SIEMENS

SIMATIC

FM 354 Positioning Module for Servo Drives

Manual

Appendices

6ES7 354-1AH01-8BG0
08/2008 Edition

Safety Information

This Manual contains information which you should carefully observe to ensure your own personal safety and the prevention of damage to the system. This information is highlighted by a warning triangle and presented in one of the following ways depending on the degree of risk involved:

Danger

indicates that death or severe personal injury damage **will** result if proper precautions are not taken.

Warning

indicates that death or severe personal injury damage **can** result if proper precautions are not taken.

Caution

indicates that minor personal injury can result if proper precautions are not taken.

Caution

without warning triangle means that material damage can occur if the appropriate precautions are not taken.

Attention

means that an undesired result or a condition may occur if the appropriate note is not observed.

If more than one level of hazard can occur, the warning note of the correspondingly highest level is used in all cases. If a warning note with a warning triangle warns of personal injury, an additional warning of material damage can be included in the same warning note.

Qualified personnel

The unit may only be started up and operated by **qualified personnel.** Qualified personnel as referred to in the safety guidelines in this document are those who are authorized to start up, earth and label units, systems and circuits in accordance with relevant safety standards.

Proper use

Please note the following:

Warning

The unit may be used only for the applications described in the catalog or the technical description, and only in combination with the equipment, components and devices of other manufacturers as far as this is recommended or permitted by Siemens.

It is assumed that this product be transported, stored and installed as intended and maintained and operated with care to ensure that the product functions correctly and safely.

Trademarks

All names marked with the copyright notice ® are registered trademarks of SIEMENS AG.
Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the rights of the registered holder.

Siemens AG
Automation & Drives
90437 Nürnberg
Federal Republic of Germany

Exclusion of liability

We have checked that the contents of this publication agree with the hardware and software described herein. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information given in this publication is reviewed at regular intervals and any corrections that might be necessary are made in the subsequent printings.

Preface

Information

This manual contains all information about the FM 354 module:

- Hardware and functions
- Parameterization
- Human-machine interface
- S7 blocks
- Safe design

Information blocks in this manual

The following information blocks describe the purpose and application of this manual:

- Product overview of the module (Chapter 1)

  This section explains the purpose and possible applications of the module. It provides introductory information about the FM 354 and its functions.
- Basic principles of positioning (Chapter 2)

  Here you will find introductory information on positioning methods and associated definitions of terms.
- Installing and removing the FM 354 (Chapter 3)

  Explains the installation and removal of the FM 354.
- Wiring the FM 354 (Chapter 4)

  Describes the connection and wiring of drives and digital input/output modules.
- Defining parameters of the FM 354 (Chapter 5)

  Describes the parameterization and functions of “Parameterize FM 357.”
- Programming the FM 354 (Chapter 6)

  Describes how to program the FM 357 with STEP 7.
- Starting up the FM 354 (Chapter 7)

  Describes startup procedures for the FM 354.
- Human-machine interface (Chapter 8)

  Describes the various options for operating and monitoring the FM 354, and which data and signals can be used and monitored.

- Reference information and appendices for finding factual information (module functions, programming guide, interface signals, error handling, technical specifications, standard HMI user interface)
- List of abbreviations and index for looking up information.

User requirements

The present manual describes the hardware and functions of the FM 354

To set up, program and start up a SIMATIC S7-300 with the FM 354, you will need a knowledge of:

- The SIMATIC S7

  Installation manual *S7-400/M7–400 Programmable Controller, Hardware and Installation*
- Your programming device (PG)
- How to perform programming with STEP 7
- How to configure an operator panel interface.

FM 354 users

The structure and presentation of the information in the manual are oriented to the intended uses of the FM 354, and the user's own activity.

It distinguishes among the following:

- Installation and wiring

  These activities include installation and wiring of the FM 354.
- Parameterizing and Programming

  These activities include parameterizing and programming the FM 354.
- Troubleshooting and diagnostics

  These activities include detecting and correcting faults and errors

  - in the hardware setup of the module and its components
  - and in the programming, handling and control of module functions.
- Operation

  These users operate the FM 354. The operator accordingly deals only with the control of positioning tasks.

Note

The PROFINET functionality described in the present manual (in the current version) is only available on request. Please get in touch with your Siemens contact partner.

Standards and approvals

Our products are in compliance with the EU Guideline 89/336/EEC “Electromagnetic Compatibility” and the harmonized European standards (EN) which it embodies.

The current version of the EC Declaration of Conformity can be found on the Internet at

http://support.automation.siemens.com/WW/view/de/15257461

Recycling and disposal

For recycling in an environmentally compatible manner and for the disposal of your old SIMATIC in line with prsent state of technology, please contact your appropriate Siemens contact partner:

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

Technical support

If you have any technical question, please do no hesitate to contact our hotline:

Time zone:	Europe/Africa	Asia/Australia	America
Telephone	+49 (0) 180 5050 222	+86 1064 719 990	+1 423 262 2522
Fax	+49 (0) 180 5050 223	+86 1064 747 474	+1 423 262 2289
Internet	http://www.siemens.com/automation/support-request		
E-mail	adsupport@siemens.com		

Note

The country−specific telephone numbers for technical consultation can be found on the Internet at: **http://www.siemens.com/automation/service&support**

Questions regarding this Manual

If you have any questions regarding this Documentation (suggestions, corrections), please send a fax or an e−mail to the following address:

Fax: +49 (0) 9131 98 63 315
E-mail: docu.motioncontrol@siemens.com

Siemens Internet address

For currently updated information on the SIMATIC products, vitsit us on the Internet at: **http://www.siemens.de/simatic**.

Further support

We are offering courses to help you familiarize yourself with the operation of the SIMATIC S7 programmable controller system.

Please contact your regional or the central training center in D-90027 Nürnberg, Germany under tel. +49 911-89 53 202.

■

Table of Contents

Product Overview 1

Chapter overview

What can the FM 354 do?

The FM 354 is a microprocessor-controlled positioning module for actuating a stepper motor.

The FM 354 is a high-performance module for positioning with stepper drives.

The module works autonomously and is controlled by way of the user program in the SIMATIC S7-300 system.

It can operate rotary and linear axes.

The FM 354 has a variety of operating modes.

The module has a non-volatile data memory to store parameterization data.

- The FM 354 is low-maintenance (no battery).
- It can be linked and adapted to user circumstances by parameterizing it as required by the system.

Where can the FM 354 be used?

The FM 354 can be used for both simple positioning and complex traversing profiles demanding superior dynamic response, accuracy and speed. It is also suitable for positioning tasks in machinery with high clock-pulse rates.

Typical uses for the positioning module might include:

- Transfer lines
- Assembly lines
- Woodworking machines
- Handling equipment
- Loaders
- Auxiliary movements in milling and turning machines
- Packaging machines
- Conveyor equipment

Its range of functions is comparable to that of the WF 721 module in the SIMATIC S5 system, and the FM 354 in the SIMATIC S7 system.

1.1 The FM 354 in the S7-300 programmable controller

How is the FM 354 linked up with the S7-300?

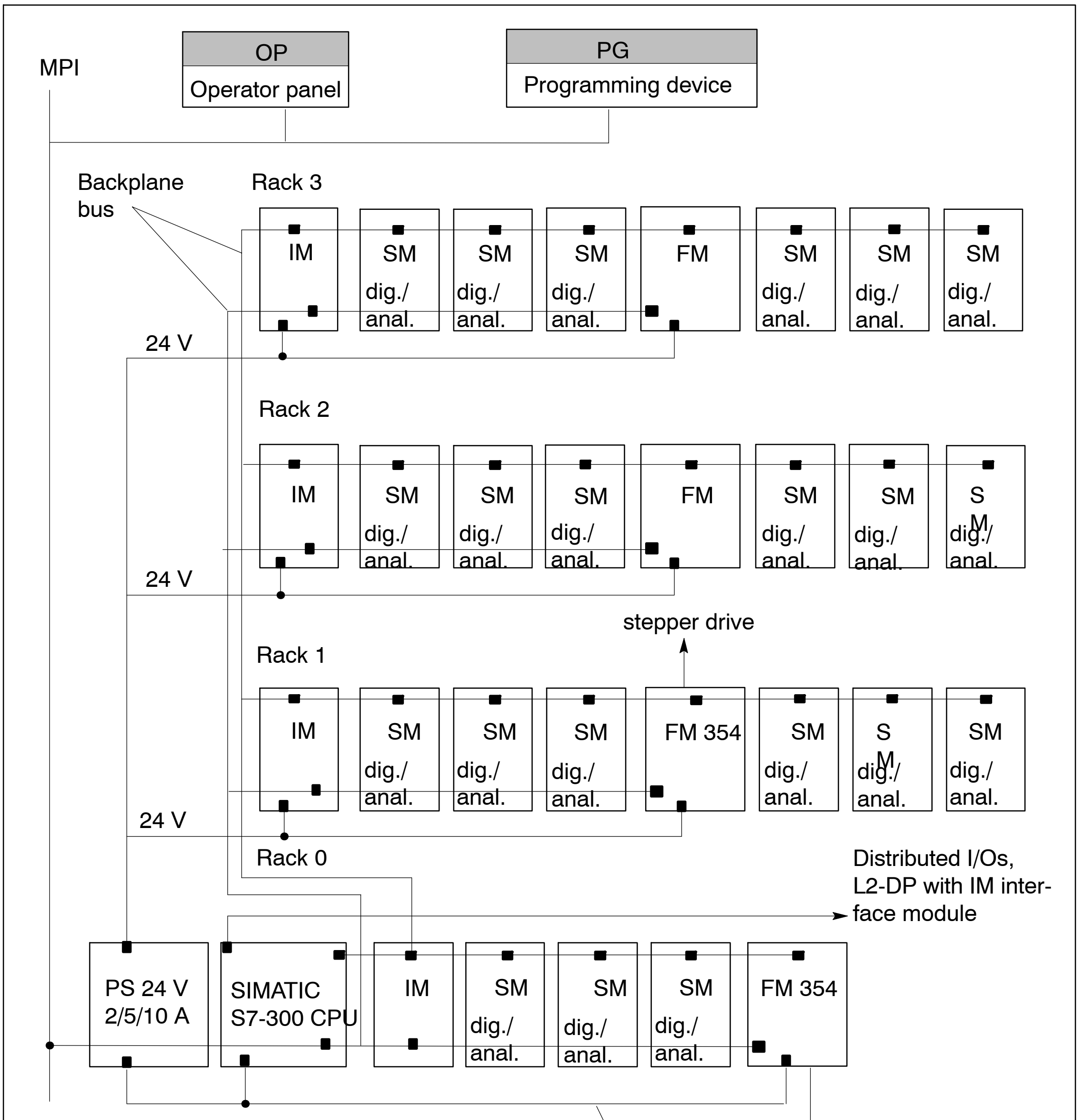
The FM 354 is designed as a function module of the SIMATIC S7-300 controller.

The S7-300 programmable controller consists of a CPU and a variety of peripheral modules mounted on a mounting rail.

The configuration may have one or more racks.

Multi-rack configurations

A SIMATIC S7-300 CPU may run up to four racks with as many as eight bus stations each (see Figure 1-1).



MPI – Multipoint interface
IM – Interface module
SM – Signal module
PS – Power supply
CPU – Central processing unit

Fig. 1-1 Multi-rack configuration of a SIMATIC S7-300 with FM 354 (example)

System overview

A complete positioning controller using the FM 354 consists of a variety of individual components, which are shown in Figure 1-2.

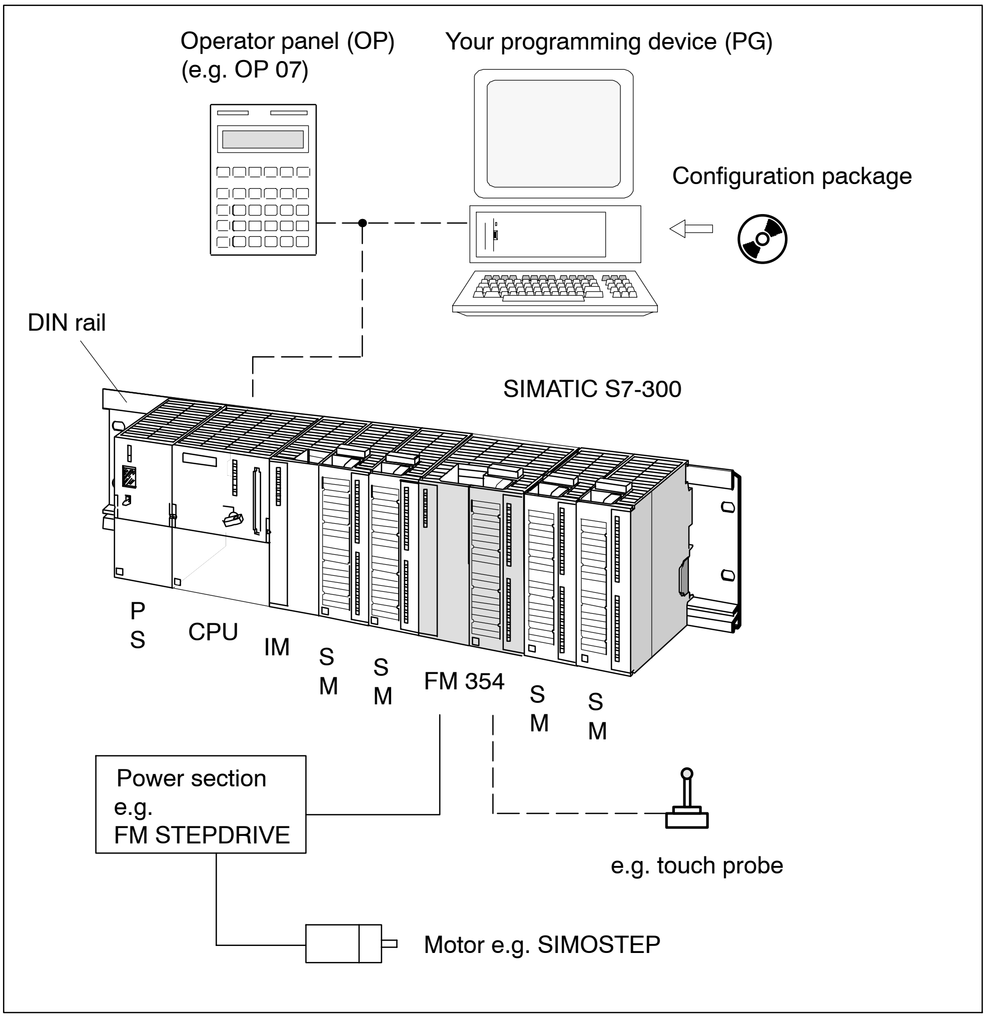


Fig. 1-2 System overview (schematic)

MPI connection

The FM can service up to 3 MPI nodes (PCs, programming devices or OPs) simultaneously.

Components

The most important components and their functions are listed in Table 1-1.

Table 1-1 Components of a positioning controller

Component	Function
DIN rail	... the module mounting rack for the S7-300.
FM 354	... the positioning module. It is controlled by the S7-300 CPU.
CPU	... executes the user program; powers the S7-300 backplane bus at 5 V; and communicates with the programming device and the operator panel via the MPI interface.
Power supply (PS)	... converts line voltage (120/230 V AC) to 24 V DC operating voltage to power the S7-300.
Signal modules (SM)	... adapts various process-signal levels to the S7-300
Interface module (IM)	... connects the individual cells of an S7-300 with one another (applies to multi-rack configuration; see Figure 1-1).
Programming device (PG)	... configures, parameterizes, programs and tests the S7-300 and the FM 354.
Operator panel (OP)	... the interface to the machine. It serves for operation and monitoring. It is not an absolute prerequisite for operation of an FM 354.
Power section	... actuates the motor.
Motor	... drives the axis.
Configuration package	... A CD-ROM containing: • An FC block package • MD DBs (for start-up of stepper motor) • The parameterization tool ”Parameterize FM 354” • A preconfigured operator interface for the COROS series OP 07 and OP 17 • A manual in PDF format • Getting Started in PDF format

System overview of data handling

The following figure gives you an overview of the data storage concept.

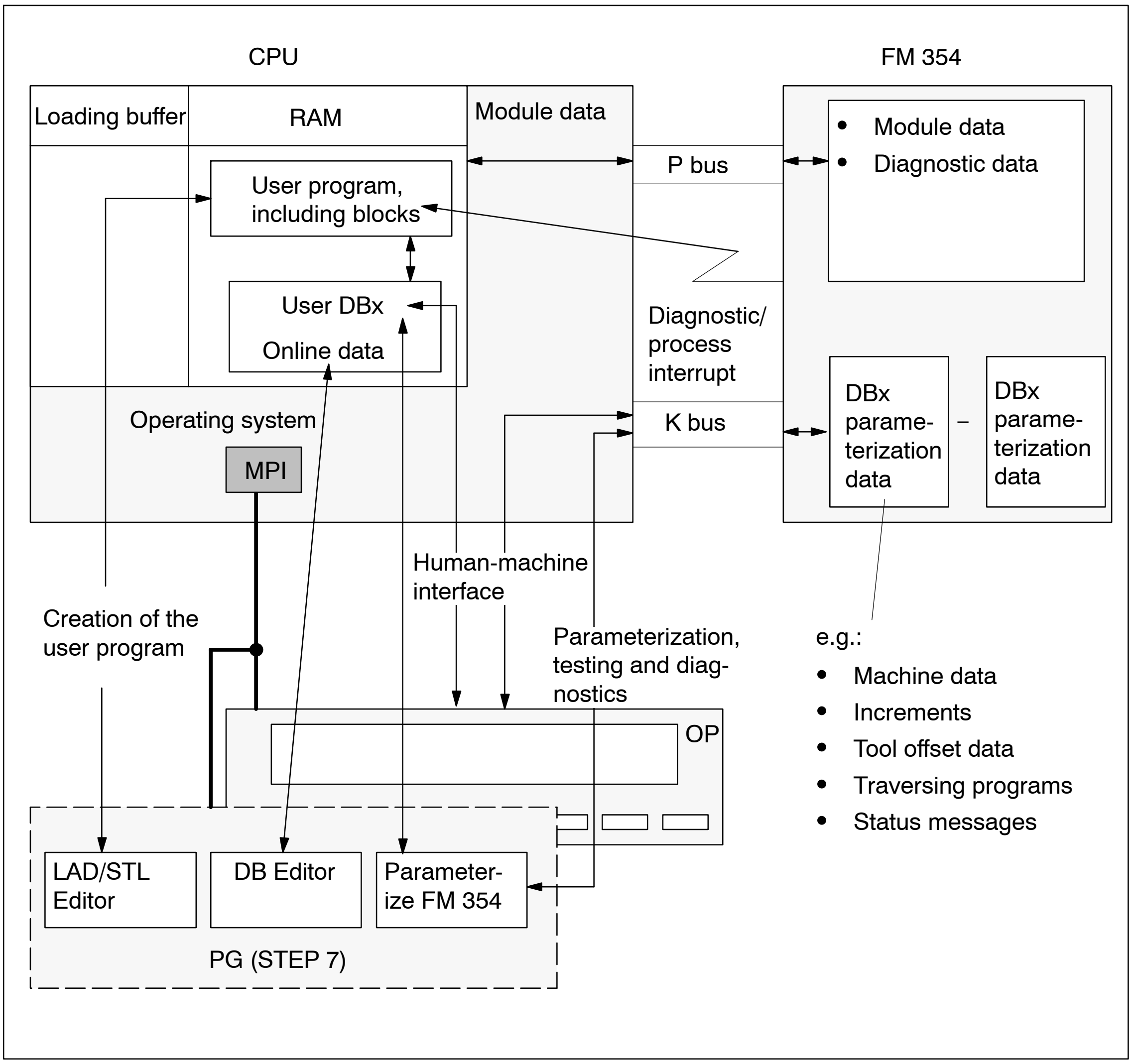


Fig. 1-3 Data storage concept

1.2 Module description

View of the FM 354

Figure 1-4 shows the FM 354 module, its interfaces and front-panel elements (including fault and status displays).

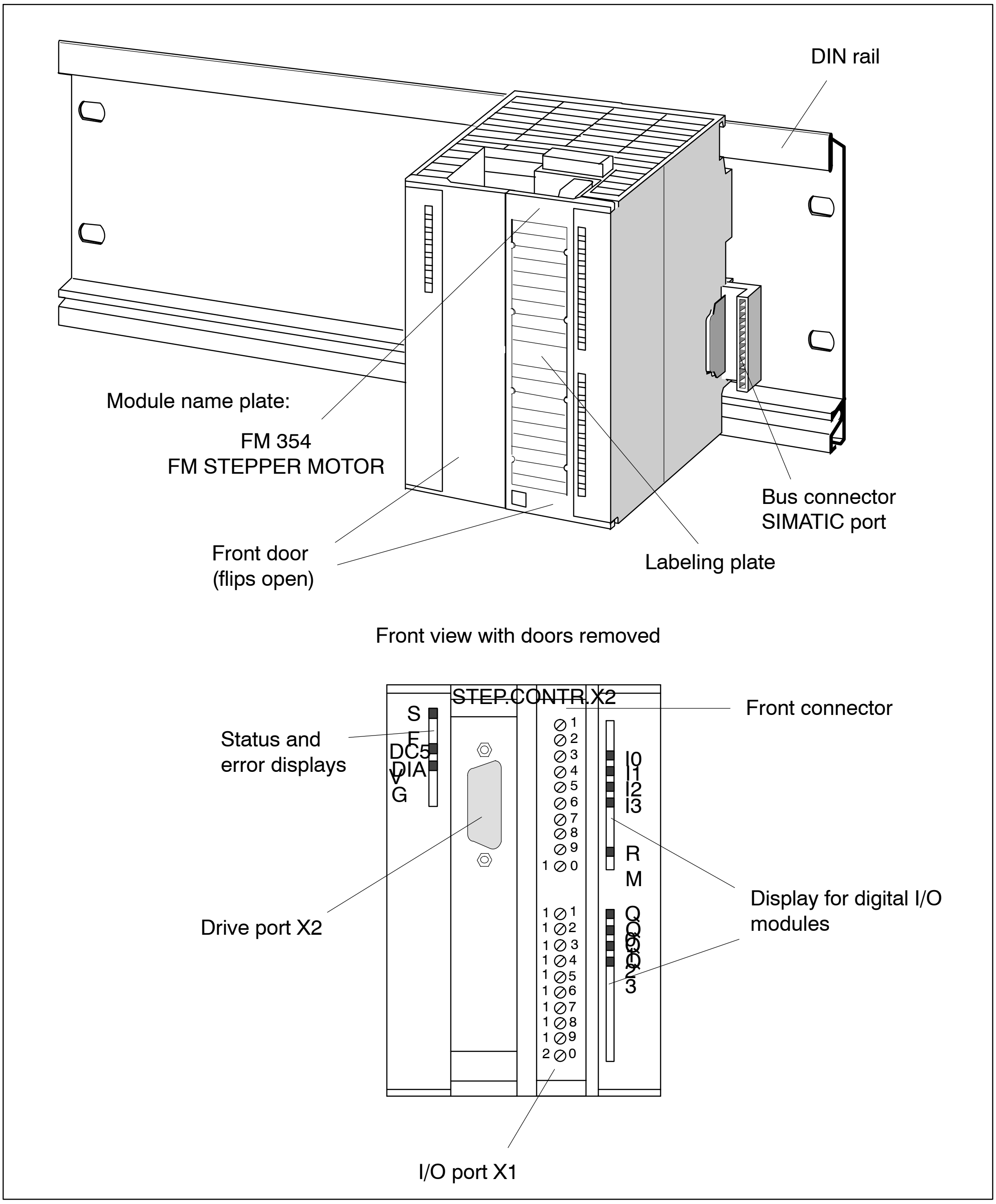


Fig. 1-4 View of the FM353

Ports

A description of the ports is provided in Table 1-2.

Table 1-2 Ports

Ports	Description
Bus connector – SIMATIC port	Back connector to continue the S7 LAN from module to module
Drive port	15-pin male sub-D connector (X2) to connect the drive unit
I/O port	20-pin male front connector (X1) to connect the load power supply and for digital input and output wiring

LED indicators

Twelve LEDs are arranged on the front panel of the FM 354. Table 1-3 describes these LEDs and what they mean.

Table 1-3 Status and error displays

LED	Significance
SF (red) - Group error	This LED indicates an error condition in the FM 354. (see Troubleshooting, Chapter 11)
5 V DC (green) - Logic power supply is ON	This LED indicates that the hardware is ready for operation.
DIAG (yellow) - Diagnostics	This LED indicates various diagnostic states (see Troubleshooting, Chapter 11)
I0 – I3 (green) - Digital inputs	These LEDs indicate which input is ON.
Q0 – Q3 (green) - Digital outputs	These LEDs indicate which output is ON.
RM (green) - Input, controller message	This LED indicates that the input is activated.

Type plate of the FM 354

Figure 1-5 describes all the information contained in the type plate of the FM 354.

