenn ungleich, springe zu "Stop".
SPB QFeh	
L B#16#2943	Marke "QFeh": Übergibt DW#16#2912 (binär 10010) an die SFC44 (REPL_VAL). SFC44 lädt diesen Wert in AKKU 1 (und ersetzt den Wert, der den Aufruf von OB 122 ausgelöst hat). Speichert den SFC-Fehlercode in #Fehler.
<> I	
SPB Stop	Vergleicht #Fehler mit 0 (wenn gleich, dann ist bei der Bearbeitung von OB 122 kein Fehler aufgetreten). Beende den Baustein, wenn kein Fehler aufgetreten ist.
Qfeh: CALL "REPL_VAL"	
VAL := DW#16#2912	Marke "Stop": Ruft SFC46 "STP" auf und versetzt die CPU in den Betriebszustand STOP.
RET_VAL :=	
#Fehler	
L #Fehler	
L 0	
==I	
BEB	
Stop: CALL "STP"	

23.9.4 Peripherie-Redundanzfehler (OB 70)

Beschreibung

Das Betriebssystem der H-CPU ruft den OB 70 auf, wenn ein Redundanzverlust am PROFIBUS DP auftritt (z. B. bei einem Busausfall am aktiven DP-Master oder bei einem Fehler in der Anschaltung des DP-Slave), oder wenn der aktive DP-Master von DP-Slaves mit geschalteter Peripherie wechselt.

Programmieren des OB 70

Den OB 70 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 70 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 70 beispielsweise dazu nutzen:

- die Startinformation des OB 70 auszuwerten und festzustellen, welches Ereignis den Peripherie-Redundanzverlust ausgelöst hat.
- mit Hilfe der SFC 51 RDSYSST den Zustand Ihres Systems zu ermitteln (SZL_ID=B#16#71).

Die CPU wechselt nicht in den Betriebszustand "STOP", wenn ein Peripherie-Redundanzfehler auftritt und der OB 70 nicht programmiert ist.

Ist der OB 70 geladen und das H-System befindet sich im redundanten Betrieb, so wird der OB 70 in beiden CPUs bearbeitet. Das H-System bleibt im redundanten Betrieb.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.5 CPU-Redundanzfehler (OB 72)

Beschreibung

Das Betriebssystem der H-CPU ruft den OB 72 auf, wenn eines der folgenden Ereignisse auftritt:

- Redundanzverlust der CPUs
- Vergleichsfehler (z. B. RAM, PAA)
- Reserve-Master-Umschaltung
- Synchronisationsfehler
- Fehler in einem SYNC-Modul
- Abbruch des Aufdatvorgangs
- Der OB 72 wird von allen CPUs ausgeführt, die sich nach einem zugehörigen Startereignis in den Betriebszuständen RUN oder ANLAUF befinden.

Programmieren des OB 72

Den OB 72 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 72 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 72 beispielsweise dazu nutzen,

- die Startinformation des OB 72 auszuwerten und festzustellen, welches Ereignis den CPU-Redundanzverlust ausgelöst hat
- mit Hilfe der SFC 51 RDSYSST den Zustand Ihres Systems zu ermitteln (SZL_ID=B#16#71)
- anlagenspezifisch auf den CPU-Redundanzverlust zu reagieren.

Die CPU wechselt nicht in den Betriebszustand "STOP", wenn ein CPU-Redundanzfehler auftritt und der OB 72 nicht programmiert ist.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.6 Zeitfehler (OB 80)

Beschreibung

Das Betriebssystem der CPU ruft OB 80 auf, wenn ein Zeitfehler auftritt. Zeitfehler können z. B. sein:

- Überschreiten der Maximalzykluszeit
- Überspringen von Uhrzeitalarmen durch Vorstellen der Uhrzeit
- zu große Verzögerung beim Bearbeiten einer Prioritätsklasse

Programmieren des OB 80

Den OB 80 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 80 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 80 beispielsweise dazu nutzen:

- die Startinformation des OB 80 auszuwerten und festzustellen, welche Uhrzeitalarme übersprungen wurden
- mit Hilfe der SFC 29 CAN_TINT den übersprungenen Uhrzeitalarm zu deaktivieren, damit er nicht ausgeführt wird und um mit der neu eingestellten Uhrzeit einen sauberen Aufsetzpunkt für die Bearbeitung der Uhrzeitalarme zu haben.

Wenn Sie übersprungene Uhrzeitalarme im OB 80 nicht deaktivieren, wird der erste übersprungene Uhrzeitalarm bearbeitet, alle anderen werden ignoriert.

Haben Sie OB 80 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Zeitfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.7 Stromversorgungsfehler (OB 81)

Beschreibung

Das Betriebssystem der CPU ruft OB 81 auf, wenn im Zentralgerät oder einem Erweiterungsgerät

- die 24V-Spannungsversorgung
- eine Batterie
- die komplette Pufferung

ausgefallen ist bzw. wenn die Störung beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 81

Den OB 81 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 81 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 81 beispielsweise dazu nutzen

- die Startinformation des OB 81 auszuwerten und festzustellen, welcher Stromversorgungsfehler vorliegt
- die Nummer des Baugruppenträgers mit der defekten Stromversorgung zu ermitteln
- eine Lampe an einer Operator Station anzusteuern, um dem Wartungspersonal anzuzeigen, dass eine Batterie auszutauschen ist.

Haben Sie OB 81 nicht programmiert, wechselt die CPU nicht in den Betriebszustand "STOP", wenn ein Stromversorgungsfehler erkannt wird. Der Fehler wird aber in den Diagnosepuffer eingetragen und die entsprechende LED auf der Frontplatte zeigt den Fehler an.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.8 Diagnosealarm (OB 82)

Beschreibung

Das Betriebssystem der CPU ruft OB 82 auf, wenn eine diagnosefähige Baugruppe, bei der Sie den Diagnosealarm freigegeben haben, einen Fehler erkennt und wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 82

Den OB 82 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 82 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 82 beispielsweise dazu nutzen

- die Startinformation des OB 82 auszuwerten
- eine genaue Diagnose des aufgetretenen Fehlers vorzunehmen.

Wenn ein Diagnosealarm ausgelöst wird, trägt die gestörte Baugruppe automatisch 4 Byte Diagnosedaten und ihre Anfangsadresse in die Startinformation des Diagnosealarm-OB und in den Diagnosepuffer ein. Damit erhalten Sie die Information, auf welcher Baugruppe und wann ein Fehler aufgetreten ist.

Weitere Diagnosedaten der gestörten Baugruppe (auf welchem Kanal ist der Fehler aufgetreten, um welchen Fehler handelt es sich) können Sie mit einem entsprechenden Programm im OB 82 auswerten. Mit der SFC 51 RDSYSST können Sie Baugruppendiagnosedaten auslesen und mit der SFC 52 WR_USRMSG diese Information in den Diagnosepuffer eintragen. Außerdem können Sie zusätzlich die selbst definierte Diagnosemeldung an ein angemeldetes Beobachtungsgerät senden.

Haben Sie OB 82 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Diagnosealarm ausgelöst wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.9 Ziehen-/Stecken-Alarm (OB 83)

Beschreibung

S7-400-CPU's überwachen zyklisch im Abstand von ca. einer Sekunde das Ziehen und Stecken von Baugruppen im Zentralgerät und den Erweiterungsgeräten.

Nach dem Einschalten der Netzspannung prüft die CPU, ob alle Baugruppen, die in der mit STEP 7 erstellten Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind. Ist dies der Fall, wird diese Ist-Konfiguration gespeichert und dient als Referenzwert für die zyklische Überwachung der Baugruppen. In jedem Abfragezyklus wird die neu ermittelte Ist-Konfiguration mit der bisherigen Ist-Konfiguration verglichen. Bei Abweichungen wird ein Ziehen-/Stecken-Alarm gemeldet und es erfolgt je ein Eintrag in den Diagnosepuffer und die Systemzustandsliste. Im Betriebszustand RUN wird der Ziehen/Stecken-OB gestartet.

Hinweis

Im Betriebszustand RUN dürfen Stromversorgungsbaugruppen, CPUs und IMs nicht gezogen werden!

Zwischen Ziehen und Stecken einer Baugruppe müssen mindestens 2 Sekunden vergehen, damit das Ziehen oder Stecken von der CPU richtig erkannt werden kann.

Parametrierung einer neu gesteckten Baugruppe

Wird eine Baugruppe im Betriebszustand RUN gesteckt, prüft die CPU, ob der Baugruppentyp der neu gesteckten Baugruppe mit dem der ursprünglich gesteckten Baugruppe übereinstimmt. Bei Übereinstimmung der Baugruppentypen erfolgt die Parametrierung. Übertragen werden entweder die Default-Parameter oder die Parameter, die Sie mit STEP 7 vergeben haben.

Programmieren des OB 83

Den OB 83 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 83 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 83 beispielsweise dazu nutzen

- die Startinformation des OB 83 auszuwerten
- mit Hilfe der Systemfunktionen SFC 55 bis 59 die neu gesteckte Baugruppe nachzuparametrieren.

Haben Sie OB 83 nicht programmiert, wechselt die CPU von RUN in den Betriebszustand "STOP", wenn ein Ziehen-/Stecken-Alarm auftritt.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.10 CPU-Hardwarefehler (OB 84)

Beschreibung

Das Betriebssystem der CPU ruft OB 84 auf, wenn ein Fehler bei der Schnittstelle zum MPI-Netz, zum K-Bus oder zur Anschaltung für die Dezentrale Peripherie, z. B. fehlerhafter Signalpegel auf der Leitung erkannt wird bzw. wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 84

Den OB 84 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 84 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 84 beispielsweise dazu nutzen

- die Startinformation des OB 84 auszuwerten
- mit Hilfe der Systemfunktion SFC 52 WR_USMSG eine Meldung zum Diagnosepuffer zu senden.

Haben Sie OB 84 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein CPU-Hardwarefehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.11 Programmablauffehler (OB 85)

Beschreibung

Das Betriebssystem der CPU ruft OB 85 auf, wenn ein

- Startereignis für einen Alarm-OB vorliegt, aber der OB nicht ausgeführt werden kann, weil er nicht in die CPU geladen wurde
- Fehler beim Zugriff auf den Instanz-Datenbaustein eines Systemfunktionsbausteins aufgetreten ist
- Fehler bei der Prozessabbild-Aktualisierung (Baugruppe nicht vorhanden oder defekt) aufgetreten ist.

Programmieren des OB 85

Den OB 85 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 85 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 85 beispielsweise dazu nutzen

- die Startinformation des OB 85 auszuwerten und festzustellen, welche Baugruppe defekt ist oder fehlt (Angabe der Baugruppen-Anfangsadresse)
- mit Hilfe der SFC 49 LGC_GADR den Steckplatz der betreffenden Baugruppe zu ermitteln.

Haben Sie OB 85 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Prioritätsklassenfehler erkannt wird.

23.9.12 Baugruppenträgerausfall (OB 86)

Beschreibung

Das Betriebssystem der CPU ruft den OB 86 auf, wenn es eines der folgenden Ereignisse erkannt hat:

- Ausfall eines zentralen Erweiterungsgerätes (nicht bei S7-300), z. B. unterbrochene Verbindungsleitung, dezentraler Spannungsausfall eines Baugruppenträgers.
- Ausfall eines Mastersystems bzw. eines Slaves (PROFIBUS DP) oder Ausfall eines IO-Systems bzw. eines IO-Devices (PROFINET IO).

Der OB 86 wird auch aufgerufen, wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 86

Den OB 86 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 86 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 86 beispielsweise dazu nutzen

- die Startinformation des OB 86 auszuwerten und festzustellen, welcher Baugruppenträger defekt ist oder fehlt.
- mit Hilfe der Systemfunktion SFC 52 WR_USMSG eine Meldung in den Diagnosepuffer einzutragen und die Meldung zu einem Beobachtungsgerät zu senden.

Haben Sie OB 86 nicht programmiert, wechselt die CPU in den Betriebszustand STOP, wenn ein Baugruppenträgerausfall erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.13 Kommunikationsfehler (OB 87)

Beschreibung

Das Betriebssystem der CPU ruft OB 87 auf, wenn ein Kommunikationsfehler beim Datenaustausch über Kommunikations-Funktionsbausteine oder Globaldatenkommunikation auftritt, z. B.

- beim Empfang von Globaldaten wurde eine falsche Telegrammkennung erkannt
- der Datenbaustein für die Statusinformation der Globaldaten ist nicht vorhanden oder zu kurz.

Programmieren des OB 87

Den OB 87 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 87 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 87 beispielsweise dazu nutzen

- die Startinformation des OB 87 auszuwerten und,
- wenn der Datenbaustein für die Statusinformation der Globaldatenkommunikation fehlt, einen Datenbaustein anzulegen.

Die CPU wechselt nicht in den Betriebszustand "STOP", wenn ein Kommunikationsfehler erkannt wird und der OB 87 nicht programmiert ist.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.14 Programmierfehler (OB 121)

Beschreibung

Das Betriebssystem der CPU ruft OB 121 auf, wenn ein Programmierfehler auftritt, z. B.

- adressierte Zeiten nicht vorhanden
- aufgerufener Baustein nicht geladen.

Programmieren des OB 121

Den OB 121 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 121 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 121 beispielsweise dazu nutzen

- die Startinformation des OB 121 auszuwerten
- die Fehlerursache in einen Melde-Datenbaustein einzutragen.

Haben Sie OB 121 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Programmierfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

23.9.15 Peripheriezugriffsfehler (OB 122)

Beschreibung

Das Betriebssystem der CPU ruft OB 122 auf, wenn mit einer STEP 7-Operation auf einen Ein- oder Ausgang einer Signalbaugruppe zugegriffen wird, dem zum Zeitpunkt des letzten Neustarts keine Baugruppe zugeordnet war, z. B.

- Fehler bei Peripherie-Direktzugriff (Baugruppe defekt oder nicht vorhanden)
- Zugriff auf eine Peripherie-Adresse, die der CPU nicht bekannt ist.

Programmieren des OB 122

Den OB 122 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 122 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 122 beispielsweise dazu nutzen

- die Startinformation des OB 122 auszuwerten
- die Systemfunktion SFC 44 aufzurufen und einen Ersatzwert für eine Eingabebaugruppe vorzugeben, damit das Programm mit einem sinnvollen, prozessabhängigen Wert weiterbearbeitet werden kann.

Haben Sie OB 122 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Peripheriezugriffsfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

24 Drucken und Archivieren

24.1 Drucken einer Projektdokumentation

Wenn Sie die Programmerstellung für Ihre Automatisierungslösung abgeschlossen haben, können Sie mit der in STEP 7 integrierten Druckfunktion alle wichtigen Daten ausdrucken und sich damit eine Projektdokumentation erzeugen.

Ausdruckbare Projektbestandteile

Den Inhalt von Objekten können Sie sowohl direkt vom SIMATIC Manager aus drucken als auch durch Öffnen des entsprechenden Objekts und anschließendem Druckanstoß.

Direkt über den SIMATIC Manager lassen sich folgende Bestandteile eines Projekts ausdrucken:

- Objektbaum (Struktur des Projekts/der Bibliothek)
- Objektlisten (Inhalt eines Objektordners)
- Objektinhalt (Inhalt eines Objekts)
- Meldungen

Über Öffnen des entsprechenden Objekts können z. B. folgende Projektbestandteile ausgedruckt werden:

- Bausteine in Darstellung KOP, FUP, AWL oder anderer Sprache (Optionssoftware)
- die Symboltabelle mit dem symbolischen Namen absoluter Adressen
- die Konfigurationstabelle mit der Anordnung der Baugruppen im AS und Baugruppenparametern
- den Diagnosepufferinhalt
- die Variablen-tabelle mit Statusformaten sowie Status- und Steuerwerten
- Referenzdaten, d. h. Querverweislisten, Belegungspläne, Programmstrukturen, nicht verwendete Operanden, Operanden ohne Symbol
- die Globaldatentabelle
- Auskunftsdaten zu Baugruppen mit Baugruppenzustand
- Bedienerrelevante Texte (Anwendertexte und Textbibliotheken)
- Dokumente von Optionspaketen z. B. von Programmiersprachen

Optionspaket DOCPRO

Zum Erzeugen, Bearbeiten und Drucken von standardisierten Schaltbüchern können Sie das Optionspaket DOCPRO einsetzen. Sie erhalten so eine Anlagendokumentation, die die DIN- und ANSI-Norm erfüllt.

24.1.1 Prinzipielle Vorgehensweise beim Drucken

Zum Ausdrucken gehen Sie folgendermaßen vor:

1. Öffnen Sie das geeignete Objekt, um die zu druckende Information am Bildschirm anzuzeigen.
2. Blenden Sie das Dialogfeld "Drucken" mit dem Menübefehl **Datei > Drucken...** im jeweiligen Fenster auf. Je nach Fenster kann der erste Eintrag in der Menüleiste anstelle von "Datei" auch anders lauten, z. B. "Tabelle".
3. Ändern Sie bei Bedarf die Druckeinstellungen (z. B. Drucker, Druckbereich, Anzahl der Kopien) im Dialogfeld und schließen Sie es.

Manche Dialogfelder enthalten die Schaltfläche "Drucken", z. B. das Dialogfeld zum Baugruppenzustand. Klicken Sie auf diese Schaltfläche, um den Inhalt des Dialogfelds auszudrucken.

Bausteine brauchen nicht geöffnet werden. Sie können diese direkt im SIMATIC Manager mit dem Menübefehl **Datei > Drucken...** ausgeben.

24.1.2 Funktionen zum Drucken

Für den Ausdruck der Druckobjekte stehen folgende ergänzende Funktionen zur Verfügung:

Druckobjekte	Menübefehl	Funktion	Funktion	Funktion
		Druckvorschau	Seite einrichten, Register "Papierformat"	Seite einrichten, Register "Schriftfelder"
Bausteine, AWL-Quellen	Datei > *	•	•	•
Baugruppenzustand		–	•	•
Globaldatentabelle	GD-Tabelle > *	•	•	•
Konfigurationstabelle	Station > *	•	•	•
Objekt, Objektordner	Datei > *	–	•	•
Referenzdaten	Referenzdaten > *	•	•	•
Symboltabelle	Tabelle > *	•	•	•
Variablentabelle	Tabelle > *	–	•	•
Verbindungstabelle	Netz > *	•	•	•
Bedienerrelevante Texte (Anwendertexte, Textbibliotheken)	Texte > *	•	•	•
* : Das Zeichen * ist Platzhalter im Menübefehl für die jeweilige Funktion (z. B. Druckvorschau oder Seite einrichten)				

Die Schritt-für-Schritt-Anleitungen, wie Sie zum Ausdrucken der einzelnen Druckobjekte vorgehen, finden Sie unter:

Schritte zum Drucken.

Druckvorschau

Über die Funktion "Druckvorschau" können Sie sich die Seitenansicht des Ausdrucks anzeigen lassen.

Hinweis

Das Druckbild des optimierten Ausdrucks wird in der Druckvorschau nicht angezeigt.

Seitenformat sowie Kopf- und Fußzeilen einstellen

Mit dem Menübefehl **Datei > Seite einrichten** können Sie im Register "Papierformat" das Seitenformat (z. B. A4, A5, Letter) und die Ausrichtung (Hochformat oder Querformat) für Ihre zu druckenden Dokumente einstellen. Sie können dabei wählen, ob Ihre Einstellungen projektweit oder nur für die aktuelle Sitzung, gelten sollen.

Passen Sie bitte das zum Ausdruck verwendete Formular dem gewünschten Papierformat an. Falls das Formular zu breit ist, wird der rechte Rand auf einem Folgeblatt ausgedruckt.

Wenn Sie ein Seitenformat mit Rand wählen (z. B. "A4 Rand"), erhält das Druckdokument an der linken Seite einen Rand, den Sie zum Lochens verwenden können.

Im Register "Schriftfelder" können Sie Kopf- und Fußzeilen für Ihre zu druckenden Dokumente projektweit oder nur für die aktuelle Sitzung, einstellen.

24.1.3 Besonderheiten zum Drucken des Objektbaums

Im Dialogfeld "Objektliste drucken" können Sie durch Markieren des Optionsfelds "Baumfenster" zusätzlich zur Objektliste den Objektbaum ausdrucken.

Wenn Sie unter "Druckbereich" das Optionsfeld "Alles" markieren, so wird die gesamte Baumstruktur ausgedruckt. Bei Markieren des Auswahlknopfes "Auswahl" wird die Baumstruktur ab dem ausgewählten Objekt abwärts ausgedruckt.

Hinweis

Die in dem Dialogfeld gewählten Einstellungen gelten nur für das Drucken der Liste bzw. des Baums, nicht jedoch für das Drucken der Inhalte. Hier werden die Einstellungen der jeweilig zuständigen Applikationen verwendet.

24.2 Archivieren von Projekten und Bibliotheken

Einzelne Projekte bzw. Bibliotheken können Sie in komprimierter Form in einer Archivdatei ablegen. Dieses komprimierte Ablegen ist auf Festplatte oder transportablen Datenträgern (z. B. Diskette) möglich.

Archivier-Programme

Innerhalb von STEP 7 können Sie ihr bevorzugtes Archivier-Programm aufrufen, um Projekte zu archivieren. Die Archivier-Programme ARJ und PKZIP 4.0 sind im Lieferumfang von STEP 7 enthalten. Die Programme selbst und die zugehörige Beschreibung finden Sie unter Ihrem Installationspfad im Verzeichnis ...\\Step7\\S7bin\\.

Um ein bestimmtes Archivier-Programm einzusetzen, benötigen Sie davon die folgende (oder eine neuere) Version:

- PKZip Commandline V4.0 (wird mitgeliefert)
- WinZip ab 6.0
- JAR ab 1.02
- ARJ V2.4.1a (nur zum Dearchivieren, wird mitgeliefert)
- ARJ32 V3.x (nur zum Dearchivieren)
- LHArc ab 2.13 (nur zum Dearchivieren)

Besonderheiten

Ab STEP 7 V5.2 werden zum Archivieren nur noch die Archivier-Programme PKZip 4.0, JAR, WinZip unterstützt, zum Dearchivieren jedoch auch die übrigen oben aufgeführten Programme.

Wenn unter früheren STEP 7-Versionen ARJ32 V3.x zum Archivieren verwendet wurde, dürfen diese Archive auch nur mit ARJ32 V3.x wieder dearchiviert werden.

Auf Netzlaufwerken benötigt die Archivierung mit PKZIP V4.0 deutlich mehr Zeit als auf lokalen Laufwerken.

24.2.1 Anwendungsfälle für Speichern / Archivieren

Speichern unter

Mit dieser Funktion können Sie eine **Kopie** des Projekts mit einem anderen Namen anlegen.

Diese Funktion lässt sich einsetzen

- zur Erstellung von Sicherungskopien,
- zur Duplizierung eines vorhandenen Projekts, um es an andere Anforderungen anzupassen.

Um möglichst schnell eine Kopie zu erstellen, wählen Sie im Folgedialog das Speichern ohne Reorganisation. Dann wird der gesamte Dateibaum vom Projektverzeichnis an ohne Überprüfung kopiert und unter einem neuen Namen gespeichert.

Auf dem Datenträger muss genügend Platz zur Ablage der Sicherungskopie vorhanden sein. Versuchen Sie nicht, Projekte auf Diskette zu speichern, da der Speicherplatz in der Regel nicht ausreichen wird. Verwenden Sie zum Transport von Projektdaten auf Disketten die Funktion "Archivieren".

Das Speichern mit Reorganisation dauert länger, es wird jedoch eine Meldung ausgegeben, wenn ein Objekt nicht kopiert und gespeichert werden kann. Ursachen können z. B. ein fehlendes Optionspaket oder defekte Daten eines Objekts sein.

Archivieren

Einzelne Projekte bzw. Bibliotheken können Sie in komprimierter Form in einer Archivdatei ablegen. Dieses komprimierte Ablegen ist auf Festplatte oder transportablen Datenträgern (z. B. Diskette) möglich.

Transportieren Sie Projekte auf Disketten nur in Form von Archivdateien. Falls das Projekt zu groß ist, wählen Sie ein Archivprogramm, mit dem diskettenübergreifende Archive erstellt werden können.

Projekte bzw. Bibliotheken, die in eine Archivdatei komprimiert wurden, können nicht bearbeitet werden. Wollen Sie diese erneut bearbeiten, müssen Sie die Daten entkomprimieren, d. h. das Projekt bzw. die Bibliothek dearchivieren.

24.2.2 Voraussetzungen für das Archivieren

Um ein Projekt/eine Bibliothek zu archivieren, müssen folgende Voraussetzungen erfüllt sein:

- Das Archivier-Programm muss auf Ihrem System installiert sein. Die Einbindung in STEP 7 wird in der Online-Hilfe im Abschnitt "Vorgehensweise beim Archivieren/Dearchivieren" erklärt.
- Alle Daten des Projekts müssen ausnahmslos in bzw. unterhalb des Projektverzeichnisses untergebracht sein. Beim Arbeiten mit der C-Entwicklungsumgebung ist es zwar möglich, Daten auch an anderer Stelle abzulegen. Diese würden aber dann nicht in die Archivdatei aufgenommen werden.
- Ab STEP 7 V5.2 werden zum Archivieren nur noch die Archivier-Programme PKZip 4.0, JAR, WinZip unterstützt, zum Dearchivieren jedoch auch die Programme ARJ und LHarc.

24.2.3 Vorgehensweise beim Archivieren/Dearchivieren

Sie archivieren/dearchivieren Ihr Projekt/Ihre Bibliothek über den Menübefehl **Datei > Archivieren** bzw. **Datei > Dearchivieren**.

Hinweis

Projekte bzw. Bibliotheken, die in eine Archivdatei komprimiert wurden, können nicht bearbeitet werden. Wollen Sie diese erneut bearbeiten, müssen Sie die Daten entkomprimieren, d. h. das Projekt bzw. die Bibliothek dearchivieren.

Beim Dearchivieren werden die dearchivierten Projekte/Bibliotheken automatisch in die Projektliste/Bibliotheksliste aufgenommen.

Einstellen des Zielverzeichnisses

Um das Zielverzeichnis einzustellen, rufen Sie mit dem Menübefehl **Extras > Einstellungen** im SIMATIC Manager das Dialogfeld "Einstellungen" auf.

Im Register "Archivieren" dieses Dialogfelds können Sie die Option "Zielverzeichnis beim Dearchivieren abfragen" ein- und ausschalten.

Ist diese Option ausgeschaltet, wird beim Dearchivieren der Pfad als Zielpfad benutzt, der im Register "Allgemein" unter "Ablageort für Projekte" bzw. "Ablageort für Bibliotheken" dieses Dialogfelds angegeben ist.

Kopieren einer Archivdatei auf Disketten

Sie können ein Projekt bzw. eine Bibliothek archivieren und die Archivdatei dann auf eine Diskette kopieren. Es ist auch möglich, im Dialog "Archivieren" gleich ein Diskettenlaufwerk auszuwählen.

25 Arbeiten mit M7-Automatisierungssystemen

25.1 Vorgehensweise für M7-Systeme

Durch seine standardisierte PC-Architektur bildet der Automatisierungsrechner M7-300/400 eine frei programmierbare Erweiterung der SIMATIC-Automatisierungsplattform. Die Anwenderprogramme für SIMATIC M7 können Sie in einer Hochsprache wie C oder auch grafisch mit CFC (Continuous Funktion Chart) programmieren.

Für die Erstellung der Programme benötigen Sie zusätzlich zu STEP 7 die Systemsoftware M7-SYS RT für M7-300/400 sowie eine Entwicklungsumgebung für M7-Programme (ProC/C++ oder CFC).

Prinzipielles Vorgehen

Bei der Erstellung einer Automatisierungslösung mit SIMATIC M7 fallen grundlegende Aufgaben an. Die folgende Tabelle zeigt die Aufgaben, die in den meisten Projekten durchgeführt werden, und ordnet sie zu einer prinzipiellen Vorgehensweise in Form eines Leitfadens. Sie verweist dabei auf die jeweiligen Kapitel dieses Handbuchs oder weiterer Handbücher.

Vorgehen	Beschreibung
Automatisierungslösung entwerfen	M7 spezifisch; siehe: PHB M7-SYS RT
STEP 7 starten	wie bei STEP 7
Projektstruktur anlegen Station einrichten Hardware konfigurieren	wie bei STEP 7
Kommunikationsverbindungen projektieren	wie bei STEP 7
Symboltabelle definieren	wie bei STEP 7
C- oder CFC-Anwenderprogramm erstellen	M7 spezifisch; siehe: ProC/C++
Betriebssystem konfigurieren Betriebssystem auf M7-300/400 installieren Hardwarekonfiguration und Anwenderprogramm in M7 laden	M7 spezifisch; siehe: BHB M7-SYS Rt
Anwenderprogramm testen und debuggen	ProC/C++
Betrieb überwachen und M7 diagnostizieren	wie bei STEP 7, jedoch ohne anwenderdefinierte Diagnose
Drucken und Archivieren	wie bei STEP 7

Was ist anders?

Für M7-300/400 werden in STEP 7 folgende Funktionen nicht unterstützt:

- Multicomputing - Synchroner Betrieb mehrerer CPUs
- Forcen von Variablen
- Globaldaten-Kommunikation
- Anwenderdefinierte Diagnose

Verwalten der M7-Zielsysteme

STEP 7 unterstützt Sie speziell bei folgenden Arbeiten mit M7-Automatisierungssystemen:

- Betriebssystem auf dem M7-300/400 installieren
- Konfigurierung des Betriebssystems durch Editieren von Systemdateien
- Übertragen von Anwenderprogrammen in den M7-300/400
- Firmware aktualisieren

Um in die M7-Zielsystemverwaltung zu gelangen, rufen Sie aus dem Kontext eines Projekts, das Stationen mit M7-CPU's oder -FM's enthält, mit angewähltem M7-Programmordner, folgenden Menübefehl auf:

Zielsystem > M7-Zielsystem verwalten

Genaue Anleitungen finden Sie in der Online-Hilfe und im Benutzerhandbuch M7-SYS RT.

25.2 Optionssoftware für M7-Programmierung

M7-Optionssoftware

STEP 7 bietet Ihnen die Grundfunktionalität an, die Sie benötigen um:

- Projekte zu erzeugen und zu verwalten
- Hardware zu konfigurieren und zu parametrieren
- Netzwerke und Verbindungen zu projektieren
- Symboldaten zu verwalten

Diese Funktionalität ist unabhängig davon, ob Ihr Automatisierungssystem SIMATIC S7 oder SIMATIC M7 ist.

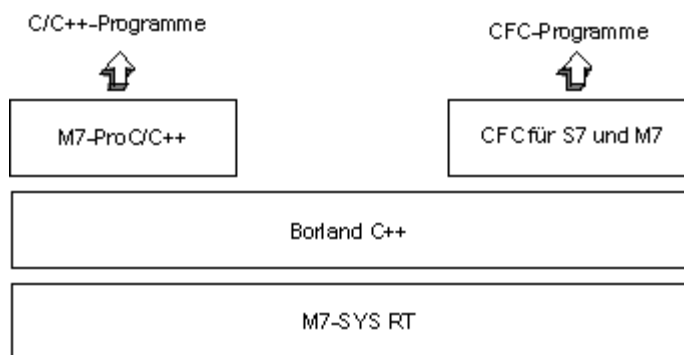
Für die Erstellung von M7-Anwenderprogrammen benötigen Sie zusätzlich zu STEP 7 noch M7-Optionssoftware:

Software	Inhalt
M7-SYS RT	<ul style="list-style-type: none"> • Betriebssystem M7 RMOS32 • Systembibliothek M7-API • MPI-Unterstützung
CFC für S7 und M7	Programmiersoftware für CFC-Anwendungen (CFC = Continuous Function Chart)
M7-ProC/C++	<ul style="list-style-type: none"> • Einbindung der Borland-Entwicklungsumgebung in STEP 7 • Symbolimport-Editor und -Generator • Hochsprachen-Debugger Organon xdb386
Borland C++	Borland C++-Entwicklungsumgebung

Zusammen mit der M7-Optionssoftware unterstützt Sie STEP 7 zusätzlich bei folgenden Tätigkeiten:

- Daten auf das den M7 über MPI übertragen
- Informationen über das M7-Automatisierungssystem abfragen
- bestimmte Einstellungen auf dem M7-Automatisierungssystem vornehmen und den M7 urlöschen

Das folgende Bild zeigt die Abhängigkeiten der M7-Optionssoftware für die M7-Programmierung.



Zusammenfassung

Für die Erstellung von ...	benötigen Sie die M7-Optionssoftware ...
C/C++-Programmen	<ul style="list-style-type: none"> • M7-SYS RT • M7-ProC/C++ • Borland C++
CFC-Programmen	<ul style="list-style-type: none"> • M7-SYS RT • CFC für S7 und M7 • Borland C++

Wo Sie Unterstützung finden

Die spezifischen Werkzeuge für die Erstellung von M7-Anwendungen sind teilweise in STEP 7 und teilweise in der M7-Optionssoftware integriert.

Aus der folgenden Tabelle erfahren Sie, welche Unterstützung Sie in den einzelnen Softwarepaketen finden:

Software ...	Unterstützt Sie ...
STEP 7	<ul style="list-style-type: none"> • beim Installieren des M7-Betriebssystems, • bei der Verwaltung des M7-Automatisierungssystems, • beim Übertragen, Starten und Löschen der M7-Anwenderprogramme • beim Abrufen von Zustands- und Diagnosedaten • beim Umräumen der CPU
M7-SYS RT	<p>durch die Dienste des M7-Betriebssystems und der -Systemsoftware bei:</p> <ul style="list-style-type: none"> • Programmablaufsteuerung, • Speicher- und Ressourcenverwaltung, • Zugriff auf Rechner- und SIMATIC-Hardware, • Alarmverwaltung, • Diagnose, • Zustandsüberwachung und • Kommunikation
M7-ProC/C++	<ul style="list-style-type: none"> • durch integrierte Code-Erzeugung (Integration der Borland-Entwicklungsumgebung in STEP 7) • durch Einbindung der Projekt-Symbolik in den Quellcode und • durch integrierte Debug-Funktionalität
Borland C++	<ul style="list-style-type: none"> • beim Erstellen von C- und C++-Programmen
CFC für S7 und M7	<ul style="list-style-type: none"> • beim Erstellen, Testen und Debuggen von CFC-Programmen und • beim Starten und Ablaufen von CFC-Programmen

25.3 Betriebssysteme für M7-300/400

Für Anwendungen, die in den Hochsprachen C und C++ erstellt werden, sind die Dienste des Betriebssystems von entscheidender Bedeutung. Das Betriebssystem übernimmt für die Anwendung folgende Aufgaben:

- Zugriff auf die Hardware
- Verwaltung der Ressourcen
- Systemeinbindung
- Kommunikation zu anderen Komponenten im System

Für die Lösung von Automatisierungsaufgaben setzen wir auf dem Automatisierungsrechner SIMATIC M7 das Echtzeitbetriebssystem M7 RMOS32 (RMOS = **R**ealtime-**M**ultitasking-**O**perating-**S**ystem) ein. Für die Einbindung in das SIMATIC-System ist M7 RMOS32 um eine Aufrufchnittstelle, das M7-API (API = **A**pplication **P**rogramming **I**nterface) erweitert.

Das Echtzeitbetriebssystem M7 RMOS32 wird für 32-Bit-Anwendungen zur Lösung zeitkritischer Echtzeit- und Multitasking-Aufgaben eingesetzt. Es ist für M7-Baugruppen in folgenden Konfigurationen verfügbar:

- M7 RMOS32
- M7 RMOS32 mit MS-DOS

Welche Betriebssystem-Konfiguration Sie für Ihr M7-Automatisierungssystem auswählen hängt von den M7-Baugruppen, die Sie einsetzen, ab:

Betriebssystem-Konfiguration	Baugruppe / Hauptspeicher	PROFIBUS-DP und TCP/IP Ja/Nein	Installation auf Massenspeicher
M7 RMOS32	FM 356-4 / 4MB FM 356-4 / 8MB CPU 388-4 / 8MB FM 456-4 / 16MB CPU 488-3 / 16MB CPU 486-3 / 16MB	nein ja ja ja ja ja	Memory Card ≥4 MB oder Festplatte
M7 RMOS32 mit MS-DOS	FM 356-4 / 8MB CPU 388-4 / 8MB FM 456-4 / 16MB CPU 488-3 / 16MB CPU 486-3 / 16MB	nein nein ja ja ja	Memory Card ≥4 MB oder Festplatte

26 Tipps und Tricks

26.1 Baugruppen tauschen in der Konfigurationstabelle

Wenn Sie eine Stationskonfiguration mit HW Konfig überarbeiten und eine Baugruppen tauschen möchten, z. B. gegen eine mit neuer Bestell-Nr., dann gehen Sie folgendermaßen vor:

1. Ziehen Sie die Baugruppe per Drag & Drop aus dem Fenster "Hardware Katalog" über die bereits platzierte ("alte") Baugruppe.
2. Lassen Sie die neue Baugruppe "fallen"; die neue Baugruppe übernimmt, soweit möglich, die Parameter von der bereits gesteckten.

Diese Vorgehensweise ist schneller als das Austauschen durch Löschen der alten und anschließendem Einfügen und Parametrieren der neuen Baugruppe.

Diese Funktion können Sie über den Menübefehl **Extras > Einstellungen** ("Baugruppen austauschen ermöglichen") in HW Konfig gezielt ein- bzw. ausschalten.

26.2 Projekte mit einer großen Anzahl vernetzter Stationen

Wenn Sie nacheinander alle Stationen konfigurieren und dann über **Extras > Netz konfigurieren** NetPro aufrufen, um Verbindungen zu projektieren, dann werden die Stationen automatisch in der Netzsicht platziert. Das hat den Nachteil, dass Sie die Stationen und Subnetze nachträglich nach topologischen Kriterien ordnen müssen.

Falls Ihr Projekt eine große Anzahl vernetzter Stationen umfasst und Sie Verbindungen zwischen diesen Stationen projektieren wollen, dann sollten Sie von Anfang an die Anlagenstruktur in der Netzsicht projektieren, um die Übersicht zu bewahren:

1. Erzeugen Sie das neue Projekt im SIMATIC Manager (Menübefehl **Datei > Neu**):
2. Starten Sie NetPro (Menübefehl **Extras > Netz konfigurieren**):

3. Erzeugen Sie in NetPro Station für Station:
 - Plazieren Sie die Station per Drag & Drop aus dem Fenster "Katalog":
 - Doppelklicken Sie auf die Station, um HW Konfig zu starten:
 - Plazieren Sie in HW Konfig per Drag & Drop die kommunikationsfähigen Baugruppen (CPUs, CPs, FMs, IF-Module):
 - Falls Sie diese Baugruppen vernetzen wollen: Doppelklicken Sie auf die entsprechenden Zeilen in der Konfigurationstabelle, um Subnetze neu anzulegen und die Schnittstellen zu vernetzen:
 - Speichern Sie die Konfiguration und wechseln Sie zu NetPro:
 - Positionieren Sie in NetPro Stationen und Subnetze (Objekt mit Mauszeiger verschieben, bis die gewünschte Position erreicht ist):
4. Projektieren Sie in NetPro die Verbindungen und korrigieren Sie ggf. die Vernetzung:

26.3 Reorganisation

Falls unerklärliche Probleme beim Arbeiten mit STEP 7 auftreten, hilft es oft, die Datenhaltung des Projekts oder der Bibliothek zu reorganisieren.

Dazu wählen Sie den Menübefehl **Datei > Reorganisieren**. Bei der Reorganisation werden die durch Löschen entstandenen Lücken beseitigt, d. h. der Speicherbedarf der Projekt-/Bibliotheksdaten wird verkleinert.

Die Funktion optimiert die Datenablage für das Projekt bzw. die Bibliothek ähnlich wie z. B. ein Programm zum Defragmentieren einer Festplatte die Datenablage auf der Festplatte optimiert.

Die Dauer des Reorganisierens hängt von den erforderlichen Datenbewegungen ab und kann längere Zeit in Anspruch nehmen. Die Funktion wird deshalb nicht automatisch (z. B. beim Schließen eines Projekts) durchgeführt, sondern muss vom Benutzer gestartet werden, wenn er das Projekt oder die Bibliothek reorganisieren möchte.

Voraussetzung

Projekte und Bibliotheken können nur dann reorganisiert werden, wenn keine Objekte daraus von anderen Applikationen bearbeitet werden und somit gesperrt sind.

26.4 Symbole über mehrere Netzwerke bearbeiten

Im Programmreditor KOP/AWL/FUP können Sie die Symbole mehrerer Netzwerke anzeigen lassen und bearbeiten.

1. Markieren Sie ein Netzwerk, indem Sie auf die Netzwerkbezeichnung (z. B. "Netzwerk 1") klicken.
2. Halten Sie die CTRL-Taste gedrückt und markieren Sie weitere Netzwerke.
3. Wählen Sie über die rechte Maustaste den Kontextmenübefehl **Symbole bearbeiten**.

Um alle Netzwerke eines Baustein zu markieren, drücken Sie CTRL+A und markieren eine Netzwerkbezeichnung.

26.5 Testen mit der Variablen-tabelle

Für das Beobachten und Steuern von Variablen in der Variablen-tabelle gibt es eine Reihe von Bearbeitungstipps:

- Symbole und Operanden können Sie sowohl in die Spalte "Symbol" als auch in die Spalte "Operand" eintragen. Der Eintrag wird automatisch in die passende Spalte geschrieben.
- Damit Ihnen der gesteuerte Wert angezeigt wird, sollten Sie den Triggerpunkt für "Beobachten" auf "Beginn Zyklus" und den Triggerpunkt für "Steuern" auf "Ende Zyklus" einstellen.
- Wenn Sie den Cursor in eine rot markierte Zeile platzieren, informiert Sie die Kurzinformation über die Ursache des Fehlers. Mit der Taste F1 erhalten Sie Hinweise zur Behebung.
- Sie können nur solche Symbole eintragen, die bereits in der Symbol-tabelle definiert sind. Ein Symbol muss genau so eingegeben werden, wie es in der Symbol-tabelle definiert ist. Symbolnamen, die Sonderzeichen enthalten, müssen in Hochkommas eingeschlossen werden (z. B. "Motor.Aus", "Motor+Aus", "Motor-Aus").
- Ausschalten von Warnhinweisen im Register "Online" (Dialogfeld "Einstellungen").
- Wechseln der Verbindung ist möglich, ohne vorher die Verbindung abgebaut zu haben.
- Der Trigger für Beobachten ist während des Beobachtens von Variablen einstellbar.
- Sie können einzelne ausgewählte Variablen steuern, indem Sie die betreffenden Zeilen markieren und die Funktion "Steuern" ausführen. Nur die sichtbaren markierten Variablen werden dann gesteuert.
- Beenden ohne Rückfrage:

Wenn Sie - bei laufendem "Beobachten", "Steuern" "PA-freischalten"- die Taste ESC drücken, werden "Beobachten" und "Steuern" ohne Rückfrage beendet.

- Eingeben eines zusammenhängenden Operandenbereichs:
Verwenden Sie den Menübefehl **Einfügen > Bereich**.
- Ein- und Ausblenden von Spalten:
Verwenden Sie die folgenden Menübefehle, um einzelne Spalten ein- oder auszublenden:
Symbol: Menübefehl **Ansicht > Symbol**
Symbolkommentar: Menübefehl **Ansicht > Symbolkommentar**
Darstellungsformat des Statuswertes: Menübefehl **Ansicht > Anzeigeformat**
Statuswert der Variablen: Menübefehl **Ansicht > Statuswert**
Steuerwert der Variablen: Menübefehl **Ansicht > Steuerwert**
- Ändern des Anzeigeformats mehrerer Tabellenzeilen gleichzeitig:
 1. Markieren Sie den Bereich der Tabelle, in dem Sie das Anzeigeformat ändern wollen, indem Sie mit gedrückter linker Maustaste über den gewünschten Tabellenbereich ziehen.
 2. Wählen Sie die Darstellung mit dem Menübefehl **Ansicht > Anzeigeformat wählen**. Es wird nur für die Zeilen der markierten Tabelle das Format gewechselt, für die der Formatwechsel erlaubt ist.
- Eingabebeispiele über die Taste F1:
 - Wenn Sie den Cursor in die Operandenspalte platzieren und F1 drücken, erhalten Sie Beispiele für die Eingabe von Operanden.
 - Wenn Sie den Cursor in die Steuerwertspalte platzieren und F1 drücken, erhalten Sie Beispiele für die Eingabe von Steuer-/Forcewerten.

26.6 Variablen vom Programmmeditor aus steuern

Sie können binäre Eingänge und Merker im Programmmeditor mit einer Schaltfläche versehen, die es Ihnen ermöglicht, diese Operanden durch einen Mausklick auf die Schaltfläche einfach und schnell zu steuern.

Voraussetzungen

- Sie haben in der Symboltabelle dem zu steuernden Operanden über den Menübefehl **Spezielle Objekteigenschaften > Bedienen am Kontakt** diese Eigenschaft zugewiesen.
- Sie haben im Programmmeditor KOP/AWL/FUP im Register "Allgemein (Menübefehl **Extras > Einstellungen**) die Option "Bedienen am Kontakt" ausgewählt.
- Sie haben den Menübefehl **Test > Beobachten** gewählt.

Triggerbedingung ist dabei "permanent/am Zyklusanfang".

In Ihrer Anlage tatsächlich vorhandene Eingänge werden solange gesteuert, wie die Schaltfläche gedrückt bleibt. Sie können über Mehrfachselektion (CTRL-Taste) auch mehrere Eingänge gleichzeitig steuern.

Bei Merkern bzw. nicht vorhandenen Eingängen wird beim Drücken der Schaltfläche der Status auf 1 gesetzt. Der Status wird erst dann wieder auf 0 zurückgesetzt, wenn Sie dies explizit über das Kontextmenü oder die Variablen-tabelle veranlassen oder wenn der Operand durch das S7-Programm zurückgesetzt wird.

Bei nicht negiertem Eingang oder Merker wird durch das Drücken der Schaltfläche mit dem Steuerwert "1" gesteuert, bei negiertem Eingang oder Merker mit "0".

Hinweis für WinCC

Wurde der Programmmeditor von WinCC aus über das Bedienen und Beobachten einer Variablen gestartet, ist nur die Steuermöglichkeit von WinCC erlaubt. Es sei denn, der Bediener hat "Instandhaltungsrechte" von WinCC, in diesem Fall sind beide Steuermöglichkeiten erlaubt.

26.7 Virtueller Arbeitsspeicher

Ein Grund für das Fehlverhalten von STEP 7 kann ein zu kleiner virtueller Arbeitsspeicher sein.

Zum Arbeiten mit STEP 7 sollten Sie die Einstellung für den virtuellen Arbeitsspeicher anpassen. Gehen Sie folgendermaßen vor:

1. Öffnen Sie die Systemsteuerung z. B. aus der Startleiste mit **Start > Einstellungen > Systemsteuerung** und doppelklicken Sie auf das Symbol "System".
Nur XP: Öffnen Sie unter **START > Arbeitsplatz > Eigenschaften > Erweitert > Systemleistung > Einstellungen**.
2. Wählen Sie unter Windows 2000 im angezeigten Dialogfeld das Register "Erweitert" und betätigen Sie die Schaltfläche "Systemleistungsoptionen".
Wählen Sie unter Windows XP/Server 2003 im Dialogfeld "Systemeinstellungen" das Register "Erweitert".
3. Betätigen Sie die Schaltfläche "Ändern".
4. Geben Sie als "Minimum" bzw. "Anfangsgröße" mindestens 40 MByte und als "Maximum" bzw. "Maximalgröße" mindestens 150 MByte an.

Hinweis

Da sich der virtuelle Arbeitsspeicher auf der Platte befindet (Default C: und dynamisch) soll darauf geachtet werden, dass für das Verzeichnis TMP bzw. TEMP ausreichend Platz zur Verfügung steht (ca. 20 - 30 MB):

- Sollte sich das S7-Projekt ebenfalls auf der selben Partition befinden, auf dem der virtuelle Arbeitsspeicher eingestellt wurde, so sollte ca. 2facher Speicherbedarf des S7-Projekts bereitgehalten werden.
 - Wenn das Projekt jedoch auf einer anderen Partition verwaltet wird, so entfällt diese Bedingung.
-

A Anhang

A.1 Betriebszustände

A.1.1 Betriebszustände und Übergänge

Betriebszustände

Betriebszustände beschreiben das Verhalten der CPU zu jedem beliebigen Zeitpunkt. Das Wissen über die Betriebszustände der CPUs ist nützlich für die Programmierung des Anlaufs, den Test der Steuerung sowie für die Fehlerdiagnose.

Die S7-300- und S7-400-CPU's können folgende Betriebszustände annehmen:

- STOP
- ANLAUF
- RUN
- HALT

Im Betriebszustand "STOP" prüft die CPU ob alle konfigurierten bzw. über Default-Adressierung eingesetzten Baugruppen vorhanden sind und setzt die Peripherie in einen vordefinierten Grundzustand. Das Anwenderprogramm wird im Betriebszustand "STOP" nicht bearbeitet.

Im Betriebszustand "ANLAUF" unterscheidet man zwischen den Anlaufarten "Neustart" (Warmstart), "Kaltstart" und Wiederanlauf:

- Bei Neustart (Warmstart) wird die Programmbearbeitung am Programmumfang mit einer "Grundstellung" der Systemdaten und der Anwenderoperandenbereiche neu begonnen (die nicht remanenten Zeiten, Zähler und Merker werden zurückgesetzt).
- Bei Kaltstart wird das Prozessabbild der Eingänge eingelesen und das STEP 7- Anwenderprogramm beginnend beim ersten Befehl im OB1 bearbeitet (gilt auch für Neustart (Warmstart)).
 - Per SFC erzeugte Datenbausteine im Arbeitsspeicher werden gelöscht, die übrigen Datenbausteine haben den vorbelegten Wert aus dem Ladespeicher
 - Das Prozessabbild sowie alle Zeiten, Zähler und Merker werden zurückgesetzt, unabhängig davon, ob sie als remanent parametrisiert worden sind.

- Bei Wiederanlauf wird die Programmbearbeitung an der unterbrochenen Stelle fortgesetzt (Zeiten, Zähler und Merker werden nicht zurückgesetzt). Wiederanlauf ist nur bei S7-400-CPUs möglich.

Im Betriebszustand "RUN" bearbeitet die CPU das Anwenderprogramm, aktualisiert die Ein- und Ausgänge, bearbeitet Alarmer und Fehlermeldungen.

Im Betriebszustand "HALT" wird die Bearbeitung des Anwenderprogramms angehalten, und Sie können das Anwenderprogramm schrittweise testen. Der Betriebszustand "HALT" kann nur beim Testen mit dem PG erreicht werden.

In all diesen Betriebszuständen ist die CPU über die MPI-Schnittstelle kommunikationsfähig.

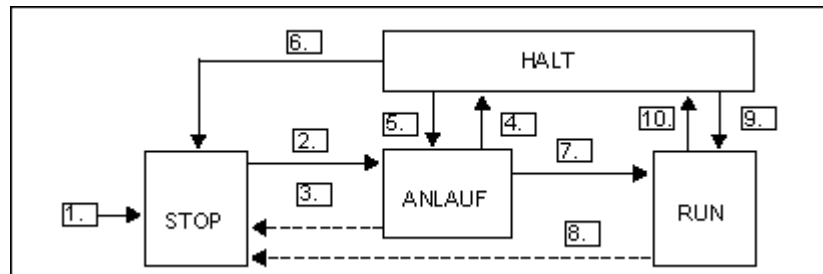
Weitere Betriebszustände

Ist die CPU nicht betriebsbereit, befindet sie sich in einem der beiden Betriebszustände:

- Spannungslos, d. h. die Netzspannung ist ausgeschaltet.
- Defekt, d. h. es ist ein nicht behebbarer Fehler aufgetreten.
Überprüfen Sie, ob die CPU tatsächlich defekt ist: Setzen Sie die CPU auf STOP, und schalten Sie den Netzschalter aus und wieder ein. Wenn die CPU anläuft, lesen Sie den Diagnosepuffer aus, um den Fehler zu analysieren. Wenn die CPU nicht anläuft, muss sie ausgetauscht werden.

Betriebszustandsübergänge

Nachfolgendes Bild zeigt die Betriebszustände und die Betriebszustandsübergänge der S7-300- und S7-400-CPUs:



Die Bedingungen, unter welchen die Betriebszustände wechseln, können Sie folgender Tabelle entnehmen.

Übergang	Beschreibung
1.	Nach dem Einschalten der Versorgungsspannung befindet sich die CPU im Betriebszustand "STOP".
2.	Die CPU geht in den Betriebszustand "ANLAUF" <ul style="list-style-type: none"> • nachdem die CPU über Betriebsartenschalter oder vom PG aus in RUN oder RUN-P gebracht wurde oder • nach automatischer Auslösung einer Anlaufart durch NETZ-EIN. • Wenn die Kommunikationsfunktion "RESUME" oder "START" ausgeführt wird. Der Betriebsartenschalter muss in beiden Fällen auf RUN oder RUN-P stehen.
3.	Die CPU geht wieder in den STOP-Zustand, wenn <ul style="list-style-type: none"> • während des Anlaufs ein Fehler erkannt wird • die CPU über Betriebsartenschalter oder vom PG aus auf STOP gesetzt wird • ein Stopp-Befehl im Anlauf-OB bearbeitet wird oder • die Kommunikationsfunktion "STOP" ausgeführt wird.
4.	Die CPU geht in den Betriebszustand "HALT", wenn im Anlaufprogramm ein Haltepunkt erreicht wird.
5.	Die CPU geht in den Betriebszustand "ANLAUF", wenn der Haltepunkt in einem Anlaufprogramm gesetzt war und der Befehl "HALT VERLASSEN" ausgeführt wird (Testfunktion).
6.	Die CPU geht wieder in den STOP-Zustand, wenn <ul style="list-style-type: none"> • die CPU über Betriebsartenschalter oder vom PG aus auf STOP gesetzt wird oder • der Kommunikationsbefehl "STOP" ausgeführt wird.
7.	Ist der Anlauf erfolgreich, dann wechselt die CPU in RUN.
8.	Die CPU geht wieder in den STOP-Zustand, wenn <ul style="list-style-type: none"> • im RUN-Zustand ein Fehler erkannt wird und der zugehörige OB nicht geladen ist • die CPU über Betriebsartenschalter oder vom PG aus auf STOP gesetzt wird • ein Stopp-Befehl im Anwenderprogramm bearbeitet wird oder • die Kommunikationsfunktion "STOP" ausgeführt wird.
9.	Die CPU geht in den Betriebszustand "HALT", wenn im Anwenderprogramm ein Haltepunkt erreicht wird.
10.	Die CPU geht in den Betriebszustand "RUN", wenn ein Haltepunkt gesetzt war und der Befehl "HALT VERLASSEN" ausgeführt wird.

Priorität der Betriebszustände

Werden mehrere Betriebszustandswechsel gleichzeitig angefordert, wird in den Betriebszustand mit der höchsten Priorität gewechselt. Steht z. B. der Betriebsartenschalter auf RUN, und vom PG aus wird versucht, die CPU in STOP zu schalten, geht die CPU in STOP, weil dieser Betriebszustand die höchste Priorität hat.

Priorität	Betriebszustand
Höchste	STOP
	HALT
	ANLAUF
Niedrigste	RUN

A.1.2 Betriebszustand STOP

Im Betriebszustand "STOP" wird das Anwenderprogramm nicht bearbeitet. Alle Ausgänge werden auf Ersatzwerte gesetzt und damit der gesteuerte Prozess in einen sicheren Betriebszustand gebracht. Die CPU prüft, ob

- Hardwareprobleme vorliegen (z. B. Baugruppen nicht verfügbar sind)
- für die CPU die Defaulteinstellung gelten soll oder Parametersätze vorliegen
- die Randbedingungen für das programmierte Anlaufverhalten stimmen
- Systemsoftwareprobleme vorliegen.

Im STOP-Zustand können auch Globaldaten empfangen werden, und es kann passiv einseitige Kommunikation über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ausgeführt werden.

Urlöschen

Im STOP-Zustand kann die CPU urlöscht werden. Urlöschen kann manuell über den Betriebsartenschalter (MRES) oder vom PG aus (z. B. vor dem Laden eines Anwenderprogramms) erfolgen.

Durch Urlöschen wird die CPU in den "Urzustand" versetzt, d. h.

- das gesamte Anwenderprogramm im Arbeitsspeicher und im RAM-Ladespeicher sowie alle Operandenbereiche werden gelöscht.
- die Systemparameter sowie die CPU- und Baugruppenparameter werden auf die Defaulteinstellung zurückgesetzt. Die vor dem Urlöschen eingestellten MPI-Parameter bleiben erhalten.
- wenn eine Memory Card (Flash-EPROM) gesteckt ist, kopiert die CPU das Anwenderprogramm aus der Memory Card in den Arbeitsspeicher (inkl. CPU- und Baugruppenparameter, falls die entsprechenden Konfigurationsdaten sich ebenfalls auf der Memory Card befinden).

Der Diagnosepuffer, die MPI-Parameter, die Uhrzeit und der Betriebsstundenzähler werden nicht zurückgesetzt.

A.1.3 Betriebszustand ANLAUF

Bevor die CPU nach dem Einschalten mit der Bearbeitung des Anwenderprogramms beginnt, wird ein Anlaufprogramm bearbeitet. Im Anlaufprogramm können Sie durch entsprechende Programmierung von Anlauf-OBs bestimmte Voreinstellungen für Ihr zyklisches Programm festlegen.

Es gibt drei Anlaufarten: Neustart (Warmstart), Kaltstart und Wiederanlauf. Ein Wiederanlauf ist grundsätzlich nur bei S7-400-CPU's möglich. Er muss im Parametersatz der CPU mit STEP 7 festgelegt worden sein.

Im Betriebszustand "ANLAUF":

- wird das Programm im Anlauf-OB (OB 100 für Neustart (Warmstart), OB 101 für Wiederanlauf, OB 102 für Kaltstart) abgearbeitet
- ist keine zeit- und alarmgesteuerte Programmbearbeitung möglich
- werden die Zeiten aktualisiert
- läuft der Betriebsstundenzähler
- sind die Digitalausgänge auf Signalbaugruppen gesperrt, können aber über Direktzugriff gesetzt werden.

Neustart (Warmstart)

Ein Neustart (Warmstart) ist immer zulässig, es sei denn, vom System wurde Urlöschen angefordert. In folgenden Fällen ist nur Neustart (Warmstart) möglich, nach:

- Urlöschen
- dem Laden des Anwenderprogramms im STOP-Zustand der CPU
- USTACK/BSTACK-Überlauf
- Abbruch des Neustart (Warmstart) (durch NETZ-AUS oder über Betriebsartenschalter)
- Überschreiten der parametrisierten Unterbrechungszeitgrenze für Wiederanlauf.

Manueller Neustart (Warmstart)

Ein manueller Neustart (Warmstart) kann ausgelöst werden:

- über den Betriebsartenschalter
(der Schalter CRST/WRST - soweit vorhanden - muss auf CRST stehen)
- über Menübefehl vom PG aus bzw. über Kommunikationsfunktionen
(wenn der Betriebsartenschalter auf RUN oder RUN-P steht)

Automatischer Neustart (Warmstart)

Ein automatischer Neustart (Warmstart) kann ausgelöst werden bei NETZ-EIN, wenn:

- die CPU bei NETZ-AUS nicht im STOP war
- der Betriebsartenschalter auf RUN oder RUN-P steht
- kein automatischer Wiederanlauf nach NETZ-EIN parametrier ist
- die CPU im Neustart (Warmstart) durch Netzausfall unterbrochen wurde (unabhängig von der Parametrierung der Anlaufart).

Der Schalter CRST/WRST ist bei automatischem Neustart (Warmstart) wirkungslos.

Ungepufferter automatischer Neustart (Warmstart)

Wird Ihre CPU ohne Pufferbatterie betrieben (falls wartungsfreier Betrieb notwendig ist), wird nach dem Einschalten oder bei Spannungswiederkehr nach NETZ-AUS die CPU automatisch urgelöscht und anschließend ein Neustart (Warmstart) durchgeführt. Das Anwenderprogramm muss auf Flash-EPROM (Memory Card) vorhanden sein.

Wiederanlauf

Nach Spannungsausfall im RUN und anschließender Spannungswiederkehr durchlaufen S7-400-CPUs eine Initialisierungsroutine und führen dann automatisch einen Wiederanlauf durch. Bei einem Wiederanlauf wird das Anwenderprogramm an der Stelle fortgesetzt, an der die Bearbeitung unterbrochen wurde. Der Teil des Anwenderprogramms, der vor Netzausfall nicht mehr bearbeitet wurde, wird als Restzyklus bezeichnet. Der Restzyklus kann auch zeit- und alarmgesteuerte Programmteile enthalten.

Ein Wiederanlauf ist prinzipiell nur zulässig, wenn das Anwenderprogramm im STOP-Zustand nicht verändert wurde (z. B. durch Nachladen eines geänderten Bausteins) bzw. wenn nicht aus anderen Gründen Neustart (Warmstart) erforderlich ist. Man unterscheidet zwischen manuellem und automatischem Wiederanlauf.

Manueller Wiederanlauf

Ein manueller Wiederanlauf ist nur möglich bei entsprechender Parametrierung im Parametersatz der CPU und nach den folgenden STOP-Ursachen:

- der Betriebsartenschalter wurde von RUN auf STOP gesetzt
- vom Anwender programmierte STOPs, STOPs nach Aufruf nicht geladener OBs
- STOP-Zustand wurde vom PG oder durch Kommunikationsfunktion herbeigeführt.

Ein manueller Wiederanlauf kann ausgelöst werden:

- über den Betriebsartenschalter
Der Schalter CRST/WRST muss auf WRST stehen.
- über Menübefehl vom PG aus bzw. über Kommunikationsfunktionen (wenn der Betriebsartenschalter auf RUN oder RUN-P steht).
- wenn im Parametersatz der CPU manueller Wiederanlauf parametrierung wurde.

Automatischer Wiederanlauf

Ein automatischer Wiederanlauf kann ausgelöst werden bei NETZ-EIN, wenn

- die CPU bei NETZ-AUS nicht im STOP oder HALT war
- der Betriebsartenschalter auf RUN oder RUN-P steht
- im Parametersatz der CPU automatischer Wiederanlauf nach NETZ-EIN parametrierung wurde.

Der Schalter CRST/WRST ist bei automatischem Wiederanlauf wirkungslos.

Remanente Datenbereiche nach Netzausfall

S7-300- und S7-400-CPU's reagieren unterschiedlich bei Spannungswiederkehr nach Netzausfall.

S7-300-CPU's (ausgenommen CPU318) kennen nur die Anlaufart "Neustart". Mit STEP 7 können Sie jedoch die Remanenz von Merkern, Zeiten, Zählern und Bereichen in Datenbausteinen festlegen, um Datenverlust bei Spannungsausfall zu vermeiden. Bei Spannungswiederkehr erfolgt ein "automatischer Neustart mit Gedächtnis".

S7-400-CPU's reagieren auf Spannungswiederkehr abhängig von der Parametrierung entweder mit Neustart (Warmstart) (nach gepuffertem oder ungepuffertem NETZ-EIN) oder mit Wiederanlauf (nur nach gepuffertem NETZ-EIN möglich).

Die folgende Tabelle zeigt das Remanenzverhalten der S7-300- und S7-400-CPU's bei Neustart (Warmstart), Kaltstart und Wiederanlauf:

X	bedeutet:	Daten bleiben erhalten
VC	bedeutet:	Codebaustein im EPROM bleibt erhalten, eventuell überladener Codebaustein geht verloren
VX	bedeutet:	Datenbaustein bleibt nur erhalten, wenn im EPROM vorhanden, remanente Daten werden aus NV-RAM übernommen (geladene oder erzeugte Datenbausteine im RAM gehen verloren)
0	bedeutet:	Daten werden zurückgesetzt bzw. gelöscht (Inhalt von DBs)
V	bedeutet:	Daten werden auf den Vorbesetzungswert aus dem EPROM-Speicher gesetzt
---	bedeutet:	nicht möglich, da kein NV-RAM vorhanden

Remanenzverhalten im Arbeitsspeicher (bei EPROM- und RAM-Ladespeicher):

			EPROM	(Mem.	Card	oder	Inte- griert)		
		CPU mit	Puffe- rung			CPU	ohne	Puffe- rung	
Daten	Bau- steine im Lade- speicher	DB im Arbeits- speicher	Merker, Zeiten, Zähler	Merker, Zeiten, Zähler	Bauste- ne im Lade- speicher	DB im Arbeits- speicher	DB im Arbeits- speicher	Merker, Zeiten, Zähler	Merker, Zeiten, Zähler
			(rema- nent para- metriert)	(nicht rema- nent para- metriert)		(rema- nent para- metriert)	(nicht rema- nent para- metriert)	(rema- nent para- metriert)	(nicht rema- nent para- metriert)
Neustart bei S7-300	X	X	X	0	VC	VX	V	X	0
Neustart bei S7-400	X	X	X	0	VC	---	V	0	0
Kaltstart bei S7-300	X	0	0	0	VC	V	V	0	0
Kaltstart bei S7-400	X	0	0	0	VC	---	V	0	0
Wieder- anlauf bei S7-400	X	X	X	X		Nur	Neustart	erlaubt	

Anlauftätigkeiten

Welche Tätigkeiten die CPU beim Anlauf durchführt, zeigt nachfolgende Tabelle:

Tätigkeiten in der Bearbeitungsreihenfolge	bei Neustart (Warmstart)	bei Kaltstart	bei Wiederanlauf
U-Stack/B-Stack löschen	X	X	0
Nicht remanente Merker, Zeiten, Zähler löschen	X	0	0
Alle Merker, Zeiten, Zähler löschen	0	X	0
Prozessabbild der Ausgänge löschen	X	X	parametrierbar
Ausgänge der Signalbaugruppen zurücksetzen	X	X	parametrierbar
Prozessalarme verwerfen	X	X	0
Verzögerungsalarme verwerfen	X	X	0
Diagnosealarme verwerfen	X	X	X
Systemzustandsliste (SZL) aktualisieren	X	X	X
Baugruppenparameter auswerten und an Baugruppen übergeben oder Defaultwerte übergeben	X	X	X
Bearbeitung des jeweiligen Anlauf-OB	X	X	X
Restzyklus (Teil des Anwenderprogramms, das aufgrund eines NETZ-AUS nicht weiterbearbeitet werden konnte) bearbeiten	0	0	X
Prozessabbild der Eingänge aktualisieren	X	X	X
Digitale Ausgänge freigeben (Signal OD aufheben) nach Betriebszustandsübergang in RUN	X	X	X
X bedeutet	wird durchgeführt		
0 bedeutet	wird nicht durchgeführt		

Abbrechen eines Anlaufs

Treten während des Anlaufs Fehler auf, wird der Anlauf abgebrochen, und die CPU geht oder bleibt in STOP.

Ein abgebrochener Neustart (Warmstart) muss wiederholt werden. Nach einem abgebrochenen Wiederanlauf ist sowohl ein Neustart (Warmstart) als auch ein Wiederanlauf möglich.

Ein Anlauf (Neustart (Warmstart) oder Wiederanlauf) wird nicht durchgeführt, oder er wird abgebrochen, wenn

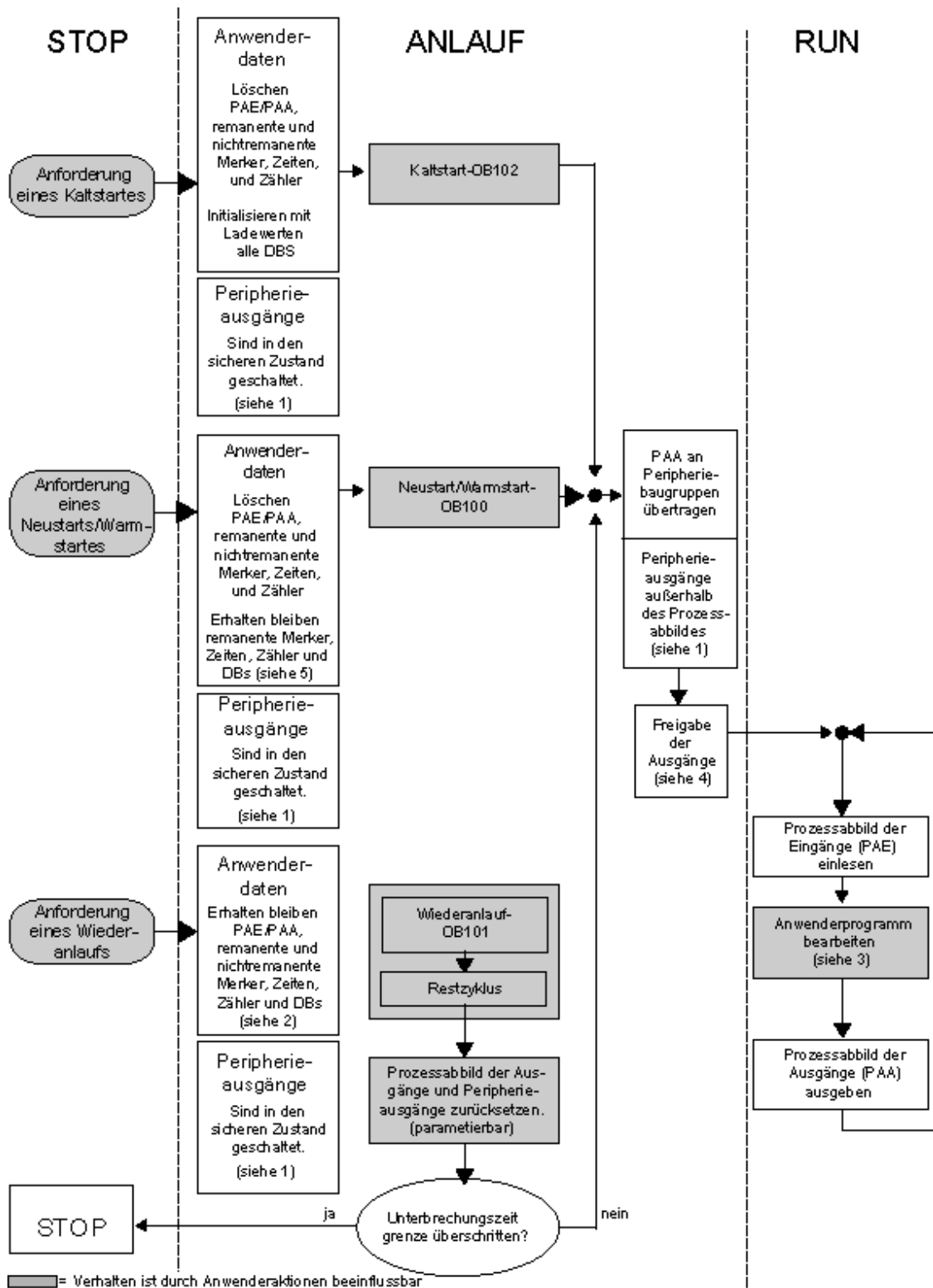
- der Betriebsartenschalter der CPU auf STOP steht
- Urlöschen angefordert wird
- eine Memory Card gesteckt ist, deren Anwendungskennung für STEP 7 nicht zulässig ist (z. B. STEP5)
- im Einzelprozessorbetrieb mehr als eine CPU gesteckt ist
- wenn das Anwenderprogramm einen OB enthält, den die CPU nicht kennt oder der gesperrt wurde
- nach dem Einschalten der Netzspannung die CPU feststellt, dass nicht alle Baugruppen, die in der mit STEP 7 erstellten Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind (Parametrierung Soll-Ist-Unterschied unzulässig)
- bei der Auswertung der Baugruppenparameter Fehler auftreten.

Ein Wiederanlauf wird auch nicht durchgeführt, oder er wird abgebrochen, wenn

- die CPU zuvor urgelöscht wurde (nach dem Urlöschen ist nur Neustart (Warmstart) zulässig)
- die Unterbrechungszeitgrenze überschritten ist (Die Unterbrechungszeit ist die Zeit, die nach dem Verlassen des RUN vergeht bis der Anlauf-OB inkl. Restzyklus bearbeitet wurde.)
- die Baugruppenkonfiguration geändert wurde (z. B. Baugruppentausch)
- durch Parametrierung nur Neustart (Warmstart) zugelassen ist
- wenn im STOP Bausteine geladen, gelöscht oder geändert wurden.

Ablauf

Nachfolgendes Bild zeigt die Tätigkeiten der CPU in den Betriebszuständen ANLAUF und RUN.



Legende zum Bild "Tätigkeiten der CPU in den Betriebszuständen ANLAUF und RUN":

1. Alle Peripherieausgänge sind durch die Peripheriebaugruppen hardwareseitig in den sicheren Zustand geschaltet (Defaultwert ="0"). Dies ist unabhängig davon, ob sie im Anwenderprogramm innerhalb oder außerhalb des Prozessabbildbereiches verwendet werden.
2. Beim Einsatz von ersatzwertfähigen Signalbaugruppen kann das Verhalten der Ausgänge parametrierbar werden, z. B. letzter Wert halten.
3. Ist notwendig für die Bearbeitung des Restzyklus.
4. Auch den Alarm-OBs steht beim erstmaligem Aufruf ein aktuelles Prozessabbild der Eingänge zur Verfügung.
5. Den Zustand der zentralen und dezentralen Peripherieausgänge im ersten Zyklus des Anwenderprogramms können Sie durch folgende Maßnahmen festlegen:
 - Parametrierbare Ausgabebaugruppen verwenden, um Ersatzwerte ausgeben zu können oder um den letzten Wert zu halten.
 - Bei Wiederanlauf: CPU-Anlauf-Parameter "Ausgänge zurücksetzen bei Wiederanlauf" aktivieren, um eine "0" auszugeben (entspricht der Voreinstellung).
 - Ausgänge im Anlauf-OB (OB 100, OB 101, OB 102) vorbesetzen.
6. In ungepufferten S7-300-Systemen bleiben nur die als remanent projektierten DB-Bereiche erhalten.

A.1.4 Betriebszustand RUN

Im Betriebszustand "RUN" erfolgt die zyklische, zeit- und alarmgesteuerte Programmbearbeitung:

- das Prozessabbild der Eingänge wird eingelesen
- das Anwenderprogramm wird abgearbeitet
- das Prozessabbild der Ausgänge wird ausgegeben.

Der aktive Austausch von Daten zwischen CPUs über die Globaldaten-Kommunikation (Globaldatentabelle) und über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ist nur im RUN-Zustand möglich.

Nachfolgende Tabelle zeigt beispielhaft, wann Datenaustausch in verschiedenen Betriebszuständen möglich ist:

Art der Kommunikation	Betriebszustand der CPU 1	Richtung des Datenaustauschs	Betriebszustand der CPU 2
Globaldaten-Kommunikation	RUN	↔	RUN
	RUN	→	STOP/HALT
	STOP	←	RUN
	STOP	X	STOP
	HALT	X	STOP/HALT
Einseitige Kommunikation	RUN	→	RUN
über Kommunikations-SFBs	RUN	→	STOP/HALT
Zweiseitige Kommunikation über Kommunikations-SFBs	RUN	↔	RUN
Einseitige Kommunikation	RUN	→	RUN
über Kommunikations-SFCs	RUN	→	STOP/HALT
Zweiseitige Kommunikation über Kommunikations-SFCs	RUN	↔	RUN
↔ bedeutet	Datenaustausch ist in beide Richtungen möglich		
→ bedeutet	Datenaustausch ist nur in eine Richtung möglich		
X	bedeutet Datenaustausch ist nicht möglich		

A.1.5 Betriebszustand HALT

Der Betriebszustand HALT nimmt eine Sonderstellung ein. Er wird nur zu Testzwecken im Anlauf oder RUN eingenommen. Im Betriebszustand HALT:

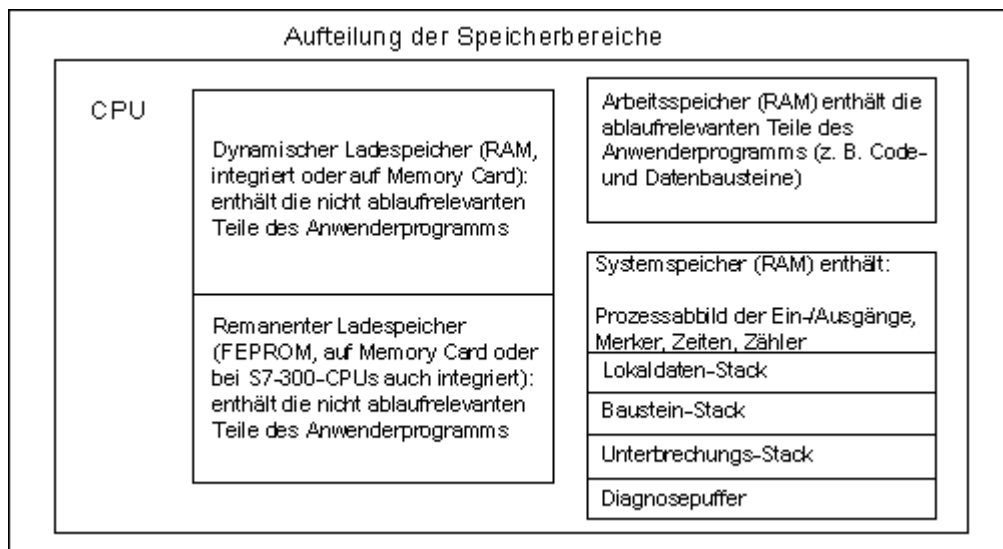
- werden alle Zeiten eingefroren: Zeiten und Betriebsstundenzähler werden nicht bearbeitet, Überwachungszeiten werden angehalten, die Grundtakte der zeitgesteuerten Ebenen werden angehalten.
- läuft die Echtzeituhr
- werden Ausgänge nicht freigeschaltet, können aber zu Testzwecken freigegeben werden
- können Ein- und Ausgänge gesteuert werden
- gehen gepufferte CPUs bei Netzausfall und -wiederkehr in HALT in STOP und führen keinen automatischen Wiederanlauf oder Neustart (Warmstart) aus. Ungepufferte CPUs führen bei Netzwiederkehr einen ungepufferten automatischen Neustart (Warmstart) aus.
- können auch Globaldaten empfangen und passiv einseitige Kommunikation über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ausgeführt werden (siehe auch Tabelle in Betriebszustand RUN)

A.2 Speicherbereiche von S7-CPU

A.2.1 Aufteilung der Speicherbereiche

Der Speicher der S7-CPU lässt sich in drei Bereiche aufteilen (siehe nachfolgendes Bild):

- Der Ladespeicher dient zur Aufnahme des Anwenderprogramms ohne symbolische Operandenzuordnung oder Kommentare (diese bleiben im Speicher des PG). Der Ladespeicher kann RAM- oder EPROM-Speicher sein.
- Bausteine, die als nicht ablaufrelevant gekennzeichnet sind, werden ausschließlich in den Ladespeicher aufgenommen.
- Der Arbeitsspeicher (integrierter RAM) dient zur Aufnahme der für den Programmablauf relevanten Teile des S7-Programms. Die Programmbearbeitung erfolgt ausschließlich im Bereich von Arbeitsspeicher und Systemspeicher.
- Der Systemspeicher (RAM) enthält die Speicherelemente, die jede CPU dem Anwenderprogramm zur Verfügung stellt, wie z. B. das Prozessabbild der Ein- und Ausgänge, Merker, Zeiten und Zähler. Außerdem enthält der Systemspeicher den Baustein-Stack und den Unterbrechungs-Stack.
- Der Systemspeicher der CPU stellt außerdem temporären Speicher (Lokaldaten-Stack) zur Verfügung, der dem Programm beim Aufrufen eines Bausteins für dessen temporäre Daten zugeordnet wird. Diese Daten sind nur solange gültig, wie der Baustein aktiv ist.



A.2.2 Lade- und Arbeitsspeicher

Wenn Sie das Anwenderprogramm vom Programmiergerät in die CPU laden, werden nur die Code- und Datenbausteine in den Lade- und den Arbeitsspeicher der CPU geladen.

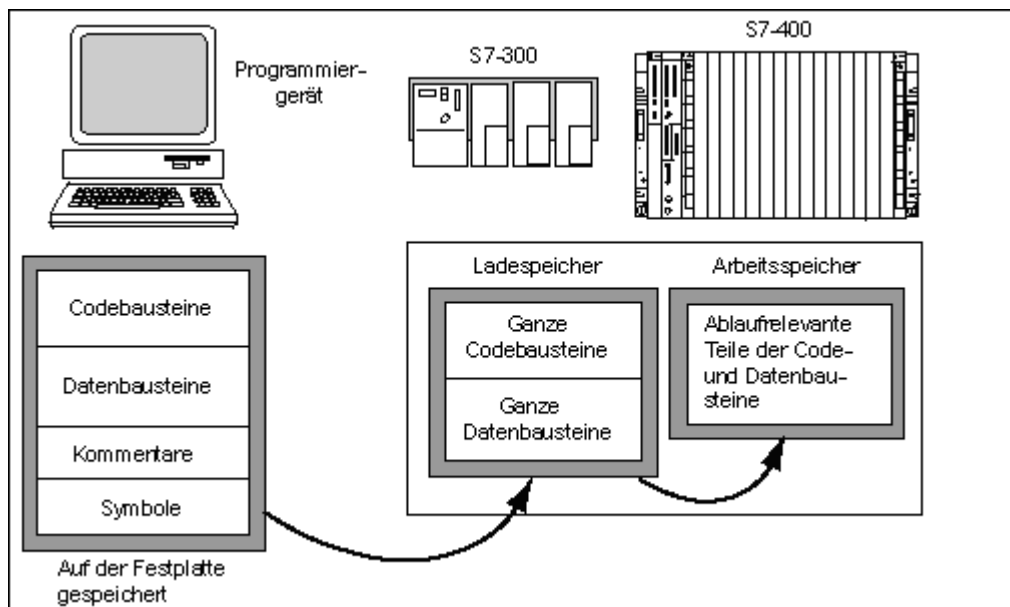
Die symbolische Operandenzuordnung (Symboltabelle) und die Bausteinkommentare bleiben im Speicherbereich des PG.

Aufteilung des Anwenderprogramms

Um eine schnelle Bearbeitung des Anwenderprogramms zu gewährleisten und den nicht erweiterbaren Arbeitsspeicher nicht unnötig zu belasten, werden nur die Teile der Bausteine, die für die Programmbearbeitung relevant sind, in den Arbeitsspeicher geladen.

Bausteinteile, die nicht erforderlich sind, um das Programm ablaufen zu lassen (z. B. Bausteinköpfe), bleiben im Ladespeicher.

Nachfolgendes Bild zeigt das Laden des Programms in den CPU-Speicher.



Hinweis

Datenbausteine, die mit Hilfe von Systemfunktionen (z. B. SFC 22 CREAT_DB) im Anwenderprogramm erstellt werden, speichert die CPU komplett in den Arbeitsspeicher.

Einige CPUs verfügen über getrennt verwaltete Bereiche für Code und Daten im Arbeitsspeicher. Größe und Belegung der Bereiche werden bei diesen CPUs auf der Registerseite "Speicher" des Baugruppenzustands angezeigt.

Kennzeichen von Datenbausteinen als "nicht ablaufrelevant"

Datenbausteine, die als Teil eines AWL-Programms in einer Quelldatei programmiert werden, können als "nicht ablaufrelevant" gekennzeichnet werden (Schlüsselwort UNLINKED). Das bedeutet, beim Laden in die CPU werden diese DBs nur im Ladespeicher abgelegt. Ihr Inhalt kann bei Bedarf mit Hilfe der SFC 20 BLKMOV in den Arbeitsspeicher kopiert werden.

Dadurch kann Platz im Arbeitsspeicher eingespart werden. Der erweiterbare Ladespeicher dient als Zwischenspeicher (z. B. für Rezepturen: nur die Rezeptur, die als nächste bearbeitet werden soll, wird in den Arbeitsspeicher geladen).

Struktur des Ladespeichers

Der Ladespeicher kann durch den Einsatz von Memory Cards erweitert werden. Die maximale Größe des Ladespeichers entnehmen Sie dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Baugruppendaten".

Bei S7-300-CPU's kann der Ladespeicher außer einem integrierten RAM- auch einen integrierten EPROM-Anteil haben. Bereiche in Datenbausteinen können durch Parametrierung mit STEP 7 remanent erklärt werden (siehe Remanente Speicherbereiche in S7-300-CPU's).

Bei S7-400-CPU's ist der Einsatz einer Memory Card (RAM oder EPROM) zur Erweiterung des Ladespeichers unerlässlich. Der integrierte Ladespeicher ist ein RAM-Speicher und dient im wesentlichen dem Nachladen und Korrigieren von Bausteinen. Bei neuen S7-400-CPU's kann auch zusätzlicher Arbeitsspeicher gesteckt werden.

Verhalten des Ladespeichers bei RAM- und EPROM-Bereichen

Je nachdem, ob Sie eine RAM- oder EPROM-Memory Card zur Erweiterung des Ladespeichers wählen, kann sich beim Laden, Nachladen und Utlöschen ein unterschiedliches Verhalten des Ladespeichers ergeben.

Nachfolgende Tabelle zeigt die Lademöglichkeiten:

Speicherart	Lademöglichkeiten	Ladeart
RAM	Laden und Löschen von einzelnen Bausteinen	PG-CPU-Verbindung
	Laden und Löschen eines gesamten S7-Programmes	PG-CPU-Verbindung
	Nachladen von einzelnen Bausteinen	PG-CPU-Verbindung
EPROM integriert (nur bei S7-300) oder steckbar	Laden von gesamten S7-Programmen	PG-CPU-Verbindung
EPROM steckbar	Laden von gesamten S7-Programmen	Laden des EPROMs auf dem PG und Stecken der Memory Card in die CPU Laden des EPROMs auf der CPU

Programme, die im RAM gespeichert sind, gehen verloren, wenn Sie die CPU urlöschen (MRES) bzw. wenn Sie die CPU oder die RAM-Memory Card ziehen.

Programme, die auf EPROM-Memory-Cards gespeichert sind, gehen beim Utlöschen nicht verloren und bleiben auch ohne Batteriepufferung erhalten (Transport, Sicherungskopien).

A.2.3 Systemspeicher

A.2.3.1 Verwenden der Systemspeicherbereiche

Der Systemspeicher der S7-CPU ist in Operandenbereiche aufgeteilt (siehe nachfolgende Tabelle). Durch Verwendung der entsprechenden Operationen adressieren Sie in Ihrem Programm die Daten direkt in den jeweiligen Operandenbereich.

Operandenbereich	Zugriff über Einheiten der folgenden Größe:	S7- Notation	Beschreibung
Prozessabbild der Eingänge	Eingang (Bit)	E	Zu Beginn jedes Zyklus liest die CPU die Eingänge aus den Eingabebaugruppen und speichert die Werte in das Prozessabbild der Eingänge.
	Eingangsbyte	EB	
	Eingangswort	EW	
	Eingangsdoppelwort	ED	
Prozessabbild der Ausgänge	Ausgang (Bit)	A	Während des Zyklus berechnet das Programm die Werte für die Ausgänge und legt sie im Prozessabbild der Ausgänge ab. Am Ende des Zyklus schreibt die CPU die errechneten Ausgangswerte in die Ausgabebaugruppen.
	Ausgangsbyte	AB	
	Ausgangswort	AW	
	Ausgangsdoppelwort	AD	
Merker	Merker (Bit)	M	Dieser Bereich stellt Speicherplatz für im Programm errechnete Zwischenergebnisse zur Verfügung.
	Merkerbyte	MB	
	Merkerwort	MW	
	Merkerdoppelwort	MD	
Zeiten	Zeit (T)	T	In diesem Bereich stehen Zeiten zur Verfügung.
Zähler	Zähler (Z)	Z	In diesem Bereich stehen Zähler zur Verfügung.
Datenbaustein	Datenbaustein, geöffnet mit "AUF DB":	DB	Datenbausteine speichern Informationen für das Programm. Sie können entweder so definiert sein, dass alle Codebausteine auf sie zugreifen können (globale DBs), oder sie sind einem bestimmten FB oder SFB zugeordnet (Instanz-DB).
	Datenbit	DBX	
	Datenbyte	DBB	
	Datenwort	DBW	
	Datendoppelwort	DBD	

Operandenbereich	Zugriff über Einheiten der folgenden Größe:	S7- Notation	Beschreibung
	Datenbaustein, geöffnet mit "AUF DI":	DI	
	Datenbit	DIX	
	Datenbyte	DIB	
	Datenwort	DIW	
	Datendoppelwort	DID	
Lokaldaten	Lokaldatenbit	L	Dieser Speicherbereich nimmt die temporären Daten eines Bausteins für die Dauer der Bearbeitung dieses Bausteins auf. Der L-Stack stellt auch Speicher zum Übertragen von Bausteinparametern und zum Speichern von Zwischenergebnissen aus KOP-Netzwerken zur Verfügung.
	Lokaldatenbyte	LB	
	Lokaldatenwort	LW	
	Lokaldatendoppelwort	LD	
Peripheriebereich: Eingänge	Peripherieeingangsbyte	PEB	Die Peripheriebereiche der Ein- und Ausgänge erlauben direkten Zugriff auf zentrale und dezentrale Ein- und Ausgabebaugruppen.
	Peripherieeingangswort	PEW	
	Peripherieeingangsdoppelwort	PED	
Peripheriebereich: Ausgänge	Peripherieausgangsbyte	PAB	
	Peripherieausgangswort	PAW	
	Peripherieausgangsdoppelwort	PAD	

Welche Adressbereiche bei Ihrer CPU möglich sind, entnehmen Sie folgenden CPU-Beschreibungen bzw. Operationslisten:

- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten".
- Referenzhandbuch "Automatisierungssysteme S7-400/M7-400. Baugruppendaten".
- Operationsliste "Automatisierungssystem S7-300".
- Operationsliste "Automatisierungssystem S7-400".

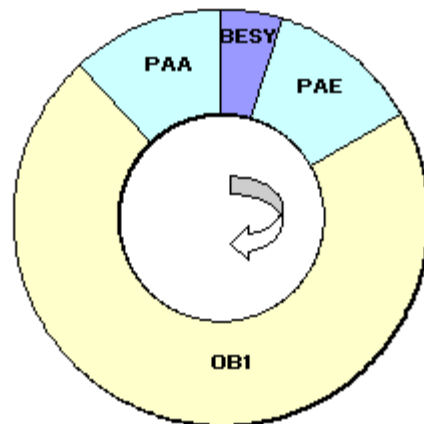
A.2.3.2 Prozessabbild der Ein-/Ausgänge

Werden im Anwenderprogramm die Operandenbereiche Eingänge (E) und Ausgänge (A) angesprochen, werden nicht die Signalzustände auf den digitalen Signalbaugruppen abgefragt, sondern es wird auf einen Speicherbereich im Systemspeicher der CPU und der dezentralen Peripherie zugegriffen. Diesen Speicherbereich bezeichnet man als Prozessabbild.

Aktualisieren des Prozessabbilds

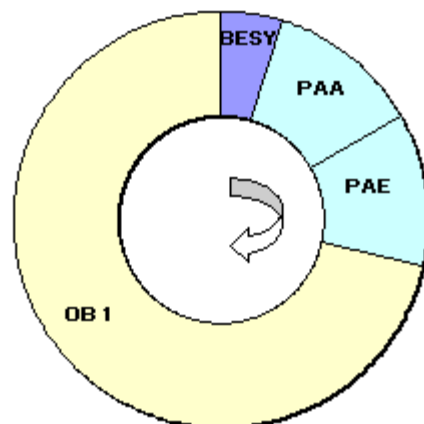
Nachfolgendes Bild zeigt die Bearbeitungsschritte innerhalb eines Zyklus.

Zyklische Programmbearbeitung (CPUs bis 10/98)



Nach den internen Aufgaben des Betriebssystems (BESY) wird der Zustand der Eingänge ins Prozessabbild der Eingänge (PAE) gelesen. Danach erfolgt die Abarbeitung des Anwenderprogramms mit allen darin aufgerufenen Bausteinen. Der Zyklus schließt mit dem Schreiben des Prozessabbilds der Ausgänge (PAA) auf die Ausgänge der Baugruppen. Das Einlesen des Prozessabbilds der Eingänge und das Schreiben des Prozessabbilds der Ausgänge auf die Ausgänge der Baugruppen wird selbständig vom Betriebssystem angesteuert.

Zyklische Programmbearbeitung (CPUs ab 10/98)



Nach den internen Aufgaben des Betriebssystems (BESY) wird das Prozessabbild der Ausgänge (PAA) auf die Ausgänge der Baugruppen geschrieben und der Zustand der Eingänge ins Prozessabbild der Eingänge (PAE) gelesen. Danach erfolgt die Abarbeitung des Anwenderprogramms mit allen darin aufgerufenen Bausteinen. Das Schreiben des Prozessabbilds der Ausgänge auf die Ausgänge der Baugruppen und das Einlesen des Prozessabbilds der Eingänge wird selbständig vom Betriebssystem angesteuert.

Vorteile des Prozessabbilds

Der Zugriff auf das Prozessabbild hat gegenüber dem direkten Zugriff auf die Ein-/Ausgabebaugruppen den Vorteil, dass der CPU für die Dauer der zyklischen Programmbearbeitung ein konsistentes Abbild der Prozesssignale zur Verfügung steht. Wenn sich während der Programmbearbeitung ein Signalzustand auf einer Eingabebaugruppe ändert, bleibt der Signalzustand im Prozessabbild erhalten bis zur Prozessabbildaktualisierung im nächsten Zyklus. Wird ein Eingangssignal innerhalb eines Anwenderprogramms mehrmalig abgefragt, so ist hiermit gewährleistet, dass die Eingangsinformation immer konsistent vorliegt.

Außerdem benötigt der Zugriff auf das Prozessabbild wesentlich weniger Zeit als der direkte Zugriff auf die Signalbaugruppen, weil sich das Prozessabbild im internen Speicher der CPU befindet.

Teilprozessabbilder

Neben dem vom Betriebssystem automatisch aktualisierten Prozessabbild (PAE und PAA) können Sie bei S7-400 für eine CPU bis zu 15 Teilprozessabbilder parametrieren (CPU-spezifisch, Nr. 1 bis max. Nr. 15; siehe Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Baugruppendaten"). Damit haben Sie die Möglichkeit, unabhängig von der zyklischen Aktualisierung des Prozessabbilds, Teilbereiche des Prozessabbildes bei Bedarf zu aktualisieren.

Jede Ein-/Ausgangsadresse, die Sie mit STEP 7 einem Teilprozessabbild zugewiesen haben, gehört nicht mehr zum OB 1-Prozessabbild der Ein-/Ausgänge! Ein-/Ausgangsadressen sind über das OB 1-Prozessabbild und alle Teilprozessabbilder nur einmalig vorgebar.

Das Teilprozessabbild definieren Sie mit STEP 7 bei der Adresszuweisung (welche Ein-/Ausgangsadressen der Baugruppen in welchem Teilprozessabbild geführt werden). Die Aktualisierung des Teilprozessabbilds erfolgt entweder anwenderseitig mit Hilfe von SFCs oder systemseitig automatisch durch Kopplung an einen OB. Ausnahme: Teilprozessabbilder der Taktsynchronalarm-OBs werden nicht systemseitig aktualisiert, obwohl sie an einen OB (OB 61 bis OB 64) gekoppelt sind.

Hinweis

Bei S7-300 CPUs können nicht belegte Prozessabbildein- und -ausgänge als zusätzliche Merkerbereiche verwendet werden. Programme, die diese Möglichkeit nutzen, sind nur unter folgender Bedingung auf älteren (d. h. vor 4/99) S7-400-CPU's ablauffähig:

Bei diesen S7-400 CPUs

- müssen die als Merker verwendeten Prozessabbilder außerhalb der parametrisierten "Größe des Prozessabbilds" liegen bzw.
- müssen in einem Teilprozessabbild liegen, das weder systemseitig noch durch SFC 26/SFC 27 aktualisiert wird!

Teilprozessabbilder aktualisieren mit SFCs

Sie können mit Hilfe von SFCs das gesamte Prozessabbild oder Teilprozessabbilder vom Anwenderprogramm aus aktualisieren.

Voraussetzung: Das entsprechende Teilprozessabbild wird nicht systemseitig aktualisiert!

- SFC 26 UPDAT_PI: Prozessabbild der Eingänge aktualisieren
- SFC 27 UPDAT_PO: Prozessabbild der Ausgänge aktualisieren

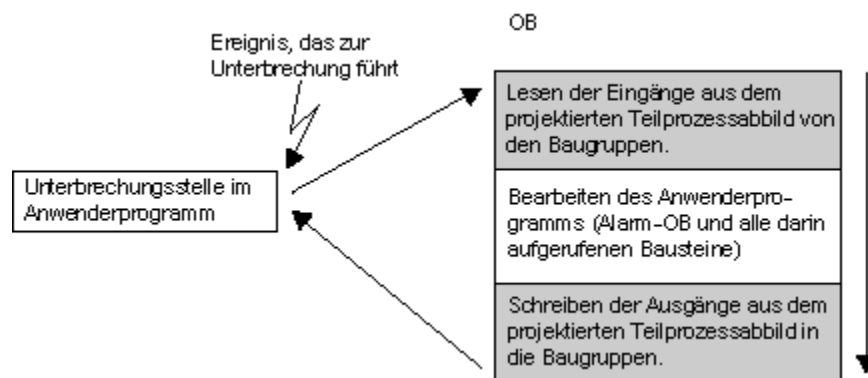
Teilprozessabbilder systemseitig aktualisieren

Sie können Teilprozessabbilder auch automatisch vom Betriebssystem bei Aufruf eines OB aktualisieren lassen - ähnlich dem (Gesamt-) Prozessabbild, das zyklisch vor bzw. nach der OB 1-Bearbeitung aktualisiert wird. Diese Funktion ist nur für bestimmte CPUs parametrierbar.

Im Betrieb wird dann automatisch das zugeordnete Teilprozessabbild aktualisiert:

- vor der OB-Bearbeitung das Teilprozessabbild der Eingänge
- nach der OB-Bearbeitung das Teilprozessabbild der Ausgänge

Welches Teilprozessabbild welchem OB zugeordnet ist, parametrieren Sie für die CPU zusammen mit der Priorität des OB.



Peripheriezugriffsfehler (PZF) bei Prozessabbild-Aktualisierung

Die voreingestellte Reaktion der CPU-Familien (S7-300 und S7-400) ist bei einem Fehler während der Prozessabbild-Aktualisierung unterschiedlich:

- S7-300: Kein Diagnosepuffer-Eintrag, kein OB-Aufruf; die entsprechenden Eingangsbytes werden auf "0" gesetzt und bleiben auf "0", bis der Fehler wieder geht.
- S7-400: Diagnosepuffer-Eintrag und Start des OB 85 bei jedem Zugriff auf die Peripherie bei jeder Aktualisierung des entsprechenden Prozessabbilds. Die fehlerhaften Eingangsbytes werden bei jedem Zugriff auf das Prozessabbild auf "0" gesetzt.

Bei neuen CPUs (ab 4/99) können Sie die Reaktion bei Peripheriezugriffsfehlern umparametrieren, so dass die CPU

- nur bei kommendem und gehendem PZF einen Diagnosepuffer-Eintrag erzeugt und den OB 85 startet (vor Aufruf des OB 85 werden die fehlerhaften Eingangsbytes auf "0" gesetzt und danach bis zum gehenden PZF nicht mehr vom Betriebssystem überschrieben) oder
- das voreingestellte Verhalten der S7-300 zeigt (kein OB-85-Aufruf, die entsprechenden Eingangsbytes werden auf "0" gesetzt und vom Betriebssystem nicht mehr überschrieben, bis der Fehler behoben ist) oder
- das voreingestellte Verhalten der S7-400 zeigt (OB-85-Aufruf bei jedem einzelnen Zugriff, die fehlerhaften Eingangsbytes werden bei jedem Zugriff auf das Prozessabbild auf "0" gesetzt)

Wie oft startet der OB 85?

Neben der parametrierten Reaktion auf PZF (kommend/gehend oder bei jedem Peripheriezugriff) hat auch der Adressraum einer Baugruppe Einfluss auf die Häufigkeit des OB-85-Starts:

Bei einer Baugruppe mit einem Adressraum bis zu einem Doppelwort startet der OB 85 einmal, z. B. bei einer Digitalbaugruppe mit bis zu 32 Eingängen oder Ausgängen, oder einer Analogbaugruppe mit 2 Kanälen.

Bei Baugruppen mit größerem Adressraum startet der OB 85 so oft, wie mit Doppelwortbefehlen darauf zugegriffen werden muss, z. B. bei einer 4-kanaligen Analogbaugruppe zweimal.