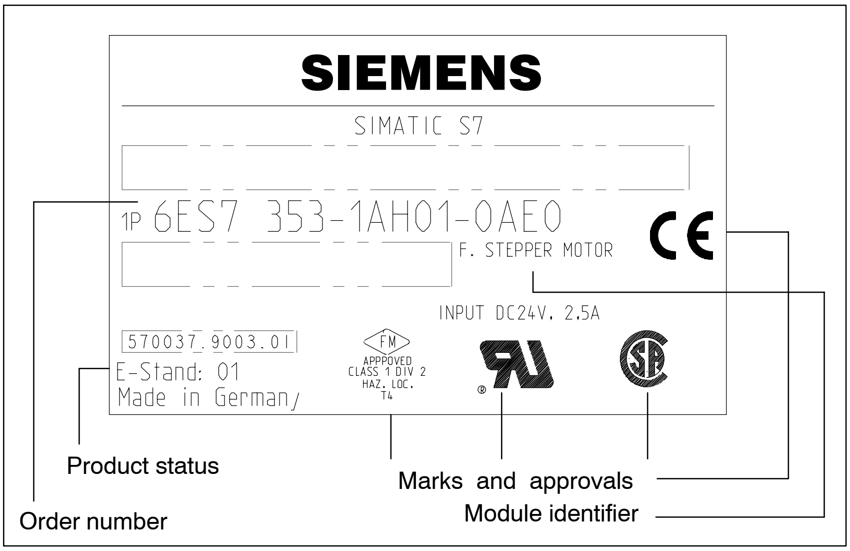


Fig. 1-5 Type plate of the FM 354

1.3 Overview of module functions

Summary

The FM 354 module performs the following functions:

- Mode control
- Determining position
- Stepper motor control
- Digital inputs and outputs
- Settings and functions that do not depend on operating mode
- Software limit switches
- Process interrupts
- Block sequence control
- Diagnostics and troubleshooting
- Data storage on the FM 354

Mode control

The operating mode is specified to the FM 354 by way of the user program.

The FM 354 has the following modes available:

- Jogging
- Open-loop control
- Reference point approach
- Incremental mode, relative
- MDI-Manual Data Input)
- Automatic
- Automatic single block

Determining position

The control frequency pulses emitted by the FM 354 are added internally to form a position actual value.

Stepper motor control

The stepper motor control performs the following tasks:

- Guidance of the drive commensurate with speed during movement sequence (e.g. adjustable acceleration and delay, start/stop operation)
- Output of step pulses in form of control frequency
- Output of a direction signal
- Accurate approach by axis into programmed target position
- Phase current control to the drive device

Digital inputs/outputs

Four digital inputs and four outputs can be used as specified by the user.

You might connect:

- Reference-point switches
- Switches for external starting
- Touch probes
- Position reached, Stop (“PEH”)
- Forward/backward rotation

The switching function is assigned to a given I/O number by way of the machine data.

Settings and functions not dependent on operating mode

Special functions can be activated by specific settings in the user program, in addition to the mode (e.g., inprocess measurement).

Software limit switches

The operating range (specified by software limit switches) is automatically monitored after synchronization is recorded.

Process interrupts

Process interrupts are triggered by such events as:

- Position reached
- Length measurement completed
- On-the-fly block change
- Inprocess measurement

Process interrupts are selected by way of machine data.

Block sequence control

Automatic processing of a traversing program, including subprograms created during the parameterization process. A number of traversing programs are available for execution on the module.

Diagnostics and troubleshooting

Startup and ongoing operation of the module are monitored by fault and diagnostic interrupts. Faults or errors are reported to the system and displayed by the LEDs on the module.

Data storage on the FM 354

Parameterization data (machine data, tool compensation data, traversing programs and increment sizes) is retained in storage on the FM 354.

■

Basic Principles of Positioning

2

What is positioning?

Positioning means moving a load to a defined position within a defined time, taking all influencing forces and torques into account.

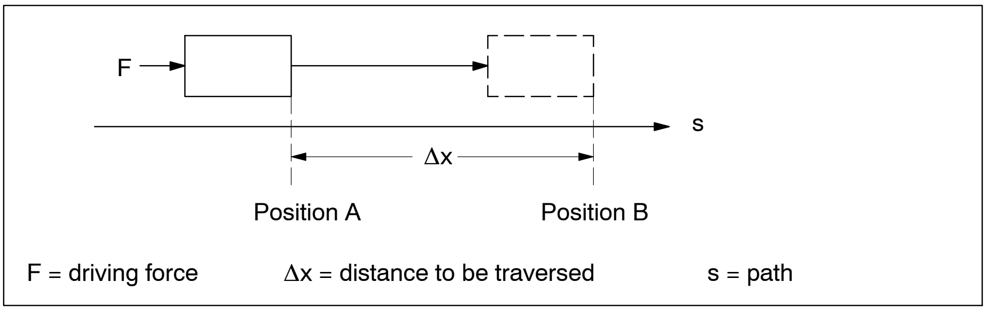


Fig. 2-1 Principle of a positioning action

Servo-controlled positioning

Servo-controlled positioning is:

- guidance of the drive at the proper speed during execution of a movement
- precise approach of the axis to the programmed target position
- maintenance of the axis in position in the face of interfering factors.

Structure of a positioning circuit

Figure 2-2 shows the structure of a position control circuit with the FM 354.

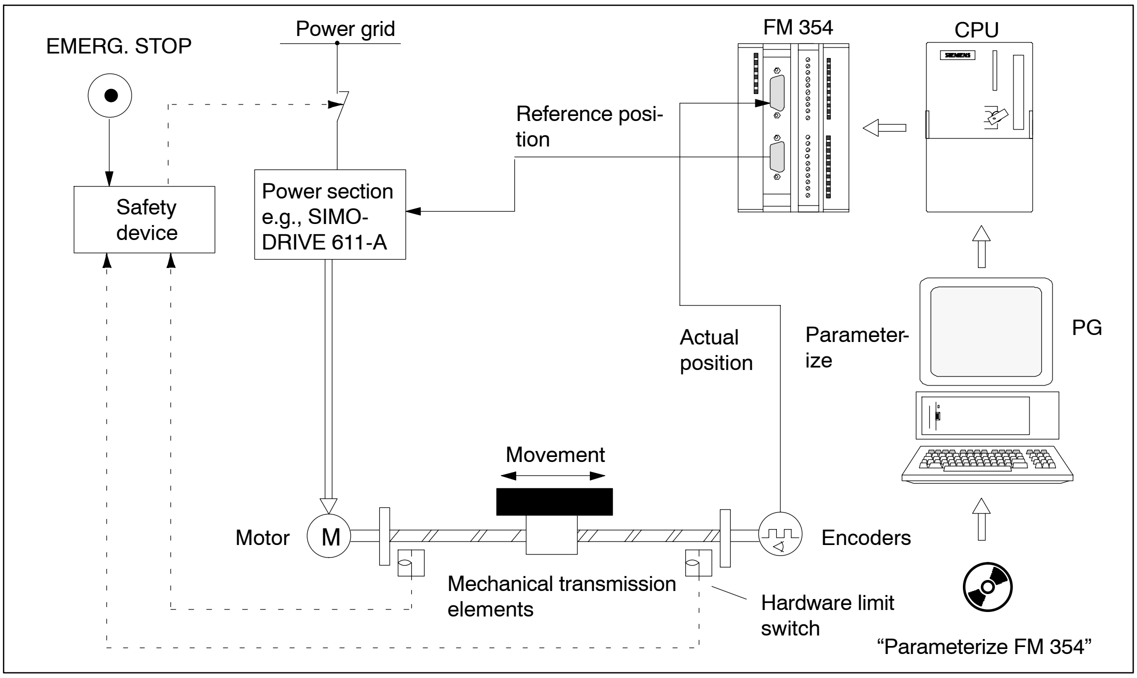


Fig. 2-2 Setup of servo-controlled positioning, example

FM 354

Servo-controlled positioning with output of an analog actuating signal for the drive.

Power section

The power section processes the analog actuating signal and delivers the proper electric power to the motor.

Motor

The motor is actuated by the power section and drives the axis.

Mechanical transmission elements

These include not only the axis, but also gear trains and clutch systems.

Encoder

The encoder detects movement of the axis. It supplies pulses to the FM 354. The number of pulses is proportional to the distance traversed

Peripherals

All other additional equipment is covered by the term peripherals.

Peripherals mainly include:

- Limit switches to limit the positioning range (safety devices).
- A programming device (PG) and the “Parameterize FM 354” parameterization software.

■

Installing and Removing 3

Chapter Overview

Overview

The FM 354 is intended for installation as an I/O module in the SIMATIC S7-300 programmable logic controller.

Important safety rules

There are important rules which you must follow when integrating an FM 354 in the S7-300 PLC in a plant or system.

These rules and specifications are described in the manual *S7-300 Programmable Controller, Hardware and Installation*.

Mechanical set-up

The options for the mechanical set-up and its configuration are described in the manual *S7-300 Programmable Controller; Hardware and Installation*.

Below, we give only a few supplementary pointers.

Installation position

The module should preferably be installed horizontally.

In vertical installations, please observe the ambient temperature restrictions (max. 40 °C).

What you should know about the mechanical layout

The FM 354 can be mounted in any of the eight available slots (slots 4 to 11) for I/O modules on the mounting rail.

In configuring the mechanical layout of your controller, you should note the following rules:

1. No more than eight SMs or FMs per tier (rack).
2. The maximum number of modules is limited by module width and by the length of your DIN rail.

   The FM 354 requires an installation width of 80 mm (3.12 inches).
3. The maximum number of modules is limited by the total power that all modules to the left of the CPU or IM, as the case may be, consume from the 5 V backplane bus.

   The CPU 314, for example, can supply a maximum of 1.2 A.

   The FM 354 requires 100 mA of this amount.

3.1 Installing the FM 354

Rules

No particular protective measures (EGB Guidelines) are necessary for the installation of the FM 354.

Warning

Install the FM 354 only after all power to the S7-300 has been turned OFF.

Tools required

A 4.5 mm (.18 inch) screwdriver.

Procedure

To install the FM 354:

1. The FM 354 comes with a bus connector. Plug this into the bus plug of the module to the left of the FM 354. (The bus plug is on the back; you may have to loosen the module already in place.)

   If further modules are to be mounted to the right, plug the bus connector of the next module into the right backplane bus connector on the FM 354.

   If the FM 354 is the last module in the rack, do not connect this bus connector.

2. Hook the FM 354 onto the rail and swing it down into position.
3. Screw the FM 354 down (torque approx. 80-110 Nm).
4. After the modules have been mounted, you can also assign each of them a slot number. Slot labels for this purpose are enclosed with the CPU.

   The numbering scheme and how to plug in the slot labels are described in the manual *S7-300 Programmable Controller, Hardware and Installation*, for the numbering scheme to follow and how to apply the slot labels.

Note

The slot determines the initial address of each module. To find out how to allocate the module start address, please refer to the manual *S7-300 Programmable Controller, Hardware and Installation*.

The FM 354 is addressed in the same way as an analog module.

3.2 Removing the FM 354

Rules

No particular protective measures (EGB Guidelines) are necessary for the removal of the FM 354.

Warning

Remove the FM 354 only after all power to the S7-300 has been turned OFF.

Tools required

A 4.5 mm (18 inch) screwdriver.

Procedure

To remove the FM 354:

1. Open the front doors. If necessary, remove the labeling strips.
2. Detach the power-supply connections from the terminal block.
3. Detach the sub-D plugs from the encoder and drive unit.
4. Release the protective device on the front connector and unplug it.
5. Loosen the fastening screws and swing the module up and out.

3.3 Replacing modules

Overview

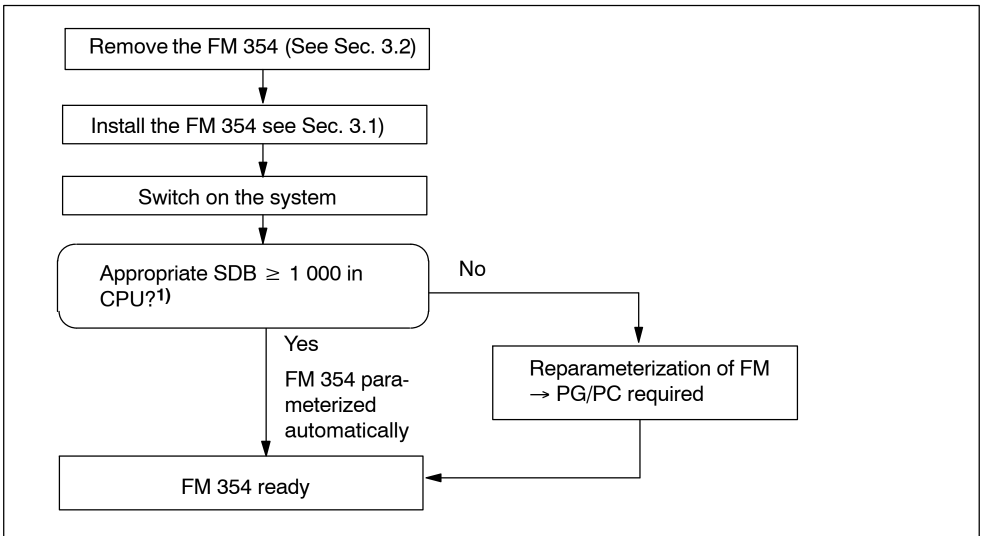
If a defective FM 354 has to be replaced, and no programming device/PC is available for parameterization, or the module is to be replaced while the system is switched on, please note the following start-up requirements (CPU, FM):

- An SDB ≥ 1 000 should be generated in order to complete the startup (for storing the parameter data); see Section 5.5.
- In the user program:
  - Interrupt communication with the FM 354 before removing the old FM, and resume communication after installing the new FM.
  - If data/parameters are modified during operation and stored modally on the FM, please follow the instructions in Section 9.3.1.

Replacing an FM 354

To replace a parameterized but defective FM 354:

1. Replace the FM 354 when the system is switched off (CPU, FM)


Remove the FM 354 (See Sec. 3.2)
Install the FM 354 see Sec. 3.1)
Switch on the system
Appropriate SDB ≥ 1 000 in CPU?1)
No
Reparameterization of FM → PG/PC required
Yes
FM 354 parameterized automatically
FM 354 ready

1) How to create an SDB ≥1 000 after startup, and how to load it in the CPU is described in Section 5.5.

Fig. 3-1 Replacing the FM 354 with the system switched off

■

Wiring 4

Chapter Overview

Safety rules

In order to ensure the safe operation of your plant, you should introduce the following additional measures, and adjust them appropriately to your system's conditions:

- An EMERGENCY STOP concept meeting appropriate safety regulations (e.g. European standards EN 60204, EN 418 and associated standards).
- Additional measures for limiting the end position of axes (e.g. hardware limit switches).
- Equipment and measures for protecting the motors and power electronics in accordance with the installation guidelines for SIMODRIVE.

We also recommend you carry out a risk analysis in accordance with basic safety requirements / Appendix 1 of the EC machine directive, in order to identify sources of danger affecting the complete system.

Further references

Please refer also to the following chapters in the manual *S7-300 Programmable Controller, Hardware and Installation*:

- Lightning protection and overvoltage protection: Chapter 4.2
- Guidelines for handling of electrostatic sensitive devices (ESDs): Appendix B.
- Configuring the electrical installation: Chapter 4

For further information about EMC guidelines, we recommend the description in: *Equipment for Machine Tools, EMC guidelines for WS/WF equipment,* Order No.: 6ZB5 440-0QX01-0BA1.

Standards and specifications

When wiring the FM 354 you must observe the relevant VDE guidelines.

4.1 Wiring an FM 354

Summary

Figure 4-1 shows how the individual components of the positioning controller with FM 354 are linked together.

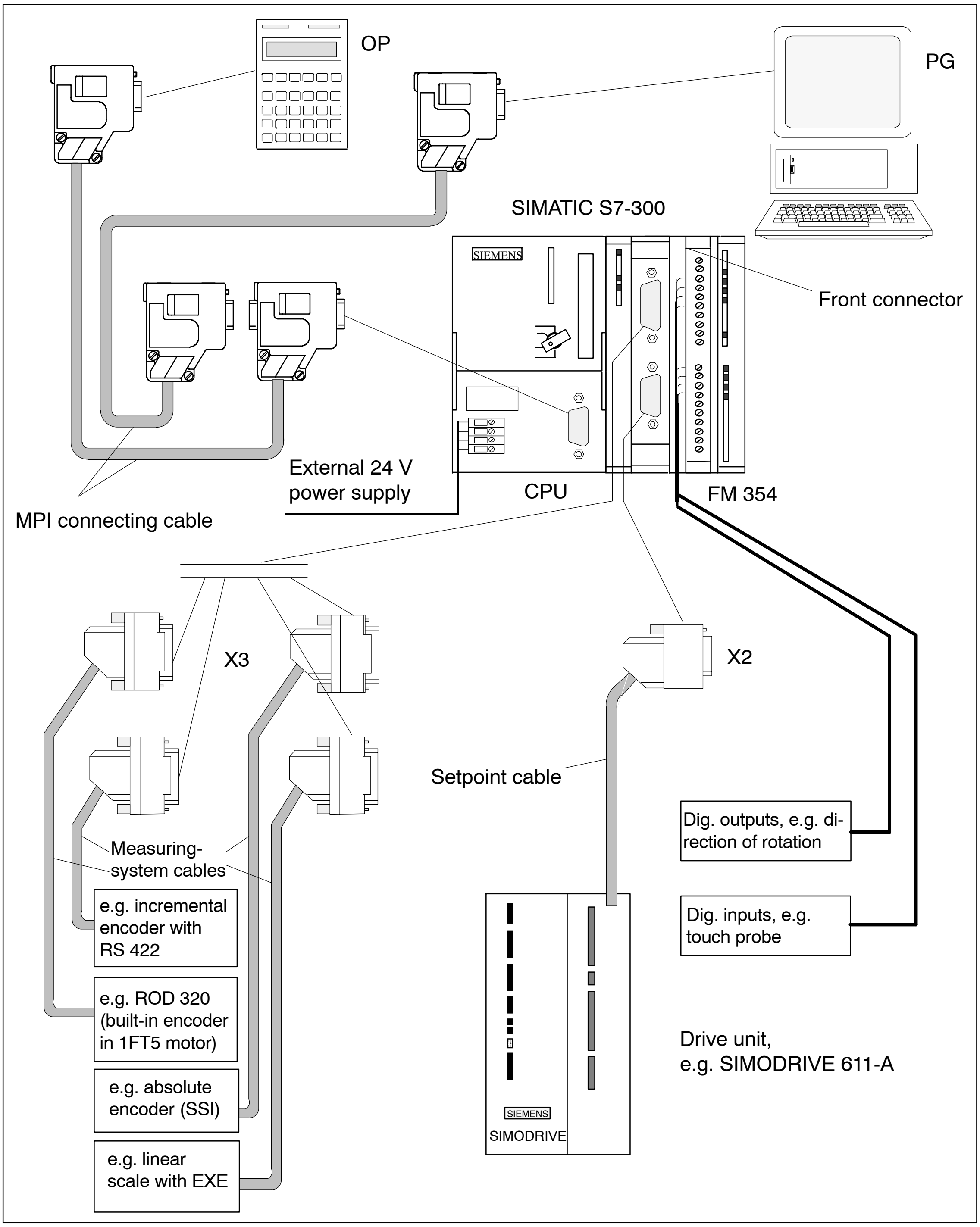

OP
PG
SIMATIC S7-300
SIEMENS
Front connector
External 24 V power supply
CPU
FM 354
MPI connecting cable
X3
X2
Setpoint cable
Dig. outputs, e.g. direction of rotation
Dig. inputs, e.g. touch probe
Measuring-system cables
e.g. incremental encoder with RS 422
e.g. ROD 320 (built-in encoder in 1FT5 motor)
e.g. absolute encoder (SSI)
e.g. linear scale with EXE
Drive unit, e.g. SIMODRIVE 611-A
SIEMENS
SIMODRIVE

Fig. 4-1 Wiring diagram of an FM 354

Note

The device is designed for operation with safety extra–low voltage (SELV). This means that only safety extra–low voltages (SELV) complying with IEC950/EN60950/VDE0805 may be connected to the power supply terminals.
The power unit for supplying the device must comply with NEC Class 2 as described by the National Electrical Code(r) (ANSI/NFPA 70).
The power of all connected power units in total must correspond to a limited power source (LPS).

Connecting cables

Table 4-1 lists the connecting cables for a positioning controller with the FM 354.

Table 4-1 Connecting cables for a positioning controller with FM 354

Type	Order No.	Description
MPI connecting cable	see *Catalog ST 70,* Order No. E86060-K4670-A101-A□	Connection between OP, PG and S7-300 CPU
Setpoint cable	6FX2 002-3AB01-1□□0 see *Catalog NC Z* Order No.: E86060-K4490-A001-A□	Setpoint cable for FM 354 and SIMODRIVE 611-A, ±10 V; one axis
Measurement system cable	6FX2 002-2CD01-1□□0 see *Catalog NC Z* Order No.: E86060-K4490-A001-A□	Incremental encoder with RS 422 and FM 354 (EXE with linear scale)
Measuring-system cables	6FX2 002-2CE01-1□□0 see *Catalog NC Z* Order No.: E86060-K4490-A001-A□	ROD 320 encoder with 1FT5 motor and FM 354
Measuring-system cables	6FX2 002-2CC01-1□□0 see *Catalog NC Z* Order No.: E86060-K4490-A001-A□	Connection of absolute encoder (SSI) and FM 354

Front connector

You need a 20-pin screw-type front connector for wiring the I/Os. It must be ordered separately.

Order No.: 6ES7 392-1AJ00-0AA0

see *Catalog ST 70,* Order No. E86060-K4670-A101-A□

see *Catalog NC 60.1,* Order No. E86060-K4460-A101-A□

4.2 Description of the drive interface

Connector for the drive unit

Power sections with an analog interface (± 10 V) can be connected to the 9-pin sub-D X2 connector of the FM 354.

The FM 354 also provides an enable signal.

Connector position

Figure 4-2 shows the installation position and identification of the plug on the module.

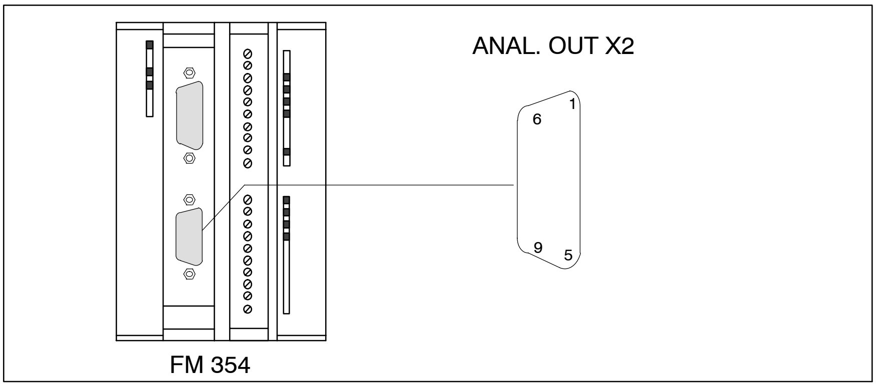


Fig. 4-2 Position of X2 connector

Connector pinout

Connector identifier:	**X2**	**ANAL. OUT X2**
Connector type:	9-pin sub-D plug connector	

Table 4-2 Pinout of the X2 connector

Pin	Name	type	Pin	Name	type
1	SW	VO	6	BS	VO
2	open		7	open	
3	open		8	open	
4	open		9	RF.1	K
5	RF.2	K			

Signal names

SW	Setpoint ± 10 V
BS	Reference potential for setpoint
RF.1...2	Contact for CL controller enable

Signal type

VO	Voltage outlet
K	Switching contact

Signals

One voltage signal and one enable signal are provided.

- **SETPOINT (SW)**

  An analog voltage signal in the range ± 10 V, for output of an rpm setpoint.
- **REFERENCE SIGNAL (BS)**

  A reference potential (analog ground) for the setpoint signal, internally connected with the logic ground.
- **SERVO ENABLE (RF)**

  A relay contact pair used to switch the axis-specific Enable signal for the power section, e.g. a SIMODRIVE drive unit. The FM 354 activates this signal when cyclic open–loop control mode is entered, that is, when startup and initialization were successfully completed and the user activated the single function “Servo Enable”. Prerequisite is, however, that MD37 is set for “Servo Enable active”.

Signal parameters

The setpoint is output as an analog differential signal.

Table 4-3 Electrical parameters of the setpoint signal

Parameters	Min	Max	Unit
Rated voltage range	–10	10	V
Output current	–3	3	mA

D/A converter resolution:	0...1 V	16 bits + sign
	1...10 V	13 bits + sign

Relay contacts

The axis enables are switched via relay outputs (“make” contacts).

Table 4-4 Electrical parameters of the relay contacts

Parameters	Max	Unit
Switching voltage	50	V
Switching current	1	A
Switching capacity	30	VA

Connecting cables to drive unit

Acceptable length: up to 35 m (115 ft)

4.3 Connecting the drive unit

To connect the connecting cables

Please note:

Note

Use only shielded twisted pairs for lines. The shielding must be connected to the metallic or metallized connector jacket on the controller side. To protect the analog setpoint signal against low-frequency interference, we recommend that you not ground the shielding on the drive-unit side

The cable set supplied as an accessory offers excellent immunity against interference

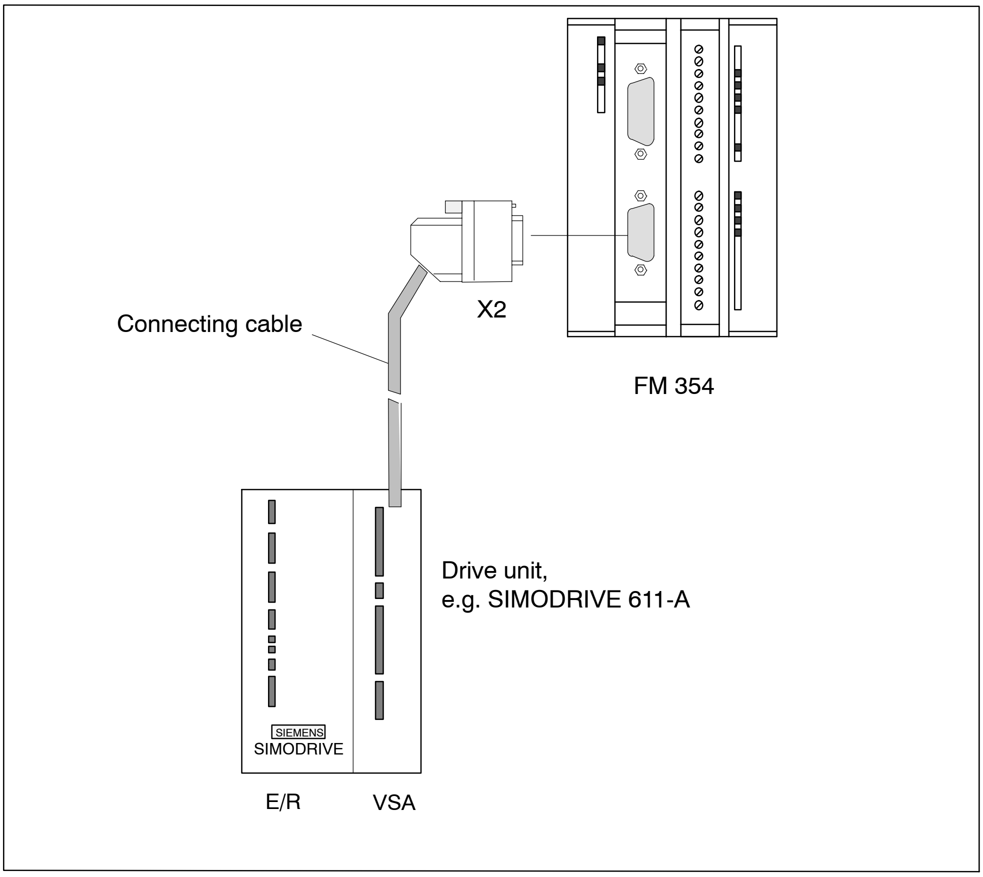


Fig. 4-3 Connecting a SIMODRIVE 611-A drive unit

Procedure to connect the connecting cable

Connect the drive unit as follows:

1. Wire the free cable end of the connecting cable to the terminals of the drive unit. (The terminal identifiers on the cable ends indicate the proper terminals for SIMODRIVE units.)
2. Open the front door and plug the sub-D connector into the module.
3. Lock the connector in place with the knurled screws. Close the front door.

Designation of the connecting cable

The connecting cable is a prefabricated cable for an axis with an analog interface, terminal designation for SIMODRIVE drive units.

Order No.: 6FX2 002-3AB01-0□□0

The connecting cable is available in a variety of lengths.

see *Catalog NC Z*, Order No.: E86060-K4490-A001-A□.

4.4 Description of the measurement system interface

Connectors for encoders

A 15-pin female sub D connector is provided for the connection of incremental encoders or absolute encoders (serial port).

Location of connector

Figure 4-4 shows where the connector is installed on the module, and how it is identified.

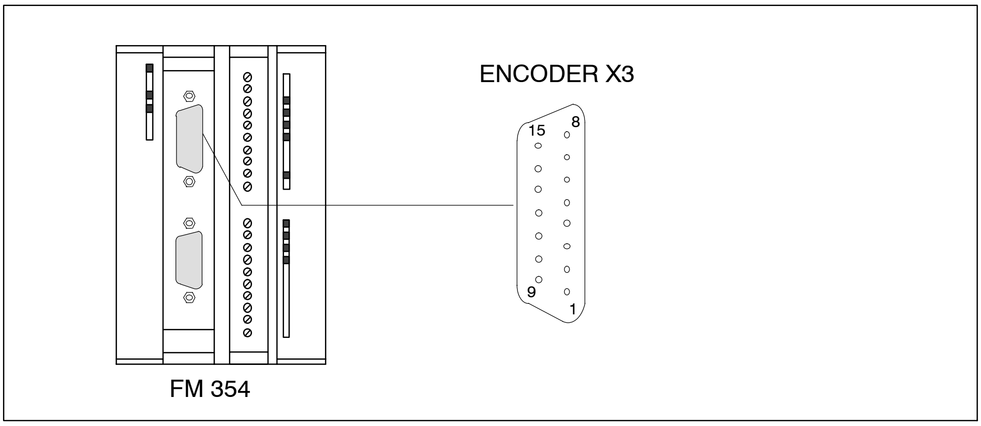


Fig. 4-4 Location of the X3 connector

Connector pinout

Identifier:	**X3**	**ENCODER X3**
Type:		15-pin female sub-D plug connector

Table 4-5 Pinout of the X3 connector

Pin	Encoders		type	Pin	Encoders		type
	Incremental	Absolute			Incremental	Absolute	
1	Unassigned		I	9	MEXT		VO
2		CLS	O	10	N		I
3		CLS_N	O	11	N_N		I
4	P5EXT		VO	12	B_N		I
5	P24EXT		VO	13	B		I
6	P5EXT		VO	14	A_N	DATA_N	I
7	MEXT		VO	15	A	DATA	I
8	Unassigned						

Signal names

A, A_N	Track A true / negated (incremental encoder)
B, B_N	Track B true / negated (incremental encoder)
N, N_N	Zero mark true / negated (incremental encoder)
CLS, CLS_N	SSI sliding pulse true / negated (absolute encoder)
DATA, DATA_N	SSI data true / negated (absolute encoder)
P5EXT	+5 V supply (pins 4 and 6 connected internally)
P24EXT	+24 V power supply
MEXT	Ground power supply

Signal type

VO	Voltage outlet (power supply)
O	Output (5 V signal)
I	Input (5 V signal)

Connectable encoder types

Incremental or absolute (SSI) encoders may be connected directly (e.g. digital-rotary encoders); they are then selected via machine data.

Encoders with SINE/COSINE signals (e.g. length scales) may be connected by way of an external electronic pulse shaper (EXE) that converts the signals to 5 V levels.

Encoder characteristics

Both encoders that can be connected directly and EXEs must meet the following requirements:

Incremental encoders

Transfer procedure:	Differential transfer with 5 V rectangular signals (such as RS422 standard)
Output signals:	Track A as true and negated signal (U_{a1}, $\overline{U_{a1}}$) Track B as true and negated signal (U_{a2}, $\overline{U_{a2}}$) Zero signal N as true and negated signal (U_{a0}, $\overline{U_{a0}}$) When connecting an incremental encoder, please note that at the instant of the zero pulse (TRUE signal) the signals of tracks A and B must also be TRUE. If necessary, the negated signal must be connected and, consequently, the direction adjusted (MD19). “1” signal $\geq$ 2.4 V “0” signal < 0.8 V
Maximum output frequency:	1 MHz
Phase shift, track A to B:	90° ±30°
Power consumption:	Not more than 300 mA

Absolute encoders (SSI)

Transfer procedure:	Synchronous-serial interface (SSI) with 5 V differential-signal transfer signals (such as RS422 standard)
Output signals:	Data as true and negated signal
Input signals:	Sliding pulse as true and negated signal
Resolution:	Not more than 25 bits
Maximum transfer frequency:	1.25 Mbps
Power consumption:	Not more than 300 mA

Encoder power supply

The 5 V or 24 V power supply to the encoders is generated within the module and is available on the female sub-D connector, and so you can power the encoders by way of the connecting cable, without additional wiring. The available voltage is electronically protected against shorting and thermal overload, and is monitored.

Table 4-6 Electrical parameters of encoder power supply

Parameters	Min	Max	Unit
5 V power supply			
Voltage	5.1	5.3	V
Ripple		50	mV_{ss}
Current carrying capacity		0.3	A
24 V power supply			
Voltage	20.4	28.8	V
Ripple		3.6	V_{ss}
Current carrying capacity		0.3	A

Note

24 V encoders that are supplied via X3 must not be inserted or removed when the FM 354 power supply is connected.

External voltage supply

If the encoders are supplied externally (encoders are not connected to FM power supply), the reference potentials of the two supplies must be interconnected.

Connecting cables to encoder

The maximum cable length depends on the specifications of the encoder power supply, and on the transfer frequency. For trouble-free operation, you should not exceed the following values when using SIEMENS cable sets:

Table 4-7 Maximum cable length as a function of encoder power supply

Supply voltage	Power consumption	Max. cable length
5 V DC	≤ 300 mA	25 m (82 ft)
5 V DC	≤ 220 mA	35 m (115 ft)
24 V DC	≤ 300 mA	100 m (328 ft)

Note

If you want to use incremental encoders with cable lengths longer than 25 or 35 m (82 or 115 ft), select a type that uses a 24 V power supply.

Table 4-8 Maximum cable length as a function of transfer frequency

Encoder type	Frequency	Max. cable length
Incremental encoder	1 MHz	10 m (32.8 ft)
	500 kHz	35 m (115 ft)
Absolute encoder (SSI)	1.25 Mbps	10 m (32.8 ft)
	125 kbps	100 m (328 ft)

For further encoder information see Section 9.6.

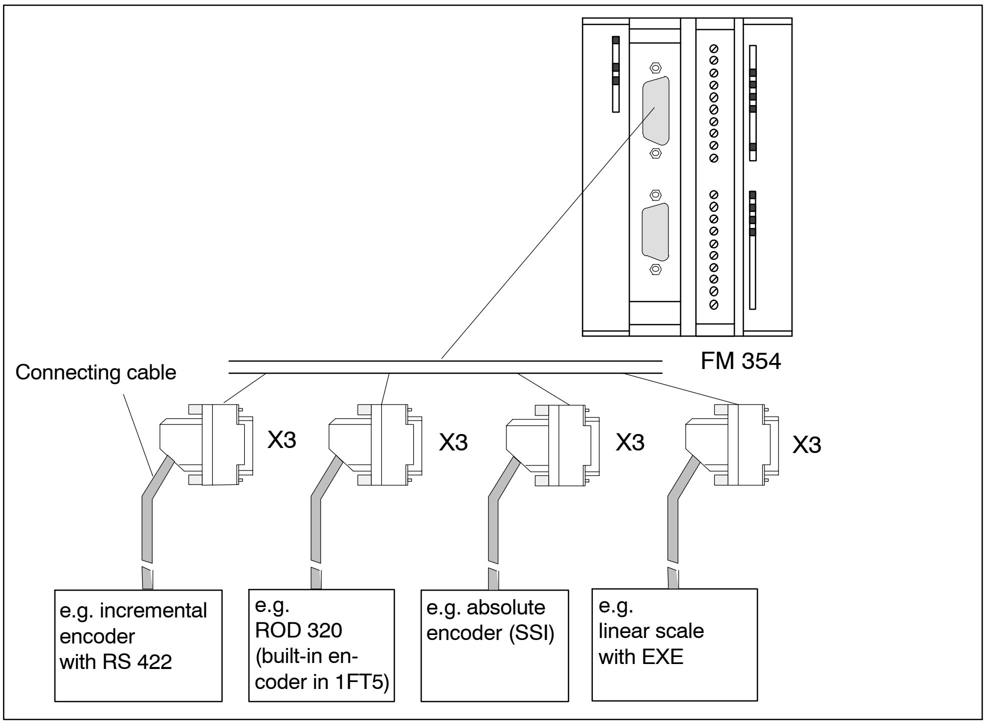
4.5 Connecting the encoders

To connect the connecting cables, please note the following**:**

Note

Use only shielded cables. The shielding must be connected to the metallic or metallized connector jacket.

The cable sets supplied as an accessory offer excellent immunity from interference, as well as cross-sections large enough for the power supply to the encoders.



FM 354
Connecting cable
X3
X3
X3
X3
e.g. incremental encoder with RS 422
e.g. ROD 320 (built-in encoder in 1FT5)
e.g. absolute encoder (SSI)
e.g. linear scale with EXE

Fig. 4-5 Connecting the encoders

Procedure for connecting encoders

To connect the encoders:

1. Connect the connecting cables to the encoders.

   For absolute encoders (SSI) it may be necessary to cut and add connectors to the cable (end of the cable to the encoder) according to the manufacturer's instructions.

2. Open the front door and plug the sub-D connector into the module.
3. Lock the connector in place with the knurled screws. Close the front door.

Available connecting cables for encoders

Cable set for add-on encoders or EXEs (for connection of linear scales)

Order No.: 6FX2 002-2CD01-1□□0

Cable set for built-in encoders with 17-pin round plugs.

Order No.: 6FX2 002-2CE01-1□□0

Cable set for absolute encoders (SSI) with a free cable end.

Order No.: 6FX2 002-2CC01-1□□0

Connecting cables are available in a variety of lengths.

see *Catalog NC Z* , Order No.: E86060-K4490-A001-A□.

4.6 Description of the I/O interface

Front connector

Four digital input/output modules and the standby signal (controller message) may be connected to the 20-pin front connector X1 with its single-wire terminal.

Location of connector

Figure 4-6 shows the front connector in position to be wired, and the labeling on the inside of the front door.

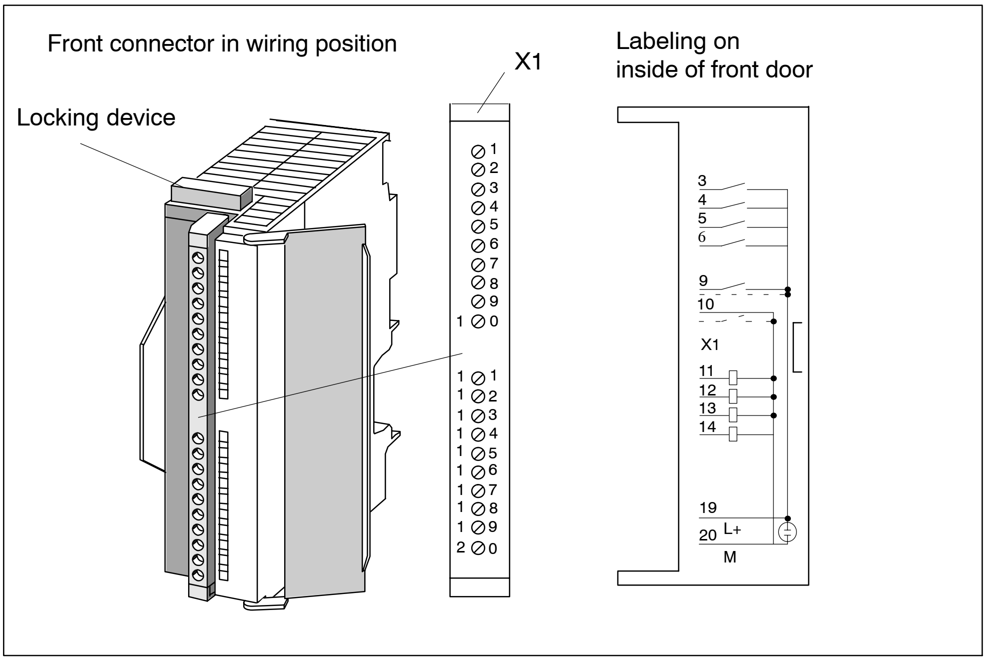


Fig. 4-6 Location of X1 connector

Connector pinout

Connector identifier:	**X1**
Connector type:	20-pin S7 front connector for single-wire terminal

Table 4-9 Pinout of the X1 connector

Pin	Name	Type	Pin	Name	Type
1	open		11	DA1	O
2	open		12	DA2	O
3	DI1	I	13	DA3	O
4	DI2	I	14	DA4	O
5	DI3	I	15	open	
6	DI4	I	16	open	
7	open		17	open	
8	open		18	open	
9	RM_P	I	19	L+	VI
10	RM_N	I	20	M	VI

Signal names

DI1...4	Digital input 1...4
DQ1...4	Digital output 1...4
RM_P	Positive input for controller message
RM_N	Negative input for controller message
L+, M	24 V load power supply / ground

Signal type

O	Output
I	Input
VI	Voltage input

4 digital inputs (DI1...4)

All inputs have equal priority. Switching functions are allocated to an input number by way of machine data; input polarity is selected in the same way (starting and shutdown slopes).

These fast inputs are PLC-compatible (24 V current-sourcing). Switches or contactless sensors (2-wire or 3-wire sensors) can be connected.

Possible uses include:

- As reference-point switches
- As switches for external Start/Stop, external block change
- As touch probes

See Section 5.3.1 for further applications.

Table 4-10 Electrical parameters of digital inputs

Parameters	Value	Unit	Notes
1 signal, voltage range	11...30	V	
1 signal, power consumption	6...15	mA	
0 signal, voltage range	–3...5	V	or input open
Signal delay 0 → 1	15	µs	
Signal delay 1 → 0	150	µs	
Internal signal delay	20	µs	for actual value transfer function

"Controller message" (RM) input

The standby signal of the drive power section (controller message) can be connected to a further input.

Note

The "controller message" input is configured as an isolated optical coupler input. This allows both a current-sourcing and a current-sinking output of the power section to be connected. See Section 4.7 for details about wiring.

Table 4-11 Electrical parameters of "controller message" input

Parameters	Value	Unit	Notes
1 signal, voltage range	15...30	V	
1 signal, power consumption	2...6	mA	
0 signal, voltage range	–3...5	V	or input open
Signal delay 0 → 1	30	µs	
Signal delay 1 → 0	150	µs	

There are two ways of powering the standby signal:

- from the controller
- from the drive unit

Power from the open-loop control

Figure 4-7 shows examples of how to power the standby signal from the controller (e.g. SIMODRIVE 611 drive unit).

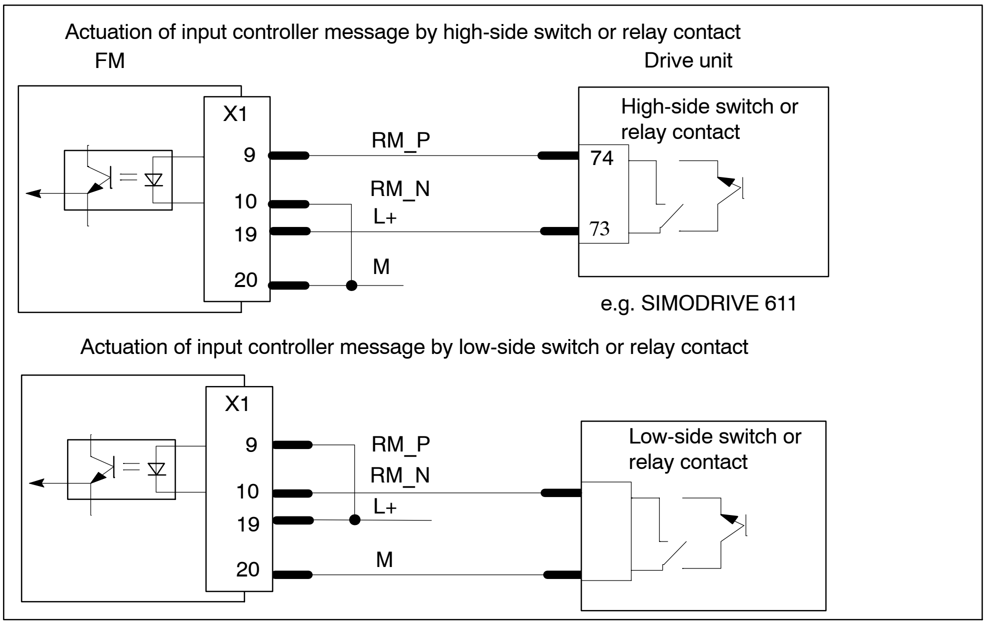

Actuation of input controller message by high-side switch or relay contact
FM
Drive unit
X1
9
10
19
20
RM_P
RM_N
L+
M
High-side switch or relay contact
74
73
e.g. SIMODRIVE 611
Actuation of input controller message by low-side switch or relay contact
X1
9
10
19
20
RM_P
RM_N
L+
M
Low-side switch or relay contact

Fig. 4-7 Actuation of the input controller message, power supply from the control

Power from the drive unit

Figure 4-8 shows examples of how to power the standby signal from the drive unit.

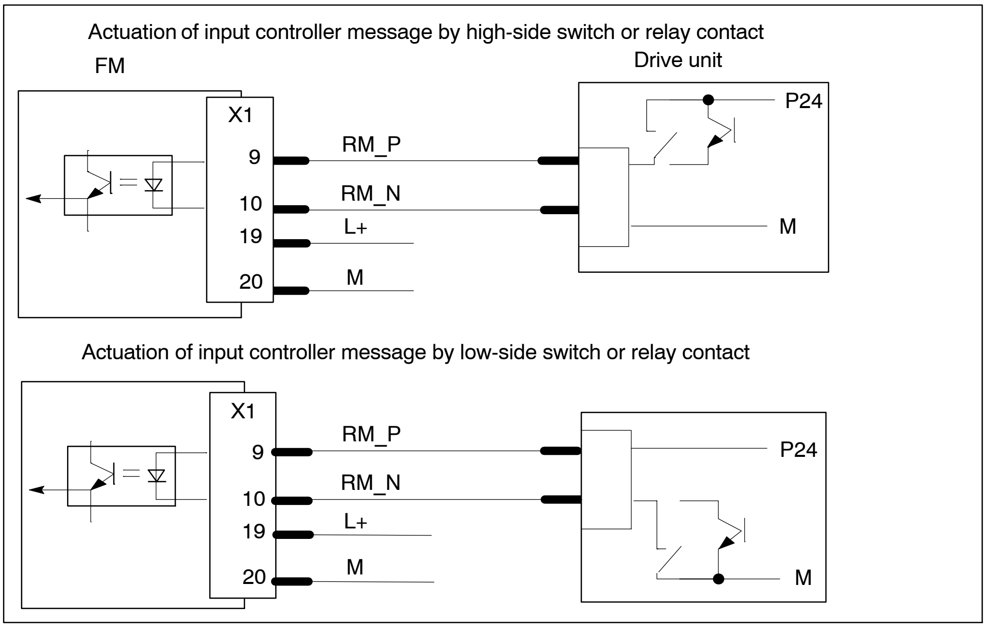

Actuation of input controller message by high-side switch or relay contact
FM
Drive unit
X1
9
10
19
20
RM_P
RM_N
L+
M
P24
M
Actuation of input controller message by low-side switch or relay contact
X1
9
10
19
20
RM_P
RM_N
L+
M
P24
M

Fig. 4-8 Actuation of the input controller message, power supply from the drive unit

4 digital outputs (DO1...4)

All outputs have equal priority. Switching functions are allocated to an output number by way of machine data.

These four outputs are intended for wiring of application-specific signals.

Possible uses include:

- Position reached and stopped
- Switching function M command
- Forward/backward rotation

See Section 5.3.1 for further applications.

Note

Make sure that the cable length between the digital outputs and the load does not exceed a permissible max. length of 30 m.

Table 4-12 Electrical parameters of digital outputs

Supply voltage	24 V DC (allowable range: 20.4...28.8 V)
Electrical isolation	No
Output voltage	• 0 Signal: Residual current max. 2 mA • 1 Signal: (Power supply –3 V)
Output current on signal “1”	
• at ambient temperature of 40°C	
– Rated value	0.5 A (total current 2 A)
– Permissible value range	5 mA...0.6 A (via supply voltage)
– Lamp load	max. 5 W
• at ambient temperature of 60°C	
– Rated value	0.1 A (total current 0.4 A)
– Permissible value range	5 mA...0.12 A (over power supply)
Switching rate	• Resistive load: max. 100 Hz • Inductive load: 0.25 Hz

Load power supply (L+, M)

Here a 24-V load power supply must be connected. The unit will not operate if the poles are reversed.

Note

Note the SIMATIC setup guidelines. In particular, the M terminal (reference potential) must be connected with the chassis ground of the programmable controller (M terminal on the terminal block of the S7-300 CPU).

see manual *S7-300 Programmable Controller, Hardware and Installation*.

Danger

The 24 V load power supply should be a functional extra-low voltage with safe isolation to EN60204-1, Section 6.4, PELV (with M grounding).

Note

Make sure that the interconnecting cable between power supply and load power supply connection L+ and appropriate reference potential M does **not** exceed a max. permissible length of 10 m.

4.7 Wiring up the front connector

Wiring the front connector

Figure 4-9 shows you how to install the conductors on the front connector and how to relieve the strain with the shield connection element.