A.2.3.3 Lokaldaten-Stack

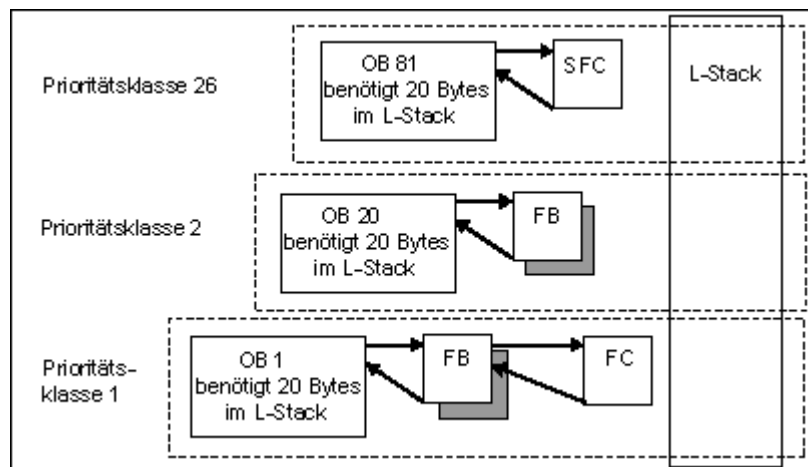
Der L-Stack speichert:

- die temporären Variablen der Lokaldaten von Bausteinen
- die Startinformation der Organisationsbausteine
- Informationen zum Übergeben von Parametern
- Zwischenergebnisse der Logik in Kontaktplan-Programmen

Beim Erstellen von Organisationsbausteinen können Sie temporäre Variablen (TEMP) deklarieren, die nur während der Bearbeitung des Bausteins zur Verfügung stehen und dann wieder überschrieben werden. Vor dem ersten Zugriff müssen die Lokaldaten initialisiert werden. Außerdem benötigt jeder Organisationsbaustein für seine Startinformation 20 Lokaldaten-Byte.

Die CPU besitzt einen begrenzten Speicher für die temporären Variablen (Lokaldaten) gerade bearbeiteter Bausteine. Die Größe dieses Speicherbereichs, des Lokaldaten-Stacks, ist CPU-abhängig. Der Lokaldaten-Stack wird zu gleichen Teilen unter den Prioritätsklassen aufgeteilt (Voreinstellung). Das bedeutet, jede Prioritätsklasse verfügt über einen eigenen Lokaldatenbereich. Damit ist gewährleistet, dass auch hochprioritäre Prioritätsklassen und ihre zugeordneten OBs Platz für ihre Lokaldaten zur Verfügung haben.

Nachfolgendes Bild zeigt die Zuordnung von Lokaldaten zu den Prioritätsklassen in einem Beispiel, in dem im L-Stack der OB 1 vom OB 10 unterbrochen wird, der dann wiederum vom OB 81 unterbrochen wird.



Vorsicht

Alle temporäre Variablen (TEMP) eines OB und seiner unterlagerten Bausteine werden im L-Stack gespeichert. Wenn Sie viele Schachtelungsebenen in Ihrer Bausteinbearbeitung verwenden, kann der L-Stack überlaufen.

S7-CPU's wechseln in den Betriebszustand "STOP", wenn Sie die zulässige L-Stack-Größe für ein Programm überschreiten.

Testen Sie den L-Stack (die temporären Variablen) in Ihrem Programm.

Berücksichtigen Sie den Lokaldatenbedarf von Synchronfehler-OBs.

Zuordnen von Lokaldaten zu Prioritätsklassen

Nicht jede Prioritätsklasse benötigt gleich viel Speicher im Lokaldaten-Stack. Durch Parametrierung mit STEP 7 können Sie bei S7-400-CPU's und bei der CPU 318 die Größe des Lokaldatenbereichs für die einzelnen Prioritätsklassen unterschiedlich festlegen. Nicht benötigte Prioritätsklassen können Sie abwählen. Bei S7-400-CPU's und der CPU 318 wird damit der Speicherbereich für andere Prioritätsklassen erweitert. Abgewählte OBs werden bei der Programmbearbeitung nicht berücksichtigt, dadurch sparen Sie Rechenzeit.

Bei den anderen S7-300-CPU's ist jeder Prioritätsklasse eine feste Anzahl von Lokaldaten zugeordnet (256 Bytes), die nicht verändert werden kann.

A.2.3.4 Unterbrechungs-Stack

Wird die Programmbearbeitung durch einen OB mit höherer Priorität unterbrochen, speichert das Betriebssystem die aktuellen Inhalte der Akkumulatoren und Adressregister sowie die Nummer und die Größe der geöffneten Datenbausteine im U-Stack.

Ist die Bearbeitung des neuen OB beendet, lädt das Betriebssystem die Informationen aus dem U-Stack und nimmt die Bearbeitung des unterbrochenen Bausteins an der Stelle wieder auf, an der die Unterbrechung eintrat.

Im Betriebszustand "STOP" können Sie mit STEP 7 am PG den U-Stack auslesen. Damit können Sie leichter die Ursache dafür erkennen, warum die CPU in den Betriebszustand "STOP" gewechselt hat.

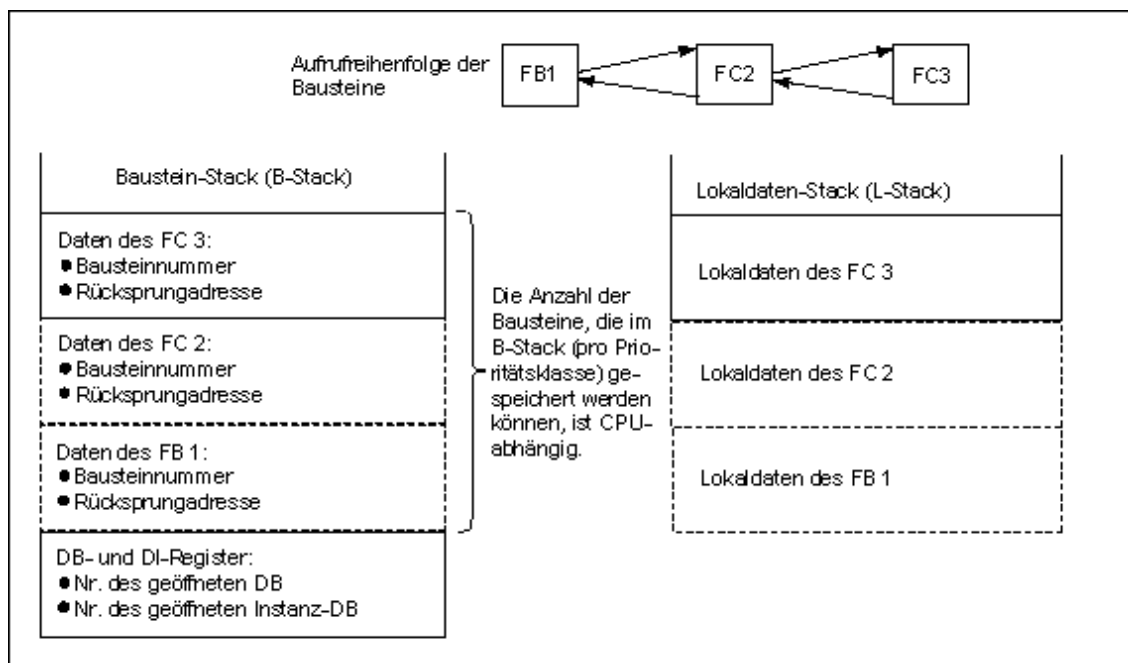
A.2.3.5 Baustein-Stack

Wird die Bearbeitung eines Bausteins durch den Aufruf eines anderen Bausteins oder durch eine höhere Prioritätsklasse (Alarm/Fehlerbehandlung) unterbrochen, speichert der B-Stack folgende Daten:

- Nummer, Art (OB, FB, FC, SFB, SFC) und Rücksprungadresse des Bausteins, der unterbrochen wurde.
- Nummer der Datenbausteine (aus DB- und DI-Register), die zum Zeitpunkt der Unterbrechung geöffnet waren.

Mit den gespeicherten Daten kann das Anwenderprogramm nach der Unterbrechung wieder fortgesetzt werden.

Befindet sich die CPU im Betriebszustand "STOP", können Sie den B-Stack mit STEP 7 am PG anzeigen lassen. Der B-Stack führt alle Bausteine auf, deren Bearbeitung zu dem Zeitpunkt, als die CPU in den Betriebszustand "STOP" versetzt wurde, nicht beendet war. Die Bausteine werden in der Reihenfolge aufgelistet, in der die Bearbeitung begonnen wurde (siehe nachfolgendes Bild).



Datenbausteinregister

Es existieren zwei Datenbausteinregister. Sie beinhalten die Nummern der aufgeschlagenen Datenbausteine

- im DB-Register steht die Nummer des aufgeschlagenen globalen Datenbausteins
- im DI-Register steht die Nummer des aufgeschlagenen Instanz-Datenbausteins.

A.2.3.6 Diagnosepuffer

Im Diagnosepuffer werden die Diagnoseereignisse in der Reihenfolge ihres Auftretens angezeigt. Der erste Eintrag enthält das neueste Ereignis. Die Anzahl der Einträge im Diagnosepuffer ist abhängig von der jeweiligen Baugruppe und ihres aktuellen Betriebszustandes.

Diagnoseereignisse können sein

- Fehler auf einer Baugruppe
- Fehler in der Prozessverdrahtung
- Systemfehler in der CPU
- Betriebszustandsübergänge der CPU
- Fehler im Anwenderprogramm
- Anwenderdefinierte Diagnoseereignisse (über die Systemfunktion SFC 52).

A.2.3.7 Auswertung des Diagnosepuffers

Ein Teil der Systemzustandsliste ist der Diagnosepuffer, in den zu System-Diagnoseereignissen und anwenderdefinierten Diagnoseereignissen in der Reihenfolge ihres Auftretens nähere Informationen eingetragen werden. Die Information, die beim Auftreten eines System-Diagnoseereignisses in den Diagnosepuffer eingetragen wird, ist identisch mit der Startinformation, die an den entsprechenden Organisationsbaustein übergeben wird.

Die Einträge im Diagnosepuffer können nicht gelöscht werden, der Inhalt des Diagnosepuffers bleibt auch nach dem Urlöschen erhalten.

Der Diagnosepuffer bietet die Möglichkeit:

- bei Anlagenstopp die letzten Ereignisse vor dem STOP auszuwerten und die STOP-Ursache zu finden
- Fehlerursachen schneller zu erkennen und dadurch die Verfügbarkeit der Anlage zu erhöhen
- das dynamische Anlagenverhalten auszuwerten und zu optimieren.

Organisation des Diagnosepuffers

Der Diagnosepuffer ist als Ringpuffer für eine baugruppen-abhängige maximale Anzahl von Einträgen ausgelegt. Wird die maximale Anzahl von Einträgen erreicht, so wird bei einem neuen Diagnosepufferereignis der älteste Eintrag gelöscht. Alle Einträge rücken entsprechend weiter. Dadurch steht der jüngste Diagnoseeintrag immer an erster Stelle. Für die S7-300 CPU 314 sind dies z. B. 100 Einträge.

Die Anzahl der angezeigten Einträge im Diagnosepuffer ist abhängig von der Baugruppe und ihrem aktuellen Betriebszustand. Bei bestimmten CPUs ist die Länge des Diagnosepuffers parametrierbar.

Inhalt des Diagnosepuffers

Das **obere** Listenfeld enthält eine Liste aller aufgetretenen Diagnoseereignisse mit folgenden Informationen:

- Laufende Nummer des Eintrags (das neueste Ereignis hat die Nr. 1)
- Uhrzeit und Datum des Diagnoseereignisses: Angezeigt werden Uhrzeit und Datum der Baugruppe, falls auf der Baugruppe eine Uhr vorhanden ist. Um diese Zeitangaben sinnvoll nutzen zu können, ist es wichtig, dass Sie die Uhrzeit und das Datum auf der Baugruppe einstellen und gelegentlich überprüfen.
- Ereignistext (Kurzdarstellung)

Im **unteren** Textfeld werden zusätzliche Informationen zu dem im oberen Fenster ausgewählten Ereignis angezeigt. Dazu gehören z. B.

- Ereignisnummer,
- Bezeichnung des Ereignisses,
- Betriebszustandswechsel, der durch das Diagnoseereignis verursacht wurde,
- Verweis auf Fehlerstelle in einem Baustein (Bausteintyp, -nummer, Relativadresse), die zum Eintrag des Ereignisses führte,
- Kommendes oder gehendes Ereignis.
- Ereignisspezifische Zusatzinformationen

Mit der Schaltfläche "Hilfe zum Ereignis" können Sie sich zusätzliche Information zu dem im Listenfeld markierten Ereignis anzeigen lassen.

Erläuterungen zu den Ereignis-IDs finden Sie in der Referenzhilfe zu Systembausteinen und Systemfunktionen (Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute).

Speichern des Inhalts in einer Textdatei

Mit der Schaltfläche "Speichern unter" im Register "Diagnosepuffer" des Dialogfelds "Baugruppenzustand" können Sie den Inhalt des Diagnosepuffers als ASCII-Text speichern.

Diagnosepuffer auslesen

Sie können sich den Inhalt des Diagnosepuffers über das Dialogfeld "Baugruppenzustand", Register "Diagnosepuffer" am PG/PC anzeigen lassen oder in einem Programm über die SFC 51 RDSYSST auslesen.

Letzter Eintrag vor STOP

Sie können festlegen, dass der letzte Diagnosepuffereintrag vor dem Übergang von RUN in STOP automatisch an ein angemeldetes Beobachtungsgerät (z. B. PG, OP, TD) gesendet wird, um sicherzustellen, dass die Ursache für den Betriebszustandswechsel in STOP schneller gefunden und behoben wird.

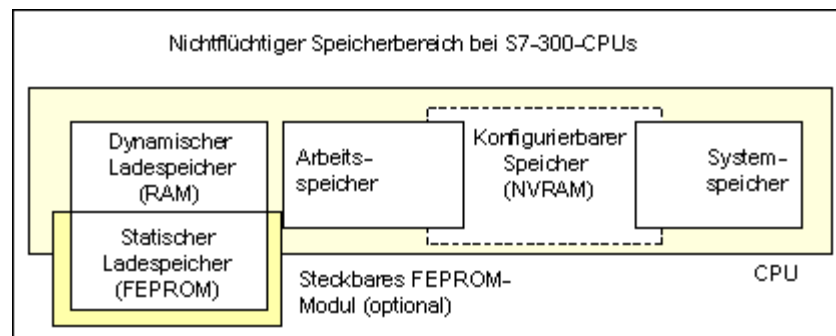
A.2.3.8 Remanente Speicherbereiche in S7-300-CPU

Bei Stromausfall oder Umlöschen (MRES) wird der Speicher der S7-300-CPU (dynamischer Ladespeicher (RAM), Arbeitsspeicher und Systemspeicher) zurückgesetzt, wobei alle Daten, die in diesen Bereichen gespeichert sind, verloren gehen. S7-300-CPU bieten die folgenden Möglichkeiten, Ihr Programm und seine Daten zu sichern:

- Sie können alle Daten, die sich im Ladespeicher, Arbeitsspeicher und in Teilen des Systemspeichers befinden, durch eine Batterie puffern.
- Sie können Ihr Programm im EPROM speichern (entweder Memory Card oder in die CPU integriert, siehe Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten").
- Sie können eine CPU-abhängige Menge Daten in einem Bereich des nichtflüchtigen NVRAM sichern.

Verwenden des NVRAM

Ihre S7-300-CPU stellt einen Bereich im NVRAM (nichtflüchtiger RAM) zur Verfügung (siehe nachfolgendes Bild). Wenn Sie Ihr Programm im EPROM des Ladespeichers abgelegt haben, können Sie einige Daten (bei Stromausfall oder beim Übergang der CPU von STOP in RUN) auch durch eine entsprechende Konfiguration speichern.



Hierzu stellen Sie Ihre CPU so ein, dass die folgenden Daten in dem nichtflüchtigen RAM gespeichert werden:

- Daten, die in einem DB gespeichert sind (dies ist nur nützlich, wenn Sie auch Ihr Programm in einem EPROM des Ladespeichers abgelegt haben)
- Werte von Zeiten und Zählern
- Daten, die in Merkern gespeichert sind

Bei jeder CPU können Sie eine bestimmte Anzahl an Zeiten, Zählern und Merkern puffern. Außerdem wird eine spezifische Anzahl an Bytes zur Verfügung gestellt, in denen die Daten, die in DBs abgelegt sind, gespeichert werden können.

Die MPI-Adresse Ihrer CPU ist im NVRAM abgelegt. Damit ist sichergestellt, dass Ihre CPU nach einem Stromausfall oder Umlöschen noch kommunikationsfähig ist.

Verwenden der Batteriepufferung zum Sichern von Daten

Durch Batteriepufferung werden der Ladespeicher und der Arbeitsspeicher bei Stromausfall remanent. Wenn durch Ihre Konfiguration Zeiten, Zähler und Merker im NVRAM gespeichert werden, gehen unabhängig von der Batteriepufferung auch diese Informationen nicht verloren.

Konfigurieren der Daten des NVRAM

Wenn Sie die CPU mit STEP 7 konfigurieren, können Sie festlegen, welche Speicherbereiche remanent sein sollen.

Wieviel Speicher im NVRAM konfiguriert werden kann, ist CPU-abhängig. Sie können nicht mehr Daten puffern, als die für Ihre CPU angegebene Menge.

A.2.3.9 Remanente Speicherbereiche in S7-400-CPU

Ungepuffertes Betrieb

Bei Stromausfall oder Umlöschen (MRES) im ungepufferten Betrieb wird der Speicher der S7-400-CPU (dynamischer Ladespeicher (RAM), Arbeitsspeicher und Systemspeicher) zurückgesetzt, wobei alle Daten, die in diesen Bereichen gespeichert sind, verloren gehen.

Im ungepufferten Betrieb ist nur Neustart (Warmstart) möglich und es gibt keine remanenten Speicherbereiche. Nur die MPI-Parameter (z. B. MPI-Adresse der CPU) bleiben nach Spannungsausfall erhalten. Damit ist die CPU nach Stromausfall oder Umlöschen noch kommunikationsfähig.

Gepuffertes Betrieb

Im gepufferten Betrieb bleibt

- beim Wiederanlauf nach Spannungsausfall der Inhalt aller RAM-Bereiche vollständig erhalten.
- beim Neustart (Warmstart) werden die Operandenbereiche Merker, Zeiten, Zähler gelöscht. Die Inhalte von Datenbausteinen bleiben erhalten.
- der Inhalt des RAM-Arbeitsspeichers bis auf nicht remanent parametrisierte Merker, Zeiten, Zähler erhalten.

Konfigurieren remanenter Datenbereiche

Sie können eine CPU-abhängige Menge von Merkern, Zeiten, Zählern remanent erklären. Beim Neustart (Warmstart) im gepufferten Betrieb bleiben diese Daten dann erhalten.

Durch Parametrierung mit STEP 7 legen Sie fest, welche Merker, Zeiten, Zähler bei einem Neustart (Warmstart) remanent sein sollen. Sie können nicht mehr Daten puffern, als die für Ihre CPU zulässige Menge.

Nähere Informationen zur Parametrierung remanenter Speicherbereiche entnehmen Sie dem Referenzhandbuch "Automatisierungssysteme S7-400/M7-400. Baugruppendaten".

A.2.3.10 Konfigurierbare Speicherobjekte im Arbeitsspeicher

Bei einigen CPUs kann die Größe von Objekten wie Lokaldaten oder Diagnosepuffer unter HW Konfig eingestellt werden. Wenn Sie beispielsweise die voreingestellten Werte verkleinern, steht ein größerer Anteil des Arbeitsspeichers anderweitig zur Verfügung. Die Einstellungen dieser CPUs sind im Register "Speicher" des Baugruppenzustands (Schaltfläche Details) ablesbar.

Nach Ändern der Speicherkonfiguration und Laden auf das Zielsystem ist ein Kaltstart des Zielsystems erforderlich, damit die Änderungen wirksam werden.

A.3 Daten- und Parametertypen

A.3.1 Einführung zu Daten- und Parametertypen

Alle in einem Anwenderprogramm verwendeten Daten müssen durch einen Datentyp gekennzeichnet sein. Man unterscheidet zwischen:

- elementaren Datentypen, die Ihnen STEP 7 zur Verfügung stellt,
- zusammengesetzten Datentypen, die Sie erzeugen können, indem Sie elementare Datentypen verknüpfen, und
- Parametertypen, mit denen Sie Parameter definieren, die an FBs oder FCs übergeben werden sollen.

Allgemeine Information

AWL-, FUP- oder KOP-Operationen arbeiten mit Datenobjekten bestimmter Größen. Bitverknüpfungsoperationen arbeiten beispielsweise mit Bits. Lade- und Transferoperationen (AWL) sowie Übertragungsoperationen (FUP und KOP) arbeiten mit Bytes, Wörtern und Doppelwörtern.

Bit nennt man eine Binärziffer "0" oder "1". Ein Byte besteht aus 8 Bits, ein Wort aus 16 Bits und ein Doppelwort aus 32 Bits.

Arithmetische Operationen arbeiten ebenfalls mit Bytes, Wörtern oder Doppelwörtern. In diesen Byte-, Wort- oder Doppelwortoperanden können Sie Zahlen verschiedener Formate codieren, wie z. B. Ganzzahlen und Gleitpunktzahlen.

Wenn Sie die symbolische Adressierung verwenden, dann definieren Sie Symbole und geben einen Datentyp für diese Symbole an (siehe nachfolgende Tabelle). Verschiedene Datentypen haben verschiedene Formatoptionen und Zahlendarstellungen.

Dieses Kapitel beschreibt nur einige der Schreibweisen, die für Zahlen und Konstanten möglich sind. Nachfolgende Tabelle listet Formate von Zahlen und Konstanten auf, auf die nicht ausführlich eingegangen wird.

Format	Größe in Bits	Zahlendarstellung
Hexadezimal	8, 16 und 32	B#16#, W#16# und DW#16#
Binär	8, 16 und 32	2#
IEC-Datum	16	D#
IEC-Zeit	32	T#
Uhrzeit	32	TOD#
Zeichen	8	'A'

A.3.2 Elementare Datentypen

Jeder elementare Datentyp hat eine definierte Länge. Die folgende Tabelle listet die elementaren Datentypen auf.

Typ und Beschreibung	Größe in Bits	Formatoptionen	Bereich und Zahlendarstellung (niedrigster bis höchster Wert)	Beispiel
BOOL (Bit)	1	Bool-Text	TRUE/FALSE	TRUE
BYTE (Byte)	8	Hexadezimalzahl	B#16#0 bis B#16#FF	L B#16#10 L byte#16#10
WORD (Wort)	16	Dualzahl Hexadezimalzahl BCD Dezimalzahl ohne Vorzeichen	2#0 bis 2#1111_1111_1111_1111 W#16#0 bis W#16#FFFF C#0 bis C#999 B#(0,0) bis B#(255,255)	L 2#0001_0000_0000_0000 L W#16#1000 L word#16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (Doppelwort)	32	Dualzahl Hexadezimalzahl Dezimalzahl ohne Vorzeichen	2#0 bis 2#1111_1111_1111_1111_1111_1111_1111_1111 DW#16#0000_0000 bis DW#16#FFFF_FFFF B#(0,0,0,0) bis B#(255,255,255,255)	2#1000_0001_0001_1000_1011_1011_0111_1111 L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (Ganzzahl)	16	Dezimalzahl mit Vorzeichen	-32768 bis 32767	L 1
DINT (Ganzzahl, 32 Bit)	32	Dezimalzahl mit Vorzeichen	L#-2147483648 bis L#2147483647	L L#1
REAL (Gleitpunktzahl)	32	IEEE Gleitpunktzahl	Obere Grenze: $\pm 3.402823e+38$ Untere Grenze: $\pm 1.175 495e-38$	L 1.234567e+13
S5TIME (SIMATIC-Zeit)	16	S7-Zeit in Schritten von 10 ms (Default-Wert)	S5T#0H_0M_0S_10MS bis S5T#2H_46M_30S_0MS und S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (IEC-Zeit)	32	IEC-Zeit in Schritten von 1 ms, Ganzzahl mit Vorzeichen	-T#24D_20H_31M_23S_648MS bis T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS

Typ und Beschreibung	Größe in Bits	Formatoptionen	Bereich und Zahlendarstellung (niedrigster bis höchster Wert)	Beispiel
DATE (IEC-Datum)	16	IEC-Datum in Schritten von 1 Tag	D#1990-1-1 bis D#2168-12-31	L D#1994-3-15 L DATE#1994-3-15
TIME_OF_DAY (Uhrzeit)	32	Uhrzeit in Schritten von 1 ms	TOD#0:0:0.0 bis TOD#23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
CHAR (Zeichen)	8	ASCII- Zeichen	'A','B' usw.	L 'E'

A.3.2.1 Format des Datentyps INT (16-Bit-Ganzzahlen)

Eine Ganzzahl hat ein Vorzeichen, das angibt, ob es sich um eine positive oder negative ganze Zahl handelt. Der Platz, den eine Ganzzahl (16 Bit) im Speicher belegt, beträgt ein Wort. Nachfolgende Tabelle zeigt den Bereich einer Ganzzahl (16 Bit).

Format	Bereich
Ganzzahl (16 Bit)	-32 768 bis +32 767

Nachfolgendes Bild zeigt die Ganzzahl +44 als Dualzahl.

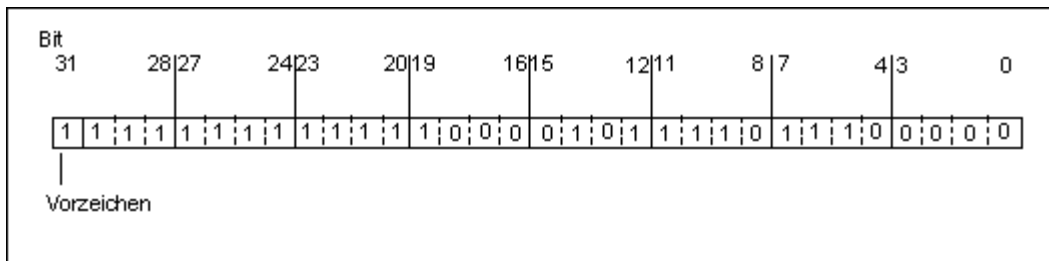


A.3.2.2 Format des Datentyps DINT (32-Bit-Ganzzahlen)

Eine Ganzzahl hat ein Vorzeichen, das angibt, ob es sich um eine positive oder negative ganze Zahl handelt. Der Platz, den eine Ganzzahl (32 Bit) im Speicher belegt, beträgt zwei Wörter. Nachfolgende Tabelle zeigt den Bereich einer Ganzzahl (32 Bit).

Format	Bereich
Ganzzahl (32 Bit)	-2 147 483 648 bis +2 147 483 647

Nachfolgendes Bild zeigt die Ganzzahl -500 000 als Dualzahl. Im Dualsystem wird die negative Form einer Ganzzahl als Zweierkomplement der positiven Ganzzahl dargestellt. Das Zweierkomplement einer Ganzzahl erhalten Sie, indem Sie die Signalzustände aller Bits umkehren und dann +1 zum Ergebnis addieren.



A.3.2.3 Format des Datentyps REAL (Gleitpunktzahlen)

Zahlen im Gleitpunktformat werden in der allgemeinen Form "Zahl = $m * b^{\text{hoch } E}$ " dargestellt. Die Basis "b" und der Exponent "E" sind ganze Zahlen, die Mantisse "m" eine rationale Zahl.

Diese Zahlendarstellung hat den Vorteil, dass auf begrenztem Raum sehr große und sehr kleine Werte dargestellt werden können. Mit der begrenzten Anzahl von Bits für Mantisse und Exponent lässt sich ein weiter Zahlenbereich abdecken.

Der Nachteil liegt in der begrenzten Rechengenauigkeit: Beispielsweise müssen bei der Summenbildung zweier Zahlen die Exponenten durch Verschieben der Mantisse (gleitender Dezimalpunkt) angeglichen werden (Addition der Mantissen zweier Zahlen mit gleichem Exponent).

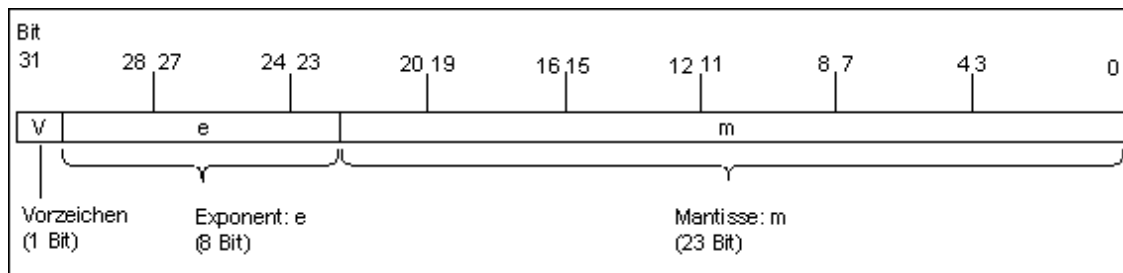
Gleitpunktformat in STEP 7

Gleitpunktzahlen in STEP 7 entsprechen dem Grundformat mit einfacher Breite, wie in der Norm ANSI/IEEE Standard 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic* beschrieben. Sie bestehen aus den folgenden Komponenten:

- dem Vorzeichen V
- dem um eine Konstante (Bias = +127) erhöhten Exponenten $e = E + \text{Bias}$
- dem gebrochenen Teil der Mantisse m.

Der ganzzahlige Anteil der Mantisse wird nicht mit abgelegt, da er innerhalb des gültigen Zahlenbereichs immer = 1 ist

Die drei Bestandteile belegen zusammen ein Doppelwort (32 Bit):



Die folgende Tabelle zeigt die Wertigkeit der einzelnen Bits im Gleitpunktformat.

Bestandteil der Gleitpunktzahl	Bit-Nummer	Wertigkeit
Vorzeichen V	31	
Exponent e	30	$2^{\text{hoch } 7}$
...
Exponent e	24	$2^{\text{hoch } 1}$
Exponent e	23	$2^{\text{hoch } 0}$
Mantisse m	22	$2^{\text{hoch } -1}$
...
Mantisse m	1	$2^{\text{hoch } -22}$
Mantisse m	0	$2^{\text{hoch } -23}$

Mit den drei Komponenten **V**, **e** und **m** ist der Wert einer in diesem Format dargestellten Zahl definiert über die Formel:

$$\text{Zahl} = 1, \mathbf{m} * 2^{\text{hoch}(\mathbf{e}-\text{Bias})}$$

Dabei gilt:

- e: $1 \leq e \leq 254$
- Bias: Bias = 127. Somit entfällt ein extra Vorzeichen für den Exponent.
- V: für eine positive Zahl ist V = 0 und für negative Zahl ist V = 1.

Wertebereich der Gleitpunktzahlen

Anhand des oben dargestellten Gleitpunktformats ergeben sich die

- kleinste Gleitpunktzahl = $1,0 * 2^{\text{hoch}(1-127)} = 1,0 * 2^{\text{hoch}(-126)}$
= 1,175 495E-38 und
- größte Gleitpunktzahl = $2^{-2} * 2^{\text{hoch}(254-127)} = 2^{-2} * 2^{\text{hoch}(126)}$
= 3,402 823E+38

Die Zahl Null wird mit e = m = 0 dargestellt; e = 255 und m = 0 steht für "unendlich".

Format	Bereich ¹⁾
Gleitpunktzahlen nach ANSI/IEEE Standard	-3,402 823E+38 bis -1,175 495E-38 und 0 und +1,175 495E-38 bis +3,402 823E+38

Die nächste Tabelle zeigt den Signalzustand der Bits im Statuswort für die Ergebnisse von Operationen mit Gleitpunktzahlen, die nicht innerhalb des gültigen Bereichs liegen.

Ungültiger Bereich für ein Ergebnis	A1	A0	OV	OS
-1,175494E-38 < Ergebnis < -1,401298E-45 (negative Zahl) Unterschreitung	0	0	1	1
+1,401298E-45 < Ergebnis < +1,175494E-38 (positive Zahl) Unterschreitung	0	0	1	1
Ergebnis < -3,402823E+38 (negative Zahl) Überlauf	0	1	1	1
Ergebnis > 3,402823E+38 (positive Zahl) Überlauf	1	0	1	1
keine gültige Gleitpunktzahl oder unzulässige Operation (Eingangswert außerhalb des gültigen Wertebereichs)	1	1	1	1

Vorsicht bei mathematischen Operationen:

Das Ergebnis "Keine gültige Gleitpunktzahl" erhält man beispielsweise bei dem Versuch, die Quadratwurzel aus -2 zu ziehen. Werten Sie deshalb bei mathematischen Operationen stets zuerst die Statusbits aus, bevor Sie mit dem Ergebnis weiterrechnen.

Vorsicht beim "Steuern von Variablen":

Legt man die Werte für Gleitkommaoperationen z. B. in Merkerdoppelwörtern ab, so kann man diese Werte mit beliebigen Bitmustern verändern. Jedoch ist nicht jedes Bitmuster eine gültige Zahl!

Genauigkeit beim Rechnen mit Gleitpunktzahlen**Vorsicht**

Bei umfangreichen Berechnungen mit Zahlen, die große Größenunterschiede (mehrere 10-er Potenzen) aufweisen, können Ungenauigkeiten im Rechenergebnis entstehen.

Die Gleitpunktzahlen in STEP 7 sind auf 6 Dezimalstellen genau. Sie können deshalb bei der Angabe von Gleitpunkt konstanten nur maximal 6 Nachkommastellen angeben.

Hinweis

Die Rechengenauigkeit von 6 Dezimalstellen bedeutet z. B., dass die Addition von $\text{Zahl1} + \text{Zahl2} = \text{Zahl1}$ ergibt, wenn Zahl1 größer $\text{Zahl2} * 10^y$ ist, mit $y > 6$:

$$100\,000\,000 + 1 = 100\,000\,000.$$

Beispiele für Zahlen in Gleitpunktdarstellung

Nachfolgendes Bild zeigt das Gleitpunktzahlenformat für die folgenden dezimalen Werte:

- 10,0
- Pi (3,141593)
- Quadratwurzel aus 2 (1,414214)

Die Zahl **10,0** im ersten Beispiel ergibt sich aus ihrem Gleitpunktformat (Darstellung in HEX: 4120 0000) wie folgt:

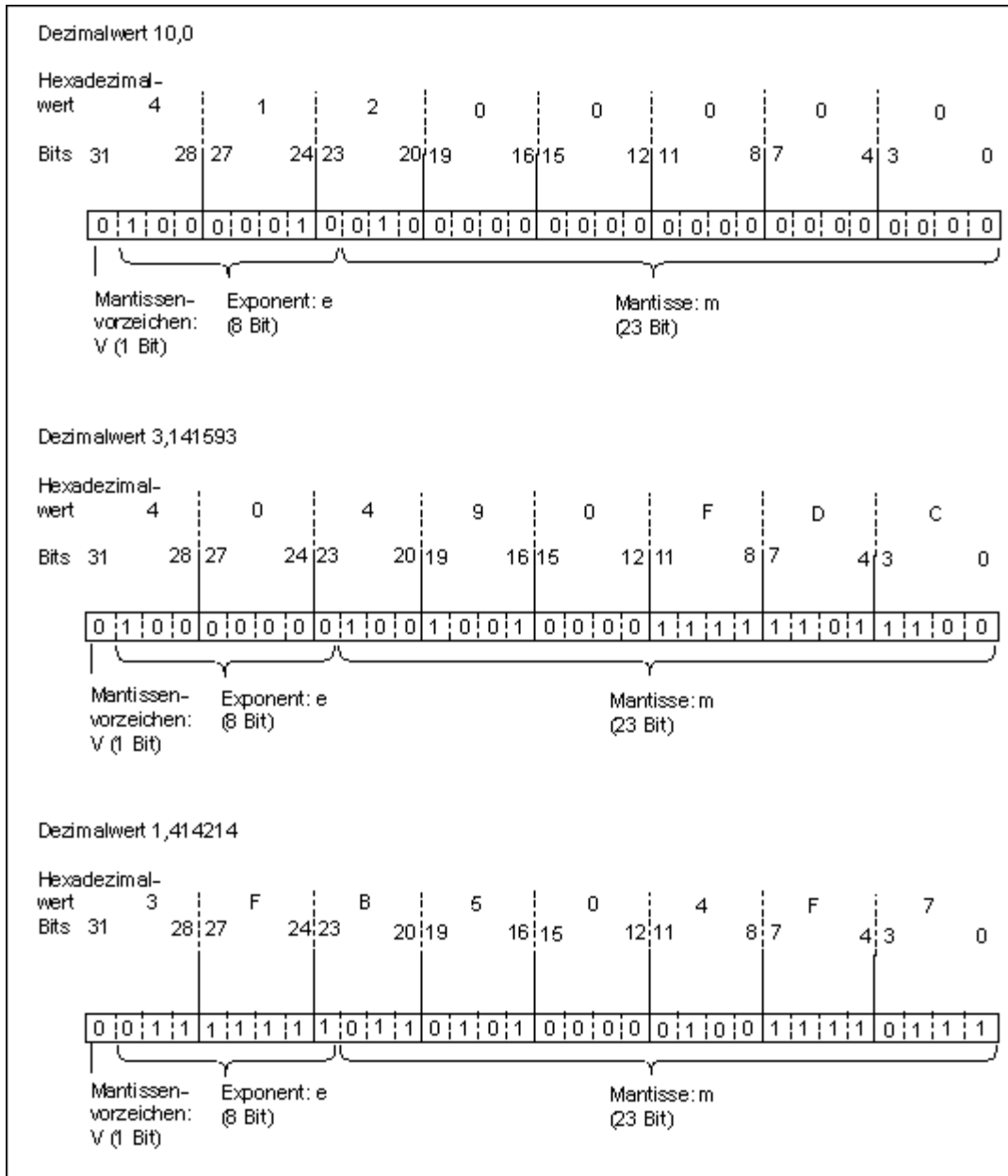
$$e = 2^7 + 2^1 = 130$$

$$m = 2^{-2} = 0,25$$

Damit ergibt sich:

$$(1 + m) * 2^{\text{e-bias}} = 1,25 * 2^3 = 10,0$$

$$[1,25 * 2^{\text{e-bias}}] = [1,25 * 2^3] = 10,0$$



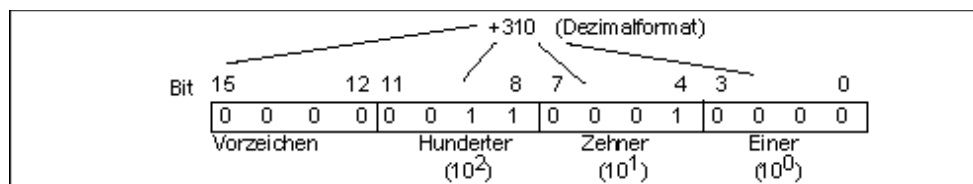
A.3.2.4 Format der Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen

Die binär-codierte Darstellung (BCD) stellt eine Dezimalzahl durch Gruppen von Binärziffern (Bits) dar. Eine Gruppe von 4 Bits stellt eine Ziffer einer Dezimalzahl oder das Vorzeichen der Dezimalzahl dar. Die Gruppen von 4 Bits bilden ein Wort (16 Bits) oder Doppelwort (32 Bits). Die vier höchstwertigen Bits geben das Vorzeichen der Zahl an ("1111" bedeutet Minus und "0000" Plus). Befehle mit BCD-codierten Operanden werten nur das höchstwertige Bit (15 bei Wort-, 31 bei Doppelwortformat) aus. Nachfolgende Tabelle zeigt Format und Bereich für die beiden Arten der BCD-Zahlen.

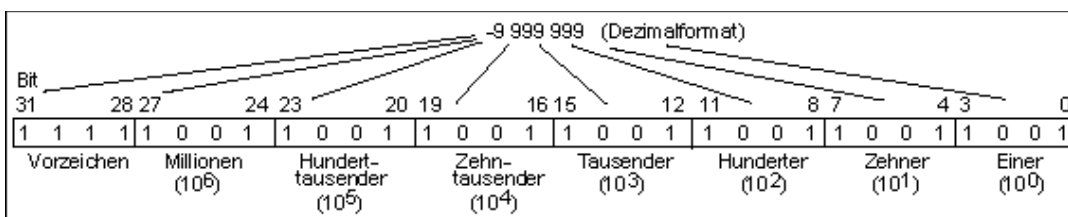
Format	Bereich
Wort (16 Bits, 3stellige BCD-Zahl mit Vorzeichen)	-999 bis +999
Doppelwort (32 Bits, 7stellige BCD-Zahl mit Vorzeichen)	-9 999 999 bis +9 999 999

Die nachfolgenden Bilder zeigen Beispiele für eine binär-codierte Dezimalzahl in folgenden Formaten:

- Wortformat

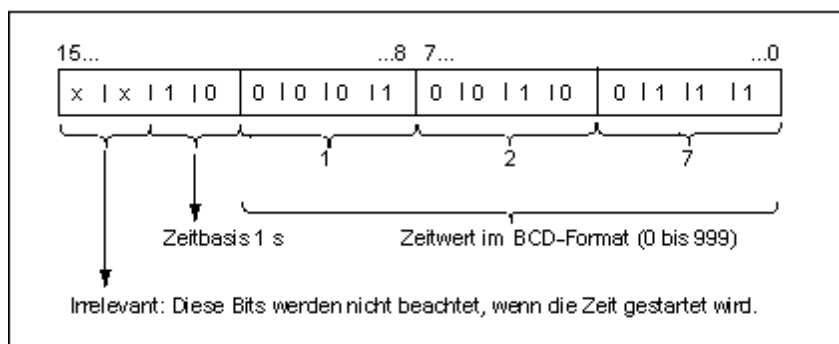


- Doppelwortformat



A.3.2.5 Format des Datentyps S5TIME (Zeitdauer)

Wenn Sie die Zeitdauer mit dem Datentyp S5TIME eingeben, werden Ihre Eingaben im BCD-Format gespeichert. Nachfolgendes Bild zeigt den Inhalt des Zeitoperanden bei einem Zeitwert von 127 und einer Zeitbasis von 1 s.



Wenn Sie mit S5TIME arbeiten, geben Sie einen Zeitwert im Bereich von 0 bis 999 ein und bezeichnen eine Zeitbasis (siehe nachfolgende Tabelle). Die Zeitbasis gibt das Intervall an, in dem eine Zeit den Zeitwert um je eine Einheit vermindert, bis er "0" erreicht.

Zeitbasis für S5TIME

Zeitbasis	Binärkode für Zeitbasis
10 ms	00
100 ms	01
1 s	10
10 s	11

Mit der folgenden Syntax können Sie einen vordefinierten Zeitwert laden:

- L¹⁾ W#16#wxyz
 - mit: w = Zeitbasis (d. h. Zeitintervall oder Auflösung)
 - xyz = Zeitwert im BCD-Format
- L¹⁾ S5T#aH_bbM_ccS_dddMS
 - mit: a = Stunden, bb = Minuten, cc = Sekunden und ddd = Millisekunden.
 - Die Zeitbasis wird automatisch gewählt und der Wert zur nächstniederen Zahl mit dieser Zeitbasis gerundet.

Sie können einen Zeitwert von max. 9 990 Sekunden bzw. 2H_46M_30S eingeben.

¹⁾ = L nur bei AWL-Programmierung angeben

A.3.3 Zusammengesetzte Datentypen

Zusammengesetzte Datentypen definieren Datengruppen, die größer sind als 32 Bits, bzw. Datengruppen, die sich aus anderen Datentypen zusammensetzen. STEP 7 lässt die folgenden zusammengesetzten Datentypen zu:

- DATE_AND_TIME
- STRING
- ARRAY (Feld)
- STRUCT (Struktur)
- UDT (Anwenderdefinierte Datentypen)
- FB und SFB

Nachfolgende Tabelle beschreibt die zusammengesetzten Datentypen. Sie definieren Strukturen und Felder entweder in der Variablendeklaration des Codebausteins oder in einem Datenbaustein.

Datentyp	Beschreibung
DATE_AND_TIME DT	Definiert einen Bereich mit 64 Bits (8 Bytes). Dieser Datentyp speichert in binärcodiertem Dezimalformat.
STRING	Definiert eine Gruppe von maximal 254 Zeichen (Datentyp CHAR). Der Standardbereich, der für einen Zeichenkette reserviert ist, besteht aus 256 Bytes. Dies ist der Platz, der benötigt wird, um 254 Zeichen und einen Kopf von 2 Bytes zu speichern. Sie können den Speicherplatz für eine Zeichenkette verringern, indem Sie auch die Anzahl der Zeichen definieren, die in der Zeichenkette gespeichert werden sollen (z. B. : string[9] 'Siemens').
ARRAY	Definiert eine multidimensionale Gruppierung eines Datentyps (entweder elementar oder zusammengesetzt). Zum Beispiel: "ARRAY [1..2,1..3] OF INT" definiert ein Feld im Format 2 x 3 aus Ganzzahlen. Sie greifen auf die Daten, die in einem Feld gespeichert sind, über den Index ("[2,2]") zu. Sie können maximal 6 Dimensionen in einem Feld definieren. Der Index kann ein beliebiger ganzzahliger Wert (-32768 bis 32767) sein.
STRUCT	Definiert eine Gruppierung von beliebig kombinierten Datentypen. Sie können z. B. ein Feld aus Strukturen oder eine Struktur aus Strukturen und Feldern definieren.
UDT	Ermöglicht es, große Datenmengen zu strukturieren und das Eingeben der Datentypen zu vereinfachen, wenn Datenbausteine erzeugt oder Variablen in der Variablendeklaration deklariert werden sollen. In STEP 7 können Sie zusammengesetzte und elementare Datentypen verknüpfen und so Ihren eigenen "anwenderdefinierten" Datentyp erstellen. UDTs haben einen eigenen Namen und sind deshalb mehrfach verwendbar.
FB, SFB	Bestimmen die Struktur des zugeordneten Instanz-Datenbausteins und ermöglichen die Übergabe von Instanzdaten für mehrere FB-Aufrufe in einem Instanz-DB.

Strukturierte Datentypen werden auf Wortgrenzen ausgerichtet (WORD aligned) abgelegt.

A.3.3.1 Format des Datentyps DATE_AND_TIME (Datum und Uhrzeit)

Wenn Sie Datum und Uhrzeit mit dem Datentyp DATE_AND_TIME (DT) eingeben, werden Ihre Eingaben im BCD-Format in 8 Bytes gespeichert. Der Datentyp DATE_AND_TIME umfasst den folgenden Bereich:

DT#1990-1-1-0:0:0.0 bis DT#2089-12-31-23:59:59.999

Die folgenden Beispiele zeigen die Syntax, mit der Sie das Datum und die Uhrzeit für Donnerstag, den 25. Dezember 1993, 8:12 Uhr und 34,567 Sekunden eingeben. Die folgenden beiden Formate sind möglich:

- DATE_AND_TIME#1993-12-25-8:12:34.567
- DT#1993-12-25-8:12:34.567

Die folgenden IEC-Standardfunktionen (International Electrotechnical Commission) stehen für das Arbeiten mit dem Datentyp DATE_AND_TIME zur Verfügung:

- Umwandeln des Datums und der Uhrzeit in das Format DATE_AND_TIME
FC3: D_TOD_DT
- Herauslösen des Datums aus dem Format DATE_AND_TIME
FC6: DT_DATE
- Herauslösen des Wochentags aus dem Format DATE_AND_TIME
FC7: DT_DAY
- Herauslösen der Uhrzeit aus dem Format DATE_AND_TIME
FC8: DT_TOD

In nachfolgender Tabelle ist der Inhalt der Bytes aufgeführt, die die Information zu Datum und Uhrzeit enthalten. Das Beispiel zeigt Datum und Uhrzeit für Donnerstag, den 25. Dezember 1993, 8:12 Uhr und 34,567 Sekunden.

Byte	Inhalt	Beispiel
0	Jahr	B#16#93
1	Monat	B#16#12
2	Tag	B#16#25
3	Stunde	B#16#08
4	Minute	B#16#12
5	Sekunde	B#16#34
6	Die beiden höchstwertigen Ziffern von MSEC	B#16#56
7 (4MSB)	Die niederwertige Ziffer von MSEC	B#16#7_
7 (4LSB)	Wochentag 1 = Sonntag 2 = Montag ... 7 = Samstag	B#16#_5

Der erlaubte Bereich für den Datentyp "DATE_AND_TIME" ist:

- min.: DT#1990-1-1-0:0:0.0
- max.: DT#2089-12-31-23:59:59.999

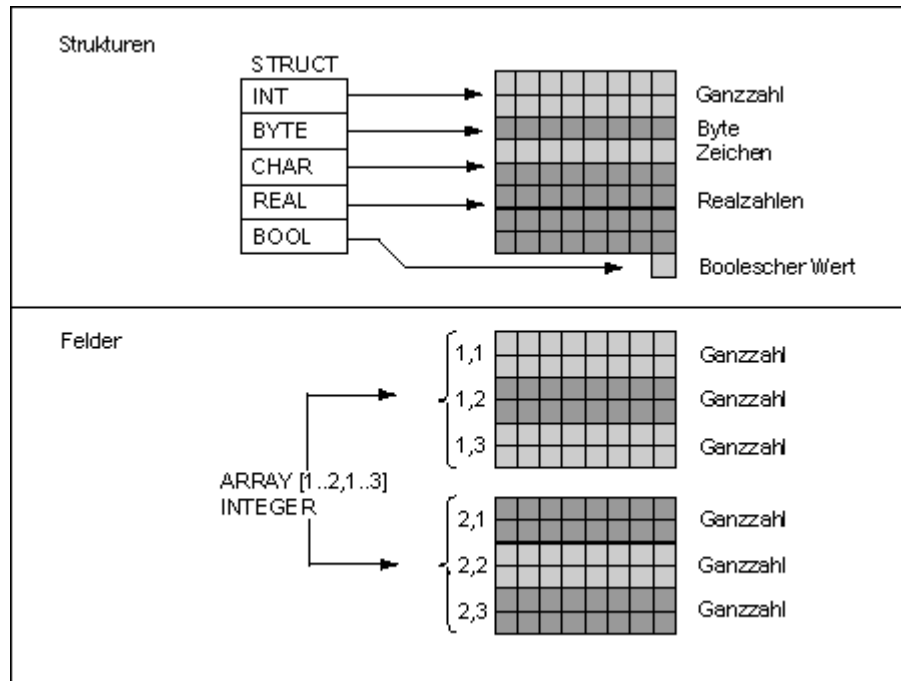
	Möglicher Wertebereich	BCD-Code
Jahr	1990 - 1999	90 - 99
	2000 - 2089	00 - 89
Monat	1 - 12	01 - 12
Tag	1 - 31	01 - 31
Stunde	00 - 23	00 - 23
Minute	00 - 59	00 - 59
Sekunde	00 - 59	00 - 59
Millisekunde	0 - 999	000 - 999
Wochentag	Sonntag - Samstag	1 - 7

A.3.3.2 Verwenden von zusammengesetzten Datentypen

Sie können neue Datentypen erstellen, indem Sie elementare und zusammengesetzte Datentypen zu den folgenden zusammengesetzten Datentypen verbinden:

- Feld (Datentyp ARRAY): Ein Feld verknüpft eine Gruppe eines Datentyps zu einer Einheit.
- Struktur (Datentyp STRUCT): Eine Struktur verknüpft verschiedene Datentypen zu einer Einheit.
- Zeichenkette (Datentyp STRING): Eine Zeichenkette definiert ein eindimensionales Feld mit maximal 254 Zeichen (Datentyp CHAR). Eine Zeichenkette kann nur als Einheit übergeben werden. Die Länge der Zeichenkette muss bei dem Formal- und dem Aktualparameter des Bausteins übereinstimmen.
- Datum und Zeit (Datentyp DATE_AND_TIME): Das Datum und die Zeit speichern Jahr, Monat, Tag, Stunden, Minuten, Sekunden, Millisekunden und Wochentag.