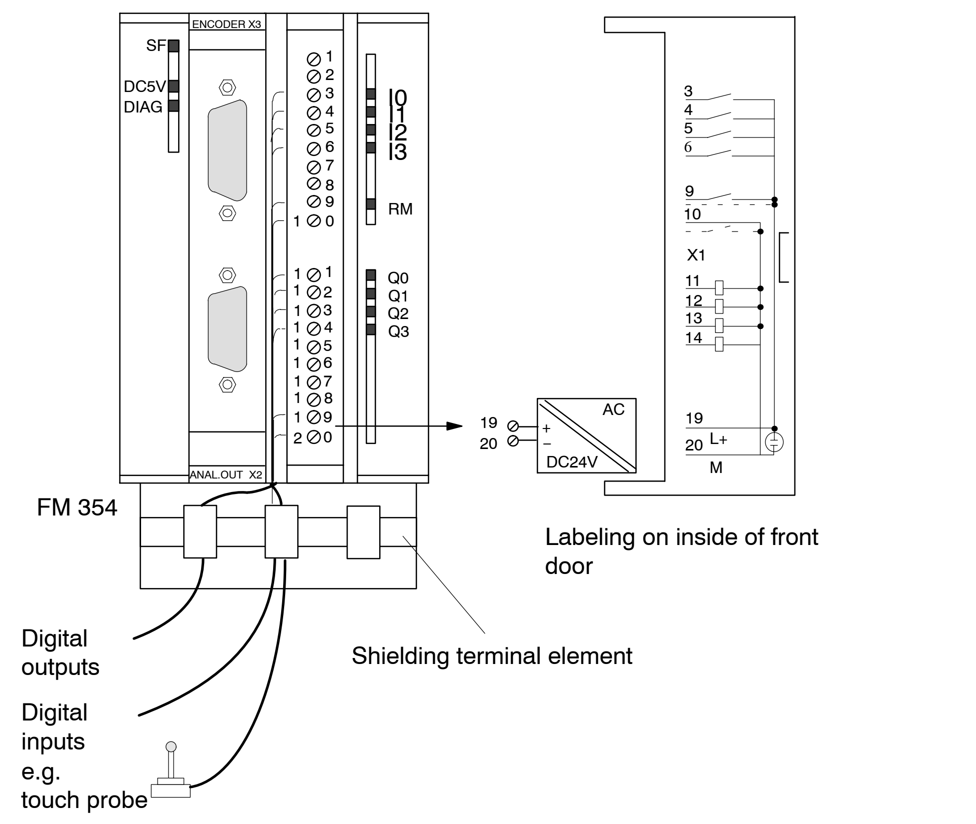


Fig. 4-9 Wiring up the front connector

Connecting cables

Flexible conductor, cross-sectional area 0.25...1.5 mm$^{2.}$

Ferrules are not necessary.

You can use ferrules without insulating collars per DIN 46228, Shape A, long configuration.

You can connect two lines measuring 0.25...0.75 mm^2 in a single ferrule.

Note

To provide optimum immunity to interference, shielded cables should be used to connect touch probes or sensors.

Tools required

A 3.5 mm (.13 inches) screwdriver or power screwdriver.

Procedure for wiring the front connector

To wire the terminal strip:

1. Strip 6 mm (.23 inches) of insulation from the cable; apply ferrules, if any.
2. Open the front door. Move the front connector into position for wiring (while pressing down on the locking elements as shown in Figure 4-6).

   Lock the connector in place without any electrical contact to the module.
3. Apply the cable grip to the connector.
4. If you are leading the lines out from below, start wiring from the bottom; otherwise start from the top. Screw down unused terminals as well.

   The tightening torque should be 60-80 Nm.
5. Tighten the cable grip on the cable strand.
6. Move the front connector into operating position (while pressing down on the locking elements).
7. You can fill out the labeling strip enclosed with the product and slip it into the front panel.

Shielded cables

When using shielded cables, the following additional steps are necessary:

1. After leading the cable into the cabinet, connect the cable shielding with a grounded shielding bus (strip the insulation from the cable itself).

   For this you can use the shielding terminal element mounted on the DIN rail; it will accept up to eight shielding terminals.

   see manual *S7-300 Programmable Controller, Hardware and Installation*.
2. Connect the shielded line to the module, but do not connect the shielding there.

Shielding terminal element

To provide a shielding end lead for shielded cables, this element can be inserted in the DIN rail. It can accept up to eight shielding terminals (KLBÜ line from Weidmüller).

Order No.:	Terminal element: 6ES7 390-5AA00-0AA0 Shielding terminal: 6ES7 390-5CA00-7AA0

see *Catalog NC 60.1,* Order No. E86060-K4460-A101-A□

see *Catalog ST 70,* Order No. E86060-K4670-A101-A□

■

Defining Parameters 5

Chapter Overview

Summary

This chapter gives you an overview of how to define the parameters of the FM 354 with the “Parameterize FM 354” tool.

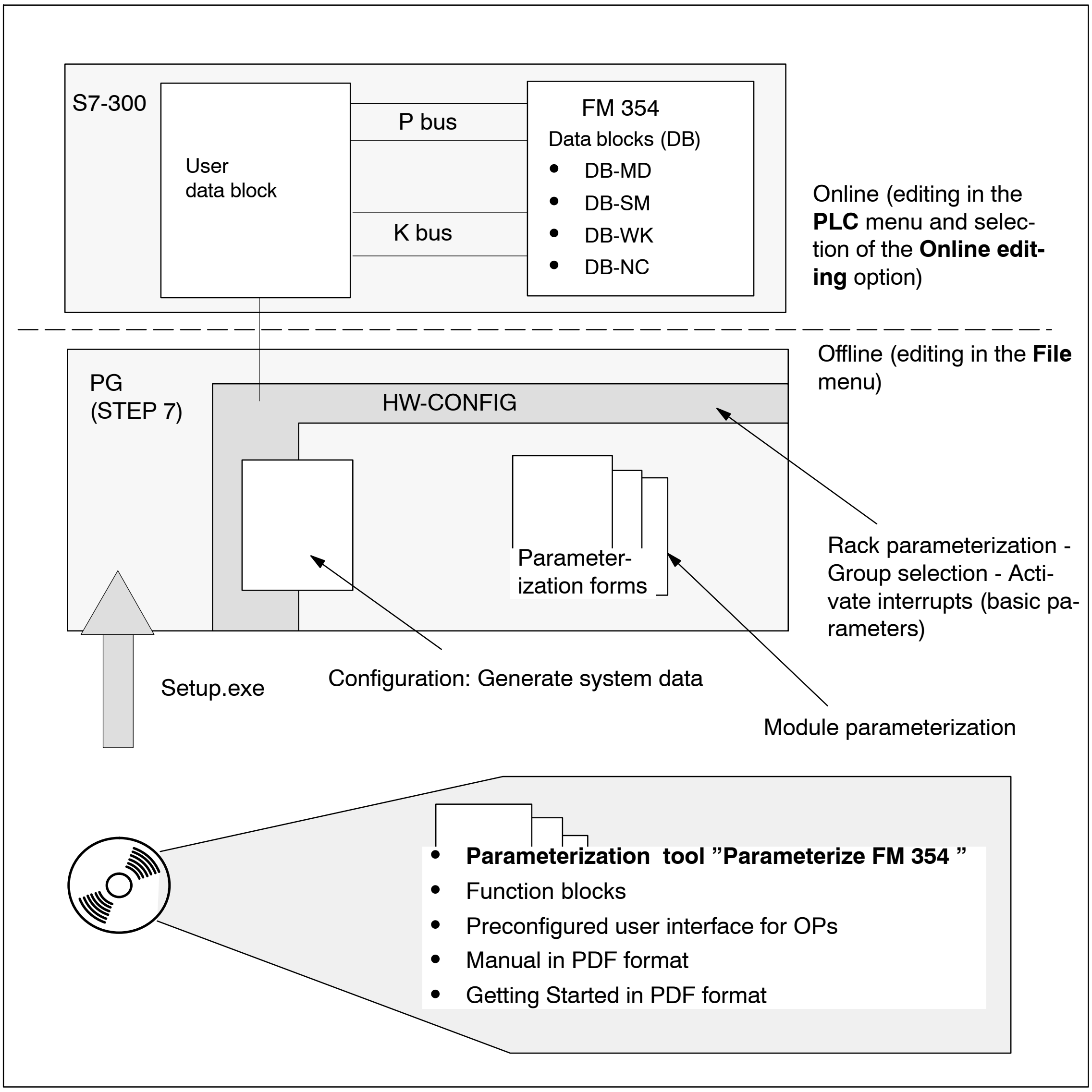

S7-300
User
data block
P bus
K bus
FM 354
Data blocks (DB)
• DB-MD
• DB-SM
• DB-WK
• DB-NC
Online (editing in the **PLC** menu and selection of the **Online editing** option)
Offline (editing in the **File** menu)
PG
(STEP 7)
HW-CONFIG
Parameterization forms
Rack parameterization - Group selection - Activate interrupts (basic parameters)
Setup.exe
Configuration: Generate system data
Module parameterization
• **Parameterization tool ”Parameterize FM 354 ”**
• Function blocks
• Preconfigured user interface for OPs
• Manual in PDF format
• Getting Started in PDF format

Fig. 5-1 Overview of parameterization

5.1 Installing “Parameterize FM 354”

Prerequisites

One of the following operating systems must be installed on the programming device (PG/PC):

- ”Windows Vista 32 Bit Ultimate”
- ”Windows Vista 32 Bit Business”
- “Windows 2000 SP4”
- “Windows 2003 Server“
- ”Windows XP-Professional”

You need the STEP 7 program (V5.3 + SP2 or higher by Windows Vista:
V5.4 + SP3 or higher).

For online operation, make sure that your programming device (PG) / PC is connected to the S7-300 CPU (see Figure 4-1). For distributed use of the FM, a connection from the PG / PC must be established to the L2 DP network. For distributed use of the FM 354 in PROFINET, a connection from the PG / PC must be established to the L2 DP network or to the Ethernet via a switch module.

Installation

The entire software (parameterization tool, function blocks and preconfigured user interface for OPs) is stored on CD ROM.

Install the software as follows:

1. Insert the CD ROM in the CD ROM drive of your PG/PC.
2. Run file **Setup.exe** on the CD ROM.
3. Follow the instructions displayed by the installation routine step for step.

   Result: The software is installed in the following directories:

   - “Parameterize FM 354” parameterization tool: **[STEP7 directory]\S7FSTEP**
   - Technology functions:
     [STEP7 directory]\S7LIBS\FMSTSV_L (library name: “FMSTSV_L”)
   - Technology functions (also for PROFINET, upon request):
     [STEP7 directory]\S7LIBS\FM353_354 (library name: “FM353_354”)
   - User interface for OPs:
     [STEP7 directory]\EXAMPLES\FM354\zDt14_02_FM354_OP_EX
   - Sample applications for blocks of the “FMSTSV_L” library:
     [STEP7 directory]\EXAMPLES\zDt14_02
     STEP7 project name: **zDt14_02_FM353_EX**
   - Sample applications for blocks of the “FM353_354” library:
     [STEP7 directory]\EXAMPLES\zDt14_03
     STEP7 project name: **zDt14_03_FM354_EX**
   - MD DBs (for commissioning of the stepper drive):
     [STEP7 directory]\EXAMPLES\FM354\MD

5.2 Getting started with “Parameterize FM 354”

Prerequisites

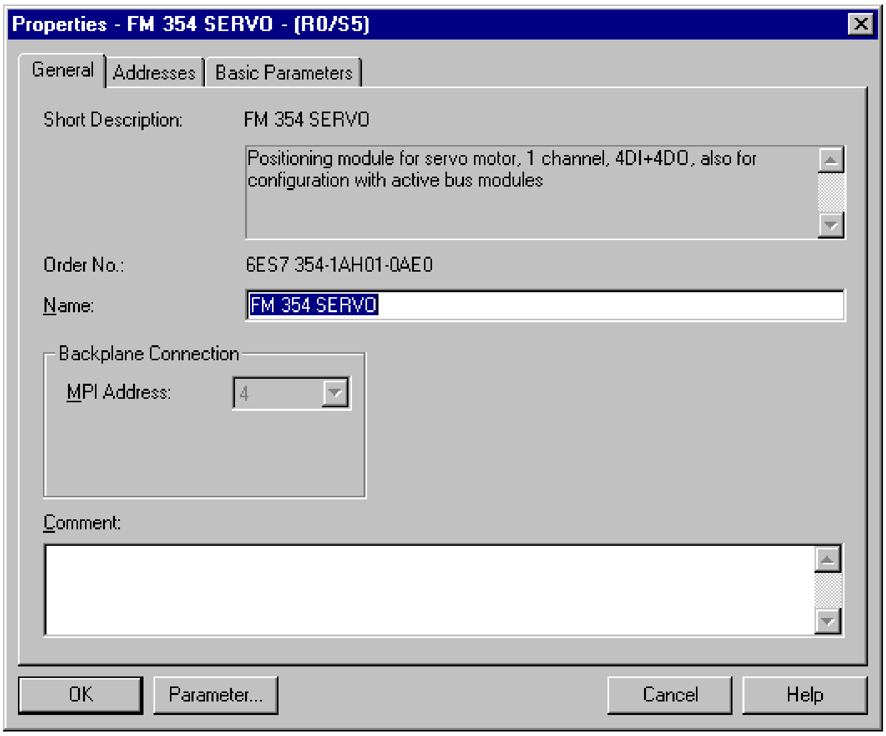
You have installed the software on your programming device/PC, as described in Section 5.1.

Configuration

Before you can configure your system, you must create a project in which to save the parameters. You will find further information on how to configure modules in your user manual *Standard Software for S7 and M7, STEP 7*. The description below outlines only the most important steps.

1. Start the *SIMATIC Manager* and open your project.
2. Insert a **SIMATIC 300 station** in the menu **Insert > Station**.
3. Select the **SIMATIC 300 station**. Call up the S7 hardware configuration from the menu **Edit > Open Object**.
4. Select a rack.
5. Select the FM 354 positioning module with the correct order number from the module catalog, and insert it in the hardware table as appropriate for your configuration.
6. Double-click a module to configure it.

   The **Properties** dialog box appears.



Properties - FM 354 SERVO - (R0/S5)

Short Description: FM 354 SERVO

Positioning module for servo motor, 1 channel, 4DI+4DO, also for configuration with active bus modules

Order No.: 6ES7 354-1AH01-0AE0

Name: FM 354 SERVO

MPI Address: 4

Comment:

Fig. 5-2 Getting started with “Parameterize FM 354”

7. By clicking on the tabs in this FM 354 window (General, Addresses and Basic Parameters), you can
   - Assign a name
   - Change the address of the FM as well as any input parameters for the POS_INIT block (see Section 6.3.2)
   - Configure the interrupts (diagnostic interrupt, process interrupt).

Note:

Further operation of the FM 354 is not possible with the CPU in the STOP state.

Click the **Parameters** button to call up the screen for setting the parameters.

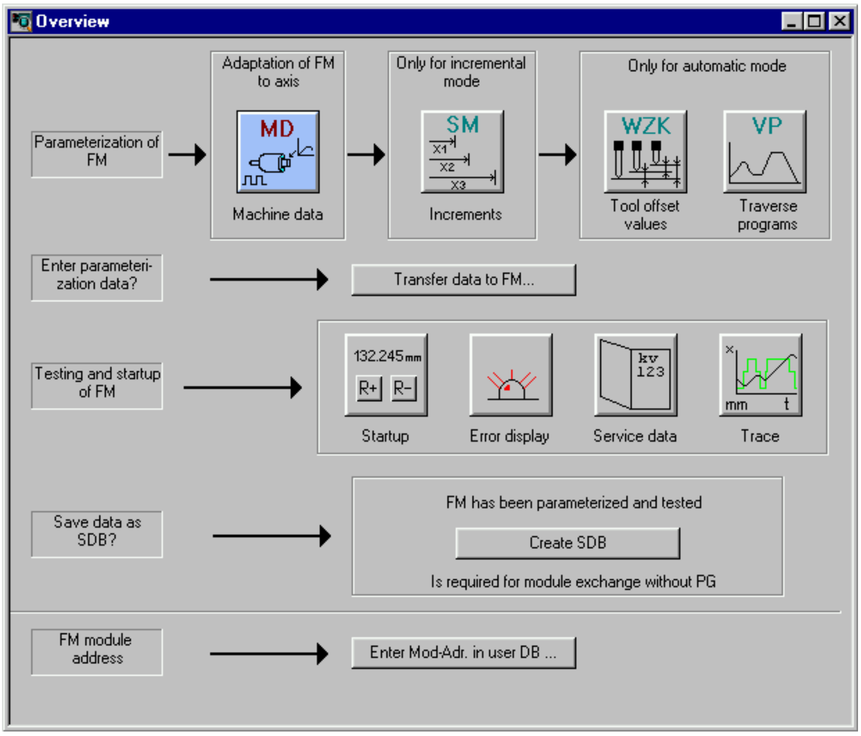


Fig. 5-3 Overview display for parameterization

You can return to this display at any point during parameterization by selecting the menu **View** > **Overview**.

The FM 354 module for servo-controlled positioning is parameterized by way of parameter DBs that reside in memory on the module. Here a key function is performed by the “Machine data” data block (DB-MD), since it is always needed, regardless of what technological function the module performs. All other parameter DBs are only needed as a function of the technology involved.

You can now set the parameters of your module. This chapter gives you an overview of the parameters that can be set.

You can use the mouse to change the size of the window for entering the parameter data and the size of the overview display.

Proceed as follows:

1. Position the mouse pointer on the top border of the window, so that it changes into an arrow.
2. Press the left mouse button, and drag the pointer downwards by moving the mouse.
3. Release the mouse button.
4. Position the mouse pointer on the bar with the name of the window.
5. Press the left mouse button, and drag the pointer upwards by moving the mouse. When you have moved the window to the correct position, release the mouse button.

When you have configured your project, you can call up the **Properties** screen in S7 Configuration by selecting the module and activating the menu command **Edit > Object Properties**.

Integrated help

The parameterization user interface has an integrated help system to support you when you set the parameters of the positioning module. To call up the integrated help:

- Select the menu command **Help > Help Topics ...** or
- press the **F1** key or
- select the symbol ↖? and then move to the element or window you want information about and press the left mouse button.

5.3 Parameter data

What can I parameterize?

You can parameterize the following data storage areas:

- Machine data (MD)
- Increment sizes (SM)
- Tool offset data (TO)
- Traversing programs (NC)
- User data (user data blocks)

This data is stored in data blocks (DBs) within the numerical range 1001 to 1239 (not including user data).

The MD, SM, TO and NC data blocks are transferred to the FM 354 and reside in memory there.

Parameterization of SM, TO and NC may be omitted if the associated functions are not used.

The user data block must be stored in the CPU. Only then can it be filled with data online (see Chapter 6).

Parameterization data (except for user data) can also be created, edited and saved offline on the PU.

Data blocks (DB) of the FM 354

Table 5-1 gives you an overview of the data blocks in the FM 354 and their meaning.

Table 5-1 Data blocks

Data block	Significance
DB-MD	**Machine data (DB No. = 1200)** **User memory requirements = 284 bytes** Machine data serves to adapt the FM 354 to the user's own specific application. Parameterization with machine data is essential in order for the FM's functions to be activated. The parameterized DB-MD should be loaded to the FM. As it is written to the FM 354, the DB-MD is checked for the input limits of the individual values and their interdependencies. It is then stored only if all values are allowed. Otherwise data error messages are displayed by way of the MPI. A defective DB will not be retained when the power is turned off. The machine data can then be activated by way of “Activate machine data” or by switching the equipment on and off.

Table 5-1 Data blocks, continued

Data block	Significance
DB-SM	**Increments (DB No. = 1230)** **User memory requirements = 468 bytes** Increments serve in the “Relative incremental” operating mode as user-definable relative path distances for individual positioning. You can define from 1 to 100 increment sizes (see Section 5.3.2). Modifications can be made in all operating modes (even in “Incremental relative” mode) during movement. The modifications of the increments must always be complete before a new movement is started in “Incremental relative” mode. If this is not the case, the error message “incremental dimensions do not exist” is output Cl. 2/No. 13.
DB-WK	**Tool offset data (DB No. = 1220)** **User memory requirements= 308 bytes** The use of tool length compensation and wear values is described in Section 10.1. Up to 20 compensation or wear values are available. Tool offset data are required for the “Automatic and Automatic single block” modes. Modifications can be made in all operating modes and during movement. If modifications are made during starting or at block transitions when the tool compensation is active (internal access to offset values), the error message “tool offset value does not exist” is output Cl.3/No.35.
DB-NC	**Traversing programs (Program No. + 1000 = DB No. = 1001...1199)** **User memory requirements= 108 bytes + (20 x number of traversing blocks)** Traversing programs are required for the “Automatic and Automatic single block” modes. • Programs which are not selected can always be modified. • If modifications are made to a preselected program, including the subprogram, preselection of the program is canceled. You must then select the program again. A modification can be made to a program when BL = 0 (start of program/end of program) and on Stop.
System data block SDB ≥ 1 000	**For module replacement without PG** All the parameter data of the FM 354 (DB-MD, DB-SM, DB-WK, DB-NC) are stored in SDB ≥ 1 000. This SDB is loaded into the CPU and is used as an additional means of data storage.
DB-SS	**Data block for status messages (DB no. 1000)** The DB-SS is an internal DB on the FM for testing, start-up and operator control and monitoring.
DB 1249	Internal DB on the FM, not relevant for user.

Data block structure

Table 5-2 gives a rough picture of data block structure.

Table 5-2 Data block structure

Addresses/ Offset	Contents	Comment
	DB header	System information, not relevant for user
0 and above	User data area / structure header	Information for labeling of data block within the system
24 and above for MD, otherwise 32	User data	Parameterization data

Detailed data block structures and parameterization data for the individual types of data blocks can be found in the following sections.

5.3.1 Machine data

DB structure

Table 5-3 gives you an overview of the structure of the “machine data” data block (DB-MD).

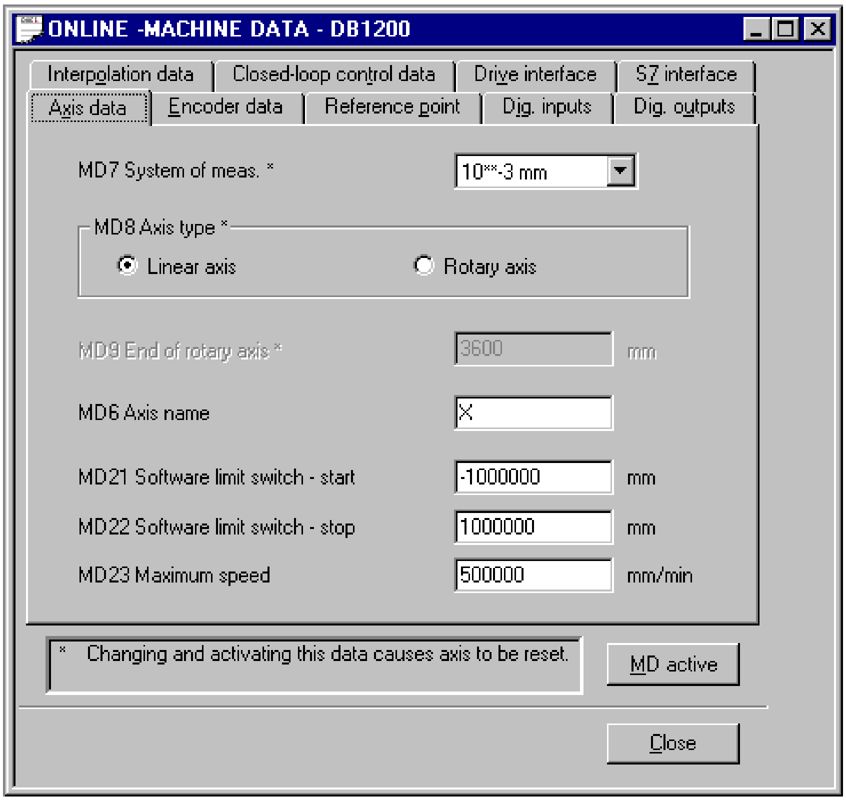
Table 5-3 DB structure – Machine data

Byte	Variable type	Value	Significance of the variables	Comment
			DB header	
0	WORD		Rack slot	Module address
2	WORD		DB No. (≥ 1000)	As in DB header
4	DWORD		reserved	
8	WORD		Error No. (from FM)	With HMI services
10	WORD	1	Channel number	
12	2 STRING	MD	DB identifier/type	2 ASCII characters
16	DWORD	354	Module identifier	FM 354
20	4 CHAR	0	Version number/block number	(DB structure)
24 and above. ..			See machine data list MD5...MD45	

Note: MD address in DB = (MD-no. –5) * 4 +24

Entering values

In “Parameterize FM 354” select the menu **File** > **New** > **Machine Data** to call up the following display.



ONLINE -MACHINE DATA - DB1200

MD7 System of meas. * 10**-3 mm

MD8 Axis type *

● Linear axis ○ Rotary axis

MD9 End of rotary axis * 3600 mm

MD6 Axis name X

MD21 Software limit switch - start -1000000 mm

MD22 Software limit switch - stop 1000000 mm

MD23 Maximum speed 500000 mm/min

* Changing and activating this data causes axis to be reset.

Fig. 5-4 Entering values for machine data

Enter the machine data in the tab windows.

You can also enter your values in a table by selecting **View** > **Table form**.

When creating the MD DBs you must follow the instructions in Section 7 “Starting up the FM 354”.

Note

The measurement system (MD7) must match the measurement system specified in the other DBs.

The measurement system raster (MSR) is the smallest distance unit in the active system of measurement.

If at some point you have failed to take this precaution:

1. Delete all data blocks (which do not match the measurement system) or clear the memory of the FM 354 completely.
2. Modify the other data blocks on the PG.
3. Reload the data blocks to the FM 354.

Machine data list

All machine data of the FM 354 are listed in Table 5-4.

Notes to the machine data list:

K stands for configuration data: see Section9.3.3

E stands for user-definable machine data settings for readjustment (startup optimization) and technology; see Section 9.3.3

The units of measurement refer to the value representation in the machine data DB.