Nachfolgendes Bild zeigt, wie Felder und Strukturen Datentypen in einem Bereich strukturieren, um so Informationen zu speichern. Sie definieren ein Feld oder eine Struktur entweder in einem DB oder in der Variablendeklaration eines FB, OB oder einer FC.



A.3.3.3 Verwenden von Feldern für den Datenzugriff

Felder

Ein Feld verknüpft eine Gruppe eines Datentyps (elementar oder zusammengesetzt) zu einer Einheit. Sie können kein Feld aus Feldern erstellen. Wenn Sie ein Feld definieren, müssen Sie folgendes angeben:

- Sie geben dem Feld einen Namen.
- Sie deklarieren ein Feld mit dem Schlüsselwort ARRAY.
- Sie geben die Größe des Felds mit Hilfe eines Index an. Sie geben die erste und die letzte Zahl der einzelnen Dimensionen (maximal 6) in dem Feld an. Den Index geben Sie in eckigen Klammern an, wobei jede Dimension durch Komma getrennt und die erste und die letzte Zahl der Dimension durch zwei Punkte voneinander getrennt werden. Der folgende Index beispielsweise definiert ein dreidimensionales Feld:

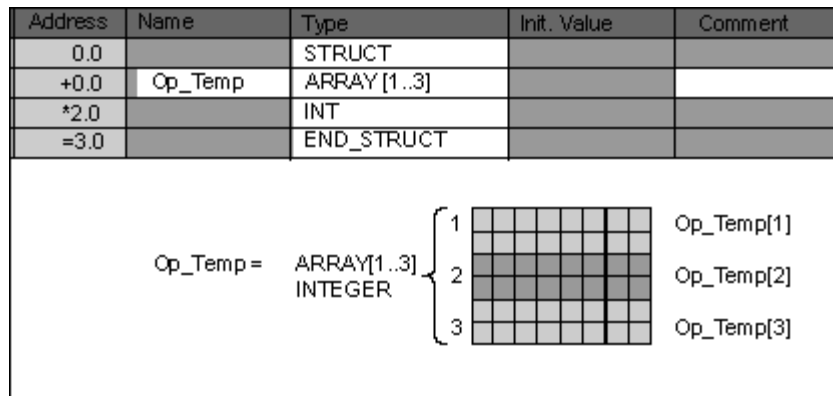
[1..5,-2..3,30..32]

- Sie geben den Datentyp der Daten an, die in dem Feld gespeichert werden sollen.

Beispiel 1

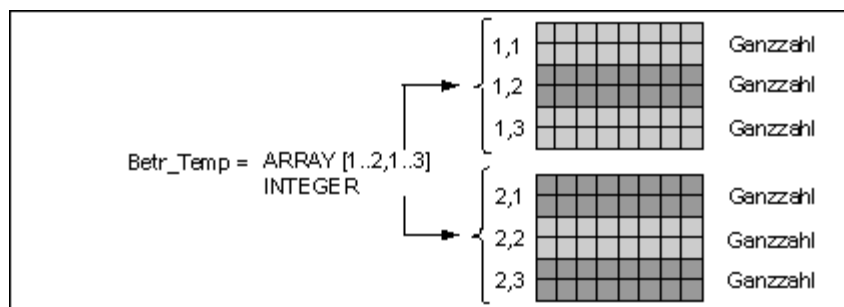
Nachfolgendes Bild zeigt ein Feld aus drei Ganzzahlen. Sie greifen über den Index auf die Daten zu, die in einem Feld gespeichert sind. Bei dem Index handelt es sich um die Zahl in eckigen Klammern. Der Index der zweiten Ganzzahl beispielsweise ist `Betr_Temp[2]`.

Ein Index kann ein beliebiger ganzzahliger Wert (-32768 bis 32767) sein, einschließlich negativer Werte. Das Feld im nachfolgenden Bild könnte auch als `ARRAY [-1..1]` definiert werden. Der Index der ersten Ganzzahl wäre dann `Betr_Temp[-1]`, der zweiten `Betr_Temp[0]` und der dritten Ganzzahl `Betr_Temp[1]`.



Beispiel 2

Ein Feld kann auch eine multidimensionale Gruppe von Datentypen beschreiben. Nachfolgendes Bild zeigt ein zweidimensionales Feld aus Ganzzahlen.



Sie greifen auf die Daten in einem multidimensionalen Feld über den Index zu. Bei dem Beispiel ist die erste Ganzzahl `Betr_Temp[1,1]`, die dritte ist `Betr_Temp[1,3]`, die vierte ist `Betr_Temp[2,1]` und die sechste ist `Betr_Temp[2,3]`.

Sie können maximal sechs Dimensionen (sechs Indizes) für ein Feld definieren. Sie könnten beispielsweise die Variable `Betr_Temp` folgendermaßen als sechsdimensionales Feld definieren:

```
ARRAY [1..3,1..2,1..3,1..4,1..3,1..4]
```

Der Index des ersten Elements in diesem Feld ist `Betr_Temp[1,1,1,1,1,1]`. Der Index des letzten Elements ist `Betr_Temp[3,2,3,4,3,4]`.

Erstellen von Feldern

Sie definieren Felder, wenn Sie Daten in einem DB oder in der Variablendeklaration deklarieren. Wenn Sie das Feld deklarieren, dann geben Sie das Schlüsselwort (ARRAY) an, danach die Größe in eckigen Klammern:

[unterer Grenzwert..oberer Grenzwert]

Bei einem multidimensionalen Feld geben Sie die zusätzlichen oberen und unteren Grenzwerte an und trennen die einzelnen Dimensionen durch ein Komma. Nachfolgendes Bild zeigt die Deklaration zum Erstellen eines Felds im Format 2 x 3.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Waerme_2X3	ARRAY[1..2,1..3]		
*2.0		INT		
=6.0		END_STRUCT		

Eingeben von Anfangswerten für ein Feld

Sie können beim Erstellen der Felder jedem Feldelement einen Anfangswert zuordnen. STEP 7 bietet Ihnen zwei Möglichkeiten zum Eingeben von Anfangswerten:

- Eingeben von individuellen Werten: Für jedes Element des Felds geben Sie einen gültigen Wert an (für den Datentyp des Felds). Geben Sie die Werte in der Reihenfolge der Elemente an: [1,1]. Beachten Sie, dass Sie die einzelnen Elemente durch Komma voneinander trennen müssen.
- Angeben eines Wiederholfaktors: Bei sequenziellen Elementen, die über den gleichen Anfangswert verfügen, können Sie die Anzahl der Elemente (den Wiederholfaktor) und den Anfangswert für diese Elemente angeben. Das Format zum Eingeben des Wiederholfaktors ist $x(y)$, wobei x der Wiederholfaktor und y der Wert ist, der wiederholt werden soll.

Wenn Sie das Feld verwenden, das im vorherigen Bild deklariert wird, können Sie den Anfangswert für alle sechs Elemente folgendermaßen angeben: 17, 23, -45, 556, 3342, 0. Sie könnten den Anfangswert von allen sechs Elementen auch auf 10 setzen, indem Sie 6(10) angeben. Sie könnten für die ersten beiden Elemente bestimmte Werte angeben und dann die übrigen vier Elemente auf 0 setzen, indem Sie 17, 23, 4(0) angeben.

Zugreifen auf Daten in einem Feld

Sie greifen auf die Daten in einem Feld über den Index des bestimmten Elements in dem Feld zu. Der Index wird mit dem symbolischen Namen verwendet.

Beispiel: Wenn das im vorherigen Bild deklarierte Feld an dem ersten Byte von DB20 (Motor) beginnt, greifen Sie auf das zweite Element in dem Feld über folgende Adresse zu:

Motor.Waerme_2x3[1,2].

Verwenden von Feldern als Parameter

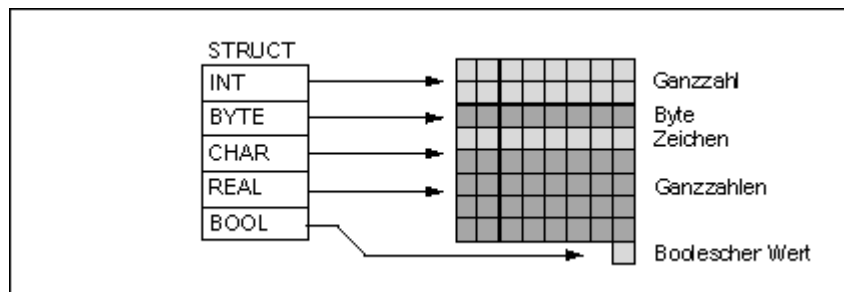
Sie können Felder als Parameter übergeben. Wenn ein Parameter in der Variablendeklaration als ARRAY deklariert ist, müssen Sie das ganze Feld übergeben (und nicht nur einzelne Elemente). Ein Element eines Felds kann jedoch einem Parameter zugeordnet werden, wenn Sie einen Baustein aufrufen, sofern das Element des Felds dem Datentyp des Parameters entspricht.

Wenn Sie Felder als Parameter verwenden, müssen die Felder nicht den gleichen Namen haben (sie brauchen gar keinen Namen zu haben). Beide Felder (der Formalparameter und der Aktualparameter) müssen aber die gleiche Struktur vorweisen. Ein Feld im Format 2 x 3 aus Ganzzahlen zum Beispiel kann als Parameter nur übergeben werden, wenn der Formalparameter des Bausteins ein Feld im Format 2 x 3 aus Ganzzahlen definiert und auch der Aktualparameter, der über die Aufrufoperation bereitgestellt wird, ein Feld im Format 2 x 3 aus Ganzzahlen ist.

A.3.3.4 Verwenden von Strukturen für den Datenzugriff

Strukturen

Eine Struktur verknüpft verschiedene Datentypen (elementare und zusammengesetzte Datentypen, einschließlich Felder und Strukturen) zu einer Einheit. So können Sie Daten entsprechend der Steuerung für Ihren Prozess gruppieren. Sie können so auch Parameter als eine Dateneinheit übergeben, und nicht als einzelne Elemente. Nachfolgendes Bild zeigt eine Struktur, die aus einer Ganzzahl, einem Byte, einem Zeichen, einer Gleitpunktzahl und einem booleschen Wert besteht.



Eine Struktur kann in maximal 8 Ebenen geschachtelt sein (z. B. eine Struktur aus Strukturen, die Felder enthält).

Erstellen einer Struktur

Sie definieren Strukturen beim Deklarieren von Daten innerhalb eines DB oder in der Variablendeklaration eines Codebausteins.

Nachfolgendes Bild zeigt die Deklaration einer Struktur (*Stapel_1*), die aus folgenden Elementen besteht: eine Ganzzahl (zum Speichern der Menge), ein Byte (zum Speichern der Ursprungsdaten), ein Zeichen (zum Speichern des Steuercode), eine Gleitpunktzahl (zum Speichern der Temperatur) und ein boolescher Merker (zum Beenden des Signals).

Adresse	Name	Typ	Anfangswert	Kommentar
0.0	Stapel_1	STRUCT		
+0.0	Menge	INT	100	
+2.0	Ursprungsdaten	BYTE		
+4.0	Steuercode	CHAR		
+6.0	Temperatur	REAL	120	
+8.0	Ende	BOOL	FALSE	
=10.0		END_STRUCT		

Zuordnen von Anfangswerten für eine Struktur

Wenn Sie jedem Element einer Struktur einen Anfangswert zuordnen möchten, geben Sie jeweils einen gültigen Wert für den Datentyp und den Namen des Elements ein. Sie können beispielsweise (der im vorherigen Bild deklarierten Struktur) die folgenden Anfangswerte zuordnen:

```
Menge           =      100
Ursprungsdaten =      B#(0)
Steuercode      =      'Z'
Temperatur      =      120
Ende            =      False
```

Speichern von und Zugreifen auf Daten in Strukturen

Sie greifen auf die einzelnen Elemente einer Struktur zu. Sie können symbolische Adressen (z. B. *Stapel_1.Temperatur*) verwenden. Sie können aber auch die absolute Adresse angeben, unter der das Element gespeichert ist (Beispiel: ist *Stapel_1* in DB20 mit Beginn bei Byte 0 gespeichert, dann ist die absolute Adresse für die *Menge* *DB20.DBW0* und die Adresse für die *Temperatur* ist *DB20.DBD6*).

Verwenden von Strukturen als Parameter

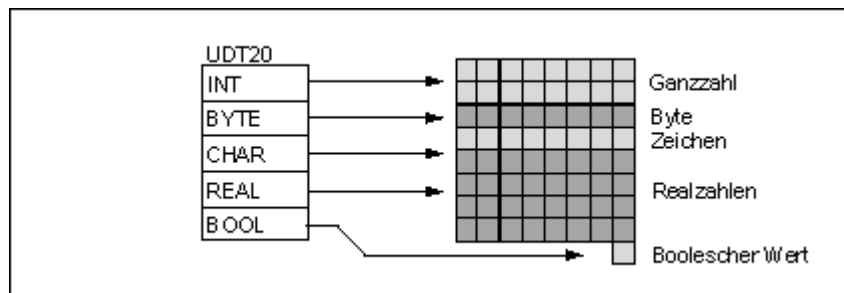
Sie können Strukturen als Parameter übergeben. Wenn ein Parameter in der Variablendeklaration als STRUCT deklariert ist, müssen Sie eine Struktur mit dem gleichen Aufbau übergeben. Ein Element einer Struktur kann jedoch einem Parameter zugeordnet werden, wenn Sie einen Baustein aufrufen, sofern das Element der Struktur dem Datentyp des Parameters entspricht.

Wenn Sie Strukturen als Parameter verwenden, müssen beide Strukturen (für den Formalparameter und den Aktualparameter) den gleichen Aufbau der Daten vorweisen, d. h. die gleichen Datentypen müssen in der gleichen Reihenfolge angeordnet sein.

A.3.3.5 Verwenden von anwenderdefinierten Datentypen für den Datenzugriff

Anwenderdefinierte Datentypen

Anwenderdefinierte Datentypen (UDT) können elementare und zusammengesetzte Datentypen verknüpfen. Sie können UDTs mit einem Namen versehen und mehrfach verwenden. Nachfolgendes Bild zeigt die Struktur eines UDT, der aus einer Ganzzahl, einem Byte, einem Zeichen, einer Gleitpunktzahl und einem booleschen Wert besteht.



Anstatt alle Datentypen einzeln oder als Struktur einzugeben, müssen Sie nur "UDT20" als Datentyp angeben und STEP 7 ordnet automatisch den entsprechenden Speicherplatz zu.

Erstellen eines anwenderdefinierten Datentyps

UDTs definieren Sie mit STEP 7. Nachfolgendes Bild zeigt einen UDT, der aus den folgenden Elementen besteht: eine Ganzzahl (zum Speichern der Menge), ein Byte (zum Speichern der Ursprungsdaten), ein Zeichen (zum Speichern des Steuercode), eine Gleitpunktzahl (zum Speichern der Temperatur) und ein boolescher Merker (zum Beenden des Signals). Sie können dem UDT in der Symboltabelle einen symbolischen Namen zuordnen (z. B. *Prozessdaten*).

Adresse	Name	Typ	Anfangswert	Kommentar
0.0	Stapel_1	STRUCT		
+0.0	Menge	INT	100	
+2.0	Ursprungsdaten	BYTE	B#16#0	
+4.0	Steuercode	CHAR		
+6.0	Temperatur	REAL	1.200000e+002	
+8.0	Ende	BOOL	FALSE	
=10.0		END_STRUCT		

Nachdem Sie einen UDT erstellt haben, können Sie den UDT wie einen Datentyp verwenden, so z. B. wenn Sie für eine Variable den Datentyp *UDT200* in einem DB deklarieren (oder in der Variablendeklaration eines FB).

Nachfolgendes Bild zeigt einen DB mit der Variablen *Prozessdaten_1* mit dem Datentyp UDT200. Sie geben nur *UDT200* und *Prozessdaten_1* an. Die kursiv dargestellten Felder werden erstellt, wenn Sie den DB übersetzen.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+10.0	<i>Prozessdaten_1</i>	UDT200		
=10.0		END_STRUCT		

Zuordnen von Anfangswerten für einen anwenderdefinierten Datentyp

Wenn Sie jedem Element eines UDT Anfangswerte zuordnen möchten, geben Sie für den Datentyp und den Namen eines jeden Elements einen gültigen Wert an. Sie können beispielsweise (dem im vorherigen Bild deklarierten UDT) die folgenden Anfangswerte zuordnen:

```
Menge           =      100
Ursprungsdaten  =      B#16#0
Steuercode=     'Z'
Temperatur      =      1.200000e+002
Ende            =      False
```

Wenn Sie eine Variable als UDT deklarieren, dann sind die Anfangswerte der Variablen die Werte, die Sie beim Erstellen des UDT angegeben haben.

Speichern von und Zugreifen auf Daten in einem anwenderdefinierten Datentyp

Sie greifen auf die einzelnen Elemente eines UDT zu. Sie können symbolische Adressen (z. B. *Stapel_1.Temperatur*) verwenden. Sie können aber auch die absolute Adresse angeben, unter der das Element gespeichert ist (Beispiel: ist *Stapel_1* in DB20 mit Beginn bei Byte 0 gespeichert, dann ist die absolute Adresse für die *Menge* *DB20.DBW0* und die Adresse für die *Temperatur* ist *DB20.DBD6*).

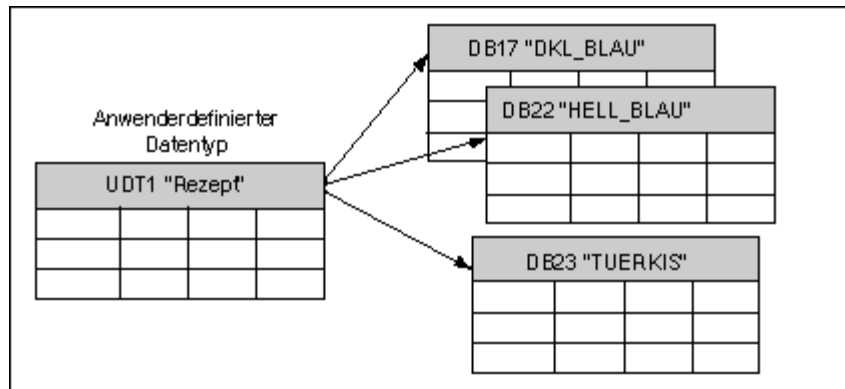
Verwenden von anwenderdefinierten Datentypen als Parameter

Sie können Variablen vom Datentyp UDT als Parameter übergeben. Wenn der Parameter in der Variablendeklaration als UDT deklariert ist, müssen Sie einen UDT übergeben, der die gleiche Struktur der Datenelemente hat. Ein Element eines UDT kann aber auch beim Aufrufen eines Bausteins einem Parameter zugeordnet werden, sofern das Element des UDT mit dem Datentyp des Parameters übereinstimmt.

Vorteile von DBs mit zugeordnetem UDT

Mit Hilfe von einmal erstellten UDTs können Sie eine Vielzahl von Datenbausteinen erzeugen, die die gleiche Datenstruktur haben. Diese Datenbausteine können von Ihnen durch Eingabe von unterschiedlichen Aktualwerten genau für die entsprechende Aufgabe angepasst werden.

Strukturieren Sie z. B. einen UDT für ein Rezept (z. B. für das Mischen von Farben), dann können Sie diesem UDT mehrere DBs zuordnen, die jeweils die anderen Mengenangaben beinhalten:



Die Struktur des Datenbausteins wird durch den zugeordneten UDT festgelegt.

A.3.4 Parametertypen

Zusätzlich zu elementaren und zusammengesetzten Datentypen können Sie Parametertypen für Formalparameter definieren, die zwischen Bausteinen übergeben werden. STEP 7 kennt folgende Parametertypen:

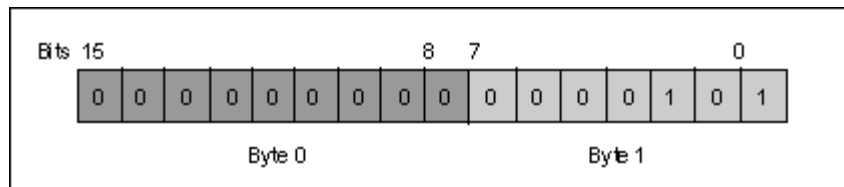
- **TIMER** oder **COUNTER**: legen eine bestimmte Zeit oder einen bestimmten Zähler fest, der bei der Bearbeitung verwendet werden soll. Wenn Sie einen Formalparameter vom Parametertyp **TIMER** oder **COUNTER** versorgen, muss der zugehörige Aktualparameter eine Zeit oder ein Zähler sein, d. h. Sie geben ein "T" oder "Z" an, auf das eine positive Ganzzahl folgt.
- **BLOCK**: legt einen bestimmten Baustein fest, der als Eingang oder als Ausgang verwendet werden soll. Die Deklaration des Parameters bestimmt die Bausteinart (FB, FC, DB usw.), die verwendet werden soll. Wenn Sie einen Formalparameter vom Parametertyp **BLOCK** versorgen, geben Sie eine Bausteinadresse als Aktualparameter an. Beispiel: "FC101" (bei absoluter Adressierung) oder "Ventil" (bei symbolischer Adressierung).
- **POINTER**: referenziert die Adresse einer Variablen. Ein Pointer enthält eine Adresse anstelle eines Werts. Wenn Sie einen Formalparameter vom Parametertyp **POINTER** versorgen, geben Sie als Aktualparameter die Adresse an. In STEP 7 können Sie einen Pointer im Pointerformat oder einfach als Adresse (z. B. M 50.0) angeben. Beispiel für ein Pointerformat zum Adressieren der Daten, die in M 50.0 beginnen: P#M50.0
- **ANY**: wird verwendet, wenn der Datentyp des Aktualparameters nicht bekannt ist oder wenn ein beliebiger Datentyp eingesetzt werden kann. Nähere Informationen zum Parameter **ANY** entnehmen Sie den Abschnitten "Format des Parametertyps **ANY**" bzw. "Verwenden des Parametertyps **ANY**".

Ein Parametertyp kann auch ein anwenderdefinierter Datentyp (UDT) sein. Nähere Informationen zu UDTs entnehmen Sie dem Abschnitt "Verwenden von anwenderdefinierten Datentypen für den Datenzugriff".

Parameter	Größe	Beschreibung
TIMER	2 Bytes	Kennzeichnet eine bestimmte Zeit, die vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: T1
COUNTER	2 Bytes	Kennzeichnet einen bestimmten Zähler, der vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: Z10
BLOCK_FB BLOCK_FC BLOCK_DB BLOCK_SDB	2 Bytes	Kennzeichnet einen bestimmten Baustein, der vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: FC101 DB42
POINTER	6 Bytes	Kennzeichnet die Adresse. Format: P#M50.0
ANY	10 Bytes	Wird verwendet, wenn der Datentyp des Aktualparameters nicht bekannt ist. Format: P#M50.0 BYTE 10 ANY-Format bei Datentypen P#M100.0 WORD 5 L#1COUNTER 10 ANY-Format bei Parametertypen

A.3.4.1 Format der Parametertypen BLOCK, COUNTER, TIMER

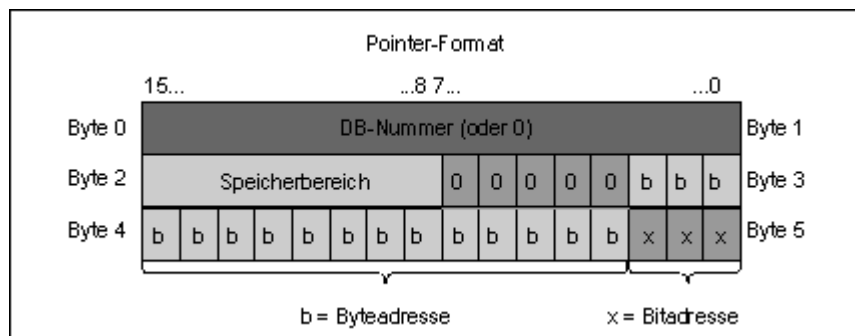
STEP 7 speichert die Parametertypen BLOCK, COUNTER und TIMER als Binärzahlen in einem Wort (32 Bits). Nachfolgendes Bild zeigt das Format dieser Parametertypen.



Die zulässige Anzahl der Bausteine, Zeiten und Zähler ist von Ihrer Variante der S7-CPU abhängig. Nähere Informationen zu der zulässigen Anzahl an Zeiten und Zählern sowie zu der maximalen Anzahl an verfügbaren Bausteinen entnehmen Sie den Datenblättern zu Ihrer CPU im Handbuch "Automatisierungssystem S7-300, Aufbau, CPU-Daten" oder im Installationshandbuch "Automatisierungssysteme S7-400/M7-400, Aufbauen".

A.3.4.2 Format des Parametertyps POINTER

STEP 7 speichert den Parametertyp POINTER in 6 Bytes (48 Bits). Nachfolgendes Bild zeigt die Art der Daten, die in jedem Byte gespeichert werden.



Der Parametertyp POINTER speichert die folgenden Informationen:

- DB-Nummer (bzw. 0, wenn die Daten nicht in einem DB gespeichert werden)
- Speicherbereich in der CPU (Nachfolgende Tabelle zeigt die Hexadezimalcodes der Speicherbereiche für den Parametertyp POINTER)

Hexadezimalcode	Speicherbereich	Beschreibung
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DB	Datenbaustein
b#16#85	DI	Instanz-Datenbaustein
b#16#86	L	Lokaldaten (L-Stack)
b#16#87	V	Vorherige Lokaldaten

- Adresse der Daten (im Format Byte.Bit)
- STEP 7 bietet das Pointer-Format: p#Speicherbereich Byte.Bit_Adresse.
 (Wenn der Formalparameter als Parametertyp POINTER deklariert wurde, dann brauchen Sie nur den Speicherbereich und die Adresse anzugeben.
 STEP 7 formatiert Ihren Eintrag dann automatisch in das Pointer-Format um.)
 Die folgenden Beispiele zeigen, wie Sie den Parametertyp POINTER für die Daten, die bei M50.0 beginnen, eingeben:
- P#M50.0
 - M50.0 (wenn der Formalparameter als POINTER deklariert wurde)

A.3.4.3 Verwenden des Parametertyps POINTER

Ein Pointer wird verwendet, um auf einen Operanden zu zeigen. Der Vorteil dieser Adressierung ist, dass Sie den Operanden der Anweisung während der Programmbearbeitung dynamisch modifizieren können.

Pointer zur speicherindirekten Adressierung

Programmanweisungen, die mit der speicherindirekten Adressierung arbeiten, setzen sich zusammen aus einer Operation, einem Operandenkennzeichen und einem Versatz (der Versatz muss in eckigen Klammern angegeben werden).

Beispiel für einen Pointer im Doppelwortformat:

L	P#8.7	Lade den Wert des Pointers in AKKU 1.
T	MD2	Transferiere den Pointer nach MD2.
U	E [MD2]	Frage den Signalzustand an Eingang E 8.7 ab,
=	A [MD2]	und weise den Signalzustand Ausgang A 8.7 zu.

Pointer zur bereichsinternen und bereichsübergreifenden Adressierung

Die Programmanweisungen, die mit diesen Adressierungen arbeiten, setzen sich zusammen aus einer Operation und den folgenden Teilen:

Operandenkennzeichen, Adressregister-Kennung, Versatz.

Das Adressregister (AR1/2) und der Versatz müssen zusammen in eckigen Klammern angegeben werden.

Beispiel zur bereichsinternen Adressierung

Der Pointer enthält keine Angabe zu einem Speicherbereich:

L	P#8.7	Lade den Wert des Pointers in AKKU 1.
LAR1		Lade den Pointer von AKKU 1 in AR1.
U	E [AR1, P#0.0]	Frage den Signalzustand an Eingang E 8.7 ab und
=	A [AR1, P#1.1]	weise den Signalzustand dem Ausgang zu A 10.0 zu

Der Versatz 0.0 hat keinen Einfluss. Ausgang 10.0 errechnet sich aus 8.7 (AR1) plus den Versatz 1.1. Das Ergebnis ist 10.0 und nicht 9.8, siehe Pointer-Format.

Beispiel zur bereichsübergreifenden Adressierung

Bei der bereichsübergreifenden Adressierung ist im Pointer der Speicherbereich angegeben (im Beispiel E bzw. A).

L	P# E8.7	Lade den Wert des Pointers und Bereichskennung in AKKU 1.
LAR1		Lade Speicherbereich E und die Adresse 8.7 in AR1.
L	P# A8.7	Lade den Wert des Pointers und Bereichskennung in AKKU 1.
LAR2		Lade Speicherbereich A und die Adresse 8.7 in AR2.
U	[AR1, P#0.0]	Frage den Signalzustand an Eingang E 8.7 ab und
=	[AR2, P#1.1]	weise den Signalzustand Ausgang A 10.0 zu.

Der Versatz 0.0 hat keinen Einfluss. Ausgang 10.0 errechnet sich aus 8.7 (AR2) plus 1.1 (Versatz). Das Ergebnis ist 10.0 und nicht 9.8, siehe Pointer-Format.

A.3.4.4 Baustein zum Verändern des Pointers

Mit Hilfe des Beispiel-Bausteins FC3 "Rangieren von Pointern" ist es möglich die Bit- oder Byteadresse eines Pointers zu verändern. Der zu verändernde Pointer wird beim FC-Aufruf an die Variable "Zeiger" übergeben (es können bereichsinterne und bereichsübergreifende Pointer im Doppelwortformat verwendet werden).

Mit dem Parameter "Bit-Byte" können Sie die Bit- oder die Byteadresse des Pointers ändern (0: Bitadresse, 1: Byteadresse). Bei der Variable "Inc_Wert" (im Format Integer) wird die Zahl angegeben, die zum Adressinhalt addiert bzw. subtrahiert werden soll. Hier können Sie auch negative Zahlen zum Dekrementieren der Adresse angeben.

Bei Bitadressenänderung erfolgt ein Übertrag in die Byteadresse (auch bei Dekrementierung); z. B.:

- P#M 5.3, Bit_Byte = 0, Inc_Wert = 6 => P#M 6.1 oder
- P#M 5.3, Bit_Byte = 0, Inc_Wert = -6 => P#M 4.5.

Die Bereichsinformation des Pointers wird durch die Funktion nicht beeinflusst.

Der FC fängt einen Über-/Unterlauf des Zeigers ab. In diesem Fall wird der Zeiger nicht verändert und die ausgangsseitige Variable "RET_VAL" (Fehlerbehandlung möglich) wird auf "1" gesetzt (bis zur nächsten korrekten Bearbeitung der FC 3). Dies ist der Fall, wenn:

- 1.Bitadresse gewählt ist und Inc_Wert >7, oder <-7
- 2.Bit- oder Byteadresse gewählt ist und die Änderung eine "negative" Byteadresse zur Folge hätte
- 3.Bit- oder Byteadresse gewählt ist und die Änderung eine unzulässig große Byteadresse zur Folge hätte.

Beispiel-Baustein in AWL zum Verändern des Pointers:

```

FUNCTION FC 3: BOOL
TITLE =Rangieren von Pointern
//Die FC 3 kann zum Verändern von Pointern verwendet werden.
AUTHOR : AUT1CS1
FAMILY : INDADR
NAME : ADRPOINT
VERSION : 0.0

VAR_INPUT
  Bit_Byte : BOOL ; //0: Bitadresse, 1: Byteadresse
  Inc_Wert : INT ; //Inkrement (wenn Wert neg.=> Dekrement/wenn Wert pos.
                  => Inkrement)
END_VAR

VAR_IN_OUT
  Zeiger : DWORD ; //der zu veränderten Pointer
END_VAR
VAR_TEMP
  Inc_Wert1 : INT ; //Zwischenwert Inkrement
  Zeiger1 : DWORD ; //Zwischenwert Zeiger
  Zw_Wert : DWORD ; //Hilfsvariable
END_VAR
BEGIN
NETWORK
TITLE =
//Der Baustein fängt Veränderungen, welche die Bereichsinformationen des
Zeigers
//verändern, oder zu "negativen" Pointern führen automatisch ab!
  SET   ; //Setze VKE auf 1 und
  R     #RET_VAL; //setze Überlauf zurück
  L     #Zeiger; //Versorgen des temporären
  T     #Zeiger1; //Zwischenwertes Zeiger
  L     #Inc_Wert; //Versorgen des temporären
  T     #Inc_Wert1; //Zwischenwertes Inkrement
  U     #Bit_Byte; //wenn =1, dann Byteadressenoperation
  SPB   Byte; //Sprung zur Byteadressenberechnung
  L     7; //Wenn Wert des Inkrementes > 7,
  L     #Inc_Wert1;
  <I    ;
  S     #RET_VAL; //dann setze RET_VAL und

```

```
SPB   Ende; //springe zum Ende
L     -7; //Wenn Wert des Inkrementes < -7,
<I   ;
S     #RET_VAL; //dann setze RET_VAL und
SPB   Ende; //springe zum Ende
U     L     1.3; //wenn Bit 4 des Wertes = 1 (Inc_Wert neg) ist
SPB   neg; //dann springe zu Bitadressen-Subtraktion
L     #Zeiger1; //Lade Adressinformation des Zeigers
L     #Inc_Wert1; //und addiere das Inkrement
+D   ;
SPA   test; //Springe zum Test auf negatives Ergebnis
neg:  L     #Zeiger1; //Lade Adressinformation des Zeigers
      L     #Inc_Wert1; //Lade das Inkrement
      NEGI  ; //negiere den negativen Wert,
      -D   ; //subtrahiere den Wert
      SPA   test; //und springe zum Test
Byte: L     0; //Beginn der Byteadressenänderung
      L     #Inc_Wert1; //Wenn Inkrement >=0, dann
      <I   ;
      SPB   pos; //springe zu Addition, sonst
      L     #Zeiger1; //lade Adressinformation des Zeigers,
      L     #Inc_Wert1; //lade das Inkrement,
      NEGI  ; //negiere den negativen Wert,
      SLD   3; //schiebe das Inkrement 3 Stellen nach links,
      -D   ; //subtrahiere den Wert
      SPA   test; //und springe zum Test
pos:  SLD   3; //schiebe Inkrement 3 Stellen nach links
      L     #Zeiger1; //lade Adressinformation des Zeigers
      +D   ; //addiere Inkrement
test: T     #Zw_Wert; //Transfer. Ergebnis Berechnungen in Zw_Wert,
      U     L     7.3; //Wenn ungültige Byteadresse (zu groß oder
      S     #RET_VAL; //negativ), dann setze RET_VAL
      SPB   Ende; //und springe zum Ende,
      L     #Zw_Wert; //sonst transferiere Ergebnis
      T     #Zeiger; //in Zeiger
Ende:  NOP  0;
END_FUNCTION
```


A.3.4.5 Format des Paramertyps ANY

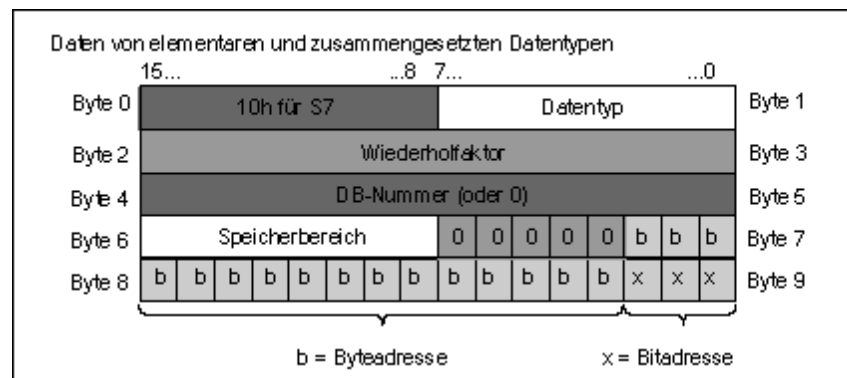
STEP 7 speichert die Daten des Paramertyps ANY in 10 Byte. Beim Aufbau eines Parameters vom Typ ANY müssen Sie darauf achten, dass alle 10 Byte belegt sind, da der aufgerufene Baustein den gesamten Inhalt des Parameters auswertet. Wenn Sie beispielsweise in Byte 4 eine DB-Nummer spezifizieren, müssen Sie auch den Speicherbereich in Byte 6 explizit angeben.

STEP 7 verwaltet die Daten von elementaren und zusammengesetzten Datentypen anders als die Daten für Paramertypen.

ANY-Format bei Datentypen

Bei elementaren und zusammengesetzten Datentypen speichert STEP 7 die folgenden Daten:

- Datentypen
- Wiederholfaktor
- DB-Nummer
- Speicherbereich, in dem die Informationen gespeichert werden
- Anfangsadresse der Daten



Der Wiederholfaktor kennzeichnet eine Menge des gekennzeichneten Datentyps zum Übergeben durch den Paramertyp ANY. Dadurch können Sie einen Datenbereich angeben und auch Felder und Strukturen in Verbindung mit dem Paramertyp ANY verwenden. STEP 7 kennzeichnet Felder und Strukturen als Anzahl (mit Hilfe des Wiederholfaktors) an Datentypen. Sollen z. B. 10 Wörter übergeben werden, muss beim Wiederholfaktor der Wert 10 und beim Datentyp der Wert 04 eingetragen sein.

Die Adresse wird im Format Byte.Bit gespeichert, wobei die Byteadresse in den Bits 0 bis 2 von Byte 7, in den Bits 0 bis 7 von Byte 8 und in den Bits 3 bis 7 von Byte 9 gespeichert wird. Die Bitadresse wird in den Bits 0 bis 2 von Byte 9 gespeichert.

Beim Nullpointer vom Datentyp NIL sind ab Byte 1 alle Bytes mit 0 belegt.

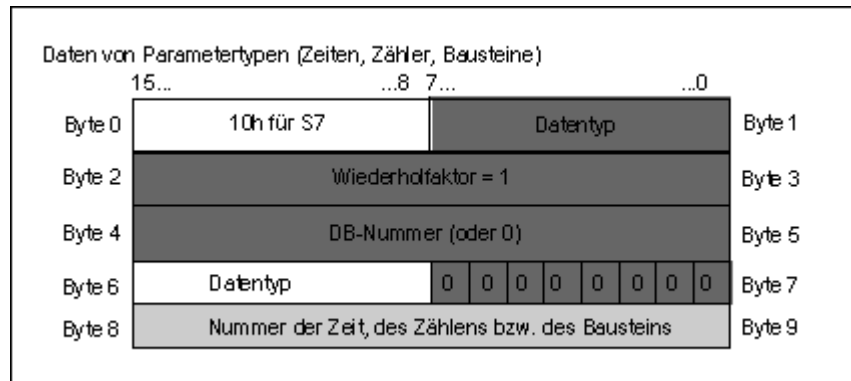
Nachfolgenden Tabellen zeigen die Codierung der Datentypen bzw. der Speicherbereiche für den Parametertyp ANY.

Codierung der Datentypen		
Hexadezimalcode	Datentyp	Beschreibung
b#16#00	NIL	Nullpointer
b#16#01	BOOL	Bits
b#16#02	BYTE	Bytes (8 Bits)
b#16#03	CHAR	Zeichen (8 Bits)
b#16#04	WORD	Wörter (16 Bits)
b#16#05	INT	Ganzzahlen (16 Bits)
b#16#06	DWORD	Wörter (32 Bits)
b#16#07	DINT	Ganzzahlen (32 Bits)
b#16#08	REAL	Gleitpunktzahlen (32 Bits)
b#16#09	DATE	Datum
b#16#0A	TIME_OF_DAY (TOD)	Uhrzeit
b#16#0B	TIME	Zeit
b#16#0C	S5TIME	Datentyp S5TIME
b#16#0E	DATE_AND_TIME (DT)	Datum und Zeit (64 Bits)
b#16#13	STRING	Zeichenkette

Codierung der Speicherbereiche		
Hexadezimalcode	Bereich	Beschreibung
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DB	Datenbaustein
b#16#85	DI	Instanz-Datenbaustein
b#16#86	L	Lokaldaten (L-Stack)
b#16#87	V	Vorherige Lokaldaten

ANY-Format bei Parametertypen

Bei Parametertypen speichert STEP 7 den Datentyp und die Adresse der Parameter. Der Wiederholfaktor ist immer 1. Byte 4, 5 und 7 sind immer 0. Byte 8 und 9 geben die Nummer der Zeit, des Zählers oder des Bausteins an.



Nachfolgende Tabelle zeigt die Codierung der Datentypen für den Parametertyp ANY bei Parametertypen.

Hexadezimalcode	Datentyp	Beschreibung
b#16#17	BLOCK_FB	FB-Nummer
b#16#18	BLOCK_FC	FC-Nummer
b#16#19	BLOCK_DB	DB-Nummer
b#16#1A	BLOCK_SDB	SDB-Nummer
b#16#1C	COUNTER	Nummer des Zählers
b#16#1D	TIMER	Nummer der Zeit

A.3.4.6 Verwenden des Parametertyps ANY

Sie können für einen Baustein Formalparameter definieren, die für Aktualparameter mit beliebigen Datentypen geeignet sind. Dies ist vor allem dann nützlich, wenn der Datentyp des Aktualparameters, der beim Aufrufen des Bausteins bereitgestellt wird, unbekannt ist oder variieren kann (und wenn ein beliebiger Datentyp zulässig ist). In der Variablendeklaration des Bausteins deklarieren Sie den Parameter als Datentyp ANY. In STEP 7 können Sie dann einen Aktualparameter eines beliebigen Datentyps zuordnen.

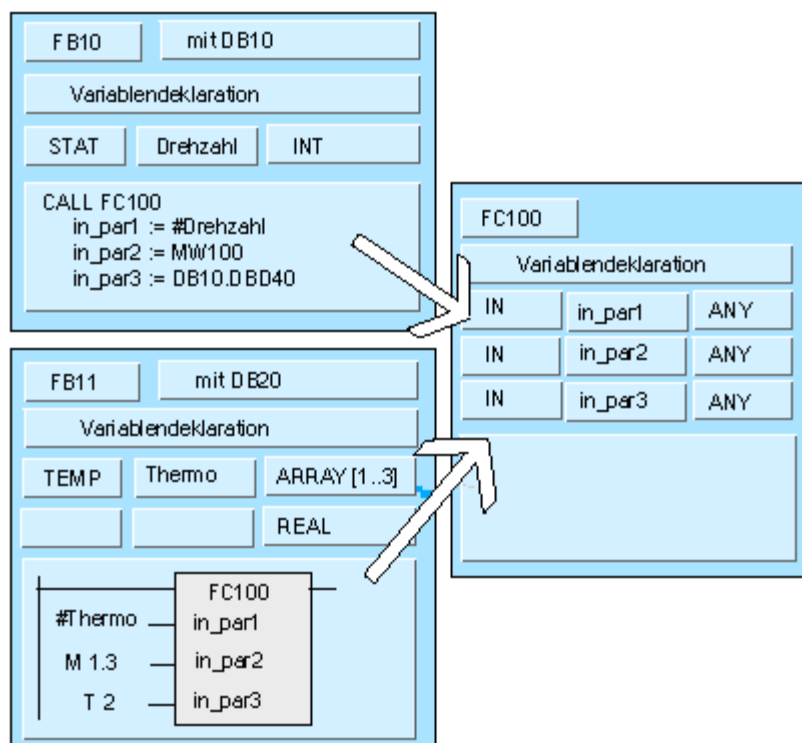
STEP 7 ordnet einer Variablen vom Datentyp ANY 80 Bits Speicherplatz zu. Wenn Sie diesem Formalparameter einen Aktualparameter zuordnen, codiert STEP 7 die Startadresse, den Datentyp und die Länge des Aktualparameters in den 80 Bits. Der aufgerufene Baustein analysiert die 80 Bits an Daten, die für den Parameter ANY gespeichert wurden, und erhält so Informationen, die für die zusätzliche Bearbeitung verwendet werden können.

Zuordnen eines Aktualparameters zu einem Parameter ANY

Wenn Sie für einen Parameter den Datentyp ANY deklarieren, können Sie diesem Formalparameter einen Aktualparameter mit beliebigem Datentyp zuordnen. In STEP 7 können Sie die folgenden Datentypen als Aktualparameter zuordnen:

- Elementare Datentypen: Sie geben die absolute Adresse oder den symbolischen Namen des Aktualparameters an.
- Zusammengesetzte Datentypen: Sie geben den symbolischen Namen der Daten mit zusammengesetztem Datentyp an (z. B. Felder und Strukturen).
- Zeiten, Zähler und Bausteine: Sie geben die Nummer an (z. B. T1, Z20 oder FB6).

Nachfolgendes Bild zeigt, wie Daten an eine FC mit Parametern vom Datentyp ANY übergeben werden können.



In diesem Beispiel hat FC 100 drei Parameter (*in_par1*, *in_par2* und *in_par3*), die als Datentyp ANY deklariert wurden.

- Wenn FB 10 die FC 100 aufruft, übergibt FB 10 eine Ganzzahl (die statische Variable Drehzahl), ein Wort (MW100) und ein Doppelwort in DB 10 (DB10.DBD40).
- Wenn FB 11 die FC 100 aufruft, übergibt FB 11 ein Feld aus Realzahlen (die temporäre Variable "Thermo"), einen booleschen Wert (M 1.3) und eine Zeit (T2).

Angeben eines Datenbereichs für einen Parameter ANY

Sie können einem Parameter ANY aber nicht nur individuelle Operanden (z. B. MW100) zuordnen, sondern Sie können auch einen Datenbereich angeben. Wenn Sie einen Datenbereich als Aktualparameter zuordnen möchten, verwenden Sie das folgende Format einer Konstanten, um die Datenmenge anzugeben, die übergeben werden soll:

p# Bereichskennung Byte.Bit Datentyp Wiederholfaktor

Sie können für das Element *Datentyp* in dem Format für Konstanten alle elementaren Datentypen und den Datentyp DATE_AND_TIME angeben. Handelt es sich bei dem Datentyp nicht um BOOL, dann muss die Bitadresse von 0 (x.0) angegeben werden. Nachfolgende Tabelle zeigt Beispiele für das Format zum Angeben von Speicherbereichen, die an einen Parameter ANY übergeben werden sollen.

Aktualparameter	Beschreibung
p# M 50.0 BYTE 10	Gibt 10 Bytes im Speicherbereich Merker an: MB50 bis MB59.
p# DB10.DBX5.0 S5TIME 3	Gibt 3 Einheiten von Daten vom Datentyp S5TIME an, die in DB10 gespeichert sind: DB Byte 5 bis DB Byte 10.
p# A 10.0 BOOL 4	Gibt 4 Bits im Speicherbereich Ausgänge an: A 10.0 bis A 10.3.

Beispiel für die Verwendung des Parametertyps ANY

Das folgende Beispiel zeigt, wie Sie mit Hilfe des Parametertyps ANY und der Systemfunktion SFC 20 BLKMOV einen Speicherbereich von 10 Byte kopieren können.

AWL	Erläuterung
FUNCTION FC 10:VOID	
VAR_TEMP	
Quelle : ANY;	
Ziel : ANY;	
END_VAR	
BEGIN	
LAR1P#Quelle;	Lade die Anfangsadresse des ANY-Pointers in AR1.
L B#16#10;	Lade die Syntax-ID und
T LB[AR1,P#0.0];	transferiere sie in den ANY-Pointer.
L B#16#02;	Lade Datentyp Byte und
T LB[AR1,P#1.0];	Transferiere ihn in den ANY-Pointer.
L 10;	Lade 10 Byte und
T LW[AR1,P#2.0];	transferiere sie in den ANY-Pointer.
L 22;	Quelle ist der DB22, DBB11
T LW[AR1,P#4.0];	
L P#DBX11.0;	
T LD[AR1,P#6.0];	
LAR1P#Ziel;	Lade die Anfangsadresse des ANY-Pointers in AR1.
L B#16#10;	Lade die Syntax-ID und
T LB[AR1,P#0.0];	transferiere sie in den ANY-Pointer.
L B#16#02;	Lade Datentyp Byte und
T LB[AR1,P#1.0];	Transferiere ihn in den ANY-Pointer.
L 10;	Lade 10 Byte und
T LW[AR1,P#2.0];	transferiere sie in den ANY-Pointer.
L 33;	Ziel ist der DB33, DBB202
T LW[AR1,P#4.0];	
L P#DBX202.0;	
T LD[AR1,P#6.0];	
CALL SFC 20 (Aufruf der Systemfunktion Blockmove
SRCBLK := Quelle,	
RET_VAL := MW 12,	Auswertung des BIE-Bit und des MW 12
DSTBLK := Ziel	
);	
END_FUNCTION	

A.3.4.7 Zuordnen von Datentypen zu Lokaldaten von Codebausteinen

STEP 7 begrenzt die Datentypen (elementare und zusammengesetzte Datentypen sowie Parametertypen), die in der Variablendeklaration den Lokaldaten eines Bausteins zugeordnet werden können.

Gültige Datentypen für die Lokaldaten eines OB

Nachfolgende Tabelle zeigt die Einschränkungen (-) beim Deklarieren von Lokaldaten für einen OB. Weil Sie einen OB nicht aufrufen können, kann ein OB auch nicht über Parameter (Ein-, Aus- oder Durchgänge) verfügen. Weil ein OB keinen Instanz-DB hat, können Sie für einen OB keine statischen Variablen deklarieren. Bei den Datentypen der temporären Variablen eines OB kann es sich um elementare oder zusammengesetzte Datentypen sowie um ANY handeln.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarations- typ	Elementare Datentypen	Zusammen- gesetzte Datentypen	Parame- tertyp	Parame- tertyp	Parame- tertyp	Parame- tertyp	Parame- tertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	—	—	—	—	—	—	—
Ausgang	—	—	—	—	—	—	—
Durchgang	—	—	—	—	—	—	—
Statisch	—	—	—	—	—	—	—
Temporär	● ⁽¹⁾	● ⁽¹⁾	—	—	—	—	● ⁽¹⁾
⁽¹⁾ Im L-Stack des OB gespeichert.							