Table 5-4 Machine data list

No.	Designation	Default values	Value/Meaning	Data type/ Unit/Comments	See Section
1...4				open	
5 E	Process interrupt generation	0	0 = Position reached 1 = Length meas. completed 3 = Change block on-the-fly 4 = Inprocess Measurement	BITFIELD32	9.10
6	Axis name	X	max. 2 ASCII characters[1]	4 bytes[3]	
7 K	System of measurement	1	$1 = 10^{-3}$ mm $2 = 10^{-4}$ inch $3 = 10^{-4}$ degrees $4 = 10^{-2}$ degrees	DWORD [MSR]	9.4
8 K	Axis type	0	0 = linear axis 1 = rotary axis	DWORD	9.5
9 K	Rotary axis end[2]	$36 \cdot 10^5$	0...1 000 000 000	DWORD [MSR]	
10 K	Encoder type	1	0 = not present 1 = incremental encoder 3 = absolute enc. (SSI, 13-bit) 4 = absolute enc. (SSI, 25-bit) 13 = absolute enc. (SSI, 13-bit) 14 = absolute enc. (SSI, 25-bit)	DWORD GRAY Code GRAY Code Binary Code Binary Code	9.6.1 9.6.2
11 K	Travel per motor revolution (division period)[2]	10 000	1...1,000,000 ,000	DWORD [MSR] (integer component)	9.6.1 9.6.2
12 K	Residual distance per encoder revolution (division period)[2]	0	$0...2^{32}-1$	DWORD [2^{-32}MSR] (fractional component)	

MSR = measurement-system grid RPS = reference-point switch

[1] The variable axis name is implemented as an axis letter (X, Y, Z, ...) with an address extension (1...9). **Permissible characters: X, Y, Z, A, B, C, U, V, W, Q, E, 1...9 e.g.: “X”, “X1”**

[2] see Dependencies

[3] The axis name is in bytes 3 and 4 (bytes 1 and 2 give the character length specification).

Table 5-4 Machine data list, continued

No.	Designation	Default values	Value/Meaning	Data type/ Unit/Comments	See Section
13 K	Increments per encoder revolution (division period)[2]	2 500	$2^1...2^{25}$ (for absolute encoders)	DWORD With incremental encoders, evaluation takes place at $4 \cdot$ MD.	9.6.1 9.6.2
14 K	Number of rotations – absolute encoder	0	0/1 = single-turn encoders $2^1...2^{12}$ for multi-turn encoders	DWORD Only powers of two are allowed.	9.6.1 9.6.2
15 K	Baud rate –absolute encoder For baud rates which lie between these values, set the next lower baud rate	2	1 = 78,000 2 = 156,000 3 = 312,000 4 = 625,000 5 = 1,250,000	DWORD	9.6.1 9.6.2
16 K	Reference-point coordinate	0	–1,000,000,000...+1,000,000,000	DINT [MSR]	9.2.3
17 K	Absolute-encoder readjustment	0	0...2^{25} – 1	DWORD [Encoder grid] absolute encoder	9.6.3
18 K	Type of reference-point approach (reference-point approach direction)	0	0 = direction +, zero pulse right 1 = direction +, zero pulse left 2 = direction –, zero pulse right 3 = direction –, zero pulse left 4 = direction +, RPS center 5 = direction –, RPS center 8 = direction +, RPS edge 9 = direction –, RPS edge	DWORD Code identifies position for synchronization point with ref. to RPS for incremental encoders only	9.2.3
19 K	Direction adjustment	0	0 = invert direction of measurand 1 = invert analog value	BITFIELD32	9.7
20 K	Hardware monitoring	0	0 = cable break (incremental encoder) 1 = error, absolute encoder 2 = pulse monitoring (incremental encoder) 3 = Voltage monitoring - encoder	BITFIELD32	9.6.1 9.6.2
21 E	Software limit switches, beginning[2)]	-10^9	–1 000 000 000...1 000 000 000	DINT [MSR]	9.7 9.9
22 E	Software limit switch – end[2)]	10^9	–1 000 000 000...1 000 000 000		

MSR = measurement-system grid RPS = reference-point switch

1) The variable axis name is implemented as an axis letter (X, Y, Z, ...) with an address extension (1...9). **Permissible characters: X, Y, Z, A, B, C, U, V, W, Q, E, 1...9 e.g.: "X", "X1"**

2) see Dependencies

3) The axis name is in bytes 3 and 4 (bytes 1 and 2 give the character length specification).

Table 5-4 Machine data list, continued

No.	Designation	Default values	Value/Meaning	Data type/ Unit/Comments	See Section
23 E	Maximum speed	$30 \cdot 10^6$	10...500,000,000	DWORD [MSR/min]	9.7
24 E	Target range (position reached, stop)	1 000	0...1 000 000	DWORD [MSR]	
25 E	Monitoring time	0	0 = no monitoring 1 ... 65534	DWORD [ms] rounded to 2-ms steps	
26 E	Stationary range	10^4	1...1 000 000	DWORD [MSR]	
27 E	Reference-point shift	0	−1,000,000,000...+1,000,000,000	DINT [MSR]	9.2.3
28 E	Referencing velocity[2]	$6 \cdot 10^6$	10...500,000,000	DWORD [MSR/min]	9.2.3
29 E	Reducing velocity[2]	$3 \cdot 10^6$	10...500,000,000		
30 E	Backlash compensation	0	0 to 1,000,000	DINT [MSR]	9.7
31 E	Directional reference of backlash	0	0 = as in search for reference (not for absolute encoders) 1 = positive 2 = negative	DWORD	9.7
32 K	M-function output type	1	**during positioning:** 1 = time-controlled 2 = acknowledgment-controlled **before positioning:** 3 = time-controlled 4 = acknowledgment-controlled **after positioning:** 5 = time-controlled 6 = acknowledgment-controlled	DWORD serial output of up to 3 M functions in NC block	10.3 9.1
33 K	M-function output time	10	1...100,000	DWORD [ms] rounded to 2-ms steps	

MSR = measurement-system grid RPS = reference-point switch

1) The variable axis name is implemented as an axis letter (X, Y, Z, ...) with an address extension (1...9). **Permissible characters: X, Y, Z, A, B, C, U, V, W, Q, E, 1...9 e.g.: “X”, “X1”**

2) see Dependencies

3) The axis name is in bytes 3 and 4 (bytes 1 and 2 give the character length specification).

Table 5-4 Machine data list, continued

No.	Designation	Default values	Value/Meaning	Data type/ Unit/Comments	See Sec- tion
34 K	Digital inputs[2]	0	0 = external start 1 = input for enable 2 = external block change 3 = set actual value on-the-fly 4 = measure 5 = RPS for search for reference 6 = reversing switch for search for reference	BITFIELD32 bit-coded function allocation: Bit No. I/O 0 Bit No. + 8 I/O 1 Bit No. + 16 I/O 2 Bit No. + 24 I/O 3 Front edge always activates the function	9.2.3 9.8
35 K	Digital outputs[2]	0	0 = Position reached, stop 1 = Axis movement forward 2 = Axis movement reverse 3 = Change M97 4 = Change M98 5 = Enable Start 7 = Direct output		9.8
36 K	Input adjustment (signal processing inverted)	0	8 = I0 inverted 9 = I1 inverted 10 = I2 inverted 11 = I3 inverted	BITFIELD32	9.8
37 K	Servo control signals	1	0 = controller enable active 2 = controller ready active 3 = controller ready inverted 7 = time override active 15 = continue operation after emergency stop (drive enable [AF]) 16 = automatic drift compensation active		9.7 9.1.1
38 E	Positioning loop amplification	1 000	1...10,000	DWORD [(MSR/min)/MSR]	
39 E	Minimum following error, dynamic	0	0 = no monitoring 1...1 000 000	DWORD [MSR]	
40 E	Acceleration	1 000	0 = without ramp 1...100,000	DWORD [10^{3}MSR/s^2]	9.7
41 E	Deceleration	1 000			
42 E	Jolt time	0	0...10,000	DWORD [ms]	
43 E	Set voltage, max.	8 000	1,000...10,000	DWORD [mV]	
44 E	Offset compensation	0	−1,000...+1,000	DINT [mV]	
45 E	Voltage ramp	0	0...10,000,000	DWORD [mV/s]	

MSR = measurement-system grid RPS = reference-point switch

[1] The variable axis name is implemented as an axis letter (X, Y, Z, ...) with an address extension (1...9). **Permissible characters: X, Y, Z, A, B, C, U, V, W, Q, E, 1...9 e.g.: “X”, “X1”**

[2] see Dependencies

[3] The axis name is in bytes 3 and 4 (bytes 1 and 2 give the character length specification).

Table 5-4 Machine data list, continued

No.	Designation	Default values	Value/Meaning	Data type/ Unit/Comments	See Sec- tion
52 E	Speed for backlash compensation	0	0 1 to 100	DWORD [%]	9.7
53 E	Mode for backlash compensation	0	0 = before positioning 1 = during positioning	DWORD	9.7
54 E	Standstill speed	0	0 Automatic zero-speed monitoring 1...1 000 000 zero-speed monitoring detection if standstill speed falls below setpoint	– [MSR/min]	7.3.8
55 E	TimeOut time for zero-speed monitoring	0	0 No TimeOut monitoring 1...100 000 Enforced zero-speed monitoring after the TimeOut time has elapsed	– [ms]	7.3.8
56 E	Standard diagnosis	0	0 unverzögerte Standarddiagnose 1...10000 Ansprechzeit für Standarddiagnose (im 2 ms-Raster aufgerundet wirksam) > 10000 Parametriertool Fehlermeldung: 0x0538 MD56(im Parametriertool < Version 4.03.03 (?) erfolgt die Fehlermeldung mit der falschen MD-Nummer MD57)	[ms]	7.3.9
60 E	Function enable for response time in MD56	0	101 Function enable for response time in MD56 Remaining values: No function enable for the response time in MD56	Code	

MSR = measurement-system grid RPS = reference-point switch

1) The variable axis name is implemented as an axis letter (X, Y, Z, ...) with an address extension (1...9). **Permissible characters: X, Y, Z, A, B, C, U, V, W, Q, E, 1...9 e.g.: “X”, “X1”**

2) see Dependencies

3) The axis name is in bytes 3 and 4 (bytes 1 and 2 give the character length specification).

Dependencies

With certain combinations of machine data, restrictions in the value range arise for non-processing of the machine data.

These dependencies are verified on acceptance of the MD DB or individual machine data, and an error message is output in the event of a violation. Some checks are performed on the basis of internally calculated reference variables.

These reference variables and the dependency checks are described in the tables below.

Reference variables generated internally from MD:

Generation of travel per encoder revolution **UMWEG**

$\mathrm{UMWEG} = \mathrm{MD11} + \mathrm{MD12} \cdot 2^{-32}$

Generation of internal measured value factor **MWFAKTOR**

MD10	Measured value factor
0	MWFAKTOR = 1
1	MWFAKTOR = UMWEG / (4 · MD13)
3, 4, 13, 14	MWFAKTOR = UMWEG / MD13

Activation of software limit switches **SEAKT**

MD21	MD22	SEAKT
$= -10^9$	$= +10^9$	0 (inactive)
$\neq -10^9$	$= +10^9$	1 (active)
$= -10^9$	$\neq +10^9$	
$\neq -10^9$	$\neq +10^9$	

Internal generation of absolute traversing range limits **VFBABS**

MWFAKTOR	VFBABS
< 1	10^9 / MWFAKTOR
≥ 1	10^9

Verification:

MD9 check

MD8	MD10	MD18	Permissible rotary axis end	
0	–	–	–	
1	0		–	(MD23/30 000) · ≤ MD9 ≤ VFBABS
	1	≥ 4	–	
		< 4	MD9 mod UMWEG == 0	
	3, 13	–	UMWEG mod MD9 == 0	
	4, 14	–	(MD14 · UMWEG) mod MD9 == 0[1]	

[1] MD9 is the ratio of a power of 2^x or 2^{-x} to the absolute value range of the encoder (see Section 9.6.2)

MD11, MD12, MD13 check → results in MWFAKTOR (see above)

Permissible measured value factor range:	2^{-14} < MWFAKTOR < 2^{14}

MD13 check

MD10	Increments per encoder revolution	
0, 1	–	
3, 4, 13, 14	2^x	x = 1, 2, 3, ...

MD14 check

MD10	No. of revolutions	
0, 1, 3, 13	–	
4, 14	2^x	x = 1, 2, 3, ...

MD21, MD22 check (Part1)

SEAKT	MD8	Permissible software limit switches
0	–	MD21 = -10^9, MD22 = $+10^9$
1	0	MD21 ≥ –VFBABS MD22 ≤ VFBABS MD21 < MD22
	1	0 ≤ MD21 < MD9 0 ≤ MD22 < MD9 MD21 ≠ MD22

MD21, MD22 check (Part2)

SEAKT	MD10	Permissible software limit switches
0	–	MD21 = -10^9, MD22 = $+10^9$
1	0, 1	–
	3, 13	MD22−MD21 ≤ UMWEG
	4, 14	MD22−MD21 ≤ MD14 · UMWEG

MD28 check

Permissible velocity:	10 ≤ MD28 ≤ MD23

MD29 check

MD10	Permissible velocity:
3, 4, 13, 14	any, not used
0, 1	10 ≤ MD29 ≤ MD23

MD31 check

MD30	MD10	Permissible directional reference of backlash
0		–
≠ 0	0, 1	
	3, 4, 13, 14	1, 2

MD34 check

Permissible:

BYTE0(MD34) ≠ BYTE1(MD34) ≠ BYTE2(MD34) ≠ BYTE3(MD34)

MD35 check

Permissible:

BYTE0(MD35)&0x7F ≠ BYTE1(MD35)&0x7F ≠ BYTE2(MD35)&0x7F ≠ BYTE3(MD35)&0x7F

MD53 check

MD53	Permissible MD52
0	0...100
1	MD52 < (10 V − MD43 [V]) · 100 / 10 V

5.3.2 Increments

DB structure

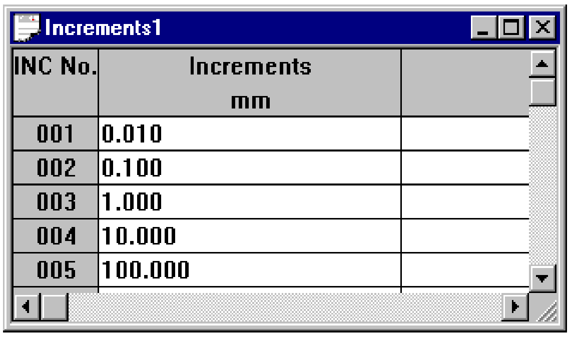
Table 5-5 gives you a general view of the structure of the “Increments” data block (DB-SM).

Table 5-5 DB structure – increments

Byte	Variable type	Value	Significance of the variables	Comment
			DB header	
0	WORD		Rack slot	Module address
2	WORD		DB No. (≥ 1000)	As in DB header
4	DWORD		Reserved	
8	WORD		Error No. (from FM)	With MMI services
10	WORD	1	Channel number	
12	2 STRING	SM	DB identifier/type	2 ASCII characters
16	DWORD	354	Module identifier	FM 354
20	4 CHAR	0	Version number/block number	(DB structure)
24	DWORD	1...3	Measurement-system grid per MD7	Unit of measurement
28	WORD	0/1	Parameter (DB) backup	Job via HMI
30	WORD		Reserved	
32	DWORD	0...10^9	Increment 1	
36	DWORD	0...10^9	Increment 2 to increment 100	see Section 9.2.5

Input of values

Values are input in the increments menu of the “Parameterize FM 354” parameterization tool.



INC No.	Increments mm	
001	0.010	
002	0.100	
003	1.000	
004	10.000	
005	100.000	

Fig. 5-5 Entering values for incremental dimensions

5.3.3 Tool offset data

DB structure

Table 5-6 gives you a general view of the structure of the “tool offset data” data block (DB-WK).

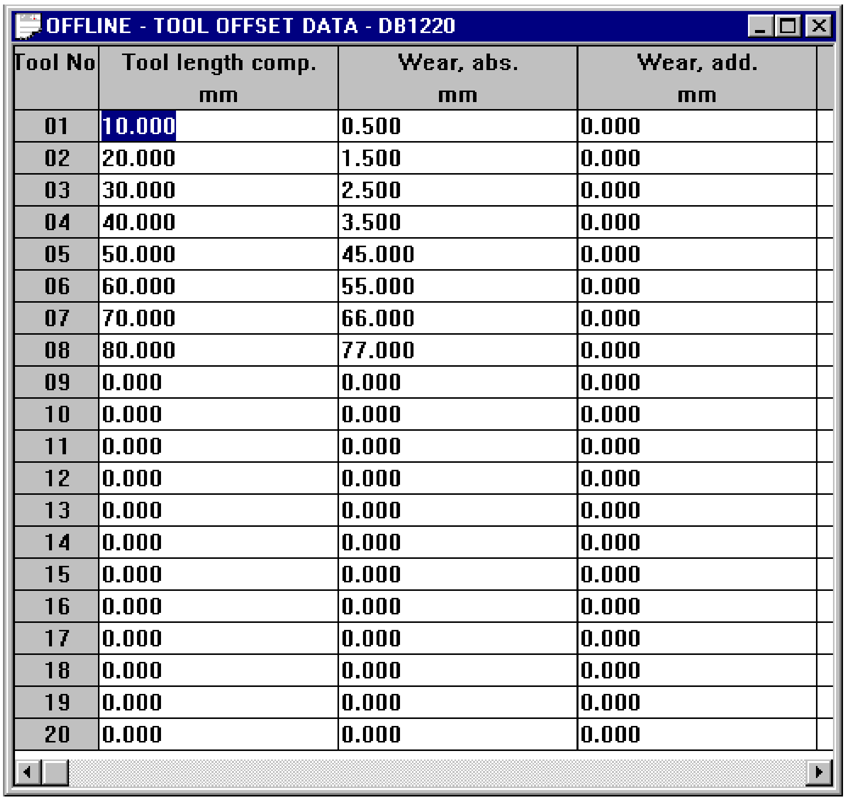
Table 5-6 DB structure – tool offset data

Byte	Variable type	Value	Significance of the variables	Comment
			DB header	
0	WORD		Rack slot	Module address
2	WORD		DB No. (≥ 1000)	As in DB header
4	DWORD		Reserved	
8	WORD		Error No. (from FM)	With MMI services
10	WORD	1	Channel number	
12	2 STRING	TO	DB identifier/type	2 ASCII characters
16	DWORD	354	Module identifier	FM 354
20	4 CHAR	0	Version number/block number	(DB structure)
24	DWORD	1...3	Measurement-system grid per MD7	Unit of measurement
28	WORD	0/1	Parameter (DB) backup	Job via MMI
30	WORD		Reserved	
32	DINT DINT DINT	$-10^9...10^9$ $-10^9...10^9$ $-10^9...10^9$	Tool length offset 1 Wear value 1 absolute Wear value 1 additive	Tool 1
44	DINT DINT DINT	$-10^9...10^9$ $-10^9...10^9$ $-10^9...10^9$	Tool length offset 2 Wear value 2 absolute Wear value 2 additive to Tool length offset 20 Wear value 20 absolute Wear value 20 additive	Tool 2 to Tool 20 see Section 10.1

Input of values

Values are input in the tool offset data menu of the “Parameterize FM 354” parameterization tool.

If the additive wear value is changed online, the FM calculates the new wear parameter as an absolute value and the additive tool wear is reset to 0.



OFFLINE - TOOL OFFSET DATA - DB1220

Tool No	Tool length comp.	Wear, abs.	Wear, add.
	mm	mm	mm
01	10.000	0.500	0.000
02	20.000	1.500	0.000
03	30.000	2.500	0.000
04	40.000	3.500	0.000
05	50.000	45.000	0.000
06	60.000	55.000	0.000
07	70.000	66.000	0.000
08	80.000	77.000	0.000
09	0.000	0.000	0.000
10	0.000	0.000	0.000
11	0.000	0.000	0.000
12	0.000	0.000	0.000
13	0.000	0.000	0.000
14	0.000	0.000	0.000
15	0.000	0.000	0.000
16	0.000	0.000	0.000
17	0.000	0.000	0.000
18	0.000	0.000	0.000
19	0.000	0.000	0.000
20	0.000	0.000	0.000

Fig. 5-6 Entering values for tool offset data

5.3.4 Traversing programs

DB structure

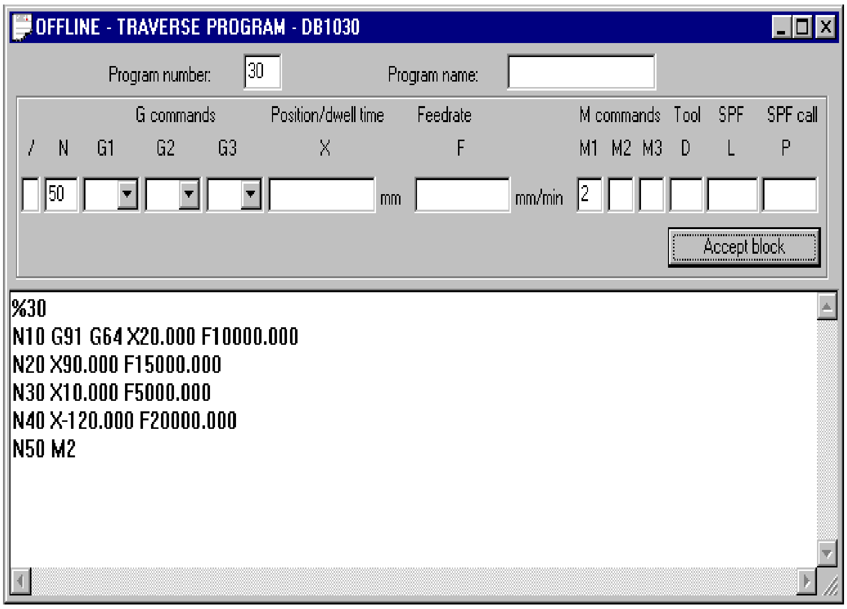
Table 5-7 gives you a general view of the structure of the “traversing programs” data block (DB-NC).

Table 5-7 DB structure – traversing programs

Byte	Variable type	Value	Significance of the variables	Comment
			DB header	
0	WORD		Rack slot	Module address
2	WORD		DB No. (≥ 1000)	As in DB header
4	DWORD		Reserved	
8	WORD		Error No. (from FM)	With MMI services
10	WORD	1	Channel number	
12	2 STRING	NC	DB identifier/type	2 ASCII characters
16	DWORD	354	Module identifier	FM 354
20	4 CHAR	0	Version number/block number	(DB structure)
24	DWORD	1...3	Measurement-system grid per MD7	Unit of measurement
28	WORD		Reserved	
30	WORD		Reserved	
32	18 STRING	ASCII characters	NC program name	max. 18 characters
52	STRUCT	NC block	NC block new (modification range)	
72	STRUCT	NC block	1st traversing block	
92	STRUCT	NC block	2nd to 100th traversing block	see Section 9.3.11, 10.1

Input of traversing programs

An empty window is provided for the input of NC traversing programs. Here you can input your traversing program as follows:



OFFLINE - TRAVERSE PROGRAM - DB1030

Program number: 30

Program name:

N: 50

M1: 2

```
%30
N10 G91 G64 X20.000 F10000.000
N20 X90.000 F15000.000
N30 X10.000 F5000.000
N40 X-120.000 F20000.000
N50 M2
```

Fig. 5-7 Entry for traversing programs

1. % Program number Program name

   The “%” can be input only in the first line. This input is mandatory. The DB number is formed from the program number.

   The program name is optional and may have up to 18 characters.

2. N<block number> – G<command> (G1, G2, G3) – X<value> – F<value> – M<command> (M1, M2, M3) – D<No.> (tool offset number) – L<No.> – P<No.> – (for NC programming see Chapter 10).

   – You must enter the block number (N) **first and in ascending order**. The rest of the inputs may be in any desired sequence.

   – Input separators as a blank.

You must enter characters in upper case letters.

You can also use the input area at the top of the screen. The program number and the program name are saved when you exit the input box. You can save the traversing blocks with the “Save Block” button.

5.4 Parameterization with “Parameterize FM 354”

Entering the values

You have a variety of options for entering your parameterization data.

1. User data

   You can input values or select texts in a table. Select input fields with the cursor and enter the values. You can select the associated texts for the values with the space key.

2. Machine data

   The values are entered in dialog boxes and windows selected by option tabs.

   To display the machine data in a table, select the menu **View > Table form**. Here you can enter the values as described in the user data section.

3. Tool compensation data and increment sizes

   You can input the values in a table. Select input fields with the cursor and enter the values.

4. Traversing programs

   Traversing programs are input in text format.

A comment column is included in the tables for MD, SM, and TO values. This comment is not stored in the data block. It can be printed out or stored with the data in the file on export.

5.5 Storing the parameter data in SDB ≥ 1 000

Overview

The FM 354 stores its parameter data internally.

In order to ensure that the parameter data are available if a fault develops on the FM 354 and no programming device/PC is at hand, the data can be stored in a system data block in the CPU (SDB ≥ 1 000). The CPU transfers the data stored in SDB ≥ 1 000 to the FM 354 on each new start. If the FM 354 has no machine data or the internal time stamp (time of creation) is invalid, the data are transferred from SDB ≥ 1 000 to the FM 354 and saved there.

The time stamp is renewed every time a DB (parameter initialization data) is opened and when a file is imported. If the contents of the DB are modified (for instance the machine data), a new time stamp is also generated when the DB is saved or loaded.

You must ensure that the parameter data in SDB ≥ 1 000 always match the parameter data on the FM 354 when start-up is complete.