Gültige Datentypen für die Lokaldaten eines FB

Nachfolgende Tabelle zeigt die Einschränkungen (-) beim Deklarieren von Lokaldaten für einen FB. Wegen des Instanz-DB gibt es weniger Einschränkungen beim Deklarieren von Lokaldaten für einen FB. Für die Deklaration von Eingangsparametern gibt es keinerlei Einschränkungen, für einen Ausgangsparameter können Sie keine Parametertypen deklarieren und für Durchgangsparameter sind nur die Parametertypen POINTER und ANY zulässig. Temporäre Variablen können Sie als Datentyp ANY deklarieren. Alle anderen Parametertypen sind unzulässig.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	●	●	●	●	●	●	●
Ausgang	●	●	—	—	—	—	—
Durchgang	●	● ⁽¹⁾⁽³⁾	—	—	—	●	●
Statisch	●	●	—	—	—	—	—
Temporär	● ⁽²⁾	● ⁽²⁾	—	—	—	—	● ⁽²⁾

⁽¹⁾ Als Verweis (48-Bit-Pointer) im Instanz-DB gespeichert.
⁽²⁾ Im L-Stack des FB gespeichert.
⁽³⁾ STRINGS können nur in der Standardlänge definiert werden.

Gültige Datentypen für die Lokaldaten einer FC

Nachfolgende Tabelle zeigt die Einschränkungen (-) beim Deklarieren von Lokaldaten für eine FC. Weil eine FC keinen Instanz-DB hat, verfügt sie auch nicht über statische Variablen. Für Eingangs-, Ausgangs- und Durchgangsparameter einer FC sind die Parametertypen POINTER und ANY zulässig. Sie können auch temporäre Variablen vom Parametertyp ANY deklarieren.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	●	● ⁽²⁾	●	●	●	●	●
Ausgang	●	● ⁽²⁾	—	—	—	●	●
Durchgang	●	● ⁽²⁾	—	—	—	●	●
Temporär	● ⁽¹⁾	● ⁽¹⁾	—	—	—	—	● ⁽¹⁾

⁽¹⁾ Im L-Stack der FC gespeichert.
⁽²⁾ STRINGS können nur in der Standardlänge definiert werden.

A.3.4.8 Zulässige Datentypen beim Übergeben von Parametern

Regeln für die Parameterübergabe zwischen Bausteinen

Wenn Sie Formalparametern Aktualparameter zuordnen, dann können Sie entweder eine absolute Adresse, einen symbolischen Namen oder eine Konstante angeben. STEP 7 begrenzt die jeweils gültigen Zuordnungen bei den verschiedenen Parametern. Ausgangs- und Durchgangparameter können zum Beispiel keinem konstanten Wert zugeordnet werden (weil der Zweck eines Ausgangs bzw. eines Durchgangs das Ändern des Werts ist). Diese Einschränkungen gelten besonders bei Parametern mit zusammengesetzten Datentypen, denen weder eine absolute Adresse noch eine Konstante zugeordnet werden kann.

Nachfolgende Tabellen zeigen die Einschränkungen (-) bei den Datentypen von Aktualparametern, die Formalparametern zugeordnet werden.

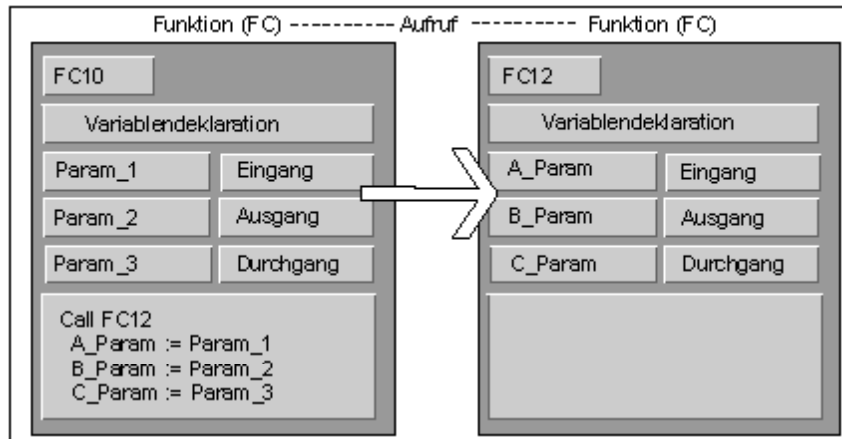
Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Elementare Datentypen				
Deklara- tionstyp	Absolute Adresse	Symbolischer Name (in der Symboltabelle)	Bausteinlokales Symbol	Konstante
Eingang	●	●	●	●
Ausgang	●	●	●	—
Durchgang	●	●	●	—

Zusammengesetzte Datentypen				
Deklara- tionstyp	Absolute Adresse	Symbolischer Name des Elements des DB (in der Symboltabelle)	Bausteinlokales Symbol	Konstante
Eingang	—	●	●	—
Ausgang	—	●	●	—
Durchgang	—	●	●	—

Zulässige Datentypen beim Aufruf einer FC durch eine andere FC

Sie können die Formalparameter einer aufrufenden FC den Formalparametern einer aufgerufenen FC zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FC 10, die als Aktualparameter den Formalparametern von FC 12 zugeordnet werden.



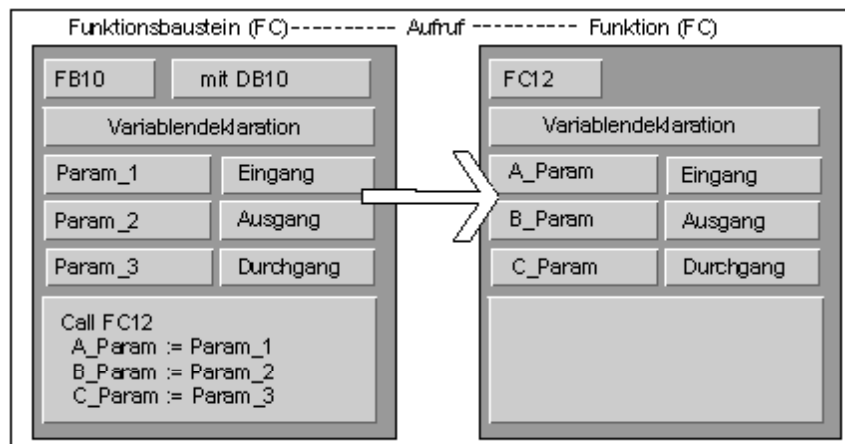
STEP 7 begrenzt die Zuordnung von Formalparametern einer FC als Aktualparameter zu den Formalparametern einer anderen FC. Sie können beispielsweise keine Parameter mit zusammengesetztem Datentyp oder Parametertyp als Aktualparameter zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn eine FC eine andere FC aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	—	—	—	—	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	—	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf einer FC durch einen FB

Sie können die Formalparameter eines aufrufenden FB den Formalparametern einer aufgerufenen FC zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FB 10, die als Aktualparameter den Formalparametern von FC 12 zugeordnet werden.

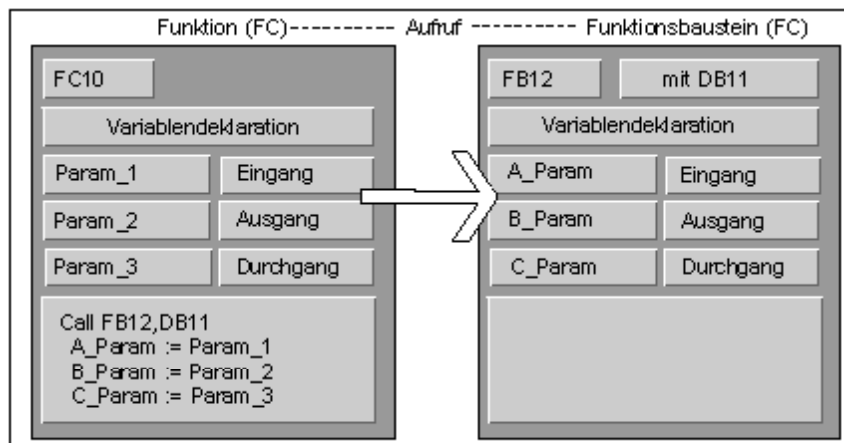


STEP 7 begrenzt die Zuordnung der Formalparameter einer FB zu den Formalparametern einer FC. Sie können beispielsweise keine Parameter mit Parametertyp als Aktualparameter zuordnen. Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn ein FB eine FC aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parameter-typ	Parameter-typ	Parameter-typ	Parameter-typ	Parameter-typ
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	●	—	—	—	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	●	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf eines FB durch eine FC

Sie können die Formalparameter einer aufrufenden FC den Formalparametern eines aufgerufenen FB zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FC 10, die als Aktualparameter den Formalparametern von FB 12 zugeordnet werden.



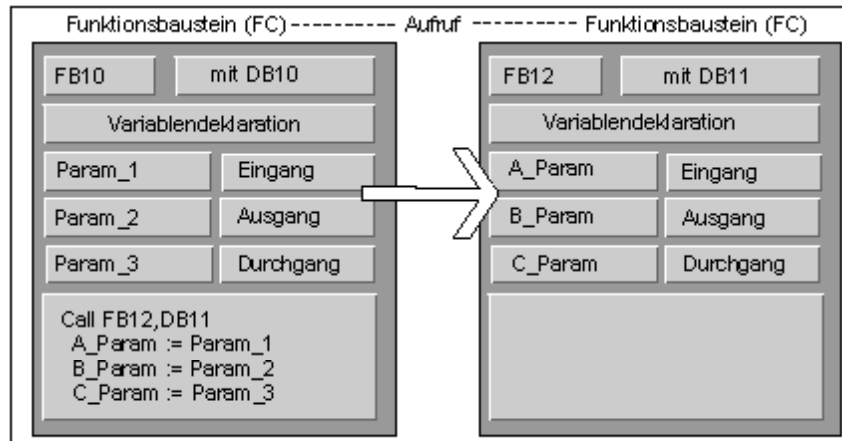
STEP 7 begrenzt die Zuordnung von Formalparametern einer FC zu den Formalparametern eines FB. Sie können beispielsweise Parameter mit zusammengesetztem Datentyp nicht als Aktualparameter zuordnen. Sie können aber Eingangsparameter mit dem Parametertyp TIMER, COUNTER oder BLOCK den Eingangsparametern des aufgerufenen FB zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn eine FC einen FB aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	—	●	●	●	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	—	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf eines FB durch einen anderen FB

Sie können die Formalparameter eines aufrufenden FB den Formalparametern des aufgerufenen FB zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FB 10, die als Aktualparameter den Formalparametern von FB 12 zugeordnet werden.



STEP 7 begrenzt die Zuordnung von Formalparametern eines FB zu den Formalparametern eines anderen FB. Sie können beispielsweise keine Eingangs- und Ausgangsparameter mit zusammengesetztem Datentyp als Aktualparameter den Eingangs- und Ausgangsparametern eines aufgerufenen FB zuordnen. Sie können aber Eingangsparameter mit dem Parametertyp TIMER, COUNTER oder BLOCK den Eingangsparametern des aufgerufenen FB zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn ein FB einen anderen FB aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	●	●	●	●	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	●	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

A.3.4.9 Übergabe an IN_OUT-Parameter eines FB

Bei der Übergabe von zusammengesetzten Datentypen an IN_OUT-Parameter eines Funktionsbausteins (FB) wird die Operandenadresse der Variable übergeben (call by reference).

Bei der Übergabe von elementaren Datentypen an IN_OUT-Parameter eines FB werden die Werte in den Instanz-Datenbaustein kopiert, bevor der FB gestartet wird und aus dem Instanz-Datenbaustein herauskopiert, nachdem der FB beendet ist.

Dadurch können IN_OUT-Variablen elementaren Datentyps mit einem Wert initialisiert werden.

Es ist jedoch nicht möglich, in einem Aufruf eine Konstante als Aktualparameter an der Position einer IN_OUT-Variablen anzugeben, da auf eine Konstante nicht geschrieben werden kann.

Variable vom Datentyp STRUCT oder ARRAY können nicht initialisiert werden, da in diesem Fall im Instanz-Datenbaustein nur eine Adresse liegt.

A.4 Arbeiten mit älteren Projekten

A.4.1 Umsetzen von Version 1-Projekten

Sie haben die Möglichkeit, Projekte wiederzuverwenden, die mit STEP 7 Version 1 erstellt wurden. Dazu wandeln Sie Projekte der Version 1 in Projekte der Version 2 um.

Folgende Komponenten eines Version 1-Projekts bleiben dabei erhalten:

- Projektstruktur mit Programmen
- Bausteine
- AWL-Quelldateien
- Symboltabelle

Die Konfiguration der Hardware wird nicht konvertiert. Die erhaltenen Programmkomponenten können Sie in andere Projekte kopieren. Sie können auch das neue Projekt um eine Station ergänzen, die Sie entsprechend konfigurieren und parametrieren.

Nach dem Umsetzen in Version 2 können Sie in einem Dialog entscheiden, ob nun dieses Version 2-Projekt in ein Projekt Ihrer aktuellen STEP 7-Version umgesetzt werden soll.

Hinweis

Die einzelnen Bausteine bleiben in ihren Eigenschaften Bausteine der Version 1. Der in Version 1 generierte Code wird nicht verändert, und die Bausteine lassen sich daher nicht im Zusammenhang mit Multiinstanzen verwenden.

Wenn Sie in den umgesetzten Bausteinen Multiinstanzen vereinbaren wollen, generieren Sie mit der Applikation "KOP/AWL: Bausteine programmieren" zunächst aus den umgesetzten Bausteinen AWL-Quellen und übersetzen diese wieder in Bausteine.

Das Programmieren von Multiinstanzen ist eine neue Möglichkeit in STEP 7 Version 2 für die Erstellung von Funktionsbausteinen (FBs). Wenn Sie die mit Version 1 erstellten FBs in der gleichen Funktionalität im Version 2-Projekt weiterverwenden wollen, brauchen Sie diese nicht umsetzen.

Vorgehensweise

Um Version 1-Projekte umzusetzen gehen Sie wie folgt vor:

1. Wählen Sie den Menübefehl **Datei > Version 1-Projekt öffnen**.
2. Wählen Sie im folgenden Dialogfeld das Version 1-Projekt aus, das Sie wiederverwenden möchten. Ein Version 1-Projekt erkennen Sie an der Dateierweiterung *.s7a.
3. Geben Sie im nachfolgenden Dialogfeld den Namen des neuen Projekts an, in welches das Version 1-Projekt umgesetzt werden soll.

A.4.2 Umsetzen von Version 2-Projekten

In STEP 7 können Sie mit dem Menübefehl **Datei > Öffnen** auch Version 2-Projekte öffnen.

Version 2-Projekte/Bibliotheken lassen sich mit dem Menübefehl **Datei > Speichern unter** und der Einstellung "mit Reorganisation" auf Ihre aktuelle STEP 7-Version umsetzen (Migration). Das Projekt wird dann als Projekt der aktuellen STEP 7-Version gespeichert.

Sie können Projekte und Bibliotheken aus älteren STEP 7-Version unter Beibehaltung des Formats bearbeiten und speichern, in dem Sie im Dialogfeld "Projekt speichern unter" den Dateityp der älteren STEP 7-Version auswählen. Um beispielsweise die Objekte mit STEP 7 Version 2.1 zu bearbeiten, wählen Sie hier Projekt 2.x bzw. Bibliothek 2.x (als Version 2 speichern ist nicht mehr möglich ab Version 5.1, siehe auch Version 2-Projekte und Bibliotheken bearbeiten).

Bezeichnungen des Dateityps

	STEP 7 V3	ab STEP 7 V4
Dateityp der aktuellen Version	Projekt3.x Bibliothek3.x	Projekt Bibliothek
Dateityp der Vorgängerversion	Projekt2.x Bibliothek2.x	Projekt2.x Bibliothek2.x

Dann steht Ihnen allerdings nur der Funktionsumfang der älteren STEP 7-Version zur Verfügung. Die Projekte bzw. Bibliotheken können danach aber weiterhin mit der älteren STEP 7-Version gepflegt werden.

Hinweis

Bei Übergang von Version 3 nach Versionen ab 4 wurde lediglich die Bezeichnung geändert, das Format blieb identisch. Deshalb gibt es in STEP 7 V4 keinen Dateityp Projekt3.x.

Vorgehensweise

Um Version 2-Projekte im Format der aktuellen STEP 7-Version umzusetzen, gehen Sie wie folgt vor:

1. Führen Sie für das Projekt die Funktion **Datei > Speichern unter...** mit Reorganisation aus.
2. Wählen Sie im Dialogfeld "Projekt speichern unter" den Dateityp "Projekt" und betätigen Sie die Schaltfläche "Speichern".

Um Version 2-Projekte unter Beibehaltung des Formats in die aktuelle STEP 7-Version umzusetzen, gehen Sie wie folgt vor:

1. Führen Sie bei Bedarf Schritt 1 wie oben beschrieben durch.
2. Wählen Sie im Dialogfeld "Projekt speichern unter" den Dateityp der älteren STEP 7-Version und betätigen Sie die Schaltfläche "Speichern".

A.4.3 Hinweise zu STEP 7 V2.1-Projekten mit GD-Kommunikation

- Wenn Sie ein Projekt mit Globaldaten aus STEP 7 V2.1 nach STEP 7 V5 umsetzen wollen, müssen Sie zuvor die GD-Tabelle mit STEP 7 V5.0 im STEP 7 V2.1 Projekt öffnen. Damit werden Ihre bisher projektierten Kommunikationsdaten durch GD-Kommunikation automatisch in die neue Struktur umgewandelt.
- Beim Archivieren von STEP 7 V2.1-Projekten kann es zu einer Fehlermeldung älterer Packprogramme (ARJ, PKZIP...) kommen, wenn im Projekt Dateien mit Namen größer acht Zeichen vorhanden sind. Diese Meldung erscheint auch, wenn in dem STEP 7 V2.1-Projekt das MPI-Netz mit einer Bezeichnung größer 8 Zeichen editiert wurde. Editieren Sie in STEP 7 V2.1-Projekten mit Globaldaten für das MPI-Netz einen Namen mit maximal 8 Zeichen Länge, bevor Sie die Konfigurierung der Globaldaten-Kommunikation erstmals starten.
- Wollen Sie ein Projekt aus STEP 7 V2.1 umbenennen, so müssen Sie die Überschriften der Spalten (CPUs) in der GD-Tabelle neu zuordnen, indem Sie die zugehörige CPU neu auswählen. Wenn Sie den alten Projektnamen wiederherstellen, sind die Zuordnungen wieder vorhanden.

A.4.4 DP-Slaves mit fehlenden oder fehlerhaften GSD-Dateien

Wenn Sie ältere Stationskonfigurationen mit STEP 7 Version 5.1 bearbeiten, dann kann es in seltenen Fällen vorkommen, dass die GSD-Datei eines DP-Slaves fehlt oder nicht übersetzbar ist (z. B. wegen Syntaxfehlern in der GSD-Datei).

In diesem Fall erzeugt STEP 7 einen "Dummy"-Slave, der den projektierten Slave repräsentiert, z. B. nach einem Stations-Laden in PG oder nach dem Öffnen und Weiterbearbeiten eines älteren Projekts. Dieser "Dummy"-Slave ist nur sehr eingeschränkt bearbeitbar; Sie können den Slave-Aufbau (DP-Kennungen) und die Slaveparameter nicht ändern. Ein erneutes Laden in die Station ist aber möglich, die ursprüngliche Konfiguration des Slaves bleibt erhalten. Ebenso kann der gesamte DP-Slave gelöscht werden.

Umkonfigurieren und -parametrieren des DP-Slaves

Wenn Sie den DP-Slave umkonfigurieren oder umparametrieren möchten, dann müssen Sie eine aktuelle GSD-Datei für diesen DP-Slave vom Hersteller anfordern und über den Menübefehl **Extras > GSD-Dateien installieren** verfügbar machen.

Nachdem die korrekte GSD-Datei installiert ist, wird sie zur Darstellung des DP-Slaves benutzt. Der DP-Slave behält seine Daten und ist wieder voll bearbeitbar.

A.5 Beispielprogramme

A.5.1 Beispielprojekte und Beispielprogramme

Die STEP 7 Installations-CD enthält mehrere nützliche Beispielprojekte, die in der folgenden Aufstellung aufgelistet sind. Soweit sie installiert sind, finden Sie die Beispielprojekte im "Öffnen"-Dialog des SIMATIC Managers (Register "Beispielprojekte"). Durch die Installation von Optionspaketen können weitere Beispielprojekte ergänzt werden. Entnehmen Sie den Inhalt dieser Beispielprojekte der Dokumentation des Optionspakets.

Beispiele und Beispielprojekte	Auf CD enthalten	In diesem Buch beschrieben	Beschreibung dt./engl. in OB 1
Projekte "ZDt01_01_STEP7_**" bis "ZDt01_06_STEP7_*" (Erste Schritte und Übungen)	•	Eigenes Handbuch	•
Projekt "ZDt01_11_STEP7_DezP" (Beispiel einer PROFIBUS DP Konfiguration)	•	-	-
Projekt "ZDt01_08_STEP7_Mischen" (Industrieller Mischprozess)	•	•	-
Projekt "ZDt01_09_STEP7_Zebra" (Ampelsteuerung am Zebrastreifen)	•	-	•
Projekt "ZDt01_10_STEP7_COM_SFB" (Datenaustausch zwischen zwei S7-400-CPU's)	•	-	•
Projekt ZXX01_14_HSystem_S7400H (Einstiegsprojekt für hochverfügbare Systeme) Projekt ZXX01_15_HSystem_RED_IO (Einstiegsprojekt für hochverfügbare Systeme mit redundanter Peripherie)	• •	Eigenes Handbuch Eigenes Handbuch	• •
Projekt "ZDt01_11_STEP7_COM_SFC1" und "ZDt01_12_STEP7_COM_SFC2" (Datenaustausch über Kommunikations-SFCs für nicht-projektierte Verbindungen)	•	-	•
Projekt "ZDt01_13_STEP7_PID-Temp" (Beispiel zu den Temperaturreglern FB 58 und FB 59)	•	-	•
Beispiel zur Hantierung mit Uhrzeitalarmen	-	•	-
Beispiel zur Hantierung mit Verzögerungsalarmen	-	•	-
Beispiel zum Maskieren und Demaskieren von Synchronfehlerereignissen	-	•	-
Beispiel zum Sperren und Freigeben von Alarm- und Asynchronfehler ereignissen	-	•	-
Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronfehler ereignissen	-	•	-

Der Schwerpunkt der Beispiele liegt nicht auf der Vermittlung eines bestimmten Programmierstils oder technischen Fachwissens, das Sie zur Steuerung eines bestimmten Prozesses benötigen. Mit Hilfe dieser Beispiele können Sie vielmehr die Schritte nachvollziehen, die beim Programmwurf ausgeführt werden müssen.

Löschen, Kopieren und Installieren von Beispielprojekten

Die mitgelieferten Beispielprojekte können Sie im SIMATIC Manager löschen und nach Bedarf wieder neu installieren. Für die Installation starten Sie das Programm Setup von STEP 7. Die Beispielprojekte lassen sich selektiv nachinstallieren. Mit "Speichern unter" erstellte Kopien der mitgelieferten Beispielprojekte und selbsterstellte "Beispielprojekte" können ausschließlich als Anwenderprojekt gespeichert werden.

Hinweis

Die mit STEP 7 mitgelieferten Beispielprojekte werden, falls diese nicht abgewählt sind, bei einer STEP 7-Installation kopiert. Wenn Sie mitgelieferte Beispielprojekte bearbeitet haben, werden bei einer erneuten STEP 7- Installation die von Ihnen geänderten Beispielprojekte mit dem Original überschrieben.

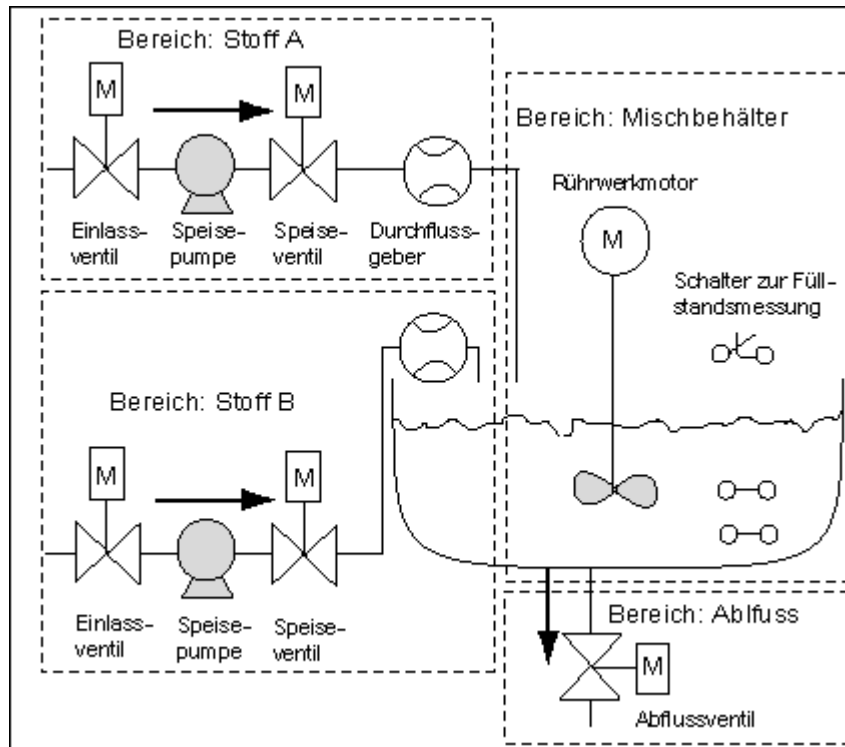
Deswegen sollten Sie mitgelieferte Beispielprojekte vor Änderungen kopieren und nur die Kopie bearbeiten.

A.5.2 Beispielprogramm für einen industriellen Mischprozess

Das Beispielprogramm baut auf den Informationen auf, die bereits im Teil 1 des Handbuchs über das Steuern eines industriellen Mischprozesses vermittelt wurden.

Aufgabenstellung

Zwei Stoffe (Stoff A und Stoff B) sollen in einem Mischbehälter durch ein Rührwerk vermengt werden. Die Masse soll durch ein Abflussventil aus dem Mischbehälter abgelassen werden. Im nachfolgenden Bild sehen Sie ein Diagramm des Beispielprozesses.



Beschreibung der Teilprozesse

Im Teil 1 des Handbuchs wurde beschrieben, wie Sie den Beispielprozess in Funktionsbereiche und einzelne Aufgaben zerlegen können. Im Folgenden finden Sie eine Beschreibung der einzelnen Teilbereiche.

Bereiche Stoff A und Stoff B:

- Die Stoffzuleitungen sind jeweils mit einem Einlass- und einem Speiseventil sowie einer Speisepumpe zu versehen.
- In den Zuleitungen befinden sich Durchflussgeber.
- Das Einschalten der Speisepumpen muss verriegelt sein, wenn der Füllstandsmesser "Behälter ist voll" anzeigt.
- Das Einschalten der Speisepumpen muss verriegelt sein, wenn das Abflussventil geöffnet ist.
- Frühestens 1 Sekunde nach Ansteuerung der Speisepumpe dürfen Einlass- und Speiseventil geöffnet werden.
- Sofort nach dem Stoppen der Speisepumpen (Signal des Durchflussgebers) müssen die Ventile geschlossen werden, um ein Abfließen des Stoffes aus der Pumpe zu verhindern.
- Die Ansteuerung der Speisepumpen ist zeitüberwacht, d. h. innerhalb von 7 Sekunden nach Ansteuerung muss vom Durchflussgeber ein Durchfluss gemeldet werden.
- Die Speisepumpen müssen in kürzester Zeit abgeschaltet werden, wenn die Durchflussgeber während des Laufs der Speisepumpen keinen Durchfluss mehr melden.
- Die Anzahl der Starts der Speisepumpen müssen gezählt werden (Wartungsintervall).

Bereich Mischbehälter:

- Das Einschalten des Rührwerkmotors muss verriegelt sein, wenn der Füllstandsmesser "Behälter unter Minimum" anzeigt oder das Abflussventil offen ist.
- Der Rührwerkmotor gibt ein Rückmeldesignal nach Erreichen der Nennzahl. Wird dieses Signal nicht innerhalb von 10 Sekunden nach Ansteuerung des Motors gemeldet, muss der Motor abgeschaltet werden.
- Die Anzahl der Starts des Rührwerkmotors müssen gezählt werden (Wartungsintervall).
- Im Mischbehälter sind drei Geber zu installieren:
 - Behälter voll: Öffner. Ist der maximale Füllstand erreicht, wird der Kontakt geöffnet.
 - Füllstand im Behälter über Minimum: Schließer. Ist der minimale Füllstand erreicht, wird der Kontakt geschlossen.
 - Behälter nicht leer: Schließer. Ist der Behälter nicht leer, ist der Kontakt geschlossen.

Bereich Abfluss:

- Der Abfluss soll über ein Magnetventil gesteuert werden.
- Das Magnetventil wird durch den Bediener gesteuert, jedoch spätestens beim Signal "Behälter leer" wieder geschlossen.
- Das Öffnen des Abflussventils ist verriegelt, wenn
 - der Rührwerkmotor läuft
 - der "Behälter leer" ist

Bedienpult

Um einem Bediener das Starten und Stoppen sowie das Beobachten des Prozesses zu ermöglichen, soll außerdem ein Bedienpult angelegt werden. Auf dem Bedienpult befinden sich

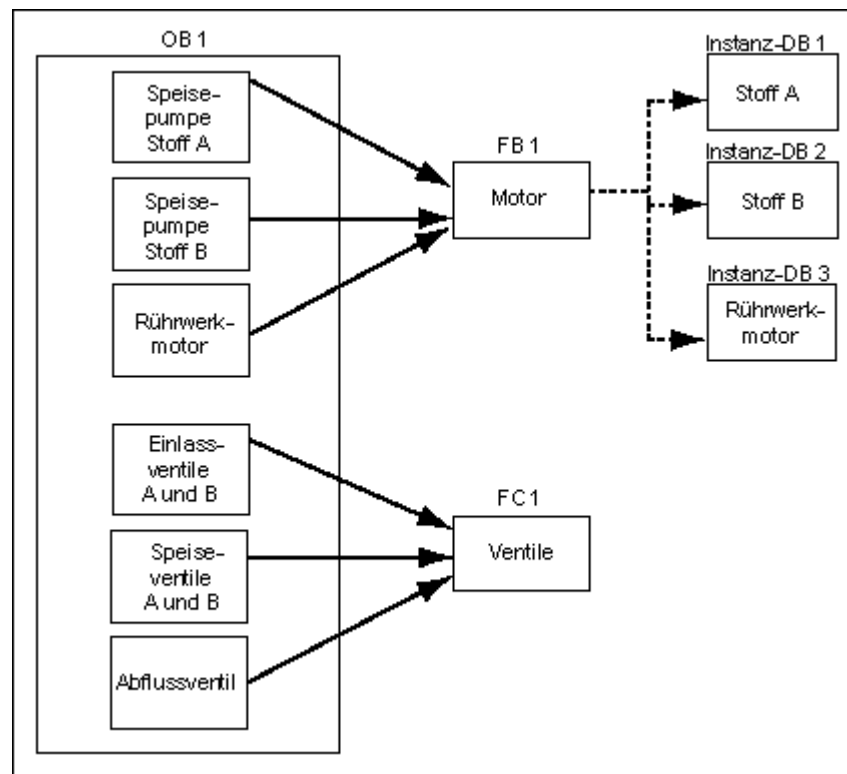
- Taster zum Steuern der wichtigsten Vorgänge. Mit dem Taster "Wartungsanzeige rücksetzen" schalten Sie die Wartungsanzeigelampen derjenigen Motoren aus, bei denen eine Wartung fällig ist, und setzen die zugehörigen Zählerstände für das Wartungsintervall auf 0.
- Anzeigelampen zum Ablesen des Betriebszustands.
- der NOT-AUS-Schalter.

A.5.2.1 Definieren von Codebausteinen

Mit dem Verteilen des Anwenderprogramms auf verschiedene Bausteine und der Hierarchie der Bausteinaufrufe legen Sie die Struktur des Programms fest.

Hierarchie der Bausteinaufrufe

Nachfolgendes Bild zeigt die Hierarchie der Bausteine, die im strukturierten Programm aufgerufen werden sollen.



- OB 1: Er ist die Schnittstelle zum Betriebssystem der CPU und enthält das Hauptprogramm. Im OB 1 werden die Bausteine FB 1 und FC 1 aufgerufen und die spezifischen Parameter übergeben, die für die Steuerung des Prozesses erforderlich sind.
- FB 1: Die Speisepumpe für Stoff A, die Speisepumpe für Stoff B und der Rührwerk-motor können durch einen einzigen Funktionsbaustein gesteuert werden, da die Anforderungen (Ein-, Ausschalten, Einsätze zählen, etc.) identisch sind.
- Instanz-DB 1-3: Die Aktualparameter und die statischen Daten für die Steuerung der Speisepumpen für Stoff A, Stoff B und für den Rührwerk-motor sind unterschiedlich und werden deshalb in drei dem FB 1 zugeordneten Instanz-DBs abgelegt.
- FC 1: Die Einlass- und Speiseventile für die Stoffe A und B sowie das Abflussventil verwenden ebenfalls einen gemeinsamen Codebaustein. Da nur die Funktion Öffnen und Schließen programmiert werden muss, genügt ein einziger FC.

A.5.2.2 Zuordnen von symbolischen Namen

Definieren symbolischer Namen

In dem Beispielprogramm werden Symbole verwendet, die Sie mit STEP 7 in der Symboltabelle definieren müssen. Nachfolgende Tabellen zeigen die symbolischen Namen und die absoluten Adressen der verwendeten Programmelemente.

Symbolische Adressen von Speisepumpe, Rührwerkmotor und Einlassventilen			
Symbolischer Name	Adresse	Datentyp	Beschreibung
Feed_pump_A_start	E 0.0	BOOL	Starttaster Speisepumpe für Stoff A
Feed_pump_A_stop	E 0.1	BOOL	Stopptaster Speisepumpe für Stoff A
Flow_A	E 0.2	BOOL	Stoff A fließt
Inlet_valve_A	A 4.0	BOOL	Ansteuerung des Einlassventils für Stoff A
Feed_valve_A	A 4.1	BOOL	Ansteuerung des Speiseventils für Stoff A
Feed_pump_A_on	A 4.2	BOOL	Anzeigelampe "Speisepumpe Stoff A läuft"
Feed_pump_A_off	A 4.3	BOOL	Anzeigelampe "Speisepumpe Stoff A läuft nicht"
Feed_pump_A	A 4.4	BOOL	Ansteuerung der Speisepumpe für Stoff A
Feed_pump_A_fault	A 4.5	BOOL	Anzeigelampe "Fehler der Speisepumpe A"
Feed_pump_A_maint	A 4.6	BOOL	Anzeigelampe "Wartung der Speisepumpe A erforderlich"
Feed_pump_B_start	E 0.3	BOOL	Starttaster Speisepumpe für Stoff B
Feed_pump_B_stop	E 0.4	BOOL	Stopptaster Speisepumpe für Stoff B
Flow_B	E 0.5	BOOL	Stoff B fließt
Inlet_valve_B	A 5.0	BOOL	Ansteuerung des Einlassventils für Stoff B
Feed_valve_B	A 5.1	BOOL	Ansteuerung des Speiseventils für Stoff B
Feed_pump_B_on	A 5.2	BOOL	Anzeigelampe "Speisepumpe Stoff B läuft"
Feed_pump_B_off	A 5.3	BOOL	Anzeigelampe "Speisepumpe Stoff B läuft nicht"
Feed_pump_B	A 5.4	BOOL	Ansteuerung der Speisepumpe für Stoff B
Feed_pump_B_fault	A 5.5	BOOL	Anzeigelampe "Fehler der Speisepumpe B"
Feed_pump_B_maint	A 5.6	BOOL	Anzeigelampe "Wartung der Speisepumpe B erforderlich"
Agitator_running	E 1.0	BOOL	Rückmeldesignal des Rührwerkmotors
Agitator_start	E 1.1	BOOL	Starttaster Rührwerk
Agitator_stop	E 1.2	BOOL	Stopptaster Rührwerk
Agitator	A 8.0	BOOL	Ansteuerung des Rührwerks
Agitator_on	A 8.1	BOOL	Anzeigelampe "Rührwerk läuft"
Agitator_off	A 8.2	BOOL	Anzeigelampe "Rührwerk läuft nicht"
Agitator_fault	A 8.3	BOOL	Anzeigelampe "Fehler des Rührwerkmotors"
Agitator_maint	A 8.4	BOOL	Anzeigelampe "Wartung des Rührwerkmotors erforderlich"

Symbolische Adressen von Sensoren und Behälterfüllstandsanzeige			
Symbolischer Name	Adresse	Datentyp	Beschreibung
Tank_below_max	E 1.3	BOOL	Sensor "Mischbehälter nicht voll"
Tank_above_min	E 1.4	BOOL	Sensor "Mischbehälter über Minimum"
Tank_not_empty	E 1.5	BOOL	Sensor "Mischbehälter nicht leer"
Tank_max_disp	A 9.0	BOOL	Anzeigelampe "Mischbehälter voll"
Tank_min_disp	A 9.1	BOOL	Anzeigelampe "Mischbehälter unter Minimum"
Tank_empty_disp	A 9.2	BOOL	Anzeigelampe "Mischbehälter leer"

Symbolische Adressen des Abflussventils			
Symbolischer Name	Adresse	Datentyp	Beschreibung
Drain_open	E 0.6	BOOL	Taster zum Öffnen des Abflussventils
Drain_closed	E 0.7	BOOL	Taster zum Schließen des Abflussventils
Drain	A 9.5	BOOL	Ansteuerung des Abflussventils
Drain_open_disp	A 9.6	BOOL	Anzeigelampe "Abflussventil offen"
Drain_closed_disp	A 9.7	BOOL	Anzeigelampe "Abflussventil geschlossen"

Symbolische Adressen der übrigen Programmelemente			
Symbolischer Name	Adresse	Datentyp	Beschreibung
EMER_STOP_off	E 1.6	BOOL	NOT-AUS-Schalter
Reset_maint	E 1.7	BOOL	Resettaster für die Wartungsanzeigelampen aller Motoren
Motor_block	FB 1	FB 1	FB zum Steuern der Pumpen und des Rührwerk-motors
Valve_block	FC 1	FC 1	FC zum Steuern der Ventile
DB_feed_pump_A	DB 1	FB 1	Instanz-DB für die Steuerung der Speisepumpe A
DB_feed_pump_B	DB 2	FB 1	Instanz-DB für die Steuerung der Speisepumpe B
DB_agitator	DB 3	FB 1	Instanz-DB für die Steuerung des Rührwerk-motors

A.5.2.3 Erstellen des FB für den Motor

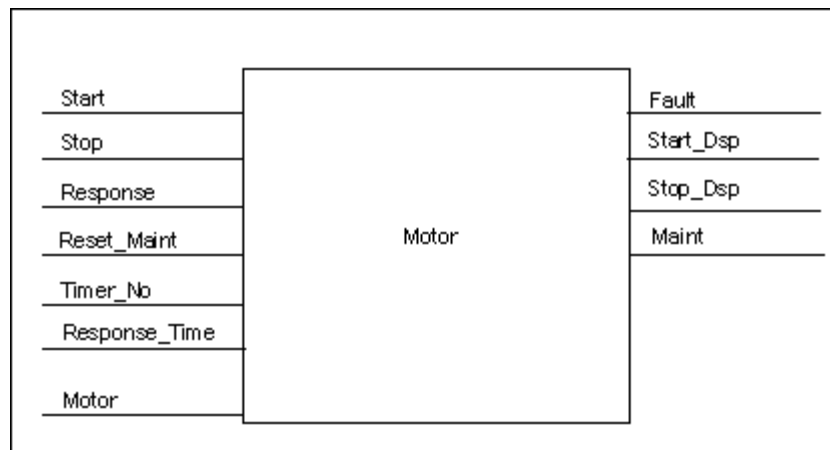
Anforderungen an den FB

Der FB für den Motor enthält die folgenden logischen Funktionen:

- Es gibt einen Start- und einen Stoppeingang.
- Verriegelungen ermöglichen den Betrieb der Geräte (Pumpen und Rührwerkmotor). Der Zustand der Verriegelungen wird in den temporären Lokaldaten (L-Stack) von OB 1 gespeichert (z. B. "Enable_Motor") und mit den Start- und Stoppeingängen verknüpft, wenn der FB für den Motor bearbeitet wird.
- Eine Rückmeldung von den Geräten muss innerhalb einer bestimmten Zeit erscheinen. Andernfalls wird angenommen, dass ein Fehler aufgetreten ist. Der FB stoppt daraufhin den Motor.
- Der Timer und die Zeitdauer für den Rückmelde- bzw. Fehlerzyklus müssen festgelegt werden.
- Ist der Starttaster gedrückt und die Freigabe erteilt, dann schaltet sich das Gerät ein und läuft solange, bis der Stopptaster gedrückt wird.
- Wird das Gerät eingeschaltet, wird ein Timer gestartet. Erscheint das Rückmeldesignal des Geräts nicht vor Ablauf des Timers, dann stoppt das Gerät.

Festlegen der Ein- und Ausgänge

Nachfolgendes Bild zeigt die Ein- und Ausgänge des allgemeinen FB für den Motor.



Definieren der Parameter für den FB

Wenn Sie einen wiederverwendbaren FB für den Motor (zum Steuern der beiden Pumpen und des Rührwerkmotors) erstellen möchten, müssen Sie allgemeine Parameternamen für die Eingänge und die Ausgänge definieren.

Der FB für den Motor im Beispielprozess hat die folgenden Anforderungen:

- Er benötigt Signale vom Bedienpult zum Stoppen bzw. Starten des Motors und der Pumpen.
- Er benötigt ein Rückmeldesignal von Motor und Pumpen, dass der Motor läuft.
- Er muss die Zeit zwischen dem Senden des Signals, das den Motor einschaltet, und dem Empfangen des Rückmeldesignals ermitteln. Erscheint nach einer bestimmten Zeit kein Rückmeldesignal, muss der Motor ausgeschaltet werden.
- Er muss die entsprechenden Anzeigen auf dem Bedienpult ein- bzw. ausschalten.
- Er liefert ein Signal zur Ansteuerung des Motors.

Diese Anforderungen können als Eingänge und Ausgänge des FB festgelegt werden. Nachfolgende Tabelle zeigt die Parameter des FB für den Motor in unserem Beispielprozess.

Parametername	Eingang	Ausgang	Durchgang
Start	n		
Stop	n		
Response	n		
Reset_Maint	n		
Timer_No	n		
Response_Time	n		
Fault		n	
Start_Dsp		n	
Stop_Dsp		n	
Maint		n	
Motor			n

Deklarieren der Variablen des FB für den Motor

Sie müssen die Eingangs-, Ausgangs- und Durchgangsparameter des FB für den Motor deklarieren.

Adresse	Deklaration	Name	Typ	Anfangswert
0.0	IN	Start	BOOL	FALSE
0.1	IN	Stop	BOOL	FALSE
0.2	IN	Response	BOOL	FALSE
0.3	IN	Reset_Maint	BOOL	FALSE
2.0	IN	Time_No	TIMER	
4.0	IN	Response_Time	S5TIME	S5T#0MS
6.0	OUT	Fault	BOOL	FALSE
6.1	OUT	Start_Dsp	BOOL	FALSE
6.2	OUT	Stop_Dsp	BOOL	FALSE
6.3	OUT	Maint	BOOL	FALSE
8.0	IN_OUT	Motor	BOOL	FALSE
10.0	STAT	Time_bin	WORD	W#16#0
12.0	STAT	Time_BCD	WORD	W#16#0
14.0	STAT	Starts	INT	0
16.0	STAT	Start_Edge	BOOL	FALSE

Bei FBs werden die Eingangs-, Ausgangs-, Durchgangsparameter und die statischen Variablen in dem Instanz-DB gespeichert, der in der Aufrufanweisung angegeben wird. Die temporären Variablen werden im L-Stack gespeichert.

Programmieren des FB für den Motor

In STEP 7 muss jeder Baustein, der von einem anderen Baustein aufgerufen wird, vor dem Baustein erstellt werden, der den Aufruf enthält. Im Beispielprogramm müssen Sie also den FB für den Motor vor dem OB 1 erstellen.

Der Anweisungsteil des FB 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Start/Stop und Selbsthaltung

```
U(
O #Start
O #Motor
)
UN#Stop
= #Motor
```

Netzwerk 2 Anlaufüberwachung

```
U #Motor
L #Response_Time
SE #Timer_No
UN#Motor
R #Timer_No
L #Timer_No
T #Timer_bin
LC #Timer_No
T #Timer_BCD
U #Timer_No
UN#Response
S #Fault
R #Motor
```

Netzwerk 3 Startlampe und Fehlerrücksetzung

```
U #Response
= #Start_Dsp
R #Fault
```

Netzwerk 4 Stoplampe

```
UN#Response
= #Stop_Dsp
```

Netzwerk 5 Zählung der Starts

```
U #Motor
FP #Start_Edge
SPBN    lab1
L #Starts
+ 1
T #Starts

lab1: NOP 0
```

Netzwerk 6 Wartungsanzeigelampe

```
L #Starts
L 50
>=|
= #Maint
```

Netzwerk 7 Rücksetzen des Zählers für die Anzahl der Starts

```
U #Reset_Maint
U #Maint
SPBN     END
L 0
T #Starts
END: NOP 0
```

Erstellen der Instanz-Datenbausteine

Legen Sie drei Datenbausteine an und öffnen Sie diese nacheinander. Im Dialogfeld "Neuer Datenbaustein" wählen Sie das Optionsfeld "Datenbaustein mit zugeordnetem Funktionsbaustein". Im Listenfeld "Zuordnung" wählen Sie "FB 1". Damit sind die Datenbausteine als Instanz-Datenbausteine mit fester Zuordnung zum FB 1 festgelegt.

A.5.2.4 Erstellen der FC für die Ventile

Anforderungen an die FC

Die Funktion für die Einlass- und Speiseventile sowie für das Abflussventil enthält die folgenden logischen Funktionen:

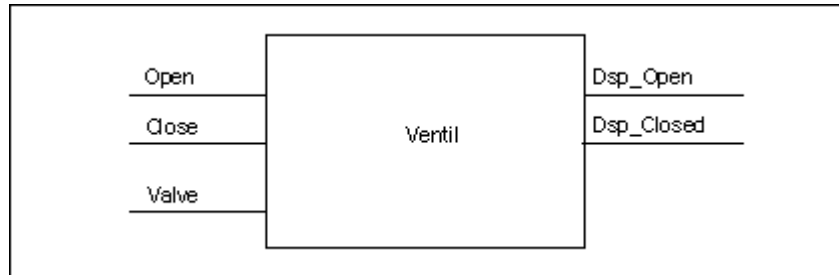
- Es gibt je einen Eingang zum Öffnen und zum Schließen der Ventile.
- Verriegelungen ermöglichen das Öffnen und Schließen der Ventile. Der Zustand der Verriegelungen wird in den temporären Lokaldaten (L-Stack) von OB 1 gespeichert (z. B. "Enable_Valve") und mit den Eingängen zum Öffnen und Schließen verknüpft, wenn die FC für die Ventile bearbeitet wird.

Nachfolgende Tabelle zeigt die Parameter, die an die FC übergeben werden müssen.

Parameter für die Ventile	Eingang	Ausgang	Durchgang
Open	✓		
Close	✓		
Dsp_Open		✓	
Dsp_Closed		✓	
Valve			✓

Festlegen der Ein- und Ausgänge

Nachfolgende Bild zeigt die Ein- und Ausgänge der allgemeinen FC für die Ventile. Die Geräte, die den FB für den Motor aufrufen, übergeben Eingangsparameter. Die FC für die Ventile gibt Ausgangsparameter zurück.



Deklariere die Variablen der FC für die Ventile

Wie bei dem FB für den Motor müssen Sie auch bei der FC für die Ventile die Eingangs-, Ausgangs- und Durchgangparameter deklarieren.

Adresse	Deklaration	Name	Typ	Anfangswert
0.0	IN	Open	BOOL	FALSE
0.1	IN	Close	BOOL	FALSE
2.0	OUT	Dsp_Open	BOOL	FALSE
2.1	OUT	Dsp_Closed	BOOL	FALSE
4.0	IN_OUT	Valve	BOOL	FALSE

Bei FCs werden die temporären Variablen im L-Stack gespeichert. Die Eingangs-, Ausgangs- und Durchgangvariablen werden als Pointer auf den Codebaustein gespeichert, der die FC aufgerufen hat. Zusätzlicher Speicherplatz im L-Stack (nach den temporären Variablen) wird für diese Variablen verwendet.

Programmieren der FC für die Ventile

Auch die Funktion FC 1 für die Ventile müssen Sie vor dem OB 1 erstellen, da aufgerufene Bausteine vor aufrufenden Bausteinen erstellt werden müssen.

Der Anweisungsteil des FC 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Öffnen/Schließen und Selbsthaltung

```
U(  
O #Open  
O #Valve  
)  
UN#Close  
= #Valve
```

Netzwerk 2 Anzeige, ob Ventil geöffnet

```
U #Valve  
= #Dsp_Open
```

Netzwerk 3 Anzeige, ob Ventil geschlossen

```
UN#Valve  
= #Dsp_Closed
```

A.5.2.5 Erstellen des OB 1

OB 1 bestimmt die Struktur des Beispielprogramms. Außerdem enthält OB 1 die Parameter, die an die verschiedenen Bausteine übergeben werden, z. B.:

- Die AWL-Netzwerke für die Speisepumpen und den Rührwerkmotor liefern dem FB für den Motor die Eingangsparameter zum Starten ("Start"), Stoppen ("Stop"), für die Rückmeldung ("Response") und für das Rücksetzen der Wartungsanzeige ("Reset_Maint"). Der FB für den Motor wird in jedem Zyklus des AS bearbeitet.
- Wird der FB für den Motor bearbeitet, legen die Eingänge "Timer_No" und "Response_Time" fest, welcher Timer verwendet wird und innerhalb welcher Zeitdauer eine Rückmeldung erfolgen muss.
- Die FC für die Ventile und der FB für die Motoren werden in jedem Programmzyklus des Automatisierungssystems bearbeitet, da sie im OB1 aufgerufen werden.

Das Programm verwendet den FB für den Motor mit verschiedenen Instanz-DBs, um die Aufgaben für die Steuerung der Speisepumpen und des Rührwerkmotors zu erfüllen.