Note

If parameter initialization data are modified again in the FM following creation of SDB ≥ 1 000, they are overwritten when the CPU is restarted (see “Time stamp”, above).

SDB ≥ 1 000 should therefore not be created until start-up is finished.

If you need to modify the data subsequently, you should generate SDB ≥ 1 000 again and load it into the CPU. You can delete the previous SDB before you load the new one, however the new SDB automatically overwrites the old one when it is generated. The old SDB and the new SDB do not have to be allocated the same number.

Creating the SDB

Prerequisite: Online connection with the FM 354

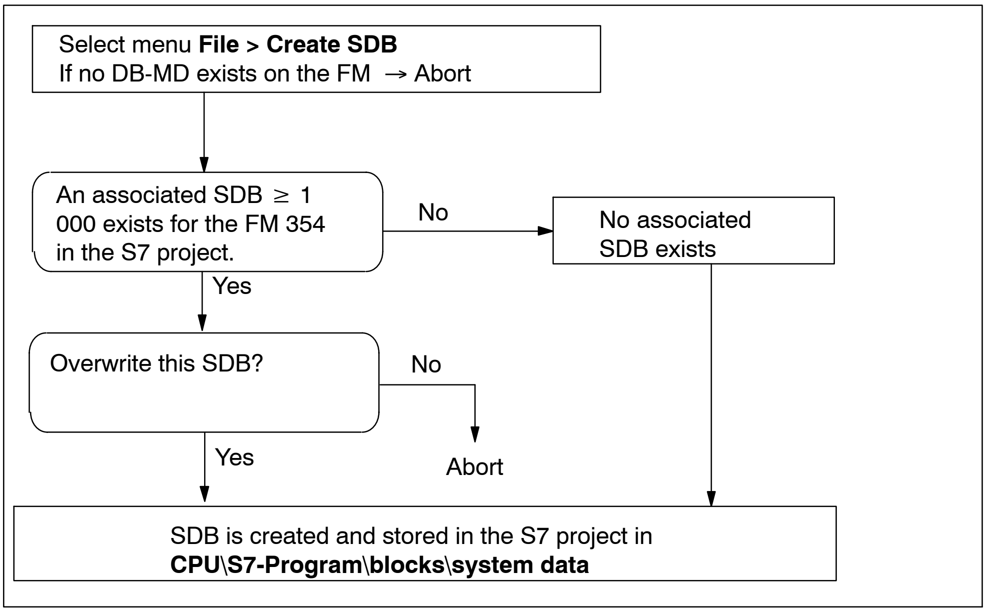

Select menu **File > Create SDB**
If no DB-MD exists on the FM → Abort
An associated SDB ≥ 1 000 exists for the FM 354 in the S7 project.
No
No associated SDB exists
Yes
Overwrite this SDB?
No
Yes
Abort
SDB is created and stored in the S7 project in **CPU\S7-Program\blocks\system data**

Fig. 5-8 Creating SDB ≥ 1 000

Display/delete SDB in the S7 project

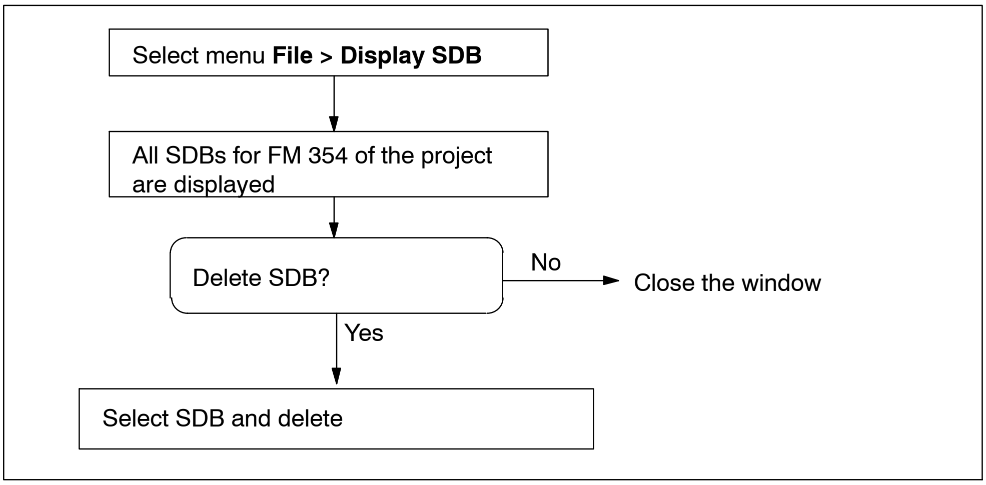

Select menu **File > Display SDB**
All SDBs for FM 354 of the project are displayed
Delete SDB?
No
Close the window
Yes
Select SDB and delete

Fig. 5-9 Displaying/deleting SDB ≥ 1 000

Loading the SDB in the CPU

When you have created the SDB, you must load the “system data” of the project into the CPU.

There are two ways of proceeding:

1. Method

   Select the online window in the *SIMATIC Manager* (the online and offline windows must be open)

   Copy the system data from the offline project in **CPU\S7-Program\Blocks\System data** into the online project (drag with the mouse or select Copy/Paste).

2. Method

   Select the system data in the *SIMATIC Manager* in **CPU\S7-Program\Blocks\System data**.

   Activate the menu **PLC** > **Load** (or the right mouse button) to load the system data into the CPU

   or

   Use the menu **PLC** > **Load in EPROM memory card on CPU**

You can also program the memory card for the CPU on a programming device/PC.

If the configuration is loaded from HW-CONFIG, this SDB is **not** loaded into the CPU.

Deleting SDBs in the CPU

To delete the SDBs in the CPU:

1. Select “Parameterize FM 354”.
2. Select menu **File** > **Display SDB**. Delete the SDB(s).
3. Close “Parameterize FM 354” and in the *SIMATIC Manager* in Online Project **select CPU\S7-Program\Blocks\System data**. Delete the system data.
4. Transfer the system data to the CPU again (see above)

■

Programming the Technological Functions 6

Chapter Overview

General remarks

The purpose of the function description of the blocks and of the interface is to illustrate the communication between the CPU and the FM 354 in the SIMATIC S7 programmable controller. The programmable blocks and the AW-DB (which is the interface to the FM 354) make it possible for you to write your user program to suit your particular application.

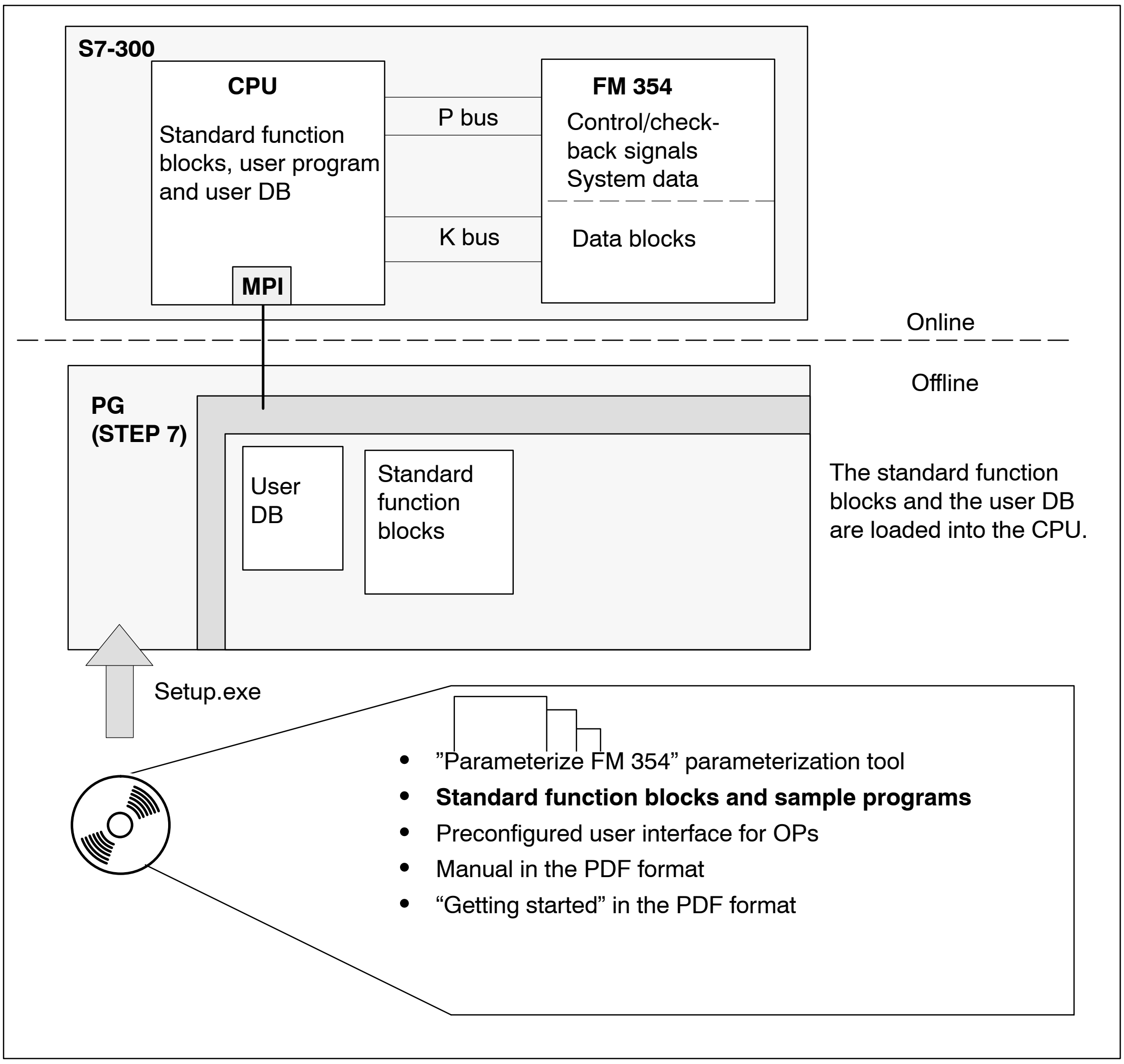

S7-300
CPU
Standard function blocks, user program and user DB
MPI
P bus
K bus
FM 354
Control/check-back signals
System data
Data blocks
Online
Offline
PG (STEP 7)
User DB
Standard function blocks
The standard function blocks and the user DB are loaded into the CPU.
Setup.exe
”Parameterize FM 354” parameterization tool
Standard function blocks and sample programs
Preconfigured user interface for OPs
Manual in the PDF format
“Getting started” in the PDF format

Fig. 6-1 Programming overview

Prerequisites

The following prerequisites must be fulfilled for the development of your user program if you want to control the FM 354:

- You must have installed the software on the PG/PC as per Section 5.1.

  The block libraries containing the basic functions are stored in the following directories by default:

  - Blocks of the “FMSTSV_L” library:
    [STEP7 directory]\S7LIBS\FMSTSV_L
  - Blocks of the “FM353_354” library (also for PROFINET, upon request):
    [STEP7 directory]\S7LIBS\FM353_354
- The link from PG/PC to the S7 CPU must be established (see Figure 4-1).
- You must have already created your project for the SIMATIC S7 (see “FM 354, First Steps”).

6.1 Programming fundamentals

Overview

In this chapter you will find information on the following:

- Communication between the CPU and the FM 354, Section 6.1.1, p. 6-4
- Structure of a user program, Section 6.1.2, p. 6-5
- Distributed use of OB 86, Section 6.1.3, p. 6-6
- Embedding an OP, Section 6.1.4, p. 6-6
- Procedure for creating a user program, Section 6.1.5, p. 6-7

6.1.1 Communication between the CPU and the FM 354

Embedding the FM 354 in the user program

The diagram below shows you how the FM 354, the user DB and the technological functions communicate.

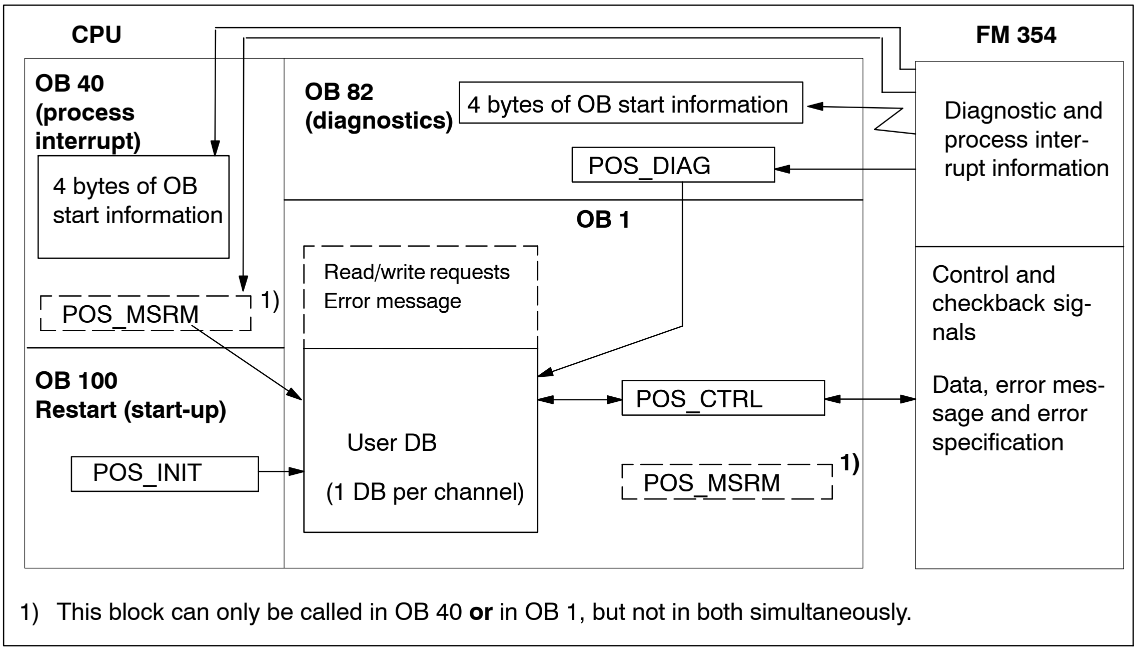
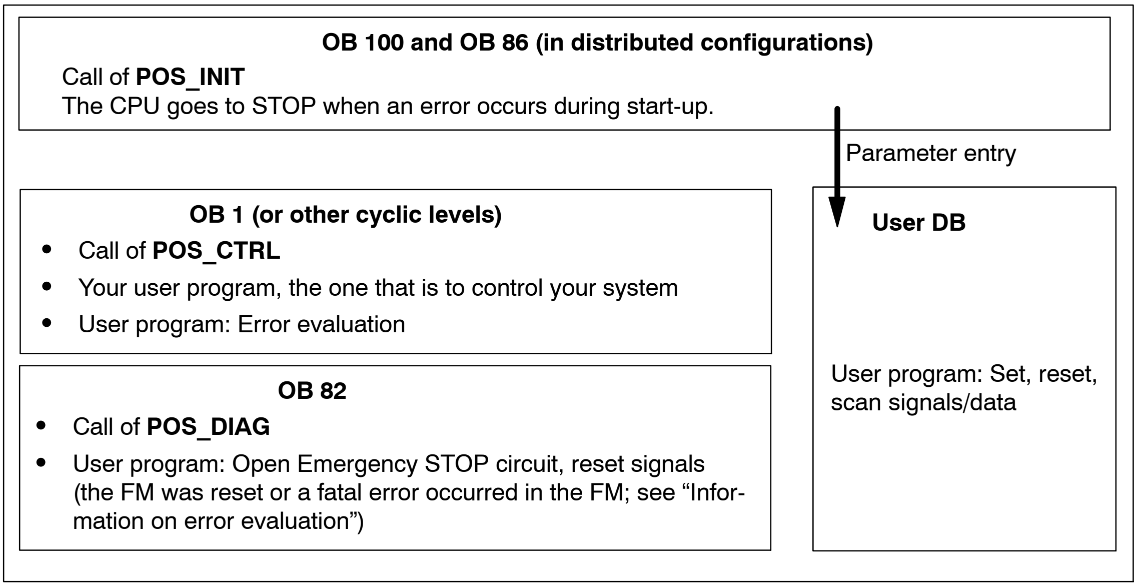

CPU
FM 354
OB 40
(process
interrupt)
4 bytes of OB start information
POS_MSRM
1)
OB 82
(diagnostics)
4 bytes of OB start information
POS_DIAG
OB 1
Read/write requests
Error message
User DB
(1 DB per channel)
POS_CTRL
POS_MSRM
1)
OB 100
Restart (start-up)
POS_INIT
Diagnostic and process interrupt information
Control and checkback signals
Data, error message and error specification
1) This block can only be called in OB 40 **or** in OB 1, but not in both simultaneously.

Fig. 6-2 Overview diagram for linking the FM 354 into the user program

6.1.2 Structure of a user program

The diagram below provides an overview of the structure of the user program (AWP).



OB 100 and OB 86 (in distributed configurations)

Call of **POS_INIT**
The CPU goes to STOP when an error occurs during start-up.

Parameter entry

OB 1 (or other cyclic levels)

- Call of **POS_CTRL**
- Your user program, the one that is to control your system
- User program: Error evaluation

OB 82

- Call of **POS_DIAG**
- User program: Open Emergency STOP circuit, reset signals (the FM was reset or a fatal error occurred in the FM; see “Information on error evaluation”)

User DB

User program: Set, reset, scan signals/data

Note

GET/PUT functions (SFC 72/73) from/to the FM are not guaranteed to work properly, that is to say, these functions are not supported, since they are not required. Parameter initialization data can be modified via the “Modify parameters/data” signal (user DB, DBX39.3).

Information on signal processing:

The FM 354’s cycle (= 2 ms) and the user cycle (OB 1) are asynchronous to one another. Depending on the instant of signal transfer to the FM 354, the time it takes to process signals may be = 1 to < 2 x the FM cycle. This must be taken into account particularly when user cycles are short. If necessary, the processing status of the FM 354 should be queried before activating a new action.

Please note that signal/data transmission takes longer/several user cycles in distributed configurations (see Section 6.9).

Information on testing the user program

When testing the user program with “Set breakpoint”, please note that it is not always possible to resume the program scan with the FM 354 after the breakpoint has been reached (for technical reasons).

For example, movements activated by the user program cannot be halted when the user program has reached the breakpoint.

The program can be resumed by executing a restart (CPU: STOP/RUN), by resetting the axis, or by changing the operating mode.

6.1.3 Distributed configuration, OB 86

If system operation must be maintained with an FM 354 despite a faulty link between CPU and distributed I/Os (DP), OB 86 is to be linked into the user program. In OB 86, provisions (e.g. set flag and evaluate in OB) must be made to prevent OB 1 communication with the FM 354 in case of a failure. To enable synchronization of the user program and the FM, the POS_INIT block (corresponding to execution in OB 100) is to be linked on re-establishing the link. Additionally, OB 122 (I/O access error) must be loaded in the CPU.

6.1.4 Embedding an OP

Part of the AW-DB, namely the “Data field for operator control/monitoring” (DBB496 to DBB515), is used to store signals/data for an OP as per the preconfigured operator interface. In order to initiate actions, the relevant signals/data have to be transferred to the interface (relevant area in the AW-DB) via the user program (see Section 6.7, example 4).

6.1.5 Procedure for writing the user program (AWP)

The sample projects “zDt13_02_FM353_EX” included in the configuring package (for the blocks of the “FMSTSV_L” library) and “zDt13_03_FM353_EX” (for blocks of the “FM353_354” library) will help you when creating your user program and can be used as a template.

Suggested procedure:

1. Open your project in the SIMATIC Manager.
2. Select **SIMATIC xxx > CPUxxx > S7 Program**.
3. In the SIMATIC Manager, select **File > Open > Projects** to load the “zDt13_02_FM353_EX” or the “zDt13_03_FM353_EX” project.
4. Select the “EXAMPLES” directory.
5. Select the “Symbols” file and copy it to your project under **SIMATIC xxx > CPUxxx > S7 Program** (replacing the existing object).
6. Open the “Sources” directory and copy from all STL sources to your project’s “Sources” directory.
7. Open the ”Blocks“ directory and copy all blocks from it into the ”Blocks” directory of your project.
8. Select the “Sources” directory in your project. Start the “LAD/STL/FBD Editor” by double-clicking on the “OB_EXAMPLE” file.
9. **Modify the appropriate input parameters (see “Block description”, Sections 6.3 and 6.4) in the POS_INIT call in OB 100 and in the POS_DIAG call in OB 82**.
10. **You can insert the relevant functions from the sample project “zDt13_02_FM353_EX” (see Section 6.7) in the “EXAMPLE CALLS” network in OB 1. The functions can be activated by writing your user program to set/reset the signals in the DB 100 (user DB for the examples) supplied.**

    The call for the POS_CTRL must be modified accordingly (input parameters or appropriate instance DB).
11. The organization blocks (OB 1, OB 82, OB 100) are generated from the STL source with the menu commands **File > Save** and **File > Compile** (warnings from the compilation run can be ignored).
12. Close the editor.
13. Set the CPU to “STOP” and switch the CPU on.
14. In the SIMATIC Manager, select **SIMATIC xxx > CPUxxx > S7 Program > Blocks**.
15. Load all the S7 blocks (including system data) into your CPU (with the CPU at STOP) with **PLC > Load**.

6.2 Putting the FM 354 into operation with the parameter initialization tool

To put the FM 354 into operation with the parameter initialization tool “Parameterize FM 354”, the CPU must be at “STOP”. It can also be at “RUN”, for example if you want to automate part of your plant or connect the drives, in which case the control/checkback signals “Switch P bus interface to start-up” (AW-DB, DBX14.1) and “Switching of P bus interface concluded” (AW-DB, DBX22.1) must be observed. For a description of these signals, see Section 9.1).

Also please observe Section 7.3 “Testing and Optimization”.

Note

Observe the relevant safety measures if you want to move the axis.

6.3 Standard function blocks of the “FMSTSV_L” block library

Overview

This chapter contains information on the following:

- Overview of the “FMSTSV_L” block library, Section 6.3.1, p. 6-9
- POS_INIT (FC 0) – Initializing the user DB, Section 6.3.2, p. 6-10
- POS_CTRL (FC 1) – Data exchange, Section 6.3.3, p. 6-12
- POS_DIAG (FC 2) – Reading diagnostic alarm data, Section 6.3.4, p. 6-22
- POS_MSRM (FC 3) – Reading measurement values, Sections 6.3.5, p. 6-25
- Interface, user data blocks (AW-DB), Section 6.3.6, p. 6-26

6.3.1 Overview of the “FMSTSV_L” function block library

You can use the blocks of the “FMSTSV_L” block library as follows:

- Central configuration of the FM
- Distributed configuration of the FM via PROFIBUS DP

The table below provides an overview of the functions (FC), data blocks (DB) and organization blocks (OB) required for communication and control of the FM 354.

Table 6-1 Standard function blocks of the “FMSTSV_L” library (overview)

Block	Block Name	Description/Function	Remarks
FC 0 Page 6-10	POS_INIT	Call in OB 100 and OB 83, start-up/initialization	Required for application, no. can be changed [1)]
FC 1 Page 6-12	POS_CTRL	Call in OB 1, cyclic operation (synchronization with FM 354) Basic functions and operating modes, interface processing, read and write requests	
FC 2 Page 6-22	POS_DIAG	Call in OB 82, internal errors, external errors, and external channel errors of the FM	
FC 3 Page 6-25	POS_MSRM	Call in OB 40 or OB 1, reading out of the measured values	To be used only if the function in question is required for the application in question; the number can be changed [1)]
DB (UDT)	AW-DB	Interface to the FM	Required for application
OB 1	–	Cyclic level	Required for application
OB 82	–	Diagnostic interrupt level	
OB 100	–	Start-up level	
OB 86	–	Rack failure	For distributed configurations
OB 122	–	I/O access error	

1) – **The block number is a default setting**, block number can be changed in the SIMATIC Manager
– Symbol table entries can be changed in conjunction with symbolic programming only

Note

The symbolic block identifier is used from here on.

6.3.2 The POS_INIT (FC 0) block – Initialization

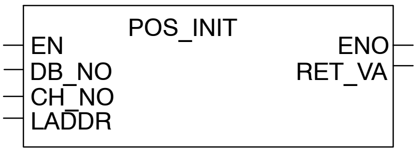
Function

Use the POS_INIT block to initialize specific areas of your AW-DB.

Call options

The POS_INIT block must be called once in the start-up OB 100 and in OB 86 for ”distributed configuration”.

Call in LAD representation (ladder diagram)	Call in STL representation (statement list)

	`CALL POS_INIT`
	`DB_NO :=`
	`CH_NO :=`
	`LADDR :=`

Parameters

The Table below lists the parameters for this block.

Name	Data Type	Param. Type	Description
DB_NO	INT	I	Data block number
CH_NO	BYTE	I	Number of the axis: 0 – Only one channel/axis on the module } Same meaning internally 1 – First channel/axis on the module } 2...255 – Illegal
LADDR	INT	I	• Logical base address of the module; use entry from “HW-CONFIG”, → “Properties”, → “Address” (see Section 5.2) • 0 – No entry of addresses in the user DB

Parameter types: I = input parameter

Function description

The POS_INIT block carries out the following actions:

1. Entry of addressing values in user DB AW-DB, If parameter LADDR ≠ 0

   Module address

2. Deletion of the following structures in user DB AW-DB:

   - Control signals
   - Checkback signals
   - Initiate, Ready and Error signals for the job requests
   - Single functions and single commands and their Ready and Error signals

3. If the input parameter LADDR = 0, no address is entered in the user DB. It is assumed that the addressing values (module address) have been entered manually via ”Parameterize FM 354” (button ”Enter Mod-Adr in user DB” in main display).