Deklarieren von Variablen für OB 1

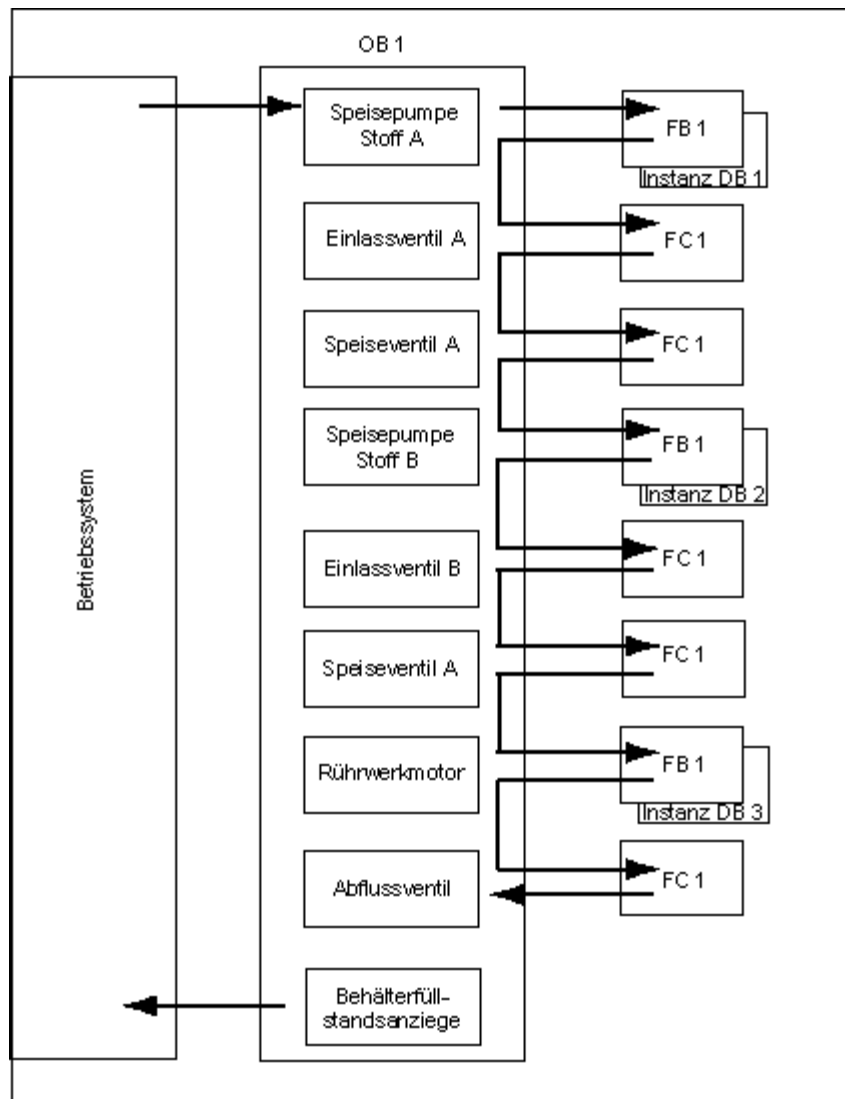
Nachfolgend wird die Variablendeklaration für den OB 1 abgebildet. Die ersten 20 Bytes enthalten die Startinformationen des OB 1 und dürfen nicht geändert werden.

Adresse	Deklaration	Name	Typ
0.0	TEMP	OB1_EV_CLASS	BYTE
1.0	TEMP	OB1_SCAN1	BYTE
2.0	TEMP	OB1_PRIORITY	BYTE
3.0	TEMP	OB1_OB_NUMBR	BYTE
4.0	TEMP	OB1_RESERVED_1	BYTE
5.0	TEMP	OB1_RESERVED_2	BYTE
6.0	TEMP	OB1_PREV_CYCLE	INT
8.0	TEMP	OB1_MIN_CYCLE	INT
10.0	TEMP	OB1_MAX_CYCLE	INT
12.0	TEMP	OB1_DATE_TIME	DATE_AND_TIME
20.0	TEMP	Enable_Motor	BOOL
20.1	TEMP	Enable_Valve	BOOL
20.2	TEMP	Start_Fulfilled	BOOL
20.3	TEMP	Stop_Fulfilled	BOOL
20.4	TEMP	Inlet_Valve_A_Open	BOOL
20.5	TEMP	Inlet_Valve_A_Closed	BOOL
20.6	TEMP	Feed_Valve_A_Open	BOOL
20.7	TEMP	Feed_Valve_A_Closed	BOOL
21.0	TEMP	Inlet_Valve_B_Open	BOOL
21.1	TEMP	Inlet_Valve_B_Closed	BOOL
21.2	TEMP	Feed_Valve_B_Open	BOOL
21.3	TEMP	Feed_Valve_B_Closed	BOOL
21.4	TEMP	Open_Drain	BOOL
21.5	TEMP	Close_Drain	BOOL
21.6	TEMP	Close_Valve_Fulfilled	BOOL

Erstellen des Programms für OB 1

In STEP 7 muss jeder Baustein, der von einem anderen Baustein aufgerufen wird, vor dem Baustein erstellt werden, der den Aufruf enthält. In dem Beispielprogramm müssen Sie sowohl den FB für den Motor als auch die FC für die Ventile vor dem Programm in OB 1 erstellen.

Die Bausteine FB 1 und FC 1 werden im OB 1 mehrfach aufgerufen, der FB 1 mit verschiedenen Instanz-DBs:



Der Anweisungsteil des OB 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Verriegelungen für Speisepumpe A

```
U "EMER_STOP_off"
U "Tank_below_max"
UN"Drain"
= #Enable_Motor
```

Netzwerk 2 Aufruf FB Motor für Stoff A

```
U      "Feed_pump_A_start"
U      #Enable_Motor
=      #Start_Fulfilled
U(
O      "Feed_pump_A_stop"
ON#Enable_Motor
)
=      #Stop_Fulfilled
CALL   "Motor_block", "DB_feed_pump_A"
      Start :=#Start_Fulfilled
      Stop  :=#Stop_Fulfilled
      Response :="Flow_A"
      Reset_Maint :="Reset_maint"
      Timer_No :=T12
      Reponse_Time:=S5T#7S
      Fault :="Feed_pump_A_fault"
      Start_Dsp :="Feed_pump_A_on"
      Stop_Dsp :="Feed_pump_A_off"
      Maint :="Feed_pump_A_maint"
      Motor :="Feed_pump_A"
```

Netzwerk 3 Verzögerung der Ventilfeigabe Stoff A

```
U "Feed_pump_A"
L S5T#1S
SE T 13
UN"Feed_pump_A"
R T 13
U T 13
= #Enable_Valve
```

Netzwerk 4 Einlassventilsteuerung für Stoff A

```
UN"Flow_A"
UN"Feed_pump_A"
=      #Close_Valve_Fulfilled
CALL   "Valve_block"
      Open :=#Enable_Valve
      Close :=#Close_Valve_Fulfilled
      Dsp_Open :=#Inlet_Valve_A_Open
      Dsp_Closed:=#Inlet_Valve_A_Closed
      Valve :="Inlet_Valve_A"
```

Netzwerk 5 Speiseventilsteuerung für Stoff A

```

UN"Flow_A"
UN"Feed_pump_A"
=      #Close_Valve_Fulfilled
CALL   "Valve_block"
      Open  :=#Enable_Valve
      Close :=#Close_Valve_Fulfilled
      Dsp_Open   :=#Feed_Valve_A_Open
      Dsp_Closed:=#Feed_Valve_A_Closed
      Valve  :="Feed_Valve_A"

```

Netzwerk 6 Verriegelungen für Speisepumpe B

```

U "EMER_STOP_off"
U "Tank_below_max"
UN"Drain"
= "Enable_Motor"

```

Netzwerk 7 Aufruf FB Motor für Stoff B

```

U      "Feed_pump_B_start"
U      #Enable_Motor
=      #Start_Fulfilled
U(
O      "Feed_pump_B_stop"
ON#Enable_Motor
)
=      #Stop_Fulfilled
CALL   "Motor_block", "DB_feed_pump_B"
      Start :=#Start_Fulfilled
      Stop  :=#Stop_Fulfilled
      Response :="Flow_B"
      Reset_Maint :="Reset_maint"
      Timer_No :=T14
      Reponse_Time:=S5T#7S
      Fault :="Feed_pump_B_fault"
      Start_Dsp :="Feed_pump_B_on"
      Stop_Dsp :="Feed_pump_B_off"
      Maint :="Feed_pump_B_maint"
      Motor :="Feed_pump_B"

```

Netzwerk 8 Verzögerung der Ventilfeigabe Stoff B

```

U "Feed_pump_B"
L S5T#1S
SE T 15
UN"Feed_pump_B"
R T 15
U T 15
= #Enable_Valve

```

Netzwerk 9 Einlassventilsteuerung für Stoff B

```

UN"Flow_B"
UN"Feed_pump_B"
= #Close_Valve_Fulfilled
CALL "Valve_block"
  Open :=#Enable_Valve
  Close :=#Close_Valve_Fulfilled
  Dsp_Open :=#Inlet_Valve_B_Open
  Dsp_Closed:=#Inlet_Valve_B_Closed
  Valve :="Inlet_Valve_B"

```

Netzwerk 10 Speiseventilsteuerung für Stoff B

```

UN"Flow_B"
UN"Feed_pump_B"
= #Close_Valve_Fulfilled
CALL "Valve_block"
  Open :=#Enable_Valve
  Close :=#Close_Valve_Fulfilled
  Dsp_Open :=#Feed_Valve_B_Open
  Dsp_Closed:=#Feed_Valve_B_Closed
  Valve :="Feed_Valve_B"

```

Netzwerk 11 Verriegelungen für Rührwerk

```

U "EMER_STOP_off"
U "Tank_above_min"
UN"Drain"
= #Enable_Motor

```

Netzwerk 12 Aufruf FB Motor für Rührwerk

```

U "Agitator_start"
U #Enable_Motor
= #Start_Fulfilled
U(
O "Agitator_stop"
ON#Enable_Motor
)
= #Stop_Fulfilled
CALL "Motor_block", "DB_Agitator"
  Start :=#Start_Fulfilled
  Stop :=#Stop_Fulfilled
  Response :="Agitator_running"
  Reset_Maint :="Reset_maint"
  Timer_No :=T16
  Reponse_Time:=S5T#10S
  Fault :="Agitator_fault"
  Start_Dsp :="Agitator_on"
  Stop_Dsp :="Agitator_off"
  Maint :="Agitator_maint"
  Motor :="Agitator"

```

Netzwerk 13 Verriegelungen für Abflussventil

```
U "EMER_STOP_off"  
U "Tank_not_empty"  
UN"Agitator"  
= "Enable_Valve
```

Netzwerk 14 Abflussventilsteuerung

```
U        "Drain_open"  
U        #Enable_Valve  
=        #Open_Drain  
U(  
O        "Drain_closed"  
ON#Enable_Valve  
)  
=        #Close_Drain  
CALL     "Valve_block"  
    Open  :=#Open_Drain  
    Close :=#Close_Drain  
    Dsp_Open    :="Drain_open_disp"  
    Dsp_Closed :="Drain_closed_disp"  
    Valve    :="Drain"
```

Netzwerk 15 Behälterfüllstandsanzeige

```
UN"Tank_below_max"  
= "Tank_max_disp"  
UN"Tank_above_min"  
= "Tank_min_disp"  
UN"Tank_not_empty"  
= "Tank_empty_disp"
```

A.5.3 Beispiel zur Hantierung mit Uhrzeitalarmen

Struktur des Anwenderprogramms "Uhrzeitalarme"

FC 12

OB 10

OB 1 und OB 80

A.5.3.1 Struktur des Anwenderprogramms Uhrzeitalarme

Aufgabenstellung

Ausgang A 4.0 soll in der Zeit von Montag, 5.00 Uhr bis Freitag, 20.00 Uhr gesetzt sein. In der Zeit von Freitag, 20.00 Uhr bis Montag, 5.00 Uhr soll der Ausgang A 4.0 zurückgesetzt sein.

Umsetzung im Anwenderprogramm

Nachfolgende Tabelle zeigt die Teilaufgaben der verwendeten Bausteine.

Baustein	Teilaufgabe
OB 1	Aufruf der Funktion FC 12
FC 12	In Abhängigkeit vom Zustand des Ausgangs A 4.0, des Uhrzeitalarm-Status und der Eingänge E 0.0 und E 0.1 <ul style="list-style-type: none"> • Startzeitpunkt vorgeben • Uhrzeitalarm setzen • Uhrzeitalarm aktivieren • CAN_TINT
OB 10	Abhängig vom aktuellen Wochentag <ul style="list-style-type: none"> • Startzeitpunkt vorgeben • Ausgang A 4.0 setzen bzw. rücksetzen • nächsten Uhrzeitalarm setzen • nächsten Uhrzeitalarm aktivieren
OB 80	Setzen des Ausgangs A 4.1 Startereignisinformation des OB 80 speichern im Merkerbereich

Verwendete Operanden

Nachfolgende Tabelle zeigt die verwendeten globalen Operanden. Die bausteintemporären Variablen sind im Deklarationsteil des jeweiligen Bausteins deklariert.

Operand	Bedeutung
E 0.0	Eingang für die Freigabe von "Uhrzeitalarm setzen" und "Uhrzeitalarm aktivieren"
E 0.1	Eingang für die Stornierung eines Uhrzeitalarms
A 4.0	Ausgang, der vom Uhrzeitalarm-OB (OB 10) gesetzt/rückgesetzt wird
A 4.1	Ausgang, der bei einem Zeitfehler (OB 80) gesetzt wird
MW 16	STATUS des Uhrzeitalarms (SFC 31 "QRY_TINT")
MB 100 bis MB 107	Speicher für Startereignisinformation des OB 10 (nur Uhrzeit)
MB 110 bis MB 129	Speicher für Startereignisinformation des OB 80 (Zeitfehler)
MW 200	RET_VAL des SFC 28 "SET_TINT"
MB 202	Binäresultat- (Statusbit BIE)-Zwischenspeicher für SFCs
MW 204	RET_VAL des SFC 30 "ACT_TINT"
MW 208	RET_VAL des SFC 31 "QRY_TINT"

Verwendete SFCs und FCs

Im Programmbeispiel werden die folgenden Systemfunktionen verwendet:

- SFC 28 "SET_TINT" : Uhrzeitalarm stellen
- SFC 29 "CAN_TINT" : Uhrzeitalarm stornieren
- SFC 30 "ACT_TINT" : Uhrzeitalarm aktivieren
- SFC 31 "QRY_TINT" : Uhrzeitalarm abfragen
- FC 3 "D_TOD_DT" : DATE und TIME_OF_DAY zu DT zusammenfassen

A.5.3.2 FC 12

Deklarationsteil

Im Deklarationsteil des FC 12 werden folgende bausteintemporäre Variablen deklariert:

Variablenname	Datentyp	Deklaration	Kommentar
IN_UHRZEIT	TIME_OF_DAY	TEMP	Startzeit Vorgabe
IN_DATUM	DATE	TEMP	Startdatum Vorgabe
OUT_UHRZEIT_DATUM	DATE_AND_TIME	TEMP	Startdatum/-zeit gewandelt
OK_MERKER	BOOL	TEMP	Freigabe für Uhrzeitalarm-Stellen

Anweisungsteil AWL

Im Anweisungsteil des FC 12 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (FC 12)	Erläuterung
Network 1:	
CALL SFC 31	SFC QRY_TINT
OB_NR := 10	STATUS Uhrzeitalarme abfragen.
RET_VAL:= MW 208	
STATUS := MW 16	
Network 2:	Startzeitpunkt vorgeben in
UN A 4.0	Abhaengigkeit von A 4.0 (in Variable
SPB mont	#IN_DATUM und #IN_UHRZEIT)
L D#1995-1-27	Startdatum ist ein Freitag.
T #IN_DATUM	
L TOD#20:0:0.0	
T #IN_UHRZEIT	
SPA wndl	Startdatum ist ein Montag.
mont: L D#1995-1-23	
T #IN_DATUM	
L TOD#5:0:0.0	
T #IN_UHRZEIT	
wndl: NOP 0	

AWL (FC 12)	Erläuterung
Network 3:	
CALL FC 3	Format wandeln von DATE und
IN1 := #IN_DATUM	TIME_OF_DAY nach DATE_AND_TIME (für
IN2 := #IN_UHRZEIT	Uhrzeitalarm stellen)
RET_VAL := #OUT_UHRZEIT_DATUM	
Network 4:	
U E 0.0	Alle Voraussetzungen für Uhrzeitalarm
UN M 17.2	stellen erfüllt? (Eingang für
U M 17.4	Freigabe gesetzt und Uhrzeitalarm
= #OK_MERKER	nicht aktiv und Uhrzeitalarm-OB ist
	geladen)
Network 5:	Wenn ja, dann Uhrzeitalarm
U #OK_MERKER	stellen...
SPBNB m001	
CALL SFC 28	
OB_NR := 10	
SDT := #OUT_UHRZEIT_DATUM	...und Uhrzeitalarm aktivieren.
PERIOD := W#16#1201	
RET_VAL := MW 200	
m001 U BIE	
= M 202.3	
Network 6:	
U #OK_MERKER	
SPBNB m002	
CALL SFC 30	
OB_NR := 10	Wenn Eingang für die Stornierung von
RET_VAL := MW 204	Uhrzeitalarmen gesetzt ist, dann
	Uhrzeitalarm stornieren.
m002 U BIE	
= M 202.4	
Network 7:	
U E 0.1	
SPBNB m003	
CALL SFC 29	
OB_NR := 10	
RET_VAL := MW 210	
m003 U BIE	
= M 202.5	

A.5.3.3 OB 10

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 10 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)
- Weitere bausteintemporäre Variablen WTAG, IN_DATUM, IN_UHRZEIT und OUT_UHRZEIT_DATUM

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Gesamte Startereignisinformation des OB 10 als Struktur deklariert
E_ID	WORD	TEMP	Ereignis-ID
PR_KLASSE	BYTE	TEMP	Prioritätsklasse
OB_NR	BYTE	TEMP	OB-Nummer
RESERVED_1	BYTE	TEMP	Reserviert
RESERVED_2	BYTE	TEMP	Reserviert
PERIODE	WORD	TEMP	Periodizität des Uhrzeitalarms
RESERVED_3	DWORD	TEMP	Reserviert
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben
JAHR	BYTE	TEMP	
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEK_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	
WTAG	INT	TEMP	Wochentag
IN_DATUM	DATE	TEMP	Eingangsvariable für FC 3 (Wandlung des Zeitformates)
IN_UHRZEIT	TIME_OF_DAY	TEMP	Eingangsvariable für FC 3 (Wandlung des Zeitformates)
OUT_UHRZEIT_DATUM	DATE_AND_TIME	TEMP	Ausgangsvariable für FC 3 und Eingangsvariable für SFC 28

Anweisungsteil AWL

Im Anweisungsteil des OB 10 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 10)	Erläuterung
Network 1:	
L	Wochentag selektieren
#STARTINFO.T_STMP.MSEK_WTAG	
L W#16#F	
UW	und merken.
T #WTAG	
Network 2:	Wenn Wochentag nicht Montag ist, dann Montag, 5.00 Uhr als nächsten Startzeitpunkt vorgeben und Ausgang A 4.0 rücksetzen.
L #WTAG	
L 2	
<>I	
SPB mont	
Network 3:	Sonst, d. h. wenn Wochentag = Montag ist, dann Freitag, 20.00 Uhr als nächsten Startzeitpunkt vorgeben und Ausgang A 4.0 setzen.
L D#1995-1-27	
T #IN_DATUM	
L TOD#20:0:0.0	
T #IN_UHRZEIT	
SET	
= A 4.0	
SPA wndl	
mont:	
L D#1995-1-23	
T #IN_DATUM	
L TOD#5:0:0.0	
T #IN_UHRZEIT	
CLR	
= A 4.0	Startzeitpunkt vorgeben abgeschlossen.
wndl:	
NOP 0	Vorgegebenen Startzeitpunkt in Format DATE_AND_TIME wandeln (für SFC 28).
Network 4:	
CALL FC 3	
IN1 := #IN_DATUM	
IN2 := #IN_UHRZEIT	Uhrzeitalarm stellen.
RET_VAL := #OUT_UHRZEIT_DATUM	
Network 5:	
CALL SFC 28	
OB_NR := 10	
SDT := #OUT_UHRZEIT_DATUM	
PERIOD := W#16#1201	
RET_VAL := MW 200	
U BIE	
= M 202.1	
AWL (OB 10)	Erläuterung
Network 6:	
CALL SFC 30	Uhrzeitalarm aktivieren.
OB_NR := 10	
RET_VAL := MW 204	
U BIE	
= M 202.2	
Network 7:	
CALL SFC 20	Blocktransfer: Uhrzeitangabe aus Starterereignisinformation des OB 10 in den Merkerbereich MB 100 bis MB 107 retten.
SRCBLK := #STARTINFO.T_STMP	
RET_VAL := MW 206	
DSTBLK := P#M 100.0 BYTE 8	

A.5.3.4 OB 1 und OB 80

Da die Startereignisinformation des OB 1 (OB für zyklisches Programm) in diesem Beispiel nicht ausgewertet wird, ist nur die Startereignisinformation des OB 80 dargestellt.

Anweisungsteil OB 1

Im Anweisungsteil des OB 1 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 1)	Erläuterung
CALL FC 12	Aufruf der Funktion FC 12

Deklarationsteil OB 80

Abweichend vom voreingestellten Deklarationsteil des OB 80 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Gesamte Startereignisinformation des OB 80 als Struktur deklariert
E_ID	WORD	TEMP	Ereignis-ID
PR_KLASSE	BYTE	TEMP	Prioritätsklasse
OB_NR	BYTE	TEMP	OB-Nummer
RESERVED_1	BYTE	TEMP	Reserviert
RESERVED_2	BYTE	TEMP	Reserviert
Z1_INFO	WORD	TEMP	Zusatzinformation über das Ereignis, das den Fehler verursacht hat.
Z2_INFO	DWORD	TEMP	Zusatzinformation über Ereignis-ID, Prioritätsklasse und OB-Nr. des Fehlerereignisses
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben
Jahr	BYTE	TEMP	
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEK_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Anweisungsteil OB 80

Im Anweisungsteil des OB 80, der vom Betriebssystem bei einem Zeitfehler aufgerufen wird, geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 80)	Erläuterung
Network 1:	
UN A 4.1	Ausgang A 4.1 setzen, wenn Zeitfehler aufgetreten ist.
S A 4.1	
CALL SFC 20	Blocktransfer: Gesamte Startereignisinformation in den Merkerbereich MB 110 bis MB 129 retten.
SRCBLK := #STARTINFO	
RET_VAL := MW 210	
DSTBLK := P#M 110.0 Byte 20	

A.5.4 Beispiel zur Hantierung mit Verzögerungsalarmen

Struktur des Anwenderprogramms "Verzögerungsalarme"

OB 20

OB 1

A.5.4.1 Struktur des Anwenderprogramms Verzögerungsalarme

Aufgabenstellung

Wenn Eingang E 0.0 gesetzt wird, soll Ausgang A 4.0 10 Sekunden später gesetzt werden. Jedes Setzen von Eingang E 0.0 soll die Verzögerungszeit neu starten.

Als anwendungsspezifisches Kennzeichen soll der Zeitpunkt (Sekunden- und Millisekunden) des Verzögerungsalarm-Starts in der Startereignisinformation des Verzögerungsalarm-OB (OB 20) erscheinen.

Falls in diesen 10 Sekunden E 0.1 gesetzt wird, soll der Organisationsbaustein OB 20 nicht aufgerufen werden; d. h. der Ausgang A 4.0 soll nicht gesetzt werden.

Wenn Eingang E 0.2 gesetzt wird, soll Ausgang A 4.0 wieder zurückgesetzt werden.

Umsetzung im Anwenderprogramm

Nachfolgende Tabelle zeigt die Teilaufgaben der verwendeten Bausteine.

Baustein	Teilaufgabe
OB 1	Aktuelle Uhrzeit lesen und aufbereiten für Start des Verzögerungsalarms Abhängig vom Flankenwechsel am Eingang E 0.0 Verzögerungsalarm starten Abhängig vom Status des Verzögerungsalarms und vom Flankenwechsel am Eingang E 0.1 Verzögerungsalarm stornieren Abhängig vom Zustand des Eingangs E 0.2 Ausgang A 4.0 zurücksetzen
OB 20	Setzen des Ausganges A 4.0 Aktuelle Uhrzeit lesen und aufbereiten Startereignisinformation in den Merkerbereich retten

Verwendete Operanden

Nachfolgende Tabelle zeigt die verwendeten Globaldaten. Die bausteintemporären Variablen sind im Deklarationsteil des jeweiligen Bausteins deklariert.

Operand	Bedeutung
E 0.0	Eingang für die Freigabe von "Verzögerungsalarm starten"
E 0.1	Eingang für die Stornierung eines Verzögerungsalarms
E 0.2	Eingang für das Rücksetzen des Ausganges A 4.0
A 4.0	Ausgang, der vom Verzögerungsalarm-OB (OB 20) gesetzt wird
MB 1	Genutzt für Flankenmerker und Binärerergebnis- (Statusbit BIE)-Zwischenspeicher für SFCs
MW 4	STATUS des Verzögerungsalarms (SFC 34 "QRY_TINT")
MD 10	Sekunden und Millisekunden BCD-codiert aus der Startereignisinformation des OB 1
MW 100	RET_VAL des SFC 32 "SRT_DINT"
MW 102	RET_VAL des SFC 34 "QRY_DINT"
MW 104	RET_VAL des SFC 33 "CAN_DINT"
MW 106	RET_VAL des SFC 20 "BLKMOV"
MB 120 bis MB 139	Speicher für Startereignisinformation des OB 20
MD 140	Sekunden und Millisekunden BCD-codiert aus der Startereignisinformation des OB 20
MW 144	Sekunden und Millisekunden BCD-codiert aus der Startereignisinformation des OB 1; gewonnen aus Startereignisinformation des OB 20 (anwenderspezifisches Kennzeichen SIGN)

Verwendete SFCs

Im Anwenderprogramm "Verzögerungsalarme" werden die folgenden Systemfunktionen verwendet:

- SFC 32 "SRT_DINT" : Verzögerungsalarm starten
- SFC 33 "CAN_DINT" : Verzögerungsalarm stornieren
- SFC 34 "QRY_DINT" : Zustand eines Verzögerungsalarms abfragen

A.5.4.2 OB 20

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 20 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Start-Info zu OB 20
E_ID	WORD	TEMP	Ereignis-ID
AE_NR	BYTE	TEMP	Ablaufebene
OB_NR	BYTE	TEMP	OB-Nr
DK1	BYTE	TEMP	Datenkennung 1
DK2	BYTE	TEMP	Datenkennung 2
SIGN	WORD	TEMP	anwenderspezifisches Kennzeichen
DTIME	TIME	TEMP	Zeit, mit der Verzögerungsalarm gestartet wird
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben (Zeitstempel)
JAHR	BYTE	TEMP	
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEC_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Anweisungsteil

Im Anweisungsteil des OB 20 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 20)	Erläuterung
Network 1:	
SET	Ausgang A 4.0 unbedingt setzen
= A 4.0	
Network 2:	Ausgangswort sofort aktualisieren
L AW 4	
T PAW 4	
Network 3:	Sekunden aus Starterereignisinformation lesen
L	Millisekunden und Wochentag aus Starterereignisinformation lesen
#STARTINFO.T_STMP.SEKUNDEN	
T MW 140	
L	
#STARTINFO.T_STMP.MSEC_WTAG	Wochentag eliminieren und Millisekunden zurückschreiben (befinden sich nun BCD-codiert in MW 142).
T MW 142	
L MD 140	
SRD 4	Startzeitpunkt des Verzögerungsalarms (=Aufruf des SFC 32) aus Starterereignisinformation lesen
T MD 140	
Network 4:	
L #STARTINFO.SIGN	Starterereignisinformation in Merkerbereich kopieren (MB 120 bis MB 139)
T MW 144	
Network 5:	
CALL SFC 20	
SRCBLK := STARTINFO	
RET_VAL := MW 106	
DSTBLK := P#M 120.0 BYTE 20	

A.5.4.3 OB 1**Deklarationsteil**

Abweichend vom voreingestellten Deklarationsteil des OB 1 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Start-Info zu OB 1
E_ID	WORD	TEMP	Ereignis-ID
AE_NR	BYTE	TEMP	Ablaufebene
OB_NR	BYTE	TEMP	OB-Nr
DK 1	BYTE	TEMP	Datenkennung 1
DK 2	BYTE	TEMP	Datenkennung 2
AKT_ZYK	INT	TEMP	aktuelle Zykluszeit
MIN_ZYK	INT	TEMP	minimale Zykluszeit
MAX_ZYK	INT	TEMP	maximale Zykluszeit
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben (Zeitstempel)
JAHR	BYTE	TEMP	
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEC_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Anweisungsteil

Im Anweisungsteil des OB 1 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 1)	Erläuterung
Network 1:	
L #STARTINFO.T_STMP.SEKUNDEN	Sekunden aus
T MW 10	Startereignisinformation lesen
L #STARTINFO.T_STMP.MSEC_WTAG	Millisekunden und Wochentag aus
T MW 12	Startereignisinformation lesen
L MD 10	Wochentag eliminieren und
SRD 4	Millisekunden zurückschreiben
T MD 10	(befinden sich nun BCD-codiert in MW 12)
Network 2:	
U E 0.0	Positive Flanke an Eingang E 0.0?
FP M 1.0	
= M 1.1	
Network 3:	
U M 1.1	
SPBNB m001	Wenn ja, Verzögerungsalarm
CALL SFC 32	starten (Startzeitpunkt des
OB_NR := 20	Verzögerungsalarms dem Parameter
DTIME := T#10S	SIGN zugewiesen)
SIGN := MW 12	
RET_VAL:= MW 100	
m001: NOP 0	
Network 4:	
CALL SFC 34	Status des Verzögerungsalarms
OB_NR := 20	abfragen (SFC QRY_DINT)
RET_VAL:= MW 102	
STATUS := MW 4	
Network 5:	
U E 0.1	Positive Flanke am Eingang E
FP M 1.3	0.1?
= M 1.4	
Network 6:	
U M 1.4	
U M 5.2	... und Verzögerungsalarm ist
SPBNB m002	aktiviert (Bit 2 des
CALL SFC 33	Verzögerungsalarm-STATUS)?
OB_NR := 20	Dann Verzögerungsalarm
RET_VAL:= MW 104	stornieren
m002: NOP 0	
U E 0.2	
R A 4.0	Ausgang A 4.0 mit Eingang E 0.2
	zurücksetzen

A.5.4.4 Beispiel für das Maskieren und Demaskieren von Synchronfehlerereignissen

Im folgenden Beispiel für ein Anwenderprogramm zeigen wir Ihnen das Maskieren und Demaskieren von Synchronfehlerereignissen. Mit dem SFC 36 "MSK_FLT" werden in der Programmierfehlermaske folgende Fehler maskiert:

- Bereichslängenfehler beim Lesen
- Bereichslängenfehler beim Schreiben

Mit einem 2. Aufruf des SFC 36 "MSK_FLT" wird zusätzlich ein Zugriffsfehler maskiert:

- Peripheriezugriffsfehler beim Schreiben.

Mit dem SFC 38 "READ_ERR" werden die maskierten Synchronfehlerereignisse abgefragt. Der "Peripheriezugriffsfehler beim Schreiben" wird mit dem SFC 37 "DMSK_FLT" wieder demaskiert.

Anweisung

Im folgenden sehen Sie den OB 1, in dem das Beispiel für das Anwenderprogramm in AWL programmiert wurde.

AWL (Network 1)	Erläuterung
UN M 255.0	nicht remanenter Merker M 255.0 (nur beim ersten Durchlauf=0)
SPBNB m001	
CALL SFC 36	SFC 36 MSK_FLT (Maskieren von Synchronfehlerereignissen)
PRGFLT_SET_MASK :=DW#16#C	Bit2=Bit3=1 (BLFL und BLFS werden maskiert)
ACCFLT_SET_MASK :=DW#16#0	alle Bits=0 (es wird kein Zugriffsfehler maskiert)
RET_VAL :=MW 100	Returnwert
PRGFLT_MASKED:=MD 10	Ausgabe der aktuellen Programmierfehlermaske in MD 10
ACCFLT_MASKED:=MD 14	Ausgabe der aktuellen Zugriffsfehlermaske in MD 14
m001: U BIE	
S M 255.0	Setzen von M255.0, wenn Maskieren erfolgreich

AWL (Network 2)	Erläuterung
CALL SFC 36	SFC 36 MSK_FLT (Maskieren von Synchronfehlerereignissen)
PRGFLT_SET_MASK :=DW#16#0	alle Bits=0 (es wird kein weiterer Programmfehler maskiert)
ACCFLT_SET_MASK :=DW#16#8	Bit3=1 (schreibende Zugriffsfehler werden maskiert)
RET_VAL :=MW 102	Returnwert
PRGFLT_MASKED:=MD 20	Ausgabe der aktuellen Programmierfehlermaske in MD 20
ACCFLT_MASKED:=MD 24	Ausgabe der aktuellen Zugriffsfehlermaske in MD 24

AWL (Network 3)		Erläuterung
UN	M 27.3	Baustein-Ende, wenn schreibender Zugriffsfehler (Bit3 in ACCFLT_MASKED) nicht maskiert
BEB		

AWL (Network 4)		Erläuterung
L	B#16#0	Schreibender Zugriff (mit Wert 0) auf PAB 16
T	PAB 16	

AWL (Network 5)		Erläuterung
CALL	SFC 38	SFC 38 READ_ERR (Abfragen von Synchronfehlerereignissen)
PRGFLT_QUERY	:=DW#16#0	alle Bits=0 (es wird kein Programmierfehler abgefragt)
ACCFLT_QUERY	:=DW#16#8	Bit3=1 (schreibender Zugriffsfehler wird abgefragt)
		Returnwert
RET_VAL	:=MW 104	Ausgabe der aktuellen Programmierfehlermaske in MD 30
PRGFLT_CLR	:=MD 30	Ausgabe der aktuellen Zugriffsfehlermaske in MD 34
ACCFLT_CLR	:=MD 34	kein Fehler aufgetreten und schreibenden Zugriffsfehler festgestellt
U	BIE	
U	M 37.3	VKE invertieren
NOT		M 0.0=1, wenn PAB 16 vorhanden
=	M 0.0	

AWL (Network 6)		Erläuterung
L	B#16#0	Schreibender Zugriff (mit Wert 0) auf PAB 17
T	PAB 17	

AWL (Network 7)		Erläuterung
CALL	SFC 38	SFC 38 READ_ERR (Abfragen von Synchronfehlerereignissen)
PRGFLT_QUERY	:=DW#16#0	alle Bits=0 (es wird kein Programmierfehler abgefragt)
ACCFLT_QUERY	:=DW#16#8	Bit3=1 (schreibender Zugriffsfehler wird abgefragt)
		Returnwert
RET_VAL	:=MW 104	Ausgabe der aktuellen Programmierfehlermaske in MD 30
PRGFLT_CLR	:=MD 30	Ausgabe der aktuellen Zugriffsfehlermaske in MD 34
ACCFLT_CLR	:=MD 34	kein Fehler aufgetreten und schreibenden Zugriffsfehler festgestellt
U	BIE	
U	M 37.3	VKE invertieren
NOT		M 0.1=1, wenn PAB 17 vorhanden
=	M 0.1	

AWL (Network 8)		Erläuterung
L	B#16#0	Schreibender Zugriff (mit Wert 0) auf PAB 18
T	PAB 18	

AWL (Network 9)		Erläuterung
CALL	SFC 38	SFC 38 READ_ERR (Abfragen von Synchronfehlerereignissen)
PRGFLT_QUERY	:=DW#16#0	alle Bits=0 (es wird kein Programmierfehler abgefragt)
ACCFLT_QUERY	:=DW#16#8	Bit3=1 (schreibender Zugriffsfehler wird abgefragt)
RET_VAL	:=MW 104	Returnwert
PRGFLT_CLR	:=MD 30	Ausgabe der aktuellen Programmierfehlermaske in MD 30
ACCFLT_CLR	:=MD 34	Ausgabe der aktuellen Zugriffsfehlermaske in MD 34
U	BIE	kein Fehler aufgetreten und schreibenden Zugriffsfehler festgestellt
U	M 37.3	
NOT		VKE invertieren
=	M 0.2	M 0.2=1, wenn PAB 18 vorhanden

AWL (Network 10)		Erläuterung
L	B#16#0	
T	PAB 19	Schreibender Zugriff (mit Wert 0) auf PAB 19

AWL (Network 11)		Erläuterung
CALL	SFC 38	SFC 38 READ_ERR (Abfragen von Synchronfehlerereignissen)
PRGFLT_QUERY	:=DW#16#0	alle Bits=0 (es wird kein Programmierfehler abgefragt)
ACCFLT_QUERY	:=DW#16#8	Bit3=1 (schreibender Zugriffsfehler wird abgefragt)
RET_VAL	:=MW 104	Returnwert
PRGFLT_CLR	:=MD 30	Ausgabe der aktuellen Programmierfehlermaske in MD 30
ACCFLT_CLR	:=MD 34	Ausgabe der aktuellen Zugriffsfehlermaske in MD 34
U	BIE	kein Fehler aufgetreten und schreibenden Zugriffsfehler festgestellt
U	M 37.3	
NOT		VKE invertieren
=	M 0.3	M 0.3=1, wenn PAB 19 vorhanden

AWL (Network 12)		Erläuterung
CALL	SFC 37	SFC 37 DMSK_FLT (Demaskieren von Synchronfehlerereignissen)
PRGFLT_RESET_MASK	:=DW#16#0	alle Bits=0 (es wird kein Programmierfehler demaskiert)
ACCFLT_RESET_MASK	:=DW#16#8	Bit3=1 (schreibender Zugriffsfehler wird demaskiert)
RET_VAL	:=MW 102	Returnwert
PRGFLT_MASKED	:=MD 20	Ausgabe der aktuellen Programmierfehlermaske in MD 20
ACCFLT_MASKED	:=MD 24	Ausgabe der aktuellen Zugriffsfehlermaske in MD 24

AWL (Network 13)			Erläuterung
U	M 27.3		Baustein-Ende, wenn schreibender Zugriffsfehler (Bit3 in ACCFLT_MASKED) nicht demaskiert
BEB			

AWL (Network 14)			Erläuterung
U	M 0.0		EB 0 nach PAB 16 transferieren, wenn vorhanden
SPBNB	m002		
L	EB 0		
T	PAB 16		
m002: NOP	0		

AWL (Network 15)			Erläuterung
U	M 0.1		EB 1 nach PAB 17 transferieren, wenn vorhanden
SPBNB	m003		
L	EB 1		
T	PAB 17		
m003: NOP	0		

AWL (Network 16)			Erläuterung
U	M 0.2		EB 2 nach PAB 18 transferieren, wenn vorhanden
SPBNB	m004		
L	EB 2		
T	PAB 18		
m004: NOP	0		

AWL (Network 17)			Erläuterung
U	M 0.3		EB 3 nach PAB 19 trans- ferieren, wenn vorhanden
SPBNB	m005		
L	EB 3		
T	PAB 19		
m005: NOP	0		

A.5.4.5 Beispiel zum Sperren und Freigeben von Alarm- und Asynchronfehlerereignissen (SFC 39 und 40)

In diesem Beispiel für ein Anwenderprogramm wird ein Programmteil angenommen, der nicht durch Alarme unterbrochen werden darf. Für diesen Programmteil werden mit der SFC 39 "DIS_IRT" OB 35-Aufrufe (Uhrzeitalarm) gesperrt und mit dem SFC 40 "EN_IRT" werden die OB 35-Aufrufe wieder freigegeben.

Im OB 1 werden der SFC 39 und der SFC 40 aufgerufen:

AWL (OB 1)	Erläuterung
U M 0.0	Programmteil, der problemlos unterbrochen werden kann:
S M 90.1	
U M 0.1	Programmteil, der nicht durch Interrupts unterbrochen werden darf:
S M 90.0	
:	Alarme sperren und verwerfen
:	Mode 2: einzelne Alarm-OBS sperren
CALL SFC 39	OB35 sperren
MODE :=B#16#2	
OB_NR :=35	
RET_VAL :=MW 100	
:	
:	
L PEW 100	
T MW 200	
L MW 90	
T MW 92	
:	
:	Alarme freigeben
CALL SFC 40	Mode 2: einzelne Alarm-OBS freigeben
MODE :=B#16#2	OB35 freigeben
OB_NR :=35	
RET_VAL :=MW 102	Programmteil, der problemlos unterbrochen werden kann:
U M 10.0	
S M 190.1	
U M 10.1	
S M 190.0	
:	
:	

A.5.4.6 Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronfehlerereignissen (SFC 41 und 42)

In diesem Beispiel für ein Anwenderprogramm wird ein Programmteil angenommen, der nicht durch Alarme unterbrochen werden darf. Für diesen Programmteil werden mit der SFC 41 "DIS_AIRT" Alarme verzögert und später mit dem SFC 42 "EN_AIRT" wieder freigegeben.

In dem OB 1 werden der SFC 41 und der SFC 42 aufgerufen:

AWL (OB 1)	Erläuterung
U M 0.0	Programmteil, der problemlos unterbrochen werden kann:
S M 90.1	
U M 0.1	
S M 90.0	
:	Programmteil, der nicht durch Interrupts unterbrochen werden darf: Alarme sperren und verzögern
:	
CALL SFC 41	Alarm freigeben
RET_VAL :=MW 100	
L PEW 100	Im Returnwert steht die Anzahl der eingelegten Alarmsperren
T MW 200	
L MW 90	im Returnwert steht die Anzahl der eingelegten Alarmsperren die Anzahl muss nach der Alarmfreigabe den gleichen Wert haben wie vor der Alarmsperre (hier "0")
T MW 92	
:	Alarm freigeben
:	
CALL SFC 42	Im Returnwert steht die Anzahl der eingelegten Alarmsperren
RET_VAL :=MW 102	
L MW 100	im Returnwert steht die Anzahl der eingelegten Alarmsperren die Anzahl muss nach der Alarmfreigabe den gleichen Wert haben wie vor der Alarmsperre (hier "0")
DEC 1	
L MW 102	Programmteil, der problemlos unterbrochen werden kann:
<>I	
SPB fehl	Anzahl der eingelegten Alarmsperren wird angezeigt
U M 10.0	
S M 190.1	
U M 10.1	
S M 190.0	
:	
:	Anzahl der eingelegten Alarmsperren wird angezeigt
BEA	
fehl: L MW 102	
T AW 12	

A.6 Zugriff auf Prozess- und Peripheriedatenbereiche

A.6.1 Zugriff auf den Prozessdatenbereich

Die CPU kann entweder indirekt über das Prozessabbild oder über den Rückwand-/P-Bus direkt auf Ein- und Ausgänge zentraler und dezentraler Digitalein-/ausgabebaugruppen zugreifen.

Auf Ein- und Ausgänge zentraler und dezentraler Analogein-/ausgabebaugruppen greift die CPU direkt über den Rückwand-/P-Bus zu. Sie haben aber auch die Möglichkeit, die Adressen von Analogbaugruppen in den Prozessabbild-Bereich zu legen.

Baugruppenadressierung

Die Zuordnung zwischen Adressen, die im Anwenderprogramm verwendet werden, und den Baugruppen erfolgt durch die Konfigurierung der Baugruppen mit STEP 7

- bei zentraler Peripherie: Anordnung des Baugruppenträgers und Zuordnung der Baugruppen auf Steckplätzen in der Konfigurationstabelle
- bei dezentraler Peripherie (PROFIBUS DP oder PROFINET IO): Anordnung der DP-Slaves oder IO-Devices mit Vergabe der PROFIBUS-Adresse bzw. Gerätenamen und Zuordnung der Baugruppen auf Steckplätze.

Das Konfigurieren der Baugruppen löst die Adresseinstellung der einzelnen Baugruppen über Schalter ab. Die CPU erhält vom PG als Ergebnis der Konfigurierung Daten, anhand derer sie die zugeordneten Baugruppen erkennt.

Peripherieadressierung

Für Ein- und Ausgänge existiert jeweils ein eigener Adressbereich. Deshalb muss die Adresse eines Peripheriebereichs außer der Byte- oder Wortangabe zusätzlich die Kennung E für die Eingänge und A für die Ausgänge beinhalten.

Nachfolgende Tabelle zeigt die zur Verfügung stehenden Peripherie-Adressbereiche.

Operandenbereich	Zugriff über Einheiten der folgenden Größe:	S7-Notation
Peripheriebereich: Eingänge	Peripherieeingangsbyte Peripherieeingangswort Peripherieeingangs-Doppelwort	PEB PEW PED
Peripheriebereich: Ausgänge	Peripherieausgangsbyte Peripherieausgangswort Peripherieausgangs-Doppelwort	PAB PAW PAD

Welche Adressbereiche bei den einzelnen Baugruppen möglich sind, entnehmen Sie den folgenden Handbüchern.

- CPU 31xC und CPU 31x, Technische Daten
- Automatisierungssystem S7-400, CPU Daten

Baugruppen-Anfangsadresse

Die Baugruppen-Anfangsadresse ist die niedrigste Byteadresse einer Baugruppe. Sie stellt die Anfangsadresse des Nutzdatenbereichs der Baugruppe dar und wird in vielen Fällen stellvertretend für die ganze Baugruppe verwendet.

Sie wird z. B. bei Prozessalarmen, Diagnosealarmen, Ziehen/Stecken-Alarmen und Stromversorgungsfehlern in die Startinformation des zugehörigen Organisationsbausteins eingetragen und identifiziert damit die alarmgebende Baugruppe.

A.6.2 Zugriff auf den Peripheriedatenbereich

Der Peripheriedatenbereich lässt sich unterteilen in:

- Nutzdaten und
- Diagnose- und Parameterdaten.

Beide Bereiche haben einen Eingangsbereich (nur lesender Zugriff möglich) und einen Ausgangsbereich (nur schreibender Zugriff möglich).

Nutzdaten

Adressiert werden die Nutzdaten über die Byteadresse (bei digitalen Signalbaugruppen) oder die Wortadresse (bei analogen Signalbaugruppen) des Ein- bzw. Ausgangsbereichs. Auf Nutzdaten zugreifen können Sie über Lade- und Transferbefehle, Kommunikationsfunktionen (BuB-Zugriffe) bzw. über den Prozessabbildtransfer. Nutzdaten können sein:

- digitale und analoge Ein-, Ausgangssignale von Signalbaugruppen,
- Steuer- und Statusinformationen von Funktionsbaugruppen und
- Informationen für Punkt-zu-Punkt und Buskopplungen von Kommunikationsbaugruppen (nur S7-300).

Bei der Übertragung von Nutzdaten kann eine Datenkonsistenz von maximal 4 Byte erreicht werden (Ausnahme DP-Normslaves, siehe "Einstellen des Betriebsverhaltens"). Verwenden Sie die Anweisung "Transferiere Doppelwort", werden 4 Byte zusammenhängend und unverändert (konsistent) übertragen. Verwenden Sie vier einzelne Anweisungen "Transferiere Eingangsbyte", könnte an einer Befehlsgrenze ein Prozessalarm-OB gestartet werden, der Daten an die gleiche Adresse überträgt und damit den Inhalt der ursprünglichen vier Byte verändert.

Diagnose- und Parameterdaten

Die Diagnose- und Parameterdaten einer Baugruppe können nicht einzeln adressiert werden, sondern nur zusammengefasst zu ganzen Datensätzen. Diagnose- und Parameterdaten werden grundsätzlich konsistent übertragen.

Adressiert werden die Diagnose- und Parameterdaten über die Anfangsadresse der betreffenden Baugruppe und die Datensatznummer. Datensätze werden unterteilt in Eingangs- und Ausgangsdatsätze. Eingangsdatsätze können nur gelesen, Ausgangsdatsätze können nur beschrieben werden. Auf Datensätze zugreifen können Sie mit Hilfe von Systemfunktionen oder Kommunikationsfunktionen (Bedienen und Beobachten, BuB). Nachfolgende Tabelle zeigt die Zuordnung der Datensätze zu Diagnose- und Parameterdaten.

Daten	Beschreibung
Diagnosedaten	Bei diagnosefähigen Baugruppen erhalten Sie beim Lesen der Datensätze 0 und 1 die Diagnosedaten dieser Baugruppe.
Parameterdaten	Bei parametrierbaren Baugruppen übertragen Sie beim Schreiben der Datensätze 0 und 1 die Parameter dieser Baugruppe.

Zugreifen auf Datensätze

Die Informationen in den Datensätzen einer Baugruppe können Sie nutzen, um parametrierbare Baugruppen nachzuparametrieren und Diagnoseinformationen diagnosefähiger Baugruppen auszulesen.

Nachfolgende Tabelle zeigt, mit welchen Systemfunktionen Sie auf Datensätze zugreifen können.

SFC	Anwendung
Baugruppen parametrieren	
SFC 55 WR_PARM	Übertragen der änderbaren Parameter (Datensatz 1) zur adressierten Signalbaugruppe.
SFC 56 WR_DPARM	Übertragen der Parameter aus den SDBs 100 bis 129 zur adressierten Signalbaugruppe.
SFC 57 PARM_MOD	Übertragen aller Parameter aus den SDBs 100 bis 129 zur adressierten Signalbaugruppe.
SFC 58 WR_REC	Übertragen eines beliebigen Datensatzes zur adressierten Signalbaugruppe
Diagnoseinformation auslesen	
SFC 59 RD_REC	Auslesen der Diagnosedaten

Hinweis

Wenn ein DPV1-Slave über GSD-Datei projektiert ist (GSD ab Rev. 3) und die DP-Schnittstelle des DP-Masters ist auf "**S7-kompatibel**" eingestellt, dürfen im Anwenderprogramm keine Datensätze mit SFC 58/59 bzw. SFB 53/52 von den E/A-Baugruppen gelesen bzw. beschrieben werden. Der DP-Master adressiert in diesem Fall den falschen Steckplatz (projektiertes Steckplatz+3).

Abhilfe: Schnittstelle des DP-Masters auf "DPV1" umstellen.

Adressieren von S5-Baugruppen

Sie haben die Möglichkeit:

- Erweiterungsgeräte der SIMATIC S5 mit der Anschaltungsbaugruppe IM 463-2 an eine S7-400 zu koppeln und
- einige S5-Baugruppen in Adaptionenkapseln in den zentralen Baugruppenträger der S7-400 zu stecken.

Wie Sie die S5-Baugruppen bei SIMATIC S7 adressieren, entnehmen Sie bitte dem Installationshandbuch "Automatisierungssysteme S7-400, M7-400, Aufbauen" bzw. der mitgelieferten Beschreibung der Adaptionenkapsel.

A.7 Einstellen des Betriebsverhaltens

Hier wird erläutert, wie Sie die Eigenschaften von S7-300- und S7-400-Automatisierungssystemen, die nicht fest vorgegeben sind, durch Einstellen von Systemparametern oder Einsatz von Systemfunktionen SFC beeinflussen können.

Detaillierte Informationen zu den Baugruppenparametern finden Sie in der STEP 7-Online-Hilfe sowie in folgenden Handbüchern:

- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten"
- Referenzhandbuch "Automatisierungssysteme S7-300, M7-300, Baugruppendaten"
- Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten"

Alles über SFCs finden Sie im Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen".

Adressierung von DP-Normslaves

Sollen DP-Normslaves Daten größer 4 Byte übertragen oder empfangen, müssen Sie für die Übertragung dieser Daten spezielle SFCs verwenden.

Bei CPUs, die den Austausch konsistenter Daten (> 4 Byte) über den Peripheriebereich unterstützen, sind die SFCs 14/15 notwendig (siehe Konsistente Daten dezentral lesen und schreiben).

SFC	Anwendung
Baugruppen parametrieren	
SFC 15 DPWR_DAT	Übertragen beliebiger Daten zur adressierten Signalbaugruppe
Diagnoseinformation auslesen	
SFC 13 DPNRM_DG	Auslesen der Diagnosedaten (asynchroner Lesevorgang)
SFC 14 DPRD_DAT	Auslesen konsistenter Daten (Länge 3 oder größer 4 Byte)

Beim Eintreffen eines DP-Diagnosetelegramms wird ein Diagnosealarm mit 4 Byte Diagnosedaten an die CPU gemeldet. Diese 4 Byte können mit der SFC 13 DPNRM_DG ausgelesen werden.

A.7.1 Ändern des Verhaltens und der Eigenschaften von Baugruppen

Defaulteinstellungen

- Alle parametrierbaren Baugruppen des S7-Automatisierungssystems sind bei Lieferung auf Defaultwerte eingestellt, die für Standardanwendungen geeignet sind. Mit diesen Defaultwerten können Sie die Baugruppen direkt, ohne weitere Einstellungen, einsetzen. Die Defaultwerte entnehmen Sie den Baugruppen-Beschreibungen folgender Handbücher:
- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten"
- Referenzhandbuch "Automatisierungssysteme S7-300, M7-300, Baugruppendaten"
- Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten"

Welche Baugruppen können parametriert werden?

Sie können jedoch auch das Verhalten und die Eigenschaften der Baugruppen parametrieren und damit auf Ihre Erfordernisse und die Gegebenheiten Ihrer Anlage einstellen. Parametrierbare Baugruppen sind CPUs, FMs, CPs sowie einige Analogein-/ausgabebaugruppen und Digitaleingabebaugruppen.

Es gibt parametrierbare Baugruppen mit und ohne Pufferung.

Baugruppen ohne Pufferung müssen nach jedem Spannungsausfall erneut mit den entsprechenden Daten versorgt werden. Die Parameter dieser Baugruppen werden im remanenten Speicherbereich der CPU gespeichert (indirekte Parametrierung durch die CPU).

Einstellen und Laden der Parameter

Baugruppenparameter stellen Sie mit STEP 7 ein. Beim Speichern der Parameter erzeugt STEP 7 das Objekt "Systemdatenbausteine", das mit dem Anwenderprogramm in die CPU geladen und von dort beim Anlauf in die zugehörigen Baugruppen übertragen wird.

Was kann parametrierbar werden?

Die Baugruppenparameter werden eingeteilt in Parameterblöcke. Welche Parameterblöcke auf welcher CPU verfügbar sind, entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbau, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten".

Beispiele für Parameterblöcke:

- Anlaufverhalten
- Zyklus
- MPI
- Diagnose
- Remanenz
- Taktmerker
- Alarmbehandlung
- On-Board-Peripherie (nur für S7-300)
- Schutzstufe
- Lokaldaten
- Echtzeituhr
- Asynchronfehler

Parametrieren mit SFCs

Außer der Parametrierung mit STEP 7 gibt es noch die Möglichkeit, mit Hilfe von Systemfunktionen Baugruppenparameter vom S7-Programm aus zu ändern. Nachfolgende Tabelle zeigt, welche SFCs welche Baugruppenparameter ändern können.

SFC	Anwendung
SFC 55 WR_PARM	Übertragen der änderbaren Parameter (Datensatz 1) zur adressierten Signalbaugruppe.
SFC 56 WR_DPARM	Übertragen der Parameter aus den zugehörigen SDBs zur adressierten Signalbaugruppe.
SFC 57 PARM_MOD	Übertragen aller Parameter aus den zugehörigen SDBs zur adressierten Signalbaugruppe.
SFC 58 WR_REC	Übertragen eines beliebigen Datensatzes zur adressierten Signalbaugruppe.

Die Systemfunktionen sind ausführlich beschrieben im Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen".

Welche Baugruppenparameter dynamisch änderbar sind, entnehmen Sie bitte den folgenden Handbüchern:

- Handbuch "Automatisierungssystem S7-300, Aufbau, CPU-Daten"
- Referenzhandbuch "Automatisierungssysteme S7-300, M7-300, Baugruppendaten"
- Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten"

A.7.2 Offline-Aktualisierung der Firmware (des Betriebssystems) von Baugruppen und Modulen

Im folgenden ist beschrieben, wie Sie eine neue Firmware-Version (= neue Betriebssystem-Version) über Memory Card auf eine Baugruppe wie z. B. eine CPU übertragen.

Für den Update sind zwei Schritte erforderlich:

1. Herstellen einer "Update-Memory Card" (Übertragen der Update Dateien auf eine Memory Card) mit dem PG oder PC mit externem Prommer).
2. Durchführen des Betriebssystem-Updates mit Hilfe der "Update-Memory Card" an der CPU.

Voraussetzungen

- Memory Card mit ausreichender Speicherkapazität. Informationen erhalten Sie auf den Download-Seiten des Customer-Supports. Hier finden Sie auch die Update-Dateien.
- PG bzw. PC mit Vorrichtung zur Programmierung von Memory Cards.

Übertragen der Update-Dateien auf eine Memory Card

1. Erzeugen Sie sich mit dem Windows Explorer ein neues Verzeichnis.
2. Laden Sie die gewünschte Update-Datei in dieses Verzeichnis und entpacken Sie sie. In diesem Verzeichnis liegen dann die UPD-Datei.
3. Stecken Sie die S7-Memory Card in das PG bzw. in den Prommer.
4. Löschen Sie die Memory Card (Menübefehl **Datei > S7-Memory Card > Löschen** im SIMATIC Manager).
5. Wählen Sie im SIMATIC Manager den Menübefehl **Zielsystem > Betriebssystem aktualisieren**.
6. Wählen Sie im angezeigten Dialogfeld das Verzeichnis mit der UPD-Datei aus.
7. Doppelklicken Sie auf die UPD-Datei.
Hierdurch wird der Programmiervorgang gestartet. Wenn der Programmiervorgang beendet ist, erscheint die Meldung "Das Firmware-Update für die Baugruppe ... wurde erfolgreich auf die S7-Memory Card übertragen".

Durchführung des Betriebssystem-Updates

1. Stecken Sie die Memory Card mit dem Update in das Zielsystem (d. h. in die CPU).
2. Schalten Sie die Stromversorgung (PS) für die CPU aus.
3. Stecken Sie die vorbereitete Memory Card mit dem Update in die CPU.
4. Schalten Sie die Stromversorgung für die CPU wieder ein.
Das Betriebssystem wird von der S7-Memory Card in das interne FLASH-EEPROM übertragen.
Während dieser Zeit leuchten alle Anzeige-LEDs an der CPU.
5. Das Update ist nach ca. 2 Min durchgeführt und daran erkennbar, dass die STOP-LED an der CPU langsam blinkt (systemseitige Urlöschanforderung).
6. Schalten Sie die Stromversorgung aus und stecken Sie evtl. die für den Betrieb vorgesehene S7-Memory Card.
7. Schalten Sie die Stromversorgung wieder ein. Die CPU führt ein automatisches Löschen durch und ist danach betriebsbereit.

A.7.3 Nutzen der Uhrzeitfunktionen

Alle S7-300-/S7-400-CPU's sind mit einer Uhr (Echtzeituhr oder Software-Uhr) ausgestattet. Die Uhr kann im Automatisierungssystem sowohl als Uhrzeitmaster als auch als Slave mit externer Synchronisation fungieren. Sie ermöglicht die Verwendung von Uhrzeitalarmen und Betriebsstundenzählern.

Uhrzeitformat

Die Uhr zeigt immer Uhrzeit (Mindestauflösung 1 s) und Datum mit Wochentag an. Bei einigen CPU's ist auch die Anzeige von Millisekunden möglich (siehe Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten").

Stellen und Lesen der Uhrzeit

Sie stellen die Uhrzeit und das Datum der CPU-Uhr mit dem Aufruf der SFC 0 SET_CLK aus dem Anwenderprogramm oder über Menübefehl vom PG aus und starten damit die Uhr. Mit der SFC 1 READ_CLK oder über Menübefehl vom PG aus lesen Sie das aktuelle Datum und die Uhrzeit der CPU aus.

Hinweis

Um Anzeigedifferenzen in HMI-Systemen in der Uhrzeit zu vermeiden, sollten Sie auf der CPU **Winterzeit** einstellen!

Parametrieren der Uhr

Ist in einem Netz mehr als eine Baugruppe mit Uhr vorhanden, müssen Sie durch Parametrieren mit STEP 7 einstellen, welche CPU bei der Synchronisation der Uhrzeit als Master und welche als Slave fungieren soll. Durch Parametrierung stellen Sie außerdem ein, ob die Synchronisierung über den K-Bus oder über die MPI-Schnittstelle erfolgen soll und in welchen Intervallen die automatische Synchronisierung erfolgen soll.

Synchronisieren der Uhrzeit

Um sicherzustellen, dass die Uhrzeit aller Baugruppen im Netz übereinstimmt, werden die Slave-Uhren vom Systemprogramm in regelmäßigen (parametrierbaren) Abständen synchronisiert. Mit der Systemfunktion SFC 48 SNC_RTCB können Sie Datum und Uhrzeit von der Master-Uhr an die Slave-Uhren übertragen.

Einsetzen eines Betriebsstundenzählers

Ein Betriebsstundenzähler zählt die Einschaltzeiten eines angeschlossenen Betriebsmittels oder die Betriebsdauer der CPU als Summe der Betriebsstunden.

Im Betriebszustand STOP wird der Betriebsstundenzähler angehalten. Sein Wert bleibt auch nach Umräumen erhalten. Bei Neustart (Warmstart) muss der Betriebsstundenzähler vom Anwenderprogramm wieder gestartet werden, bei Wiederanlauf läuft er automatisch weiter, wenn er vorher gestartet war.

Mit der SFC 2 SET_RTM können Sie den Betriebsstundenzähler auf einen Anfangswert setzen. Mit der SFC 3 CTRL_RTM können Sie den Betriebsstundenzähler starten oder stoppen. Mit der SFC 4 READ_RTM können Sie die aktuelle Anzahl der Betriebsstunden und den Zustand des Zählers ("gestoppt" oder "zählt") auslesen.

Eine CPU kann bis zu 8 Betriebsstundenzähler haben. Die Numerierung beginnt bei 0.

A.7.4 Verwenden von Taktmerkern und Zeiten

Taktmerker

Ein Taktmerker ist ein Merker, der seinen Binärzustand periodisch im Puls-Pausen-Verhältnis 1:1 ändert. Welches Merkerbyte der CPU zum Taktmerkerbyte wird, bestimmen Sie bei der Parametrierung des Taktmerkers mit STEP 7.

Nutzen

Taktmerker können Sie im Anwenderprogramm verwenden, um z. B. Leuchtmelder mit Blinklicht anzusteuern oder periodisch wiederkehrende Vorgänge (etwa das Erfassen eines Istwertes) anzustoßen.

Mögliche Frequenzen

Jedem Bit des Taktmerkerbytes ist eine Frequenz zugeordnet. Nachfolgende Tabelle zeigt die Zuordnung:

Bit des Taktmerkerbytes	7	6	5	4	3	2	1	0
Periodendauer (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frequenz (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

Hinweis

Taktmerker laufen asynchron zum CPU-Zyklus, d. h. in langen Zyklen kann sich der Zustand des Taktmerkers mehrfach ändern.

Zeiten

Zeiten sind ein Speicherbereich des Systemspeichers. Die Funktion einer Zeit wird durch das Anwenderprogramm festgelegt (z. B. Einschaltverzögerung). Die Anzahl der verfügbaren Zeiten ist CPU-abhängig.

Hinweis

- Wenn Sie in Ihrem Anwenderprogramm mehr Zeiten einsetzen, als die CPU zulässt, wird ein synchroner Fehler gemeldet und OB 121 gestartet.
 - Bei S7-300 (Ausnahme CPU 318) können Zeiten nur im OB 1 und im OB 100 gleichzeitig gestartet und aktualisiert werden, in allen anderen OBs können Zeiten nur gestartet werden.
-

Index

*

*.awl -Datei 6-27

*.k7e -Datei 6-27

*.k7p -Datei 6-27

*.sdf -Datei 6-27

A

Abändern von Attributen für Bedienen und Beobachten mit CFC 17-5

Abfragen des Uhrzeitalarms 4-29

Ablaufsteuerungen erstellen 9-9
mit S7-GRAPH 9-9

Absolute und symbolische Adressierung 8-1

Abwärtskompatibilität 7-3

Abzweige 10-23

ACT_TINT 4-28, 4-29

Adressbelegung prüfen 2-11

Adressierung A-122
absolut 8-1, 8-2
bereichsintern A-56
bereichsübergreifend A-56
speicherindirekt A-56
symbolisch 8-1
von DP-Normslaves A-122
von S5-Baugruppen A-121

Aktivieren

Anzeige von Symbolen im Baustein 8-13

Aktualisieren 18-10, 18-13, A-125

Bausteinaufrufe 10-26

Firmware (Betriebssystem) von Baugruppen und Modulen - offline A-125

Firmware (Betriebssystem) von Baugruppen und Modulen - online 18-10

Prozessabbild 4-14, A-21, A-22, A-23, A-24

Aktualparameter 4-16, 4-17

Aktuelle Konfigurationen mit Vorgänger-Versionen von STEP 7 bearbeiten 7-3

Alarmarten 4-3

Alarmereignisse

sperrern und freigeben A-116

verzögert bearbeiten A-117

Alarmgesteuerte Programm-bearbeitung 4-3

Alarm-OBs

abwählen 4-5, 4-6

Einsatz 4-27

parametrieren 4-3, 4-28

Allgemeine Hinweise

zur Eingabe von Symbolen 8-12

Ä

Ältere Projekte verwenden A-75, A-76

Ändern

Betriebszustand 18-8

Datenwerte in der Datensicht von Datenbausteinen 11-10

Uhrzeit für Uhrzeitalarm 4-28

Ändern der Fensteranordnung 5-34

Ändern des Verhaltens und der Eigenschaften von Baugruppen A-123

Ändern von BuB-Attributen mit CFC 17-5

Ändern von Schnittstellen 10-27

Änderungsprotokoll 6-3, 6-4

A

Anfangsadresse A-119

ANLAUF A-8, A-9, A-10, A-12

abbrechen A-5

Betriebszustand der CPU A-1

CPU-Tätigkeiten A-5

Anlauf-OB A-5, A-9, A-10, A-12

Anlauf-OBs 4-34

Soll-Ist-Baugruppenüberwachung 4-35

Startereignisse 4-34

Anlaufprogramm 4-34

Anlegen

AWL-Quellen 13-14

Referenzdaten 14-11

Anlegen eines Projekts 6-9, 6-10

Anlegen von fremdsprachigen Meldetexten in 'Systemfehler melden' 16-51

Anordnen

Boxen 10-23

Anweisungen

eingeben

Vorgehensweise 10-12

- Anweisungen aus der Programmelemente-Übersicht 10-5
- Anweisungsliste 9-7
- Anweisungsteil 10-3, 10-7
 - Aufbau 10-11
 - bearbeiten 10-11
 - in KOP 10-6
 - Suchfunktion für Fehler 10-17
- Anwenderdefinierte Datentypen
 - Beschreibung A-51
 - erstellen A-51, A-52
 - Struktur eingeben 11-8
- Anwenderdefinierte Datentypen (UDT) 9-14
- Anwenderdefinierte Diagnosemeldungen
 - anlegen und bearbeiten 16-19
 - anzeigen 16-39
- Anwenderprogramm
 - Aufgaben 4-1
 - Elemente 4-2
 - im CPU-Speicher A-16
 - laden A-16, A-17, A-18
- Anwenderprogramme
 - laden in Zielsystem 19-3
- Anwenderspeicher
 - komprimieren 19-20
- Anwender-Textbibliotheken 16-34, 16-35
 - Bearbeiten 16-35
 - Erstellen 16-34
- Anwender-Textbibliotheken erstellen 16-34
- Anwendertexte
 - siehe Bedienerrelevante Texte 16-32
- Anwendertexte übersetzen und bearbeiten 16-32
- Anwendungsfälle für Speichern / Archivieren 24-5
- ANY A-54, A-61, A-62, A-63
 - Parameter
 - Beschreibung und Verwendung A-63
- Anzeige
 - als Baumstruktur 14-4
 - als paarweise Aufrufbeziehung 14-4
 - Baustein-Längen 9-18
 - Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs) 11-6
 - gelöschter Baustein 14-4
 - maximaler Lokaldatenbedarf in der Baumstruktur 14-4
 - sprachabhängiger Informationen zu
 - KOP
 - FUP
 - AWL 14-9
 - Symbole im Baustein aktivieren 8-13
- Anzeigeelemente beschreiben 3-11
- Anzeigemöglichkeiten
 - für CPU-Meldungen und anwenderdefinierte Diagnosemeldungen 16-39
- Anzeigen
 - Baugruppenzustand 23-1
 - Baustein-Längen 9-18
 - Betriebszustand 18-8
 - fehlende Symbole 14-10
 - für den Programmstatus festlegen 21-9
 - im Programmstatus 21-3
 - Listen in zusätzlichen Arbeitsfenstern 14-10
 - nicht verwendete Operanden 14-10
 - Operanden ohne Symbol 14-10
 - Programmstruktur 14-10
 - Referenzdaten 14-10, 14-11
 - Struktur von Datenbausteinen mit zugeordnetem UDT 11-9
- Anzeigen des Änderungsprotokolls 6-4
- Anzeigen von gespeicherten CPU-Meldungen 16-42
- Anzeigesprache 16-32
- Arbeiten
 - mit Bibliotheken 9-23
- Arbeiten mit Netzwerkvorlagen 10-16
- Arbeiten mit V2-Projekten und -Bibliotheken 7-1
- Arbeitsbereich eines Fensters 5-24
- Arbeitsspeicher A-16
- Arbeitsspeicher 19-3, 19-4, A-15
- Archiv
 - CPU-Meldungen 16-39, 16-41
- Archivieren
 - Anwendungsfälle 24-5
 - Projekte und Bibliotheken 24-4
 - Voraussetzungen 24-6
 - Vorgehensweise 24-6
- ARRAY A-43
- Assistent zum unterstützten Anlegen eines Projekts 6-9
- asynchrone Ereignisse 4-11
- Asynchrone Fehler
 - OB81 23-26
 - Verwenden von OBs
 - um auf Fehler zu reagieren 4-38
- Asynchronfehlerereignisse
 - sperrern und freigeben A-116
 - verzögert bearbeiten A-117
- Attribute für Bausteine und Parameter 9-22
- Aufbau
 - Anweisungsteil 10-11
 - Fenster 5-24
 - Online-Verbindung über das Fenster "Erreichbare Teilnehmer" 18-2

- Online-Verbindung über das Online-Fenster des Projekts 18-3
- Online-Verbindungen 18-1
- Querverweisliste 14-2, 14-3
- UDT 9-14
- Variablendeklarationsfenster 10-8
- Aufbau der Export-Datei 6-20
- Aufbau des Programmeditor-Fensters 10-1
- Auflisten von Ein-
 - Aus- und Durchgängen 3-7
- Aufruf
 - Baugruppenzustand aus der Projektansicht (Online) 23-6
- Aufrufen der Hilfefunktionen 5-5
- Aufrufhierarchie im
 - Anwenderprogramm 4-9
- Aufrufmöglichkeiten des
 - Baugruppenzustands 23-9
- Aufteilung der Speicherbereiche A-15
- Ausdrucken
 - zur Projektdokumentation 24-1
- Ausgänge
 - auflisten 3-7
 - Belegungsplan 14-6
 - Prozessabbild A-21
- Ausgangsdiagramm für die Motoren erstellen 3-8
- Ausgangsdiagramm für die Ventile erstellen 3-9
- Ausgangsparameter A-68
 - RET_VAL auswerten 23-25
- AuskunftsFunktionen 23-12
- AuskunftsFunktionen der
 - Schnellansicht 23-5
- AuskunftsFunktionen des
 - Baugruppenzustands 23-10
- Auswahl des Meldeverfahrens 16-3
- Auswahl von Objekten im Dialog 5-32
- Auswahldialog 5-32
- Auswählen
 - Programmerstellmethode 9-1
 - Programmiersprache 9-3
- Auswerten
 - Ausgangsparameter
 - RET_VAL 23-25, 23-26
- Auswertung des Diagnosepuffers A-28
- Automation License Manager 2-1
- Automatisierungslösung konzipieren
 - Auflisten von Ein-
 - Aus- und Durchgängen 3-7
 - Beschreiben der einzelnen Funktionsbereiche 3-5
 - Beschreiben der notwendigen Anzeige- und Bedienelemente 3-11
 - Eingangs-/ Ausgangsdiagramms für die Motoren erstellen 3-8
 - Erstellen eines Eingangs-/ Ausgangsdiagramms für die Ventile 3-9
 - Konfigurationsplan erstellen 3-12
 - Prinzipielle Vorgehensweise 3-1
 - Sicherheitsanforderungen festlegen 3-10
 - Zerlegen des Prozesses in Aufgaben und Bereiche 3-2
- Autorisierung installieren 2-4
- Autorisierungsdiskette 2-4
- Autorisierungsprogramm 2-4
- AWL 9-3, 9-4, 9-7
 - Anzeige sprachabhängiger Information 14-9
 - Bausteine eingeben 10-12
 - Einstellungen 10-25
- AWL-Anweisungen
 - Regeln für die Eingabe 10-25
 - Regeln zur Eingabe 13-2
- AWL-Quelle
 - Formate für Bausteine 13-10
 - Inhalt anderer AWL-Quellen einfügen 13-15
- AWL-Quellen
 - anlegen 13-14
 - aus Bausteinen generieren 13-17
 - Bausteinvorlagen einfügen 13-15
 - Beispiel für FBs 13-26
 - Beispiel für FCs 13-24
 - Beispiel für OBs 13-23
 - Beispiel für UDTs 13-29
 - Beispiele für DBs 13-28
 - Beispiele für Variablendeklaration 13-22
 - externe Quellen einfügen 13-16
 - Fehlersuche 13-20
 - Grundlage zum Programmieren 13-1
 - Konsistenz prüfen 13-19
 - Quellcodes vorhandener Bausteine einfügen 13-16
 - Regeln für Variablendeklaration 13-3
 - Regeln zum Festlegen von Bausteineigenschaften 13-5
 - Regeln zum Festlegen von Systemattribute 13-4
 - Regeln zur Reihenfolge der Bausteine 13-4
 - speichern 13-19
 - Struktur von anwenderdefinierten Datentypen 13-9
 - Struktur von Bausteinen 13-8
 - Struktur von Codebausteinen 13-8
 - Struktur von Datenbausteinen 13-9

- Syntax für Bausteine 13-10
- übersetzen 13-20, 13-21
- B**
- Basissoftware 1-6
- Basiszeit (siehe Baugruppenzeit) 18-9
- Baugruppen 26-1
 - parametrieren A-123, A-124
 - simulieren 22-1
 - tauschen 26-1
- Baugruppen tauschen in der Konfigurationstabelle 26-1
- Baugruppenadressierung A-118
- Baugruppen-Anfangsadresse A-119
- Baugruppenparameter A-123, A-124
 - übertragen mit SFCs A-123
 - übertragen mit STEP 7 A-123
- Baugruppen-Symbol für unbekannte Baugruppen 7-6
- Baugruppenträgerausfall (OB 86) 23-41
- Baugruppentyp-abhängiger Umfang der Auskunftsfunktionen im Baugruppenzustand 23-12
- Baugruppenzeit 18-9, 18-10
- Baugruppenzustand 23-2, 23-7, 23-12, 23-13
 - Aktualisieren 23-11
 - anzeigen 23-1
 - Aufrufmöglichkeiten 23-9
 - aus der Projektsicht (Online) aufrufen 23-6
 - Auskunftsfunktionen 23-10
 - von DP-Slaves hinter Y-Link anzeigen 23-13
 - von PA-Feldgeräten anzeigen 23-13
- Baumstruktur 14-4
- Baustein
 - zum Verändern des Pointers A-58
- Bausteinaufrufe 4-9, 4-10
 - aktualisieren 10-26
- Bausteinbezogene Meldungen
 - zuordnen und bearbeiten 16-12
- Bausteinbezogene Meldungen (CPU-weit)
 - anlegen 16-21, 16-23
 - bearbeiten 16-24
- Bausteinbezogene Meldungen (projektweit)
 - anlegen 16-13, 16-15
 - bearbeiten 16-16
- Bausteine 4-2, 15-1, 15-2
 - Attribute 9-22
 - auf dem Zielsystem löschen 19-19
 - aus S7-CPU zurückladen 19-15
 - eingeben in AWL 10-12
 - erzeugen mit S7-GRAPH 9-9
 - im Anwenderprogramm 4-2
 - in das Zielsystem nachladen 19-6
 - speichern 10-28
 - umverdrahten 9-22
 - Zugriffsrechte 10-4
- Bausteine (geladene)
 - auf integriertem EPROM speichern 19-7
 - im PG/PC bearbeiten 19-16
- Bausteine für die Meldung von Systemfehlern generieren 16-49
- Bausteine vergleichen 9-19, 9-21
- Bausteineigenschaft
 - Zeitstempel 15-3
- Bausteineigenschaften 9-15, 9-17, 10-3
- Baustein-Eigenschaften
 - Baustein-Längen anzeigen 9-18
- Bausteincommentar 10-14
- Bausteincommentare
 - eingeben 10-15
- Bausteinconsistenz prüfen 15-1
- Baustein-Längen
 - anzeigen 9-18
- Bausteinordner 9-13
- Baustein-Ordner 5-17
- Bausteinordner-Eigenschaften
 - Baustein-Längen anzeigen 9-18
- Baustein-Stack A-15, A-27
- Bausteintitel 10-14
- Bausteinvorlagen in AWL-Quellen
 - einfügen 13-15
- BCD A-41
- BCD-Format A-42
- Bearbeiten 16-35
 - Anwender-Textbibliotheken 16-35
 - geladener Bausteine
 - falls Anwenderprogramm im PG/PC vorliegt 19-17
 - falls Anwenderprogramm nicht im PG/PC vorliegt 19-17
 - geladener Bausteine im PG/PC 19-16
 - Projekt 6-15
 - S7-Quellen 13-14
- Bearbeiten von Bereichen in Symboltabellen 8-20
- Bearbeiten von Symboltabellen 8-20
- Bedienelemente beschreiben am Beispiel eines industriellen Mischprozesses 3-11
- Bedienen am Kontakt 26-5
- Bedienen und Beobachten
 - von Variablen 17-1
- Bedienerrelevante Texte
 - exportieren/importieren 16-32
 - übersetzen und bearbeiten 16-32
 - Voraussetzungen 16-32
- Bedienphilosophie 5-23

- Bedienpult Beschreiben 3-11
- Bedienung über Tastatureingabe 5-36
- Begleitwert
 - löschen 16-31
 - Meldungen einfügen 16-28
- Beispiel
 - Eingabe eines zusammenhängenden Operandenbereichs 20-11
 - Eingabe von Operanden in Variablentabellen 20-10
 - FBs in AWL-Quellen 13-26
 - FCs in AWL-Quellen 13-24
 - Gleitpunktzahlenformat A-39
 - Maskieren und Demaskieren von Synchronfehlerereignissen A-112
 - OBs in AWL-Quellen 13-23
 - Sperrern und Freigeben von Alarm- und Asynchronereignissen (SFC 39 und 40) A-116
 - UDTs in AWL-Quellen 13-29
 - verzögerte Bearbeitung von Alarm- und Asynchronereignissen (SFC 41 und 42) A-117
 - zur Hantierung mit Uhrzeitalarmen A-99
 - zur Hantierung mit Verzögerungsalarmen A-106
- Beispiel zum Arbeiten mit Verwendungsstellen 14-13
- Beispiele
 - DBs in AWL-Quellen 13-28
 - Eingabe von Steuer-/Forcewerten 20-11
 - Variablendeklarationen in AWL-Quellen 13-22
- Beispielprogramm
 - Industrieller Mischprozess A-80
- Beispielprojekte und Beispielprogramme A-78
- Belegungsplan 14-6
- Benutzeroberfläche (siehe Benutzungsoberfläche) 5-24
- Benutzungsoberfläche 5-24
- Beobachten
 - Prinzipielle Vorgehensweise 20-2
- Beobachten von Variablen
 - Einführung 20-15
 - Trigger festlegen 20-15
- Beschreiben
 - der einzelnen Funktionsbereiche 3-5
 - der notwendigen Anzeige- und Bedienelemente 3-11
 - Sicherheitsanforderungen für das Beispiel eines industriellen Mischprozesses 3-10
- Beschreiben des Bedienpultes am Beispiel eines industriellen Mischprozesses 3-11
- Besonderheiten zum Drucken des Objektbaums 24-3
- Bestandteile einer Meldung 16-5
- Betriebsart
 - für den Test festlegen 21-10
- Betriebsstundenzähler A-127
- Betriebssystem
 - Aufgaben 4-1
- Betriebssystem der CPU 4-15, A-125
- Betriebssysteme für M7-300/400 25-6
- Betriebssystem-Update (siehe Online-Aktualisierung der Firmware von Baugruppen und Modulen) 18-10
- Betriebszustand 21-7
 - ANLAUF A-1, A-3, A-4, A-5 anzeigen und ändern 18-8
 - HALT A-1, A-4
 - RUN A-1, A-3, A-4
 - STOP A-2, A-3, A-4 Stack-Inhalte 23-15
- Betriebszustand "RUN" A-13
- Betriebszustand HALT A-14
- Betriebszustände
 - der CPU A-1
 - Priorität A-4
- Betriebszustände und Übergänge A-1
- Betriebszustandsübergänge A-2
- Bibliothek 5-10
- Bibliotheken 6-14
 - arbeiten mit 9-23
 - archivieren 24-4
 - hierarchischer Aufbau 9-25
 - reorganisieren 26-2
- Bibliotheken (Version 2) bearbeiten 7-1
- Binär-codierte Dezimalzahlen (BCD) A-41
- Bitmeldeverfahren 16-1, 16-2
- Blinktest 18-2
- BLKMOV A-17
- BLOCK A-55
 - Parametertyp A-54
- BLOCK_DB A-54
- BLOCK_FB A-54
- BLOCK_FC A-54
- BLOCK_SDB A-54
- BOOL
 - als Datentyp A-33
 - Bereich A-34
- Boxen
 - entfernen ändern 10-23
 - platzieren 10-18, 10-23
- Browser 5-32

B-Stack

- im B-Stack gespeicherte Daten A-27
- verschachtelte Aufrufe A-27

BuB-Attribute 17-1

- ändern mit CFC 17-5
- projektieren mit AWL
 - FUP
 - KOP 17-3
- projektieren über die Symboltabelle 17-4

Byte

- als Datentyp A-33
- Bereich A-34

C**CAN_TINT 4-29****Certificate of License 2-2, 2-3****CFC 9-4****CFC-Programm 25-1****Codebausteine**

- Aufbau 10-3
- definieren
 - Beispiel A-83
- im inkrementellen Editor 10-3
- speichern 10-28
- Zeitstempel 15-4

COUNTER A-54, A-55

- Parametertyp A-54

CPU (Central Processing Unit)

- Betriebszustände A-1, A-2, A-3, A-4

CPU 31xC 6-25, 6-26, 6-27**CPU urlöschen 19-18****CPU-Baugruppe simulieren 22-1****CPU-Hardwarefehler (OB 84) 23-39****CPU-Hardwarefehler-OB 23-39****CPU-Meldungen**

- anzeigen 16-39
- Archivgröße 16-39

CPU-Meldungen konfigurieren 16-42**CPU-Meldungen und anwenderdefinierte**

- Diagnosemeldungen anzeigen 16-39

CPU-Parameter "Zyklusbelastung durch

- Kommunikation" 4-14

CPU-Redundanzfehler (OB 72) 23-34**CPU-Uhren mit Zeitzonen-Einstellung 18-9****CPU-weit 16-11****CREAT_DB A-16****CRST/WRST A-5, A-6, A-7****CTRL_RTM A-127****D****Darstellung**

- FUP-Elemente 10-22
- globale oder lokale Symbole 8-4
- KOP-Elemente 10-18

Darstellung von unbekanntem

- Baugruppen 7-6

Das STEP 7 - Basispaket 1-6**DATE AND TIME (Datum und Uhrzeit)**

- Bereich A-45
- Format A-44

DATE_AND_TIME A-43**Dateiformate für den Import/Export einer**

- Symboltabelle 8-17

Datenaustausch

- in verschiedenen
 - Betriebszuständen A-13

Datenbaustein (DB) 4-2

- global 4-24
- Instanz-Datenbausteine 4-18, 4-21
- remanent A-30
- Struktur 4-24

Datenbaustein im Ladespeicher

- erzeugen 6-25

Datenbausteine 12-1

- Datensicht 11-3, 11-4
- Datenstruktur mit zugeordnetem FB (Instanz-DBs) eingeben/anzeigen 11-6
- Datenwerte auf die Anfangswerte rücksetzen 11-10
- Datenwerte in der Datensicht ändern 11-10
- Deklarationssicht 11-2
- Grundlagen 11-1
- parametrieren 12-1
- speichern 11-11

Datenbausteinregister A-27**Datensatz**

- lesen A-120
- schreiben A-120
- zugreifen auf A-120, A-123

Datensicht von Datenbausteinen 11-3**Datenträger 6-26****Datentyp**

- anwenderdefiniert 9-14
- DATE AND TIME: Datum und Uhrzeit A-44
- DWORD A-41
- REAL: Gleitpunktzahl A-37, A-38
- S5TIME A-42
- UDT 9-14
- WORD A-41

- Datentypen
 - anwenderdefiniert A-43
 - ANY A-61
 - ARRAY A-43
 - Beschreibung A-34
 - BOOL A-33, A-34
 - Byte A-33
 - BYTE A-34
 - DATE_AND_TIME A-43
 - Datum A-34
 - DINT: Ganzzahl (32 Bit) A-36
 - Doppelwort A-33
 - Doppelwort (DWORD) A-34
 - Einführung A-33
 - elementar A-34
 - FB
 - SFB 4-18, A-43
 - Ganzzahl (16 Bit) (INT) A-34
 - Ganzzahl (32 Bit) (DINT) A-34
 - INT: Ganzzahl (16 Bit) A-35
 - Parametertypen
 - ANY
 - Parameter A-63, A-64, A-65
 - Realzahl (REAL) A-34
 - S5TIME A-34
 - STRING A-43
 - STRUCT A-43
 - UDT A-43
 - Uhrzeit (TIME OF DAY) A-34
 - Wort A-33
 - Wort (WORD) A-34
 - Zeichen (CHAR) A-34
 - Zeit (TIME) A-34
 - zusammengesetzt A-43
- Datentypen zu Lokaldaten von
 - Codebausteinen zuordnen A-67
- Datenwerte
 - auf die Anfangswerte rücksetzen 11-10
 - in der Datensicht von Datenbausteinen
 - ändern 11-10
- DB 4-24
 - Formattabelle 13-13
- DBs in AWL-Quellen
 - Beispiele 13-28
- Deaktivieren
 - Uhrzeitalarm 4-28
- Dearchivieren
 - Vorgehensweise 24-6
- Defekt
 - Betriebszustand der CPU A-1
- Definieren
 - Codebausteine A-83
 - Symbole bei der Programmeingabe 8-13
- Deinstallieren
 - der Nutzungsberechtigung 2-5
- STEP 7 2-13
- Deklarationssicht von
 - Datenbausteinen 11-2
- Deklarationstyp
 - ändern 10-8
- Deklarieren von Lokaldaten A-67, A-68
- Deklarieren von lokalen Variablen
 - FB für das Beispiel eines industriellen Mischprozesses A-86
 - OB für das Beispiel eines industriellen Mischprozesses A-92
- Deklarieren von Parametern
 - FC für das Beispiel eines industriellen Mischprozesses A-90
- Demaskieren
 - Startereignisse 4-40
- Demaskieren von
 - Synchronfehlerereignissen
 - Beispiel A-112
- Detailvergleich 9-21
- Dezentrale Peripherie 7-1, 7-3
- Diagnose der Hardware 23-1
- Diagnosealarm (OB 82) 23-37
- Diagnosealarm-OB 23-37, 23-39
- Diagnosedaten auf Baugruppen 23-21
- Diagnoseereignis 23-23, A-28
- Diagnosefunktionen 23-23
- Diagnosemeldung
 - an Teilnehmer senden 23-22
 - eigene schreiben 23-22
- Diagnosepuffer A-28, A-29
 - auslesen 23-18, A-29
 - auswerten A-28
 - Definition A-28
 - Inhalt 23-23, A-28, A-29
- Diagnosesymbole
 - in der Online-Sicht 23-3
- Diagnosezustandsdaten 23-21
- Dialogfelder 5-26
- DINT
 - Datentyp A-36
- Direkt am PG angeschlossenen Teilnehmer
 - identifizieren 18-2
- Direkter Datenaustausch
 - (Querverkehr) 7-3
- DIS_AIRT 4-40
- DIS_IRT 4-38
- DMSK_FLT 4-38
- DOCPRO 24-1
- Dokumentation 1-1, 5-6, 5-8, 24-1
 - eines ganzen Projekts drucken 24-1
 - von Projektbestandteilen drucken 24-1
- Doppelwort (DWORD)
 - Bereich A-34
 - Datentyp A-33

- DP/PA-Link (IM 157) 23-13
- DP-Normslaves A-122
- DPNRM_DG A-122
- DPRD_DAT A-122
- DP-Slaves 7-1, 7-2
- DP-Slaves mit fehlenden oder fehlerhaften GSD-Dateien A-77
- DPWR_DAT A-122
- Drucken
 - Bausteine 24-1
 - Diagnosepufferinhalt 24-1
 - Globaldatentabelle 24-1
 - Konfigurationstabelle 24-1
 - Projektbestandteile 24-1
 - Projektdokumentation 24-1
 - Referenzdaten 24-1
 - Symboltabelle 24-1
 - Variablentabelle 24-1
- Drucker
 - einrichten 24-2
- Dummy-Slave A-77
- Durchgänge
 - auflisten 3-7
- Durchgangsparameter A-68
- Durchklingeln (siehe Blinktest) 18-2
- DWORD
 - Datentyp A-41
- E**
- Editieren
 - in der Symboltabelle 8-13
- Editor
 - Einstellungen für AWL 10-4
- Eigene Diagnosemeldungen senden 23-22
- Einfügen
 - Bausteinvorlagen in AWL-Quellen 13-15
 - Bereich zusammenhängender Operanden in eine Variablentabelle 20-7
 - Ersatzwerte bei Fehlererkennung 23-31
 - externer Quellen 13-16
 - Inhalt anderer AWL-Quellen 13-15
 - Kommentarzeilen 20-10
 - Operanden oder Symbole in eine Variablentabelle 20-5
 - Quellcode vorhandener Bausteine in AWL-Quellen 13-16
 - S7-/M7-Programm 6-13
 - Steuerwerte 20-8
- Einfügen von Stationen 6-11
- Einführung zu Daten- und Parametertypen A-33
- Einführung zum Forcen von Variablen 20-21
- Einführung zum Testen mit der Variablentabelle 20-1
- Eingabe in Dialogfelder 5-25
- Eingabemöglichkeiten von globalen Symbolen 8-12
- Eingänge
 - auflisten 3-7
 - Belegungsplan 14-6
 - Prozessabbild A-21
- Eingangsdiagramm für die Motoren erstellen 3-8
- Eingangsdiagramm für die Ventile erstellen 3-9
- Eingangsparameter A-67
- Eingeben
 - Baustein-/Netzwerkcommentare 10-15
 - Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs) 11-6
 - Datenstruktur von globalen Datenbausteinen 11-5
 - einzelner globaler Symbole im Dialog 8-13
 - globale Symbole in ein Programm 10-13
 - Multiinstanz im Variablendeklarationsfenster 10-10
 - Struktur von anwenderdefinierten Datentypen (UDT) 11-8
 - Struktur von Datenbausteinen mit zugeordnetem UDT 11-9
 - von Symbolen 8-14
- Eingeben mehrerer globaler Symbole in der Symboltabelle 8-14
- Eingrenzen von Störungen 23-2
- Einrichten des Zugriffsschutzes 6-3
- Einsetzen
 - SFC A-23
- Einstellen
 - Betriebsverhalten A-122
 - PG/PC-Schnittstelle 2-11
 - virtueller Arbeitsspeicher 26-6
- Einstellen der Windows-Sprache 6-8
- Einstellen des Operandenvorrangs (symbolisch/absolut) 8-5
- Einstellungen
 - AWL-Editor 10-4
 - für die Programmiersprache AWL 10-25
 - für die Programmiersprache FUP 10-22
- Einstellungen für das Melden von Systemfehlern 16-48
- Einstellungen für Programmiersprache KOP 10-18
- Elementare Datentypen A-34
- Elemente in Dialogfeldern 5-25
- EN / ENO
 - Beschaltung 10-24

- EN_AIRT 4-38
 - EN_IRT 4-38
 - Engineering Tools 1-15
 - Entfernen des Zugriffsschutzes 6-3
 - Entstehung von Lücken im
 - Anwenderspeicher (RAM) 19-20
 - Entwurfsmethoden
 - Entwerfen eines strukturierten Programms A-83
 - EPROM A-30
 - EPROM-Bereich A-16
 - Ereignisse 4-15
 - Ereignisse
 - asynchron 4-11
 - Erreichbare Teilnehmer 18-2
 - Ersatzwert
 - Verwenden von SFC44 (RPL_VAL) 23-31
 - Erstellen
 - eines anwenderdefinierten Datentyps A-51
 - Eingangs-/Ausgangsdiagramm für die Motoren 3-8
 - Eingangs-/Ausgangsdiagramm für die Ventile 3-9
 - FB für den Motor A-86, A-87, A-88, A-89
 - FC für die Ventile A-90, A-91, A-92
 - Felder A-46, A-48, A-49
 - Konfigurationsplan 3-12
 - OB 1 für das Beispiel eines industriellen Mischprozesses A-92
 - Struktur A-49, A-50
 - Variablen-tabelle 20-3
 - von Anwenderprogrammen 10-3
 - Erstellen eines Konfigurationsplans am Beispiel industrieller Mischprozess 3-12
 - Erstellen von Anwender-Textbibliotheken 16-34
 - Erweitern von DP-Slaves (mit Vorgänger-Versionen von STEP 7 erstellt) 7-1
 - Erweiterungsmöglichkeiten des STEP7 - Basispakets 1-14
 - Erzeugen 5-27, 5-28
 - Objekte 5-27
 - Referenzdaten 14-11
 - und Manipulieren von Objekten 5-27
 - Erzeugen und Manipulieren von Objekten 5-27
 - Export-Datei 6-20
 - Exportieren
 - Quellen 13-18
 - Symbol-tabelle 8-17
 - Externe Quellen einfügen 13-16
- F**
- FB 4-18, 4-19, 4-20, A-43
 - Formattabelle 13-11
 - Korrigieren der Schnittstelle 15-6
 - FBs in AWL-Quellen
 - Beispiel 13-26
 - FC 4-16, 4-17
 - Formattabelle 13-12
 - Korrigieren der Schnittstelle 15-6
 - FC 12 A-101
 - FCs in AWL-Quellen
 - Beispiel 13-24, 13-25
 - Fehlende Symbole anzeigen 14-10
 - Fehler
 - während der Installation 2-8
 - Fehler vermeiden beim Aufruf von Bausteinen 15-7
 - Fehlerbehebung
 - Programmbeispiele 23-26
 - Fehlererkennung
 - OB-Typen
 - OB81 23-26
 - Programmbeispiele
 - Ersatzwerte 23-31
 - Verwenden von Fehler-OBs
 - um auf Fehler zu reagieren 4-38
 - Fehler-OB 23-26, 23-27, 23-28, 23-29, 23-30
 - OB-Typen
 - OB 121 und OB 122 4-38
 - OB 70 und OB 72 4-38
 - OB 80 bis OB 87 4-38
 - Verwenden von Fehler-OBs zur Reaktion auf Ereignisse 4-38
 - Fehler-OBs als Reaktion auf die Erkennung eines Fehlers 23-26
 - Fehlersuche 23-1
 - in Bausteinen 10-17
 - Fehlersuche in AWL-Quellen 13-20
 - Feld (Datentyp ARRAY)
 - Anzahl verschachtelter Ebenen A-45
 - Beschreibung A-46
 - Felder
 - erstellen A-46, A-48
 - für den Datenzugriff verwenden A-46
 - Fenster "Erreichbare Teilnehmer" 18-2
 - Fensteranordnung
 - wiederherstellen 5-35
 - Fensteranordnung ändern 5-34
 - Fenstertypen umschalten 5-41
 - FEPROM A-30
 - Festlegen
 - Anzeige für den Programmstatus 21-9
 - Betriebsart für den Test 21-10

- Sicherheitsanforderungen 3-10
 - Trigger zum Beobachten von Variablen 20-15
 - Trigger zum Steuern von Variablen 20-18
 - Feststellbare Fehler 23-26
 - Filtern
 - von Symbolen 8-14
 - Firmware-Update 18-10, 18-12, A-125
 - Flash-File-System 2-8
 - FORCE-LED blinkt 18-2
 - Forcen von Variablen 20-21
 - Einführung 20-21
 - Sicherheitsvorkehrungen 20-20
 - Forcewerte
 - Beispiele für die Eingabe 20-11
 - Formalparameter
 - Systemattribute und Meldebausteine 16-8
 - Format
 - BLOCK A-55
 - COUNTER A-55
 - Datentyp DATE_AND_TIME (Datum und Uhrzeit) A-44
 - Datentyp DINT (32-Bit-Ganzzahlen) A-36
 - Datentyp INT (16-Bit-Ganzzahlen) A-35
 - Datentyp REAL (Gleitpunktzahlen) A-37
 - Datentyp S5TIME (Zeitdauer) A-42
 - Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen A-41
 - Parametertyp ANY A-61
 - Parametertyp POINTER A-55, A-56
 - Parametertypen BLOCK
 - COUNTER
 - TIMER A-55
 - TIMER A-55
 - Formate für Bausteine in AWL-Quellen 13-10
 - Formattabelle von DBs 13-13
 - Formattabelle von FBs 13-11
 - Formattabelle von FCs 13-12
 - Formattabelle von OBs 13-10
 - Freigeben von Alarm- und Asynchronfehlerereignissen
 - Beispiel A-116
 - Fremdsprachige Zeichensätze verwenden 6-5
 - Funktion (FC) 4-2, 4-16
 - Anwendungsbereich 4-16
 - Erstellen
 - Beispiel-FC für das Beispiel eines industriellen Mischprozesses A-90
 - Funktionen 24-2
 - Funktionen (FC) 4-17
 - Funktionsbausteine (FB) 4-18
 - Funktionsbausteine (FBs) 4-2, 4-18
 - Aktualparameter 4-19, 4-20
 - Anwendungsbereich 4-18
 - Erstellen
 - Beispiel-FB für das Beispiel eines industriellen Mischprozesses A-86
 - Funktionsleiste
 - Symbole 5-24
 - Funktionsplan 9-6
 - Funktionsumfang von "Systemfehler melden" 16-45
 - FUP 9-6
 - Anzeige sprachabhängiger Information 14-9
 - FUP-Anweisungen
 - Regeln für die Eingabe 10-23
 - FUP-Elemente
 - Darstellung 10-22
 - FUP-Layout 10-22
 - FW-Update 18-10
- ## G
- Ganzzahl (16 Bit)
 - Format A-35
 - Ganzzahl (16 Bit) (INT)
 - Bereich A-34
 - Ganzzahl (32 Bit)
 - Format A-36
 - Ganzzahl (32 Bit) (DINT)
 - Bereich A-34
 - GD-Kommunikation A-77
 - Geladene Bausteine auf integriertem EPROM speichern 19-7
 - Geladene Bausteine im PG/PC bearbeiten 19-16
 - Generieren
 - AWL-Quellen aus Bausteinen 13-17
 - Generieren von Bausteinen für die Meldung von Systemfehlern 16-49
 - Gepufferter Betrieb
 - remanente Speicherbereiche A-31
 - Gleitpunktzahl
 - Beispiel A-39
 - Grundelemente A-37
 - Komponentenfelder A-37
 - Parameter A-37
 - Gleitpunktzahlen
 - Format A-38
 - Globaldaten-Kommunikation A-77
 - Globale Datenbausteine
 - Datenstruktur eingeben 11-5
 - Zeitstempel 15-5
 - Globale Datenbausteine (DB) 4-24

- Globale Symbole
 - einzel einzugeben im Dialog 8-13
 - in ein Programm eingeben 10-13
 - mehrere eingeben in der Symboltabelle 8-14
- Globale und lokale Symbole 8-3
- GRAPH 9-4
- Groß-/Kleinschreibung bei Symbolen 8-15
- Große Projekte 26-1
- Grundlagen
 - zu Datenbausteinen 11-1
- Grundlagen des Meldekonzeptes 16-1
- Grundlagen zum Programmieren in AWL-Quellen 13-1
- GSD-Datei 7-1, 7-2, A-77

- H**
- HALT 21-7, A-14
 - Betriebszustand der CPU A-1
- Haltepunktliste 21-6
- Hantierung von großen Projekten 26-1
- Hardware diagnostizieren 23-1
 - Ausführliche Diagnosesicht 23-8
 - Schnellansicht 23-5
- Herstellen einer Verbindung
 - zur CPU 20-14
- Hierarchischer Aufbau von Bibliotheken 9-25
- HiGraph 9-4
- Hilfe (online)
 - aufrufen 5-5
 - Themen 5-5
- Hintergrund-OB
 - Priorität 4-36
 - Programmieren 4-37
- Hinweis
 - zur Aktualisierung des Fensterinhalts 18-7
- Human Machine Interface 1-19