Error evaluation

An error is signalled by the binary result BR = 0 or by RET_VAL < 0.

Possible errors are:

- Unknown channel number CH_NO and DB no. = 0 as input parameters; the AW-DB is not initialized.
- If no AW-DB is found, the CPU goes to STOP; view the CPU’s diagnostic buffer.

RET_VAL	Error
–1	Unknown channel number
–2	DB number = 0

6.3.3 The POS_CTRL (FC 1) block – Data exchange

Function

The POS_CTRL block is the basic block for controlling the FM 354.

With the POS_CTRL block, you can:

- process read and write requests
- control the operatign modes (control and checkback signals)

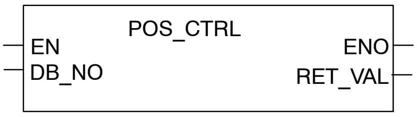
The POS_CTRL block performs the following actions:

1. Synchronization with the module (only then is the exchange of signals/data possible).
2. Reading of the checkback signals. The values/signals read by the POS_CTRL block are stored in the user data block AW-DB.
3. Transfer of the control signals from user data block AW-DB to the FM 354.
4. Carrying out of Write requests from user data block AW-DB, which includes the transfer of associated data from AW-DB and setting of the job status for the Write. Before the function is activated, all data required for the execution of the intended functions must be entered in AW-DB.
5. Carrying out of Read requests from user data block AW-DB, which includes transfer of the associated data to AW-DB and setting of the Read job status.
6. Automatic transfer of all single functions from user data block AW-DB to the FM 354 when one or more than one setting has been changed and setting of the Write job status (Set or Reset).
7. Automatic transfer of all single commands from user data block AW-DB to the FM 354 and setting of the Write job status. The single commands are reset following the transfer.
8. Automatic reading of the error number when an operator input error, traversing error or data error has occurred. The error number is entered in user data block AW-DB (DBB90 to DBB97) and the Read job status set.

Call options

POS_CTRL must be called cyclically (once in the OB 1 cycle, for instance) for each channel. Before calling the function, enter all data/signals required to execute the intended functions in user data block AW-DB.

Call in LAD Representation (ladder diagram)	**Call in STL Representation (statement list)**

	`CALL POS_CTRL`
	`DB_NO :=`
	`RET_VAL :=`

Parameters

The Table below lists the parameters for this block.

Name	Data Type	Param. Type	Description
DB_NO	INT	I	Data block number
RET_VAL	INT	Q	Return value

Parameter types: I = input parameter, Q = output parameter

Return values

The function returns the following values:

RET_VAL	BR	Description
1	1	At least 1 job/transfer in progress
0	1	No job/transfer in progress, no error
< 0	0	Error: • Data error (AW-DB, DBX22.4) • Communication error (AW-DB, DBW66)

Function description

The function works together with an AW-DB user data block. The DB number is passed to the function in the DB_NO parameter when the block is called.

- **Start-up**

  POS_CTRL acknowledges start-up of the module. During this time, the “RET_VAL” parameter and the “Write/Read job in progress” signals (AW-DB, DBX68.0 and DBX68.2) are TRUE.

- **Control and checkback signals**

  When POS_CTRL is called, the checkback signals are read from the FM 354 immediately (via direct access). Since the control signals and job requests are processed after these signals have been read, the checkback signals reflect the status of the module before the block was called. The control signals are also written to the FM 354 using direct access.

  Depending on the chosen mode, the control signals “Start, “Negative direction” and “Positive direction” (AW-DB, DBX15.0, 15.2 and 15.3) are reset once start-up has actually taken place (edge formation of the signals for the FM).

  For information on the generation of the checkback signals “Process” (AW-DB, DBX13.6) and “Position” (AW-DB, DBX13.7), see “Mode control”.

- **Job requests**

  Data interchange with the module that goes beyond control and checkback signals is handled using job requests. Simultaneously pending Write or Read requests, however, can only be executed in succession, whereby one Read and one Write request are processed in one call.

  To issue a request, set the relevant initiation signal in user data block AW-DB (DBB38 to DBB43). In the case of Write requests, you must also make the appropriate data available.

  The request is serviced when POS_CTRL is called. When the FM is used centralized, a Read request is performed during the call. A Write request requires at least three calls (or OB cycles) due to the acknowledgements required from the module. The time interval between the calls should exceed the length of one FM cycle.

  When a request has been serviced, the Initiate signal is removed (does not apply to single functions).

  The next job request is not determined or executed until the next block call has been made.

  For each job request, there is a Ready signal (AW-DB, DBX44.0 to 53.7) and an Error signal (AW-DB, DBX54.0 to 63.7) in addition to the Initiate signal.

  You should reset the Ready and Error signals for a job request following evaluation or prior to issuing the request.

- **Order in which job requests are serviced/priority**

  You may submit several job requests simultaneously, even together with Write requests for single commands and single functions.

  As soon as a Write request is detected (also on a signal change in the case of single functions), it is serviced immediately upon completion of the transfer currently in progress, if any. Be sure that signals for single commands are not set cyclically, as this could prevent other job requests from being serviced (priority).

  Order/priority of Write requests:

  1. Write single commands
  2. Write single functions
  3. Write requests.
     The Write requests are serviced in the order of the Initiate signals,
     which is stipulated in user data block AW-DB (from DBX38.0 to 39.7).

  Order/priority of Read requests:

  1. Read error code, operator/traverse errors or data errors
  2. Read requests
     The Read requests are serviced in the order of the Initiate signals,
     which is stipulated in user data block AW-DB (from DBX42.0 to 43.6).

- **Job request status**

  You can read the status of the job request in return value RET_VAL and in the “Write/read job in progress” signals in user data block AW-DB (DBX68.0 and DBX68.2). You can evaluate the status of an individual job request by evaluating the Initiate, Ready and Error signals for that job request.

Table 6-2 Job request status

Job Request Status	RET_VAL (integer)	Jobs in Progress (DBX68.0 DBX68.2)	Initiate Signals (DBB34...43)	Ready Signals (DBB44...53)	Error Signals (DBB54...63)
1. Job in progress	1	1	1	–	–
2. Job terminated without error	0	–	–	1	–
3. Write job terminated with error in this job	–1	–	–	1	1
4. Write job aborted or not executed	–1	–	–	–	1
5. Read job aborted	–2	–	–	–	1
6. Write and Read aborted or not executed (in the case of simultaneous job requests)	–3	–	–	–	1

– Irrelevant for error evaluation

Processing status

Signal	Description
Write not possible (AW-DB, DBX68.1)	= TRUE; Write request cannot be serviced in this cycle because: • The axis is not initialized • Test mode is enabled • No operating mode is active • The selected operating mode has not yet been set In these cases, you can leave the Write request pending or you can cancel it. The POS_CTRL block resets the signal when all of the above-listed conditions are fulfilled.
Read not possible (AW-DB, DBX68.3)	= TRUE; Read request cannot be serviced at this time because: • The axis is not initialized • No mode has been selected • Test mode is enabled In these cases, you can leave the Read request pending or you can cancel it. The POS_CTRL block resets the signal when all of the above-listed conditions are fulfilled.
Reset status/error (AW-DB, DBX69.1)	With this signal you can reset **all** Ready and Error signals prior to processing of the pending job requests. The signal itself is then reset by the block.

Error evaluation

Communication errors or data interpretation errors on the FM are flagged in the Binary Result (BR = 0) and by RET_VAL < 0; see job request status.

Possible errors are:

- Data transfer error (communication is not completed) during a transfer with SFC 58/59 “WR_REC / RD_REC”. The error code is made available in user data block AW-DB, DBW66 (RET_VAL value of these internal SFCs) (job states 4, 5 and 6, see Table 6-2, see also Error List, Section 6.8).
- Data transferred with Write are checked for data errors by the module and interpreted. If a data error occurs, the checkback signal “Data error” (AW-DB, DBX22.4) is set to TRUE in user data block AW-DB (message: “Write job terminated with error”). The error number, read out via an internal job request, is entered in user data block AW-DB, DBB94 and 95 (job status 3, Table 6-2).

  You will find more information on data errors in the parameter initialization tool under the menu command **Debug > Error Evaluation** and in Chapter 11.

Performance in the event of an error during the servicing of a **Write request** (does not apply to single functions and commands):

- The Initiate signal is removed for the errored request and the Error signal (AW-DB, DBX54.0 to 63.7) and Ready signal (AW-DB, DBX44.0 to 53.7) are set (job request status, point 3, Table 6-2).
- The Initiate signal is also removed for all pending Write requests and the Error signal set (job request status, point 4, Table 6-2).
- Any pending Read requests are serviced. The error code (AW-DB, DBW66) for each request is re-set if another error occurs.

Performance in the event of an error during the servicing of a **Read request**:

- The Initiate signal is removed for the errored Read and the Error signal set (job request status, point 5, Table 6-2).
- Any pending Read requests are serviced. The error code (AW-DB, DBW66) for each request is re-set if another error occurs.

Performance in the event of an error during servicing of **single functions and commands**:

- The Write request is not serviced in its entirety, and the Error signal is set (job request status, point 4, Table 6-2).
- The function set/reset which led to initiation of the Write request is not activated.

Processing write requests

Before Write requests can be processed, the data area associated with the Write request must first be initialized with the relevant values and the appropriate operating mode.

A Write request is initiated by setting the relevant job request number.

The following abbreviations are used in the Table below to indicate the adjacent operating mode:

Operating mode:			
	T	–	Jogging mode
	STE	–	Control mode
	REF	–	Approach to reference point
	SM	–	Incremental mode (relative)
	MDI	–	MDI (Manual Data Input)
	A/AE	–	Automatic mode / Automatic single block

The following Write requests are available:

Operating Mode System Data	Write Request	Data	T	STE	REF	SM	MDI	A/AE	See Sect.
Speed levels 1, 2	DBX38.0	DBB160...167	□	○	○	□	○	○	9.2.1
Frequency levels 1, 2	DBX38.1	DBB168...175	○	□	○	○	○	○	9.2.2
Setpoint for incremental dimension	DBX38.2	DBB156...159	○	○	○	□	○	○	9.2.5
MDI block	DBX38.3	DBB176...195	○	○	○	○	□	○	9.2.6
MDI block, on-the-fly	DBX38.4	DBB222...241	–	–	–	–	x	–	9.2.6
Reserved	DBX38.5								
Set reference point	DBX38.6	DBB152...155	x	x	x	x	x	–	9.3.9
Set actual value	DBX38.7	DBB144...147	x	x	–	x	x	x	9.3.5
Set on-the-fly actual value	DBX39.0	DBB148...151	x	x	–	x	x	–	9.3.6
Zero offset	DBX39.1	DBB140...143	x	x	–	x	x	x	9.3.4
Reserved	DBX39.2		x	x	x	x	x	x	
Modify parameters/data	DBX39.3	DBB196...219	x	x	x	x	x	x	9.3.1
Digital outputs	DBX39.4	DBB220...221	x	x	x	x	x	x	9.8.2
Program selection	DBX39.5	DBB242...245	–	–	–	–	–	□	9.2.7
Application request	DBX39.6	DBB246...249	x	x	x	x	x	x	9.3.7
Teach-in	DBX39.7	DBB250...251	x	–	–	x	x	–	9.3.8

○ Data are accepted, then processed in the relevant operating mode.
x Data are accepted or processed.
– Data are rejected with error (see error handling, Table 11-8, column 4, No. 1).
□ Data required to move the axis.

Processing read requests

A Read request is initiated by setting the relevant job request number. The relevant operating mode must be activated.

The following Read requests are available:

Operating Mode System Data	Read Request	Data	T	STE	REF	SM	MDI	A/AE	See Sect.
Basic operating data	DBX42.0	DBB310...333	x	x	x	x	x	x	9.3.11
Active NC block	DBX42.1	DBB342...361						x	9.3.12
Next NC block	DBX42.2	DBB362...381						x	
Actual value for block change	DBX42.3	DBB398...401						x	9.3.14
Service data	DBX42.4	DBB402...433	x	x	x	x	x	x	9.3.15
Operating error number	DBX42.5	DBB86...89	x	x	x	x	x	x	6.3.4
Suppl. operating data	DBX43.5	DBB434...442	x	x	x	x	x	x	9.3.16
Parameters/data	DBX43.3	DBB446...469	x	x	x	x	x	x	9.3.17
Digital inputs/outputs	DBX43.4	DBB220...221	x	x	x	x	x	x	9.8
Application data	DBX43.6	DBB382...397	x	x	x	x	x	x	9.3.13
Read measured values	DBX43.7	DBB486...497	x	x	x	x	x	x	9.3.10 6.3.5

x Data are accepted or processed.

Operating mode control

The operating modes are discussed in detail in Section 9.2, the control-/checkback signals and handling information in Section 9.1.

The user must write the control signals to the user data block (AW-DB). POS_CTRL transfers the control signals from user data block AW-DB to the FM 354 and the checkback signals from the FM 354 to user data block AW-DB. The FM must be initialized.

The Table below lists the control and checkback signals, with symbols in German and English.

Table 6-3 Control/checkback signals

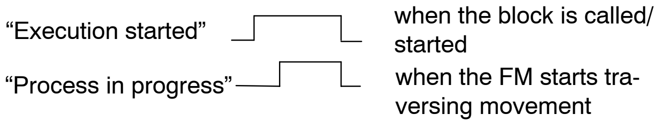
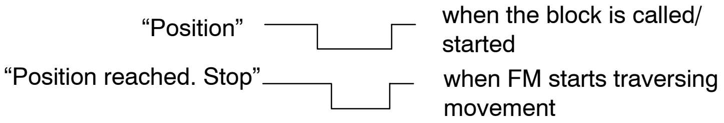
German	English	AW-DB	Description
Control signals			
TFB	TEST_EN	DBX14.1	Switch P bus interface to “Start-up”
BFQ/FSQ	OT_ERR_A	DBX14.3	Acknowledge operator errors and traversing errors
ST	START	DBX15.0	Start
STP	STOP	DBX15.1	Stop

Table 6-3 Control/checkback signals

German	English	AW-DB	Description
R–	DIR_M	DBX15.2	Negative direction
R+	DIR_P	DBX15.3	Positive direction
QMF	ACK_MF	DBX15.4	Acknowledge M function
EFG	READ_EN	DBX15.5	Read Enable
SA	SKIP_BLK	DBX15.6	Skip block
AF	DRV_EN	DBX15.7	Drive enable
BA	MODE_IN	DBB16	**Operating mode** **Code** Jog 01 Control 02 Approach to reference point 03 Incremental mode, relative 04 MDI 06 Automatic 08 Automatic single block 09
BP	MODE_TYPE	DBB17	**Operating mode parameters** **Code** Speed levels 1 and 2 Frequency levels 1 and 2 Incremental dimension selection 1...100, 254
OVERR	OVERRIDE	DBB18	Override
Checkback signals			
TFGS	TST_STAT	DBX22.1	Switching of P BUS interface completed
BF/FS	OT_ERR	DBX22.3	Operator-/traversing error
DF	DATA_ERR	DBX22.4	Data error
PARA	PARA	DBX22.7	Channel initialized
SFG	ST_ENBLD	DBX23.0	Start Enable
BL	WORKING	DBX23.1	Process in progress
WFG	WAIT_EI	DBX23.2	Wait for external Enable
T-L	DT_RUN	DBX23.5	Dwell time running
PBR	PR_BACK	DBX23.6	Reverse program scanning
BAR	MODE_OUT	DBB24	Active operating mode
SYN	SYNC	DBX25.0	Channel synchronized
ME	MSR_DONE	DBX25.1	End of measurement
FR–	GO_M	DBX25.2	Travel in negative direction
FR+	GO_P	DBX25.3	Travel in positive direction
SRFG	ST_SERVO	DBX25.4	Servo enable status
FIWS	FVAL_DONE	DBX25.5	Setting of on-the-fly actual value successfully completed
PEH	POS_RCD	DBX25.7	Position reached. Stop.
MNR	NUM_MF	DBB26	M function number
AMF	STR_MF	DBX27.4	M function modification

The checkback signals “Process in progress” and “Position reached. Stop” are not reported back to the user program until the FM has detected and processed the Start signal (≤ 2 FM cycles).

When calling the POS_CTRL block and the relevant control/checkback signals, the subsequent signals are generated so that starting of the procedure can be detected earlier than would otherwise be the case.

Signal	Description
Execution started (AW-DB, DBX13.6)	= TRUE When a mode/movement is started with the relevant control signals or when the status for “Process in progress” (AW-DB, DBX23.1) = 1  “Execution started” when the block is called/ started “Process in progress” when the FM starts traversing movement
Position (AW-DB, DBX13.7)	= FALSE When status signal “Position reached. Stop.” (AW-DB, DBX25.7) = 0 is returned or when a mode is started with the relevant control signals.  “Position” when the block is called/ started “Position reached. Stop” when FM starts traversing movement

Single functions and single commands are also requred to control the FM 354.

All single commands and single functions that are active when the POS_CTRL block is called are transferred. The single commands are cancelled following transfer, even in the event of an error.

Operating Modes System Data	Write Request	Function	T	STE	REF	SM	MDI	A/AE	See Sect.
Single functions	Internal	DBB34 and 35	□	x	□	□	□	□	9.3.2
Single commands	Internal	DBB36 and 37	x	x	x	x	x	x	9.3.3

x Data are accepted or processed.

□ Data required to move the axis.

The functions which can be activated in the FM using single settings or single commands are listed below.

Single Settings	Single Commands
Servo enable On-the-fly measuring Rotational speed monitoring Parking axis Simulation Length measuring Retriggering of reference point Reset Enable input Disable software limit position monitoring	Activate machine data Delete distance to go Automatic block return Automatic block advance Restart Rescind set actual value

Error messages from the FM

When an operator error, traversing error or data error occurs, the error number is read automatically via a Read request. The error number is entered in the AW-DB user data block and the Read status set.

An operating error, reported via a diagnostic interrupt, can be read out with the Read request “Operating error no.” (AW-DB, DBX42.5).

Table 6-4 Error messages from the FM

Error	Message	Error No.	Error Acknowledgement
Data error	Status signal (AW-DB, DBX22.4)	Is read out via Read request (AW-DB, DBB94 and 95)	New Write request
Operator/traversing error	Status signal (AW-DB, DBX22.3)	Is read out via Read request (AW-DB, DBB 90 and 91)	Set/reset control signal “Acknowledge operator/traversing error” (AW-DB, DBX14.3)
Diagnostic interrupt	Activated with OB 82, the data must be read out using the POS_DIAG block	In case of an operating error read out via the POS_DIAG block, the error no. is read out via the Read request DBX42.5 (AW-DB, DBB86 and 87)	Single command: Restart

For additional specific information, please refer to Chapter 11 “Error Handling”.

6.3.4 The POS_DIAG (FC 2) block – Read diagnostic interrupt data

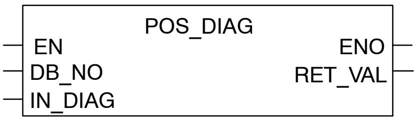
In the event of a fatal error, the FM 354 generates a diagnostic interrupt (OB 82 must be embedded in the user program and the interrupt parameterization of the FM 354 must be activated) and provides the relevant information to the local data area. For information on diagnostic interrupts, see Section 6.5.

For further information on external channel errors (operating errors), please call the POS_DIAG block.

Call options

POS_DIAG can be called in interrupt OB 82 or in OB 1.

Call in LAD representation (ladder diagram)	Call in STL representation (statement list)

	`CALL POS_DIAG`
	`DB_NO :=`
	`RET_VAL :=`
	`IN_DIAG :=`

Parameters

The Table below lists the parameters for the POS_DIAG block.

Name	Data type	Param. type	Description
DB_NO	INT	I	Data block number
RET_VAL	INT	Q	–1
IN_DIAG	BOOL	I/Q	Initiation signal for reading of the diagnostic data; is reset following execution of POS_DIAG.

Parameter types: I = input parameter, Q = output parameter,
I/Q = throughput parameter (initiation parameter)

Function description

The function works together with an AW-DB user data block. When the function is called, the AW-DB is forwarded with a DB_NO parameter.

Reading of the diagnostic data is started by setting the IN_DIAG parameter to TRUE. The block resets the parameter when the request has been serviced.

The IN_DIAG parameter remains set while the request is being serviced. Transfer of the data is terminated when the parameter is reset (IN_DIAG = FALSE).

Error evaluation

Errors are flagged in the Binary Result (BR = 0) and by RET_VAL < 0.

Possible errors are as follows:

Data transfer error during transfers with SFC 51 “RDSYSST”. The error is made available in the user DB AW-DB, DBW96 (see Error List, Section 6.8).

Diagnostic data

The prerequisite for the generation of a diagnostic interrupt is activation of the interrupt with the aid of the appropriate parameters (see Section 5.2).

If the user program does not contain an OB 82, the CPU goes to STOP.

The Table below contains the diagnostic information for the FM 354.

Table 6-5 Diagnostic information

Data Format	Message	User DB	Description
4 x 1 byte	When a diagnostic interrupt is generated, the relevant information is made available in the CPU (local data area, OB 82) and is entered in AW-DB by calling POS_DIAG.	DBX70.0	Module/group errors
		DBX70.1	Internal error/hardware error (group error bytes 72, 73)
		DBX70.2	External error
		DBX70.3	External channel error (group error byte 78)
		DBX70.6	Module not initialized
		DBX71.0...3	Module type class for FM 354 = 08H
		DBX71.4	Channel information available
		DBX72.1	Communication error (K bus)
		DBX72.3	Response from watchdog timer
		DBX72.4	Internal supply voltage to the module failed (NMI)
		DBX73.2	FEPROM error
		DBX73.3	RAM error
		DBX73.6	Process interrupt lost

Table 6-5 Diagnostic information, continued

Data Format	Message	User DB	Description
10 x byte	When the POS_DIAG block is called, the information (incl. bytes 0 to 3) is read and entered in the AW-DB (DBB70).	DBB74	FM pos. ID (74H)
		DBB75	Length of the diagnostic information (16)
		DBB76	Number of channels (1)
		DBX77	Channel vector error (1)
		DBX78.0...5	Unassigned
	When an operating error occurs, the error number can be read via Read request DBX42.5 (AW-DB, DBB86 to 89)	DBX78.7	**Operating error** (see Chapter 11, Troubleshooting)
		DBB79...83	Unassigned

Tips for the user

Following a diagnostic interrupt, the diagnostic information and the associated module address (OB82_MDL_ADDR) is made available in the local data area of OB 82 for quick analysis.

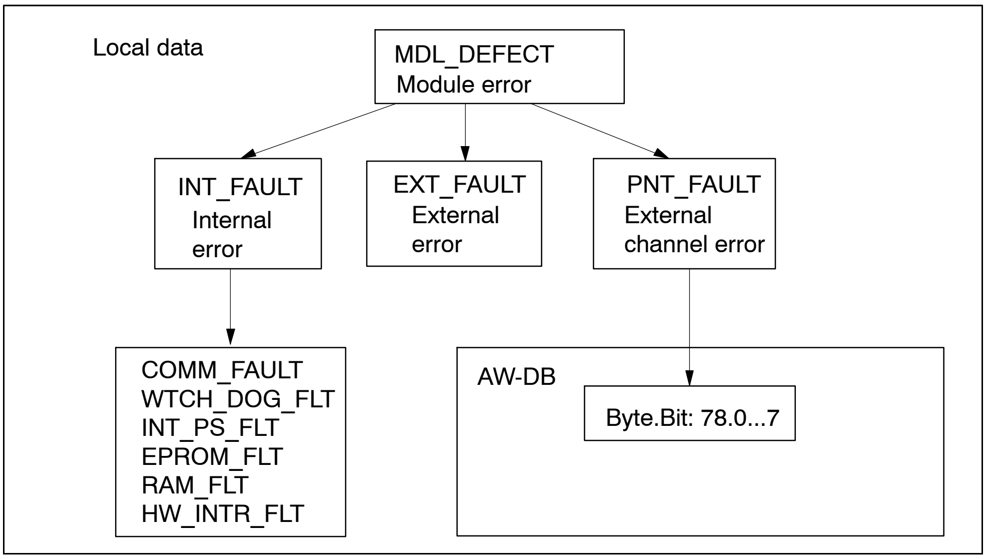


Fig. 6-3 Evaluating diagnostic data

6.3.5 The POS_MSRM (FC 3) block – Read measured values

Function

Use the POS_MSRM block to read the measured values from the AW-DB user data block.

For information on process interrupts, please see Section 6.5.

For information on measured values 9.3.10.

Note

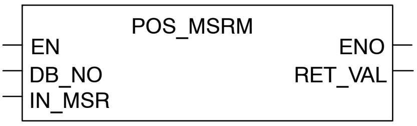
The measured values can also be read by means of POS_CTRL (Read request). If more than one Read request is pending, this request will be processed in the relevant order.

If the POS_MSRM block is called, the measured values are obtained irrespective of any other Read requests currently present.

Call options

POS_MSRM can be called in OB 40 (only when FM is used centralized) if the process interrupt was triggered (see Section 5.2), or in OB 1. POS_MSRM must not be called in both OBs at the same time.