- I**
- Icons für Objekte im SIMATIC Manager 5-6
- Ident-Nummer eintragen 2-8
- IM 157 (DP/PA-Link) 23-13
- Importieren
 - externe Quelle 6-13
 - Quellen 13-17
- Importieren Symboltabelle 8-17
- IN (Variablendeklaration) A-67
- IN_OUT (Variablendeklaration) A-67
- IN_OUT-Parameter eines Funktionsbausteins A-74
- Index 16-30, 16-31
 - in Textbibliotheken 16-30
- Indirekte Parametrierung A-123
- Informationen zur Protokoll-Datei 6-22
- Installationsvoraussetzungen 2-6
- Installieren
 - von STEP 7 2-6
- Installieren des Automation License Managers 2-4
- Installieren von STEP 7 2-7
- Instanz 4-21
- Instanz-Datenbaustein 4-21, 4-22, 4-23
 - mehrere Instanzen für einen FB erstellen 4-18
 - remanent A-30
- Instanz-Datenbausteine 4-21
 - Zeitstempel 15-5
- INT
 - Datentyp A-35
- Interruptbelegung
 - prüfen 2-12

- K**
- k7e 6-27
- k7p 6-27
- Kaltstart A-1
- Kennzeichnung
 - von Symbolen 8-4
- Kombinationsfeld
 - Definition 5-25
- Kommentare
 - zu Bausteinen 10-14
 - zu Netzwerken 10-14
- Kommentarzeichen 20-5
- Kommentarzeile 20-5
- Kommentarzeilen einfügen 20-10
- Kommunikationsfehler (OB 87) 23-42
- Kommunikationsfehler-OB 23-42
- Kommunikationslast 4-14, 4-15
- Kommunikationsprozesse 4-11
- Kompatibilität (Direkter Datenaustausch) 7-3
- Kompatibilität (DP-Slaves) 7-1, A-77
- Kompatibilität (Online-Verbindung über PROFIBUS-DP-Schnittstelle) 7-3
- Kompatibilität (V2-Projekte und -Bibliotheken) 7-1
- Komprimieren
 - des Anwenderspeichers 19-20
 - des Speicherinhalts einer S7-CPU 19-21
- Konfigurierbare Speicherobjekte im Arbeitsspeicher A-32
- Konfigurieren der Hardware 26-1
- Konfigurieren von CPU-Meldungen 16-42
- Konfigurationsplan erstellen 3-12
- Konsistenz in AWL-Quellen prüfen 13-19

- Kontaktplan 9-5
- Kontextabhängige Hilfe 5-5
- Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern 23-17
- Konzeption einer Automatisierungslösung
 - Anzeige- und Bedienelemente beschreiben 3-11
 - Ein-
 - Aus- und Durchgänge auflisten 3-7
 - Eingangs-/ Ausgangsdiagramm für die Ventile erstellen 3-9
 - Eingangs-/ Ausgangsdiagramms für die Motoren erstellen 3-8
 - Funktionsbereiche beschreiben 3-5
 - Konfigurationsplan erstellen 3-12
 - Prinzipielle Vorgehensweise 3-1
 - Prozess in Aufgaben und Bereiche zerlegen 3-2
 - Sicherheitsanforderungen festlegen 3-10
- KOP 9-3, 9-4, 9-5
 - Anzeige sprachabhängiger Information 14-9
 - unzulässige Verschaltungen 10-21
- KOP-Anweisungen
 - Regeln für die Eingabe 10-18
- KOP-Elemente
 - Darstellung 10-18
- Kopf- und Fußzeilen 24-3
- Kopieren/Verschieben von Variablen tabellen 20-4
- KOP-Layout 10-18
- Korrekturfaktor 18-9
- Korrigieren
 - der Schnittstellen in einem FC FB oder UDT 15-6
- Kurzschluss 10-21

- L**
- Lade- und Arbeitsspeicher A-16
- Lade- und Arbeitsspeicher in der CPU 19-3
- Lade-/Arbeitsspeicher löschen 19-18
- Lademöglichkeiten A-16
- Lademöglichkeiten abhängig vom Ladespeicher 19-4
- Laden 19-9, 19-10
 - Anwenderprogramm A-16
 - Anwenderprogramme in Zielsystem 19-3
 - aus dem Zielsystem in das PG 19-14
 - der aktuellen Konfiguration und aller Bausteine in das PG 19-14
 - mehrerer Objekte 19-9
 - mit Projektverwaltung 19-5
 - ohne Projektverwaltung 19-6
 - über EPROM-Memory Cards 19-7
 - Voraussetzungen 19-1
 - Laden von Objekten 19-11
 - Ladespeicher 19-3, A-15, A-16
 - nicht ablaufrelevante DBs speichern A-16
 - Layout des Quelltextes festlegen 13-15
 - Leitfaden durch STEP 7 1-1
 - Leittechnikdialog
 - siehe PCS 7-Meldungsprojektierung (CPU-weit) 16-24
 - siehe PCS 7-Meldungsprojektierung (projektweit) 16-16
 - Lesen aus Datenbaustein im Ladespeicher 6-25
 - Lesen und Stellen der Uhrzeit und des Uhrzeitstatus 18-9
 - License Key 2-1, 2-2, 2-5
 - License Manager 2-1, 2-2
 - Linear programmieren 4-3
 - Listen bedienerrelevanter Texte 16-32
 - Listenfeld 5-25
 - Lizenz 2-2, 2-3
 - Lizenz-Typen 2-3
 - Enterprise License 2-1
 - Floating License 2-3
 - Rental License 2-3
 - Single License 2-3
 - Trial License 2-3
 - Upgrade License 2-3
 - Lokaldatenbedarf 14-4, 14-5
 - Lokaldaten-Stack A-15, A-26
 - Lokalzeit 18-9
 - Löschen
 - Lade-/Arbeitsspeicher 19-18
 - S7-Bausteine auf dem Zielsystem 19-19
 - STEP 7-Objekte 5-27
 - L-Stack
 - Bearbeitung von Daten in einem verschachtelten Aufruf A-27
 - Speichern von temporären Variablen 4-18
 - Überschreiben des L-Stack A-25
 - Zuordnen von Speicher zu lokalen Variablen A-25

M

- M7-Programm
 - einfügen 6-13, 6-14
- Make (siehe Bausteinkonsistenz prüfen) 15-1
- Make (siehe Objekte übersetzen und laden) 19-9, 19-11
- Manipulieren
 - Objekte 5-27, 5-28, 5-29, 5-30, 5-31, 5-32
- Maskieren
 - Startereignisse 4-38
- Maskieren von Synchronfehlerereignissen
 - Beispiel A-112
- Maßnahmen im Programm zur
 - Störungsbehandlung 23-24
- Maximalzykluszeit 4-11
- Meldebausteine
 - Übersicht 16-6
- Meldefähiger Baustein 16-13, 16-21
- Meldekonzert
 - Grundlagen 16-1
- Melden von Systemfehlern 16-49
- Melden von Systemfehlern
 - projektieren 16-43
- Meldenummern 16-11
 - vergeben 16-10
- Meldenummernverfahren 16-2
- Meldenummernvergabe 16-11
- Meldeverfahren
 - auswählen 16-3
- Meldung
 - anwenderdefinierte Diagnosemeldung 16-19
 - bausteinbezogen 16-12
 - Beispiel 16-6
 - Bestandteile 16-5
- Meldung (CPU-weit)
 - symbolbezogen 16-26
- Meldung (projektweit)
 - symbolbezogen 16-18
- Meldungen 16-30
 - Texte aus Textbibliotheken integrieren 16-30
- Meldungsprojektierung
 - Daten nach WinCC transferieren 16-38
 - SIMATIC-Komponenten 16-5
- Meldungstyp 16-9, 16-10
- Meldungstyp und Meldungen 16-9
- Meldenummernvergabe 16-11
- Memory Card A-17, A-18
 - parametrieren 2-8
- Memory Card-Datei 6-26
- Menüleiste 5-24
- Merker
 - Belegungsplan 14-6
 - Speicherbereich
 - remanent A-31
- Micro Memory Card (MMC) 6-26, 6-27
- Micro Memory Card als Datenträger
 - verwenden 6-26
- Micro Memory Cards (MMC) 6-25
- Mindestzykluszeit 4-13
- MMC 6-25, 6-26, 6-27
- Mnemonik
 - einstellen 10-25
- Möglichkeiten zum Abändern der
 - Meldenummernvergabe eines Projekts 16-11
- Motoren
 - Eingangs-/Ausgangsdiagramm erstellen 3-8
- MPI 18-10
- MPI-ISA-Card (Auto) 2-11
- MPI-Karte im PG/PC 2-11
- MPI-Schnittstelle 2-6, 2-7
- MSK_FLT 4-38
- Multiinstanz 4-18, 4-21
 - im Variablendeklarationsfenster eingeben 10-10
- Multiinstanzen
 - Regeln 10-10
 - Verwendung 10-9
- Multiprojekt
 - Online-Zugriff auf Zielsysteme im 18-4
- Multiprojekte mit Zugriffsschutz 6-3

N

- Nachladen von Bausteinen in das
 - Zielsystem 19-6
- Namenskonventionen
 - für Projektierungsdaten 17-1
- Netzausfall A-6, A-7
- Netzwerke 9-6
 - Abschluss in KOP 10-18
- Netzwerkcommentar 10-14
- Netzwerkcommentare
 - eingeben 10-15
- Netzwerktitel 10-14
- Neustart A-1, A-5, A-6, A-7, A-8, A-9, A-10
 - abbrechen A-10
 - automatisch A-5
 - ungepuffert A-5
 - manuell A-5
- Nicht verwendete Operanden
 - anzeigen 14-10
- Nicht verwendete Symbole 14-8

- Nichtflüchtiger RAM A-30
- Non-Retain 9-17
- Nullpointer A-61
- Nutzdaten A-119
- Nutzen der Uhrzeitfunktionen A-126
- Nutzungsberechtigung durch den Automation License Manager 2-1
- NVRAM A-30, A-31

- O**
- OB 4-3, 4-4, 4-5, 4-6, 4-7, 4-8
 - Formattabelle 13-10
- OB 1 A-110, A-111
- OB 1 und OB 80 A-105
- OB 10 A-103, A-104
- OB 100 A-5
- OB 101 A-5, A-12
- OB 102 A-5
- OB 121 23-43
- OB 122 23-44
- OB 1-Prozessabbild A-22
- OB 1-Zykluszeit 4-15
- OB 20 A-108, A-109
- OB 70 23-33
- OB 72 23-34
- OB 80 23-35
- OB 81 23-36
- OB 82 23-37
- OB 83 23-38
- OB 84 23-39
- OB 85 23-40, A-24
- OB 86
 - Baugruppenträgerausfall 23-41
- OB 87 23-42
- Obergrenzen für die Eingabe von Zählern 20-9
- Obergrenzen für die Eingabe von Zeiten 20-8
- Objekt
 - ausschneiden
 - kopieren
 - einfügen 5-27, 5-28, 5-29
 - auswählen 5-32
 - Eigenschaften 5-28, 5-29
 - Löschen 5-32
 - öffnen 5-28, 5-31
 - umbenennen 5-30
 - verschieben 5-31
 - Objekt Baustein-Ordner 5-17
 - Objekt Bibliothek 5-10
 - Objekt erzeugen 5-27
 - Objekt Programmierbare Baugruppe 5-13
 - Objekt Projekt 5-8
 - Objekt Quellen-Ordner 5-21
 - Objekt S7/M7-Programm 5-15
 - Objekt Station 5-11
 - Objekte 5-6, 5-7
 - als Ordner 5-6
 - als Träger von Eigenschaften 5-6
 - als Träger von Funktionen 5-6
 - manipulieren 5-27
 - Objekte übersetzen und laden 19-9, 19-11
 - Objekte und Objekt-Hierarchie 5-6
 - Objekthierarchie 5-6
 - aufbauen 5-28
 - OBs in AWL-Quellen
 - Beispiel 13-23
 - Offline-Aktualisierung der Firmware von Baugruppen und Modulen A-125
 - Offline-Aktualisierung des Betriebssystems von Baugruppen und Modulen A-125
 - Öffnen
 - Variablentabelle 20-3
 - von Symboltabellen 8-14
 - On-/Offline-Partner vergleichen 9-19
 - Online-Aktualisierung der Firmware von Baugruppen und Modulen 18-10
 - Online-Hilfe
 - aufrufen 5-5
 - Schriftgröße ändern 5-5
 - Themen 5-5
 - Online-Sicht
 - Diagnosesymbole 23-3
 - Online-Verbindung
 - aufbauen über das Fenster "Erreichbare Teilnehmer" 18-2
 - aufbauen über das Online-Fenster des Projekts 18-3
 - Online-Verbindung über DP-Schnittstelle 7-3
 - Online-Zugriff auf Zielsysteme im Multiprojekt 18-4
 - Operanden
 - in eine Variablentabelle einfügen 20-5
 - nicht verwendete 14-10
 - ohne Symbol 14-10
 - umverdrahten 9-22
 - Operanden ohne Symbol 14-8, 14-9
 - Operandenbereiche
 - Beschreibung A-19
 - Operandenfeldbreite 10-18, 10-22
 - Operandenvorrang (symbolisch/absolut) 8-5
 - Optimieren der Vorlage für die Übersetzung 6-23
 - Optimierung des Übersetzungsvorgangs 6-24
 - Optionspaket 22-1
 - Optionssoftware 25-3

- Optionssoftware für
 - M7-Programmierung 25-3
- Ordner
 - Bausteine 9-13
- Organisationsbaustein (OB)
 - Hintergrund-OB (OB 90) 4-3, 4-36
- Organisationsbaustein für die
 - Programmbearbeitung im Hintergrund (OB 90) 4-36
- Organisationsbaustein für zyklische
 - Programmbearbeitung (OB1) 4-11
- Organisationsbausteine 4-2
 - Definition 4-3
 - Erstellen eines OB für das Beispiel eines industriellen Mischprozesses A-92
 - Fehlererkennung
 - OB122
 - Ersatzwerte 23-31
 - Prioritätsklassen 4-3, 4-5, 4-6
 - Reagieren auf Fehler 4-38
 - Startinformation 4-5
 - Organisationsbausteine für alarmgesteuerte
 - Programmbearbeitung 4-27
 - Organisationsbausteine für den Anlauf (OB100/OB101/OB102) 4-34
 - Organisationsbausteine für den
 - Prozessalarm(OB 40 bis OB 47) 4-33
 - Organisationsbausteine für den
 - Uhrzeitalarm (OB 10 bis OB 17) 4-28
 - Organisationsbausteine für den
 - Verzögerungsalarm (OB 20 bis OB 23) 4-30
 - Organisationsbausteine für den Weckalarm (OB 30 bis OB 38) 4-31
 - Organisationsbausteine für die
 - Fehlerbearbeitung (OB70bisOB87 / OB 121 bis OB 122) 4-38
 - Organisationsbausteine und
 - Programmstruktur 4-3
 - OUT (Variablendeklaration) A-67

- P**
- PAA (Prozessabbild der Ausgänge) A-21
- PAE (Prozessabbild der Eingänge) A-21
- PA-Feldgeräte 23-13
- Parameter
 - Attribute 9-22
- Parametertyp
 - ANY A-54
 - BLOCK_DB A-54
 - BLOCK_FB A-54
 - BLOCK_FC A-54
 - BLOCK_SDB A-54
 - COUNTER A-54
 - POINTER A-54
 - TIMER A-54
- Parametertyp POINTER verwenden A-56
- Parametertypen A-54
 - ANY A-63, A-64, A-65, A-66
 - Einführung A-33
- Parametrierbare Baugruppen A-123
- Parametrieren
 - indirekt A-123
 - mit SFCs A-124
 - mit STEP 7 A-123, A-124
 - prozessalarmfähige
 - Signalbaugruppen 4-33
 - Uhr A-126
- Parametrieren der
 - PG/PC-Schnittstelle 2-11
- Parametrieren von Datenbausteinen 12-1
- Parametrieren von Technologischen
 - Funktionen 12-2
- PARAM_MOD A-120, A-124
- Passwort 18-6
- Passwortschutz für Zugriff
 - auf Zielsysteme 18-6
- PCS 7-Meldungsprojektierung
 - (CPU-weit) 16-24
- PCS 7-Meldungsprojektierung
 - (projektweit) 16-16
- PC-Station 7-4, 7-5
- Peripherie
 - Adressbereiche A-118
- Peripheriedaten A-119
- Peripherie-Redundanzfehler (OB 70) 23-33
- Peripherie-Redundanzfehler-OB 23-33, 23-34
- Peripheriezugriffsfehler (OB 122) 23-44
- Peripheriezugriffsfehler (PZF) bei
 - Prozessabbild-Aktualisierung A-24
- Peripheriezugriffsfehler-OB 23-44
- Personenschaden verhüten 20-21
- PG/PC-Schnittstelle 2-11
 - Parametrieren 2-11
- Phasenverschiebung 4-31, 4-32
- Platzieren
 - Boxen 10-23, 10-24
- Pointer A-56, A-57, A-59
- POINTER A-54
 - Parametertyp A-54
- Pointerformat A-54, A-55
- Prinzipielle Vorgehensweise
 - beim Beobachten und Steuern 20-2
 - beim Drucken 24-2
 - beim Erstellen von Codebausteinen 10-3
 - zur Konzeption einer
 - Automatisierungslösung 3-1

- Prinzipielles Vorgehen
 - zum Ermitteln einer STOP-Ursache 23-15
- Priorität
 - ändern 4-3
 - Hintergrund-OB 4-36
 - Prozessalarm 4-33
 - Uhrzeitalarm 4-28
 - Verzögerungsalarm 4-30
- PROFIBUS 18-10, 18-12
- PROFIBUS-DP 7-1, 7-3
- PROFIBUS-PA-Geräte 23-13
- PROFINET-Teilnehmer 18-2
- Programmablauffehler (OB 85) 23-40
- Programmablauffehler-OB 23-40
- Programmbearbeitung
 - alarmgesteuert 4-3, 4-27
 - zyklisch 4-7
- Programmbeispiele
 - Beispiel eines industriellen Mischprozesses
 - Beschreiben der Sicherheitsanforderungen 3-10
 - Beschreiben des Bedienpultes 3-11
 - Beschreibung der einzelnen Aufgaben und Bereiche 3-5
 - Erstellen eines E/A-Diagramms 3-7
 - Konfigurationsplan erstellen 3-12
 - Zerlegen eines Prozesses in Aufgabenbereiche 3-2
 - Einfügen von Ersatzwerten 23-31
 - Ersatzwerte 23-31
 - FB für das Beispiel eines industriellen Mischprozesses A-86
 - FC für das Beispiel eines industriellen Mischprozesses A-90
 - OB für das Beispiel eines industriellen Mischprozesses A-92
 - Reagieren auf Batteriefehler 23-26
- Programme in einer CPU 4-1
- Programmeditor 9-19, 9-20, 9-21, 10-1
- Programmelemente
 - einfügen 10-5
- Programmerstellmethode auswählen 9-1
- Programmerstellung
 - allgemeine Vorgehensweise 1-1
- Programmieren
 - einer FC
 - Beispiel A-90
 - eines OB 1
 - Beispiel A-92
 - FB
 - Beispiel A-86
 - Hintergrund-OB 4-36
- Programmierfehler (OB 121) 23-43
- Programmierfehler-OB 23-43
- Programmiersprache
 - auswählen 9-3
 - festlegen 9-3
 - FUP (Funktionsplan) 9-6
 - KOP (Kontaktplan) 9-5
 - S7-CFC 9-12
 - S7-Graph (Ablaufsteuerung) 9-9
 - S7-HiGraph (Zustandsgraph) 9-10
 - S7-SCL 9-8
- Programmiersprache AWL (Anweisungsliste) 9-7
- Programmierung
 - Entwerfen eines strukturierten Programms A-83
 - Übergeben von Parametern 4-18
 - Verwendung von Datenbausteinen 4-18
- Programmstatus 21-1, 21-2
 - anzeigen 21-3
 - Anzeigen festlegen 21-9
- Programmstatus von Datenbausteinen 21-8
- Programmstruktur 14-4, 14-5
 - anzeigen 14-10
- Projekt 5-8
 - anlegen manuell 6-9
 - anlegen über Assistenten 6-9
 - kopieren 6-15
 - löschen 6-15
 - mit Globaldaten-Kommunikation
 - archivieren
 - umsetzen
 - umbenennen A-77
 - öffnen 6-15
- Projekt auf verwendete Softwarepakete prüfen 6-16
- Projektbestandteile drucken 24-1
- Projektdateien auf Micro Memory Card (MMC) ablegen 6-27
- Projektdokumentation drucken 24-1
- Projekte
 - archivieren 24-4
 - Bearbeitungsreihenfolge 6-10
 - reorganisieren 26-2
 - umbenennen 5-27
- Projekte (Version 2) bearbeiten 7-1
- Projekte mit einer großen Anzahl vernetzter Stationen 26-1
- Projekte mit Zugriffsschutz 6-3
- Projektfenster 6-1, 6-2
- Projektieren von bedien- und beobachtbaren Variablen 17-1
- Projektieren von BuB-Attributen mit AWL KOP und FUP 17-3

- Projektieren von BuB-Attributen über die Symboltabelle 17-4
 - Projektieren von Meldungen bei Systemfehlern 16-43
 - Projektierungsdaten 17-1, 17-2
 - transferieren 16-38, 17-6
 - Voraussetzungen zum Transferieren 16-38, 17-6
 - Projektierungsdaten zum Zielsystem transferieren 16-38
 - Projektsicht 6-2
 - Projektsprache 6-5, 6-6, 6-7
 - Projektstruktur 6-2
 - projektweit 16-11
 - Prorammierschritte
 - S7 1-1
 - Prozess
 - untergliedern 3-2, A-80
 - Prozess zerlegen in Aufgaben und Bereiche 3-2
 - Prozess zerlegen in Aufgabenbereiche für das Beispiel eines industriellen Mischprozesses 3-2
 - Prozessabbild 4-11, A-21
 - aktualisieren 4-14
 - Ein-/Ausgänge A-22
 - löschen 4-35
 - Prozessalarm
 - Priorität 4-33
 - Regeln 4-33
 - starten 4-33
 - Prozessalarme 4-33
 - Prozessalarmfähige Signalbaugruppen parametrieren 4-33
 - Prozessbetrieb 4-14, 4-16
 - Prozessüberwachung 17-1, 20-2
 - Prüfen
 - Konsistenz in AWL-Quellen 13-19
 - Referenzdaten 14-11
 - PZF (Peripheriezugriffsfehler) A-21
- Q**
- QRY_TINT 4-29
 - Quellcodes vorhandener Bausteine in AWL-Quellen einfügen 13-16
 - Quelldateien in S7-GRAPH 9-9
 - Quellen
 - Einfügen von externen Quellen 13-16
 - exportieren 13-18
 - externe 6-13
 - Generieren von AWL-Quellen aus Bausteinen 13-17
 - importieren 13-17
 - Regeln für Variablendeklaration in AWL-Quellen 13-3
 - Regeln zum Festlegen von Bausteineigenschaften in AWL-Quellen 13-5
 - Regeln zum Festlegen von Systemattributen in AWL-Quellen 13-4
 - Regeln zur Eingabe von Anweisungen in AWL-Quellen 13-2
 - Regeln zur Reihenfolge der Bausteine in AWL-Quellen 13-4
 - Speichern von AWL-Quellen 13-19
 - Zugriffsrechte 10-4
 - Quellen-Ordner 5-21
 - Quelltext 13-15
 - Layout festlegen 13-15
 - Querverkehr (siehe Direkter Datenaustausch) 7-3
 - Querverweisliste 14-2
- R**
- RAM A-15, A-30
 - RAM-Bereich A-16, A-31
 - RDSYSST 23-18, 23-20, A-29
 - READ_CLK A-126
 - READ_RTM A-127
 - Real
 - Datentyp A-37
 - Realzahl
 - Bereich A-34
 - Datentyp A-34
 - Redundanter Betrieb 18-10
 - Referenzdaten 14-1
 - Anwendung 14-1
 - anzeigen 14-10, 14-11
 - erzeugen 14-11
 - Regeln
 - für den Umgang mit License Keys 2-5
 - für die Bildung von Multiinstanzen 10-10
 - für die Eingabe von AWL-Anweisungen 10-25
 - für die Eingabe von FUP-Anweisungen 10-23
 - für die Eingabe von KOP-Anweisungen 10-18
 - für FUP 10-23
 - für KOP 10-18
 - für Variablendeklaration in AWL-Quellen 13-3
 - Prozessalarm 4-33
 - Uhrzeitalarm 4-28
 - Verzögerungsalarm 4-30
 - Weckalarm 4-31

- zum Exportieren von
 - Symboltabellen 8-17
- zum Festlegen von
 - Bausteineigenschaften in AWL-Quellen 13-5
- zum Festlegen von Systemattributen in
 - AWL-Quellen 13-4
- zum Importieren von
 - Symboltabellen 8-17
- zur Eingabe von Anweisungen in
 - AWL-Quellen 13-2
- zur Reihenfolge der Bausteine in
 - AWL-Quellen 13-4
- Regeln für den Umgang mit
 - License Keys 2-5
- Registerdialog 5-25
- Remanente Speicherbereiche in
 - S7-300-CPU's A-30
- Remanente Speicherbereiche in
 - S7-400-CPU's A-31
- Remanenter Speicher
 - bei S7-300-CPU's A-30
 - bei S7-400-CPU's A-31
- Remanenz
 - nach Netzausfall A-5
- Reorganisieren von Projekten und Bibliotheken 26-2
- Restzyklus A-6, A-9, A-10, A-12
- Ringpuffer (Diagnosepuffer) A-28
- roter Faden 1-1
- RPL_VAL 23-31
- Rücksetzen
 - Datenwerte auf die Anfangswerte 11-10
- RUN A-13
 - Betriebszustand der CPU A-1
 - CPU-Tätigkeiten A-5
- Runtime Software 1-17

- S**
- S5TIME
 - Bereich A-34
 - Datentyp A-42
 - Format A-42
 - Zeitbasis A-42
- S7/M7-Programm ohne Station und CPU
 - 5-22
- S7-CFC 9-12
- S7-Exportdatei 6-27
- S7-GRAPH 9-9
- S7-HiGraph 9-10
- S7-Programm
 - einfügen 6-13
- S7-Quellen
 - bearbeiten 13-14
- S7-Routing 18-10
- S7-SCL 9-8
- Sachschaden verhüten 20-21
- SCAN-Meldung
 - siehe Symbolbezogene Meldung (projektweit) 16-18
- SCAN-Meldung (CPU-weit)
 - siehe Symbolbezogene Meldung 16-26
- Schachtelungstiefe 4-9
- Schaltbücher drucken 24-1
- Schnellansicht der Diagnoseinformation
 - 23-5
- Schnelles Positionieren auf
 - Verwendungsstellen im Programm 14-12
- Schreiben in einen Datenbaustein im Ladespeicher 6-25
- Schritte
 - zum Anzeigen und Ändern des Betriebszustands 18-8
 - zum Erstellen und Öffnen einer Variablen-tabelle 20-3
 - zum Komprimieren des Speicherinhalts einer S7-CPU 19-21
- SCL 9-4
- sdf 6-27
- Seitenformat
 - einstellen 24-3
- Senden
 - eigener Diagnosemeldungen 23-22
- SET_CLK 4-29, A-126
- SET_CLKS 18-9
- SET_RTM A-127
- SET_TINT 4-28, 4-29
- Setup
 - Flash-File-System 2-10
 - Ident-Nummer eintragen 2-8
 - Memory Card-Parametrierung 2-9, 2-10
- SFB 4-25, A-43
- SFB 20 STOP 4-11
- SFB 33 16-7
- SFB 34 16-7
- SFB 35 16-6, 16-7
- SFB 36 16-6, 16-7
- SFB 37 16-7
- SFC 4-25
 - einsetzen A-21
- SFC 0 SET_CLK 4-28, A-126
- SFC 1 READ_CLK A-126
- SFC 2 SET_RTM A-126
- SFC 3 CTRL_RTM A-126
- SFC 4 READ_RTM A-126
- SFC 100 'SET_CLKS' 18-9
- SFC 13 DPNRM_DG A-122
- SFC 14 DPRD_DAT A-122
- SFC 15 DPWR_DAT A-122

- SFC 17/18 16-6
- SFC 20 BLKMOV A-16
- SFC 22 CREAT_DB A-16
- SFC 26 UPDAT_PI 4-11, A-21
- SFC 27 UPDAT_PO 4-14, A-21
- SFC 28 SET_TINT 4-28
 - Beispiel in AWL A-99
- SFC 29 CAN_TINT 4-28
 - Beispiel in AWL A-99
- SFC 30 ACT_TINT 4-28
 - Beispiel in AWL A-99
- SFC 31 QRY_TINT 4-28
 - Beispiel in AWL A-99
- SFC 32 SRT_DINT 4-30
 - Beispiel in AWL A-106
- SFC 33 CAN_DINT
 - Beispiel in AWL A-106
- SFC 34 QRY_DINT
 - Beispiel in AWL A-106
- SFC 36 MSK_FLT 4-40
 - Beispiel in AWL A-112
 - Beispiel in KOP A-112
- SFC 37 DMSK_FLT 4-40
 - Beispiel in AWL A-112
 - Beispiel in KOP A-112
- SFC 38 READ_ERR
 - Beispiel in AWL A-112
 - Beispiel in KOP A-112
- SFC 39 DIS_IRT 4-40
 - Beispiel in AWL A-116
- SFC 40 EN_IRT 4-40
 - Beispiel in AWL A-116
- SFC 41 DIS_AIRT 4-38
 - Beispiel in AWL A-117
- SFC 42 EN_AIRT 4-40
 - Beispiel in AWL A-117
- SFC 44 RPL_VAL 23-31
- SFC 46 STP 4-12
- SFC 48 SNC_RTCB A-126
- SFC 51 RDSYSST 23-18, 23-19, A-28
- SFC 52 WR_USMSG 23-22
- SFC 55 WR_PARM A-119, A-123
- SFC 56 WR_DPARM A-119, A-123
- SFC 57 PARM_MOD A-119, A-123
- SFC 82 6-25
- SFC 83 6-25
- SFC 84 6-25
- Sicherheitsanforderungen
 - Beschreibung für das Beispiel eines industriellen Mischprozesses 3-10
- Sicherheitshinweise
 - Überschreiten des L-Stack A-25
- Sicherheitsvorkehrungen zum Forcen von Variablen 20-20
- Signalbaugruppe simulieren 22-1
- SIMATIC Manager 5-1, 9-19, 9-21
 - Baustein-Längen anzeigen 9-18
- SIMATIC PC - Konfigurationen aus Vorgänger-Versionen nutzen 7-4
- SIMATIC PC-Station 7-4
- SIMATIC-Komponenten für die Meldungsprojektierung 16-5
- Simulation einer CPU- oder Signal-Baugruppe 22-1
- Simulationsprogramm 22-1
- Sitzungsgedächtnis 5-34
- Slaves mit fehlenden oder fehlerhaften GSD-Dateien A-77
- SlotPLC 6-26
- SNC_RTCB A-127
- Software PLC 6-26
- Softwarepakete 6-16
- Soll-Ist-Baugruppenüberwachung
 - Anlauf-OBs 4-34
- Soll-Ist-Vergleich A-5
- Sommer/Winterzeit 18-9
- Sortieren
 - in der Querverweisliste 14-3
 - Symbole 8-14
- Spannungslos
 - Betriebszustand der CPU A-1
- Speicher A-32
 - konfigurierbar A-32
- Speicherbereiche A-15
 - Adressbereiche A-20
 - Arbeitsspeicher A-15
 - Besonderheiten bei S7-300 A-16
 - Besonderheiten bei S7-400 A-16
 - Ladespeicher A-15
 - remanenter Speicher A-30
 - Systemspeicher A-15
- Speicherengpass beheben 19-20
- Speichern
 - Anwendungsfälle 24-5
 - AWL-Quellen 13-19
 - Bausteine 10-28
 - Datenbausteine 11-11
 - Fensteranordnung 5-35
 - geladener Bausteine auf integriertem EPROM 19-7
 - Variablentabelle 20-4
 - von Codebausteinen 10-28
- Speichern unter 6-27
- Sperrern von Alarm- und Asynchronfehlerereignissen
 - Beispiel A-116
- Sprache für Anzeige 16-32
- Spracheditor
 - starten 9-3

- Spulen
 - platzieren 10-18
- SRT_DINT 4-30
- Stack-Inhalte im Betriebszustand STOP 23-15
- Standardbibliothek 6-13
- Standardbibliotheken
 - Übersicht 9-25
- Starten
 - Prozessalarm 4-33
 - STEP 7 mit vorgegebenen Startparametern 5-3
 - STEP 7-Installation 2-8
 - STEP7 5-1
 - Uhrzeitalarm 4-28, 4-29
 - Verzögerungsalarm 4-30
 - Weckalarm 4-31
- Startereignisse
 - Anlauf-OBs 4-34
 - maskieren 4-40
 - verzögern 4-40
- STAT (Variablendeklaration) A-67
- Station 5-11
 - einfügen 6-11, 6-12
 - laden in PG 19-15
- Statuszeile
 - Beispiel 5-24
- Stellvertreter-Baugruppe 7-6, 7-7
- STEP 7
 - aufrufen der Software 5-1
 - Basissoftware 1-6
 - Benutzungsoberfläche 5-24
 - deinstallieren 2-13
 - Fehler während der Installation 2-10
 - Fehler-OBs
 - Reagieren auf Fehler 4-38
 - Installation 2-6
 - Programmiersprachen 1-9
- Steuern
 - Prinzipielle Vorgehensweise 20-2
- Steuern von Variablen
 - Einführung 20-17
 - Trigger festlegen 20-18
- Steuerwerte
 - Beispiele für die Eingabe 20-11
 - einfügen 20-8
- STOP A-4
 - Betriebszustand der CPU A-1
- STOP-Ursache ermitteln 23-15
- Störungen eingrenzen 23-1
- Störungsbehandlung 23-24
- STRING A-43
- Stromfluss 10-21
- Stromversorgungsfehler (OB 81) 23-36
- STRUCT A-43
- Struktur
 - des Anwenderprogramms
 - "Uhrzeitalarme" A-99
 - erstellen A-50
 - Ladespeicher A-16, A-17
 - von anwenderdefinierten Datentypen in AWL-Quellen 13-9
 - von Bausteinen in AWL-Quellen 13-8
 - von Codebausteinen in AWL-Quellen 13-8
 - von Datenbausteinen in AWL-Quellen 13-9
 - Struktur des Anwenderprogramms "Verzögerungsalarms" A-106
 - Struktur und Bestandteile der Symboltabelle 8-8
 - Struktur-Datentyp (STRUCT)
 - Anzahl verschachtelter Ebenen A-45
 - Beschreibung A-49
 - Strukturiert programmieren 4-8
 - Strukturierte Datentypen A-43, A-45
 - Feld A-46, A-47, A-48
 - Verschachtelungsstrukturen und Felder A-45
 - Struktur
 - Verschachtelungsstrukturen und Felder A-45
 - Strukturiertes Programm
 - Entwerfen A-83
 - Vorteile 4-2
 - Suchfunktion für Fehler im Anweisungsteil 10-17
 - Symbol für unbekannte Baugruppen 7-6
 - Symbolbezogene Meldungen (CPU-weit)
 - zulässige Signale 16-26
 - Zuordnung zur Symboltabelle 16-26
 - Symbolbezogene Meldungen (projektweit)
 - zulässige Signale 16-18
 - Zuordnung zur Symboltabelle 16-18
 - Symbolbezogene Meldungen zuordnen und bearbeiten (CPU-weit) 16-26
 - Symbolbezogene Meldungen zuordnen und bearbeiten (projektweit) 16-18
- Symbole 8-15, 8-16
 - bei der Programmeingabe definieren 8-13
 - eingeben 8-14
 - filtern 8-14
 - global 8-3
 - Groß-/Kleinschreibung 8-15, 8-16
 - in eine Variablentabelle einfügen 20-5
 - lokal 8-3
 - nicht verwendete 14-8
 - sortieren 8-14
- Symbole der Funktionsleiste 5-24

- Symbole der Programmstruktur 14-4
 - Symbole für Objekte im
 - SIMATIC Manager 5-6
 - Symbole über mehrere Netzwerke
 - bearbeiten 26-3
 - Symbolik 8-4
 - Symbolinformation 8-2
 - Symbolische Adressierung 8-4
 - im Beispielprogramm A-84
 - Symbolischen Namen zuordnen A-84
 - Symboltabelle 8-4
 - BuB-Attribute projektieren 17-4
 - Dateiformate für den Import/Export 8-17
 - für globale Symbole 8-8
 - importieren/exportieren 8-17
 - öffnen 8-14
 - Struktur und Bestandteile 8-8
 - zulässige Adressen 8-10
 - zulässige Datentypen 8-10
 - Synchrone Fehler
 - Verwenden von OBs
 - um auf Fehler zu reagieren 4-38
 - Synchronfehlerereignisse
 - maskieren und demaskieren A-112
 - Synchronisieren
 - Uhr A-126, A-127
 - Syntax für Bausteine in
 - AWL-Quellen 13-10
 - Systemarchitektur
 - Betriebszustände der CPU A-1
 - Zyklus 4-11
 - Systemattribute
 - für die Meldungsprojektierung 16-8
 - für die PCS 7-Meldungsprojektierung (CPU-weit) 16-24
 - für die PCS 7-Meldungsprojektierung (projektweit) 16-16
 - für Parameter 10-6
 - in der Symboltabelle 8-8, 8-9
 - Systemdaten 23-20
 - Systemdiagnose
 - erweitern 23-22
 - Systemfehler 23-23
 - Systemfehler melden 16-43, 16-45, 16-48, 16-49
 - Bausteine generieren 16-49
 - Einstellungen 16-48
 - erzeugter FB
 - DB 16-49
 - Systemfehler melden
 - unterstützte Komponenten 16-45
 - Systemfunktionen 4-2, 4-25
 - Arten 4-25
 - Systemfunktionsbausteine 4-2, 4-25
 - Arten 4-25
 - Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC) 4-25
 - Systemparameter A-122
 - Systemspeicher A-15
 - System-Textbibliotheken 16-36
 - Systemzustandsliste 23-19, 23-20, 23-21
 - Inhalt 23-19
 - lesen 23-20
 - Systemzustandsliste SZL 23-19
 - SZL 23-19, 23-20
- T**
- Taktmerker A-128
 - Tastenkombinationen
 - für das Bewegen des Cursors 5-38
 - für das Markieren von Texten 5-40
 - für das Umschalten zwischen Fenstertypen 5-41
 - für den Zugriff auf Online-Hilfe 5-40
 - für Menübefehle 5-37
 - Tauschen von Baugruppen 26-1
 - Teilnehmer am Subnetz ermitteln 18-2
 - Teilprozessabbild A-22, A-23
 - aktualisieren mit SFCs A-23
 - systemseitig aktualisieren A-23
 - TEMP (Variablendeklaration) A-67
 - Temporäre Variablen A-68
 - Test
 - Betriebsart festlegen 21-10
 - Testen
 - mit der Variablen-tabelle 20-1
 - mit Programmstatus 21-1
 - mit Simulationsprogramm (Optionspaket) 22-1
 - Testen im Einzelschrittmodus 21-5, 21-6
 - Testen mit der Variablen-tabelle 26-3
 - Textbibliothek 16-30, 16-31
 - Texte in Meldungen integrieren 16-30
 - Textbibliotheken 16-36
 - übersetzen 16-36, 16-37
 - Texte mehrsprachig verwalten 6-16
 - Textlisten
 - siehe Listen bedienerrelevanter Texte 16-32
 - Texttypen mehrsprachig verwalteter Texte 6-18
 - TIMER A-54, A-55
 - Parametertyp A-54
 - Tipps und Tricks 26-1, 26-2, 26-3, 26-6
 - Titel
 - zu Bausteinen 10-14
 - zu Netzwerken 10-14
 - Titelleiste 5-24
 - TPA (siehe Teilprozessabbild) A-21

- Transferieren der Projektierungsdaten zum BuB-Zielsystem 17-6
- Trigger zum Beobachten von Variablen festlegen 20-15
- Trigger zum Steuern von Variablen festlegen 20-18
- Triggerbedingung 20-15
- Triggerpunkte einstellen 20-15
- Typdatei 7-1

- U**
- Übergabe an IN_OUT-Parameter eines FB A-74
- Übergänge
 - von Betriebszuständen A-1
- Übergeben von Parametern
 - Beispiel-FB für das Beispiel eines industriellen Mischprozesses A-86
 - Entwerfen von Parametern für ein strukturiertes Programm A-86
 - Parametertypen A-54
 - Speichern der übergebenen Werte 4-18
- Übermittlung von Diagnoseinformationen 23-18
- Überschreibmodus 10-17
- Überschreiten des L-Stack A-25
- Übersetzen
 - AWL-Quellen 13-20
- Übersetzen und Bearbeiten
 - von bedienerrelevanten Texten 16-32
- Übersetzen und Laden 19-9
- Übersetzen von Objekten 19-11
- Übersicht
 - der möglichen Referenzdaten 14-1
 - Meldebausteine 16-6
 - Standardbibliotheken 9-25
- Überwachungszeiten 4-34
- UDT 9-14, A-43
 - Korrigieren der Schnittstelle 15-6
- UDTs
 - Struktur eingeben 11-8
- UDTs in AWL-Quellen
 - Beispiel 13-29
- Uhr
 - parametrieren A-127
 - synchronisieren A-127
- Uhrzeit
 - ändern 4-28
 - lesen A-126
 - stellen A-126, A-127
- Uhrzeit (TIME OF DAY)
 - Bereich A-34
- Uhrzeit stellen 18-9
- Uhrzeitalarm 18-9
 - abfragen 4-29
 - deaktivieren 4-29
 - Priorität 4-29
 - Regeln 4-28
 - starten 4-28
 - Uhrzeit ändern 4-28
- Uhrzeitalarme 4-28, 4-29
 - Hantierung A-99
 - Struktur A-99
- Uhrzeitformat A-126
- Uhrzeitfunktionen A-126
- Uhrzeitstatus 18-9, 18-10
- Uhrzeitsynchronisation 18-10
- Umbenennen
 - Projekte 5-27
- Umschalten zwischen Fenstertypen 5-41
- Umsetzen A-77
 - Version 1-Projekte A-75
 - Version 2-Projekte A-76
- Umverdrahten
 - Bausteine 9-22
 - Operanden 9-22
- Ungepufferter Betrieb
 - remanente Speicherbereiche A-31
- Unterbrechungs-Stack A-15, A-26
- Unterbrechungszeit A-5
- Untergliedern des Prozesses am Beispiel eines industriellen Mischprozesses 3-2
- Unterschied zwischen Speichern und Laden von Bausteinen 19-2
- Unterschiede zwischen Forcen und Steuern von Variablen 20-23
- Unterschiede zwischen projektweiter und CPU-weiter Vergabe von Meldenummern 16-11
- Unterstützte Komponenten und Funktionsumfang 16-45
- Unvollständige und mehrdeutige Symbole in der Symboltabelle 8-11
- Unzulässige Verschaltungen in KOP 10-21
- UPDAT_PI 4-14, A-21
- UPDAT_PO 4-11, A-21
- Update (Betriebssystem der CPU) A-125
- Urlöschen A-4
 - der CPU 19-18
- U-Stack
 - Beschreibung A-26
 - Verwendung durch den Systemspeicher A-26

V

- Variablen
 - bedienen und beobachten 17-1
 - beobachten 20-15
 - steuern 20-17
- Variablen vom Programmierer
 - aus steuern 26-5
- Variablendeklaration
 - FB für das Beispiel eines industriellen Mischprozesses A-86
 - FC für das Beispiel eines industriellen Mischprozesses A-90
 - OB für das Beispiel eines industriellen Mischprozesses A-92
- Variablendeklaration in AWL-Quellen
 - Beispiele 13-22
- Variablendeklarationsfenster
 - Multiinstanz eingeben 10-10
- Variablendeklarationstabelle 10-3, 10-6
 - Aufgabe 10-6
 - für OB81 23-26
 - Systemattribute für Parameter 10-7
- Variablendetailsicht 10-7
 - Struktur 10-8
- Variablentabelle 20-4
 - bearbeiten 20-7
 - Beispiel 20-6
 - Beispiel für die Eingabe von Operanden 20-10
 - Bereich zusammenhängender Operanden einfügen 20-7
 - erstellen und öffnen 20-3
 - kopieren/verschieben 20-4
 - Maximale Größe einer 20-5
 - Nutzen 20-1
 - Operanden oder Symbole einfügen 20-5
 - speichern 20-1, 20-4
 - Syntaxprüfung 20-7
- Ventile
 - Eingangs-/Ausgangsdiagramm erstellen 3-9
- Verbindung
 - zur CPU herstellen 20-14
- Vergabe von Meldenummern 16-10
- Vergleichen von Bausteinen 9-19
- Verhältnis zwischen Betriebszuständen der CPU A-1
- Vermeiden von Fehlern beim Aufrufen von Bausteinen 15-7
- Verschachtelte Aufrufe von Codebausteinen
 - Auswirkungen auf den B-Stack und den L-Stack A-27
- Verschieben
 - Objekt 5-27, 5-28, 5-29, 5-30, 5-32
- Version 1-Projekte umsetzen A-75
- Version 2 - Projekte und -Bibliotheken weiterbearbeiten 7-1
- Version 2-Projekte umsetzen A-76
- Verwalten von Anwendertexten deren Sprach-Font nicht installiert ist 6-21
- Verwenden
 - anwenderdefinierte Datentypen für den Datenzugriff A-51
 - Felder für den Datenzugriff A-46
 - Parametertyp ANY A-63
 - Parametertyp POINTER A-56
 - Strukturen für den Datenzugriff A-49
 - Systemspeicherbereiche A-19
 - Taktmerkern und Zeiten A-128
 - von älteren Projekten A-75, A-76
 - zusammengesetzte Datentypen A-45
- Verwendung der Variablendeklaration in Codebausteinen 10-6
- Verwendung von Multiinstanzen 10-9
- Verzögern
 - Startereignisse 4-38
- Verzögerte Bearbeitung von Alarm- und Asynchronfehlerereignisse
 - Beispiel A-117
- Verzögerungsalarm
 - Hantierung A-106
 - Priorität 4-30
 - Regeln 4-30
 - starten 4-30
 - Struktur A-106
- Virtuellen Arbeitsspeicher einstellen 26-6
- Voraussetzungen
 - für das Archivieren 24-6
 - für das Laden 19-1
- Voraussetzungen und Hinweise zum Laden 19-9
- Voreinstellungen für Programmierer KOP/FUP/AWL 10-4
- Vorgefertigte Bausteine 4-25
- Vorgehen beim Installieren 2-8
- Vorgehensweise
 - beim Archivieren/Deaktivieren 24-6
 - beim Eingeben von Anweisungen 10-12
 - für M7-Systeme 25-1
- Vorrang (Symbol/Absoluter Operand) 8-5

W

Warmstart A-5
Warnung
 Überschreiten des L-Stack A-25
Was ist neu in STEP 7 Version 5.4 1-11
Wechsel der Betriebszustände A-1
Weckalarm
 Regeln 4-31
 starten 4-31
Weckalarme 4-31
Wegweiser 1-1
Welche Meldebausteine gibt es? 16-6
Welche Meldeverfahren gibt es? 16-1
Wiederanlauf A-2, A-5, A-6, A-7, A-9, A-10
 abbrechen A-5
 automatisch A-6
 manuell A-5
Wiederherstellen
 Fensteranordnung Fensteranordnung
 speichern 5-35
WinAC 6-26
Windows-Spracheinstellung 6-5, 6-6, 6-7
WinLC 6-26
Winterzeit 18-9, 18-10
Wissenswertes zu Micro Memory Cards
 (MMC) 6-25
Wissenswertes zum
 Änderungsprotokoll 6-4
Wissenswertes zum
 Betriebszustand HALT 21-7
Wissenswertes zum Testen im
 Einzelschrittmodus / Haltepunkte 21-5
Wissenswertes zum Zugriffsschutz 6-3
WORD
 Datentyp A-41
Wort (WORD)
 Bereich A-34
 Datentyp A-33
WR_DPARM A-120, A-124
WR_PARM A-120, A-124
WR_USMSG 23-22

Y

Y-Link 23-14

Z

Zahlendarstellung
 binär-codierte Dezimalzahlen
 (BCD) A-41
 Bit A-33
 Byte A-33
 Datum und Uhrzeit (DATE AND TIME)
 A-44
 Doppelwort A-33
 Ganzzahl (16 Bit) A-35
 Ganzzahl (32 Bit) A-36
 Gleitpunktzahlen A-37, A-38, A-39
 S5 TIME A-42
 Wort A-33
Zähler 14-6, 14-7
 Belegungsplan 14-6
 Obergrenzen für die Eingabe 20-9
 Speicherbereich
 remanent A-30
Zeichen (CHAR)
 Bereich A-34
Zeitbasis für S5 TIME A-42
Zeiten 14-6, 14-7
 Belegungsplan 14-6
 Obergrenzen für die Eingabe 20-8
Zeiten (T) A-128
 Speicherbereich
 remanent A-30
Zeitfehler (OB 80) 23-35
Zeitfehler-OB 23-35
Zeitstempel 18-10
 als Bausteineigenschaft 15-3
 bei globalen Datenbausteinen 15-5
 bei Instanz-Datenbausteinen 15-5
 bei UDTs und von UDTs abgeleiteten
 DBs 15-6
 in Codebausteinen 15-4
Zeitstempelkonflikte 15-3
Zeitzone 18-9
Zerlegen des Prozesses in Aufgaben und
 Bereiche 3-2
Zerlegen eines Prozesses in
 Aufgabenbereiche für das Beispiel eines
 industriellen Mischprozesses 3-2
Ziehen-/Stecken-Alarm (OB 83) 23-38
Ziehen-/Stecken-Alarm-OB 23-38
Zielsystem
 Bausteine nachladen 19-6
Zugangsberechtigung 18-6
Zugriff auf den Peripherie-
 datenbereich A-119
Zugriff auf den Prozessdatenbereich A-118

- Zugriffsrechte auf Bausteine
bzw. Quellen 10-4
- Zugriffsschutz 6-3
- Zugriffsschutz einrichten 6-3
- Zugriffsschutz entfernen 6-3
- Zulässige Adressen und Datentypen in der
Symboltabelle 8-10
- Zulässige Bausteineigenschaften je
Bausteinart 13-7
- Zulässige Datentypen beim Übergeben von
Parametern A-69
- Zuordnen
symbolischen Namen A-84
- Zuordnen von Datentypen zu Lokaldaten
von Codebausteinen A-67
- Zurückladen
 - Bausteine aus S7-CPU 19-15
 - Zusammengesetzte Datentypen A-43
 - Zusammenspiel zwischen
Variablendetailsicht und Anweisungsteil
10-7
 - Zustandsgraph 9-10
 - Zyklische Programm-
bearbeitung 4-3, 4-6, 4-7
 - Zyklus 4-3, 4-4, 4-11, 4-12, 4-13
 - Zyklusbelastung durch
Kommunikation 4-15
 - Zyklusüberwachungszeit 4-13
 - Zykluszeit 4-12, 4-13, 4-14, 4-15
 - Zykluszeit zur Vermeidung von Zeitfehlern
kontrollieren 23-17