Call in LAD representation (ladder diagram)	Call in STL representation (statement list)

	`CALL  POS_MSRM`
	`      DB_NO      :=`
	`      RET_VAL    :=`
	`      IN_MSR     :=`

Parameters

The Table below lists the parameters for the POS_MSRM block.

Name	Data type	Param. type	Description
DB_NO	INT	I	Data block number
RET_VAL	INT	Q	−1
IN_MSR	BOOL	I/Q	Start Read

Parameter types: I = input parameter, Q = output parameter,
I/Q = throughput parameter (initiation parameter)

Function description

The function works together with an AW-DB user data block. When the function is called, the DB number is forwarded in the DB_NO parameter.

Reading of the measured value is started by setting the IN_MSR parameter to TRUE. When the function has executed, the block resets the parameter.

The IN_MSR parameter remains set while the function is executing. Transfer of the data is terminated is complete when the parameter is reset (IN_MSR = FALSE).

Error evaluation

Errors are flagged in the Binary Result (BR = 0) and by RET_VAL < 0.

Possible errors are as follows:

Data transfer errors during transfers with SFC 59 “RD_REC”. The error is made available in the user data block AW-DB, DBW98 (see Error List, Section 6.8).

6.3.6 Interface, user data blocks (AW-DBs)

The AW-DB (interface) is created off-line.

The user can access the signals and/or data on the interface using absolute or symbolic addresses (creation of the AW-DB with UDT structure).

The interface is allocated to the relevant channel/axis via the standard function blocks’ DB_NO input parameter. The module address is part of the user DB. It is entered by POS_INIT or manually via ”Parameterize FM 354” (button ”Enter Mod-Adr in user DB“ in main display). The user DB must already exist.

Creating the AW-DB (offline)

Proceed as follows:

1. Open your project and select **SIMATIC xxx > CPUxxx > S7 Program > Blocks**.
2. The data block (for example DB 1) is generated under STEP 7 with the menu command **Insert > S7 Block > Data Block**.
3. The LAD/STL/FBD editor is started by double-clicking on this data block.
4. In the “New data block” dialog, select “Data block with assigned user-specific data type”.
5. UDT 1 is displayed.

   UDT 1 contains the structure of the AW-DB.
6. Select UDT 1 and confirm with **OK**.
7. You have now created the AW-DB.
8. Save this AW-DB with **File > Save**.
9. Close the editor.

Information about symbolic programming

Normally, the blocks are entered in the symbol table with the symbol name, address, and data type (the symbol table is supplied in the project and in the library). If you change the block number in your project with the SIMATIC Manager, the numbering in the symbol table must also be changed. Block allocation via the symbol table is always absolutely unique.

Before writing and compiling your user program, you must enter the blocks (AW-DBs, FCs) which you are using for your particular configuration in the symbol table. The symbolic structure of the interface is stored in the UDT block provided. The symbolic relationship is established via your STEP 7 project, the symbol table, and the UDT block.

Appendix B shows the UDT with symbols and absolute address.

Sample symbol table:

Symbol	Address	Data Type	Comments
DB_FM	DB 1	UDT 1	User DB for the FM 354
POS_INIT	FC 0	FC 0	User DB initialization
POS_CTRL	FC 1	FC 1	Data interchange

6.4 Standard function blocks of the “FM353_354” block library (also for PROFINET, upon request)

Overview

The present chapter provides the following information:

- Overview of the “FM353_354” block library, Section 6.4.1, p. 6-28
- POS_INIT (FC 0) – Initializing the user data block (AW-DB), Section 6.4.2, p. 6-30
- POS_CTRL (FB 1) – Data exchange, Section 6.4.3, p. 6-30
- POS_DIAG (FC 2) – Reading diagnostic interrupt data, Section 6.4.4, p. 6-41
- POS_MSRM (FB 3) – Reading measured values, Section 6.4.5, p. 6-41
- Interface, user data blocks (AW-DB), Section 6.4.6, p. 6-42

6.4.1 Overview of the “FM353_354” block library

You can use the blocks of the “FM353_354” block library as follows:

- centralized configuration of the FM
- distributed configuration of the FM via PROFIBUS DP
- distributed configuration of the FM via PROFINET

Note

The blocks of the “FM353_354” block library requires the following firmware version on the CPU:

- CPU modules with MMC with firmware V2.0 and higher
- CPU318 with firmware V3.0 and higher

The table below provides an overview of the functions (FC), function blocks (FB), data blocks (DB) and organization blocks (OB) required for communication and control of the FM 354.

Table 6-6 Standard function blocks of the FM353_354 library (overview)

Block	Block Name	Description/Function	Remarks
FC 0 Page 6-30	**POS_INIT**	Call in OB 100 and OB 86, start-up/initialization	Required for application, number can be changed [1]
FB 1 Page 6-30	**POS_CTRL**	Call in OB 1, cyclic operation (synchronization with the FM 354) Basic functions and operating modes, interface processing, read and write requests	
FB 2 Page 6-41	**POS_DIAG**	Call in OB 82, internal errors, external errors, and external channel errors of the FM	
FB 3 Page 6-41	**POS_MSRM**	Call in OB 40 or OB 1, reading out of the measured values	To be used only if the function in question is required for the application in question; the number can be changed [1]
DB 1	**IFFM_ICTRL**	Interface to the FM and instance DB for POS_CTRL (FB 1)	Required for application, number can be changed [1]
DB 3	**IMSRM**	Instance DB for POS_MSRM (FB 3)	Required for application if POS_MSRM is called; number can be changed [1]
OB 1	–	Cyclic level	Required for application
OB 82	–	Diagnostic interrupt level	
OB 100	–	Start-up level	
OB 86	–	Subrack failure	For distributed configurations
OB 122	–	I/O access error	

[1] – **The block number is a default setting**, block number can be changed in the SIMATIC Manager
– Symbol table entries can only be changed in conjunction with symbolic programming

Note

The symbolic block identifier is used from here on.

6.4.2 The POS_INIT (FC 0) block – Initialization

Block description

See Section 6.3.2.

6.4.3 The POS_CTRL (FC 1) block – Data exchange

Function

The POS_CTRL block is the basic block for controlling the FM 354.

With the POS_CTRL block, you can:

- process read and write requests
- control the operating modes (control and checkback signals)

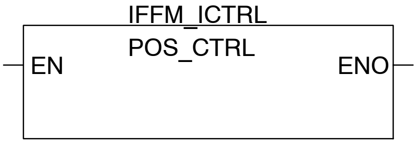
The POS_CTRL block performs the following actions:

1. Synchronization with the module (only then is the exchange of signals/data possible).
2. Reading of the checkback signals. The values/signals read by the POS_CTRL block are stored in the user data block AW-DB.
3. Transfer of the control signals from user data block AW-DB to the FM 354.
4. Carrying out of Write requests from user data block AW-DB, which includes the transfer of associated data from AW-DB and setting of the job status for the Write. Before the function is activated, all data required for the execution of the intended functions must be entered in AW-DB.
5. Carrying out of Read requests from user data block AW-DB, which includes transfer of the associated data to AW-DB and setting of the Read job status.
6. Automatic transfer of all single functions from user data block AW-DB to the FM 354 when one or more than one setting has been changed and setting of the Write job status (Set or Reset).
7. Automatic transfer of all single commands from user data block AW-DB to the FM 354 and setting of the Write job status. The single commands are reset following the transfer.
8. Automatic reading of the error number when an operator input error, traversing error or data error has occurred. The error number is entered in user data block AW-DB (DBB90 to DBB97) and the Read job status set.

Call options

The POS_CTRL block requires an instance data block (DB) when called. The DB IFFM_ICTRL (DB1) is part of the “FM353_354” library and at the same time includes the user interface. For more information, please refer to Section 6.4.6.

You must call the POS_CTRL block cyclically (e.g. once in the OB 1 cycle). Before you call the function, enter all data/signals in the user DB (AW-DB) which are required to execute the intended functions.

Call in LAD representation **(ladder diagram)**	**Call in STL representation** **(statement list)**
	`CALL  POS_CTRL, IFFM_ICTRL`

Parameters

The POS_CTRL block possesses neither input, nor output parameters.

Return values

The POS_CTRL block delivers the following return values in the RETVAL instance variable:

RET_VAL	BR	Description
1	1	At least 1 job/transfer in progress
0	1	No job/transfer in progress, no error
< 0	0	Error: • Data error (AW-DB, DBX22.4) • Communication error (AW-DB, DBW66)

Function description

The block operates with an AW-DB user data block. The DB number is determined when calling the FB using a transferred instance DB.

- **Start-up**

  POS_CTRL acknowledges start-up of the module. During this time, the RETVAL parameter and the “Write/Read job in progress” signals (AW-DB, DBX68.0 and DBX68.2) are TRUE.

- **Control and checkback signals**

  When POS_CTRL is called, the checkback signals are read from the FM 354 immediately (via direct access). Since the control signals and job requests are processed after these signals have been read, the checkback signals reflect the status of the module before the block was called. The control signals are also written to the FM 354 using direct access.

  Depending on the chosen mode, the control signals “Start, “Negative direction” and “Positive direction” (AW-DB, DBX15.0, 15.2 and 15.3) are reset once start-up has actually taken place (edge formation of the signals for the FM).

  For information on the generation of the checkback signals “Process” (AW-DB, DBX13.6) and “Position” (AW-DB, DBX13.7), see “Mode control”.

- **Job requests**

  Data interchange with the module that goes beyond control and checkback signals is handled using job requests. Simultaneously pending Write or Read requests, however, can only be executed in succession, whereby one Read and one Write request are processed in one call.

  To issue a request, set the relevant initiation signal in user data block AW-DB (DBB38 to DBB43). In the case of Write requests, you must also make the appropriate data available.

  The request is serviced when POS_CTRL is called. When the FM is used centralized, a Read request is performed during the call. A Write request requires at least three calls (or OB cycles) due to the acknowledgements required from the module. The time interval between the calls should exceed the length of one FM cycle.

  When a request has been serviced, the Initiate signal is removed (does not apply to single functions).

  The next job request is not determined or executed until the next block call has been made.

  For each job request, there is a Ready signal (AW-DB, DBX44.0 to 53.7) and an Error signal (AW-DB, DBX54.0 to 63.7) in addition to the Initiate signal.

  You should reset the Ready and Error signals for a job request following evaluation or prior to issuing the request.

- **Order in which job requests are serviced/priority**

  You may submit several job requests simultaneously, even together with Write requests for single commands and single functions.

As soon as a Write request is detected (also on a signal change in the case of single functions), it is serviced immediately upon completion of the transfer currently in progress, if any. Be sure that signals for single commands are not set cyclically, as this could prevent other job requests from being serviced (priority).

Order/priority of Write requests:

1. Write single commands
2. Write single functions
3. Write requests.
   The Write requests are serviced in the order of the Initiate signals, which is stipulated in user data block AW-DB (from DBX38.0 to 39.7).

Order/priority of Read requests:

1. Read error code, operator/traverse errors or data errors
2. Read requests
   The Read requests are serviced in the order of the Initiate signals, which is stipulated in user data block AW-DB (from DBX42.0 to 43.6).

- **Job request status**

  You can read the status of the job processing infrom the RETVAL instance variable of the block and from the "Write/read job in progress" signals in user data block AW-DB (DBX68.0 and DBX68.2). You can evaluate the status of an individual job request by evaluating the Initiate, Ready and Error signals for that job request.

Table 6-7 Job request status

Job request status	RETVAL instance variable (integer)	Jobs in progress (DBX68.0 DBX68.2)	Initiate signals (DBB34...43)	Ready signals (DBB44...53)	Error signals (DBB54...63)
1. Job in progress	1	1	1	–	–
2. Job completed without errors	0	–	–	1	–
3. Write job completed with errors in this job	–1	–	–	1	1
4. Write job aborted or not executed	–1	–	–	–	1
5. Read job aborted	–2	–	–	–	1
6. Write and Read aborted or not executed (in case of simultaneous job requests)	–3	–	–	–	1

– Irrelevant for error evaluation

Processing status

Signal	Description
Write not possible (AW-DB, DBX68.1)	= TRUE; Write request cannot be processed in this cycle because: • The axis is not initialized • Test mode is enabled • No operating mode is active • The selected operating mode has not yet been set In these cases, you can leave the Write request pending or you can cancel it. The POS_CTRL block resets the signal when all of the above-listed conditions are fulfilled.
Read not possible (AW-DB, DBX68.3)	= TRUE; Read request cannot be serviced at this time because: • The axis is not initialized • No mode has been selected • Test mode is enabled In these cases, you can leave the Read request pending or you can cancel it. The POS_CTRL block resets the signal when all of the above-listed conditions are fulfilled.
Reset status/error (AW-DB, DBX69.1)	With this signal you can reset **all** Ready and Error signals prior to processing of the pending job requests. The signal itself is then reset by the block.

Error evaluation

Communication errors or data interpretation errors on the FM are flagged in the Binary Result (BR = 0) and by instance variable RETVAL < 0; see job request status.

Possible errors are:

- Data transfer error (communication is not completed) during a transfer with SFC 58/59 “RDREC / WRREC”. The error code is made available in user data block AW-DB, DBW66 (RET_VAL value of these internal SFBs) (job states 4, 5 and 6, see Table 6-7, see also Error List, Section 6.8).
- Data transferred with Write are checked for data errors by the module and interpreted. If a data error occurs, the checkback signal “Data error” (AW-DB, DBX22.4) is set to TRUE in user data block AW-DB (message: “Write job terminated with error”). The error number, read out via an internal job request, is entered in user data block AW-DB, DBB94 and 95 (job status 3, Table 6-2).

  You will find more information on data errors in the parameter initialization tool under the menu command **Debug > Error Evaluation** and in Chapter 11.

Performance in the event of an error during the servicing of a **Write request** (does not apply to single functions and commands):

- The Initiate signal is removed for the errored request and the Error signal (AW-DB, DBX54.0 to 63.7) and Ready signal (AW-DB, DBX44.0 to 53.7) are set (job request status, point 3, Table 6-7).
- The Initiate signal is also removed for all pending Write requests and the Error signal set (job request status, point 4, Table 6-7).
- Any pending Read requests are serviced. The error code (AW-DB, DBW66) for each request is re-set if another error occurs.

Performance in the event of an error during the servicing of a **Read request**:

- The Initiate signal is removed for the errored Read and the Error signal set (job request status, point 5, Table 6-7).
- Any pending Read requests are serviced. The error code (AW-DB, DBW66) for each request is re-set if another error occurs.

Performance in the event of an error during servicing of **single functions and commands**:

- The Write request is not serviced in its entirety, and the Error signal is set (job request status, point 4, Table 6-7).
- The function set/reset which led to initiation of the Write request is not activated.

Processing write requests

Before Write requests can be processed, the data area associated with the Write request must first be initialized with the relevant values and the appropriate operating mode.

A Write request is initiated by setting the relevant job request number.

The following abbreviations are used in the Table below to indicate the adjacent operating mode:

Operating mode:			
	T	–	Jogging mode
	STE	–	Control mode
	REF	–	Approach to reference point
	SM	–	Incremental mode (relative)
	MDI	–	MDI (Manual Data Input)
	A/AE	–	Automatic mode / Automatic single block

The following Write requests are available:

Operating Mode System Data	Write Request	Data	T	STE	REF	SM	MDI	A/AE	See Sect.
Speed levels 1, 2	DBX38.0	DBB160...167	□	○	○	□	○	○	9.2.1
Frequency levels 1, 2	DBX38.1	DBB168...175	○	□	○	○	○	○	9.2.2
Setpoint for incremental dimension	DBX38.2	DBB156...159	○	○	○	□	○	○	9.2.5
MDI block	DBX38.3	DBB176...195	○	○	○	○	□	○	9.2.6
MDI block, on-the-fly	DBX38.4	DBB222...241	–	–	–	–	x	–	9.2.6
Reserved	DBX38.5								9.3.9
Set reference point	DBX38.6	DBB152...155	x	x	x	x	x	–	9.3.9
Set actual value	DBX38.7	DBB144...147	x	x	–	x	x	x	9.3.5
Set on-the-fly actual value	DBX39.0	DBB148...151	x	x	–	x	x	–	9.3.6
Zero offset	DBX39.1	DBB140...143	x	x	–	x	x	x	9.3.4
Reserved	DBX39.2		x	x	x	x	x	x	9.8
Modify parameters/data	DBX39.3	DBB196...219	x	x	x	x	x	x	9.3.1
Digital outputs	DBX39.4	DBB220...221	x	x	x	x	x	x	9.8.2
Program selection	DBX39.5	DBB242...245	–	–	–	–	–	□	9.2.7
Application request	DBX39.6	DBB246...249	x	x	x	x	x	x	9.3.7
Teach-in	DBX39.7	DBB250...251	x	–	–	x	x	–	9.3.8

○ Data are accepted, then processed in the relevant operating mode.
x Data are accepted or processed.
– Data are rejected with error (see Error Handling, Table 11-8 column 4, No. 1).
□ Data required to move the axis.

Processing read requests

A Read request is initiated by setting the relevant job request number. The relevant operating mode must be activated.

The following Read requests are available:

Operating Mode System Data	Read Request	Data	T	STE	REF	SM	MDI	A/AE	See Sect.
Basic operating data	DBX42.0	DBB310...333	x	x	x	x	x	x	9.3.11
Active NC block	DBX42.1	DBB342...361						x	9.3.12
Next NC block	DBX42.2	DBB362...381						x	
Actual value for block change	DBX42.3	DBB398...401						x	9.3.14
Service data	DBX42.4	DBB402...433	x	x	x	x	x	x	9.3.15
Operating error number	DBX42.5	DBB86...89	x	x	x	x	x	x	6.3.4
Suppl. operating data	DBX43.5	DBB434...442	x	x	x	x	x	x	9.3.16
Parameters/data	DBX43.3	DBB446...469	x	x	x	x	x	x	9.3.17
Digital inputs/outputs	DBX43.4	DBB220...221	x	x	x	x	x	x	9.3.8
Application data	DBX43.6	DBB382...397	x	x	x	x	x	x	9.3.13
Read measured values	DBX43.7	DBB486...497	x	x	x	x	x	x	9.3.10 6.3.5

x Data are accepted or processed.

Operating mode control

The operating modes are discussed in detail in Section 9.2, the control-/checkback signals and handling information in Section 9.1.

The user must write the control signals to the user data block (AW-DB). The POS_CTRL block transfers the control signals from user data block AW-DB to the FM 354 and the checkback signals from the FM 354 to user data block AW-DB. The FM must be initialized.

The Table below lists the control and checkback signals, with symbols in German and English.

Table 6-8 Control/checkback signals

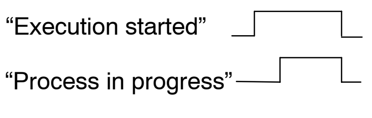
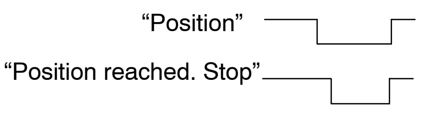
German	English	AW-DB	Description
Control signals			
TFB	TEST_EN	DBX14.1	Switch P bus interface to “Start-up”
BFQ/FSQ	OT_ERR_A	DBX14.3	Acknowledge operator and traversing errors
ST	START	DBX15.0	Start
STP	STOP	DBX15.1	Stop
R–	DIR_M	DBX15.2	Negative direction

Table 6-8 Control/checkback signals

German	English	AW-DB	Description
R+	DIR_P	DBX15.3	Positive direction
QMF	ACK_MF	DBX15.4	Acknowledge M function
EFG	READ_EN	DBX15.5	Read Enable
SA	SKIP_BLK	DBX15.6	Skip block
AF	DRV_EN	DBX15.7	Drive enable
BA	MODE_IN	DBB16	**Operating mode** **Code** Jog 01 Control 02 Approach to reference point 03 Incremental mode, relative 04 MDI 06 Automatic 08 Automatic single block 09
BP	MODE_TYPE	DBB17	**Operating mode parameters** **Code** Speed levels 1 and 2 Frequency levels 1 and 2 Incremental dimension selection 1...100, 254
OVERR	OVERRIDE	DBB18	Override
Checkback signals			
TFGS	TST_STAT	DBX22.1	Switching of P BUS interface completed
BF/FS	OT_ERR	DBX22.3	Operator-/traversing error
DF	DATA_ERR	DBX22.4	Data error
PARA	PARA	DBX22.7	Channel initialized
SFG	ST_ENBLD	DBX23.0	Start Enable
BL	WORKING	DBX23.1	Process in progress
WFG	WAIT_EI	DBX23.2	Wait for external Enable
T-L	DT_RUN	DBX23.5	Dwell time running
PBR	PR_BACK	DBX23.6	Reverse program scanning
BAR	MODE_OUT	DBB24	Active operating mode
SYN	SYNC	DBX25.0	Channel synchronized
ME	MSR_DONE	DBX25.1	End of measurement
FR–	GO_M	DBX25.2	Travel in negative direction
FR+	GO_P	DBX25.3	Travel in positive direction
SRFG	ST_SERVO	DBX25.4	Servo enable status
FIWS	FVAL_DONE	DBX25.5	Setting of on-the-fly actual value successfully completed
PEH	POS_RCD	DBX25.7	Position reached. Stop.
MNR	NUM_MF	DBB26	M function number
AMF	STR_MF	DBX27.4	M function modification
ACT_POS	ACT_POS	DBD28	Actual positon

The checkback signals “Process in progress” and “Position reached. Stop” are not reported back to the user program until the FM has detected and processed the Start signal (≤ 2 FM cycles).

When calling the POS_CTRL block and the relevant control/checkback signals, the subsequent signals are generated so that starting of the procedure can be detected earlier than would otherwise be the case.

Signal	Description
Execution started (AW-DB, DBX13.6)	= TRUE When a mode/movement is started with the relevant control signals or when the status for “Process in progress” (AW-DB, DBX23.1) = 1  “Execution started” when the block is called/ started “Process in progress” when the FM starts traversing movement
Position (AW-DB, DBX13.7)	= FALSE When status signal “Position reached. Stop.” (AW-DB, DBX25.7) = 0 is returned or when a mode is started with the relevant control signals.  “Position” when the block is called/ started “Position reached. Stop” when FM starts traversing movement

Furthermore, various settings and commands are required to control the FM 354.

In all cases, those commands and settings are transferred which are activated at the time when the POS_CTRL block is called. The commands are canceled after the transfer, also in case of error.

Operating Modes / System Data	Write Request	Function	T	STE	REF	SM	MDI	A/AE	See Sect.
Single functions	Internal	DBB34 and 35	□	x	□	□	□	□	9.3.2
Single commands	Internal	DBB36 and 37	x	x	x	x	x	x	9.3.3

x Data are accepted or processed.

□ Data required to move the axis.

The functions which can be activated in the FM using single settings or single commands are listed below.

Single Settings	Single Commands
Servo enable On-the-fly measuring Rotational speed monitoring Parking axis Simulation Length measuring Retriggering of reference point Reset Enable input Disable software limit position monitoring	Activate machine data Delete distance to go Automatic block return Automatic block advance Restart Rescind set actual value

Error messages from the FM

When an operator error, traversing error or data error occurs, the error number is read automatically via a Read request. The error number is entered in the AW-DB user data block and the Read status set.

An operating error, reported via a diagnostic interrupt, can be read out with the Read request “Operating error no.” (AW-DB, DBX42.5).

Table 6-9 Error messages from the FM

Error	Message	Error No.	Error Acknowledgement
Data error	Status signal (AW-DB, DBX22.4)	Is read out via Read request (AW-DB, DBB94 and 95)	New Write request
Operator/traversing error	Status signal (AW-DB, DBX22.3)	Is read out via Read request (AW-DB, DBB 90 and 91)	Set/reset control signal “Acknowledge operator/traversing error” (AW-DB, DBX14.3)
Diagnostic interrupt	Activated with OB 82, the data must be read out using the POS_DIAG block	In case of an operating error read out via the POS_DIAG block, the error no. is read out via the Read request DBX42.5 (AW-DB, DBB86 and 87)	Single command: Restart

For additional specific information, please refer to Chapter 11 “Error Handling”.

6.4.4 The POS_DIAG (FC 2) block – Read diagnostic interrupt data

Block description

See Section 6.3.4.

6.4.5 The POS_MSRM (FC 3) block – Read measured values

Function

Use the POS_MSRM block to read the measured values into the AW-DB user data block.

For information on process interrupts, please see Section 6.5.

For information on measured values, see Section 9.3.10.

Note

The measured values can also be read by means of POS_CTRL (Read request). If more than one Read request is present, this request is processed in the relevant order.

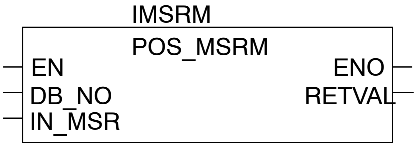
If the POS_MSRM block is called, the measured values are obtained irrespective of any other Read requests currently present.

Call options

The POS_MSRM block requires an instance DB when called.
The DB IMSRM (DB 3) is part of the “FM353_354” library.

POS_MSRM can be called in OB 40 (only when the FM is operated in the central configuration) if the process interrupt has been activated (see Section 5.2), or in OB 1. It is not possible to call POS_MSRM in both OBs simultaneously.

Call in LAD representation (ladder diagram)	Call in STL representation (statement list)

	`CALL  POS_MSRM, IMSRM`
	`      DB_NO          :=`
	`      RETVAL         :=`
	`      IN_MSR         :=`

Parameters

The Table below lists the parameters for the POS_MSRM block.

Name	Data type	Param. type	Description
DB_NO	INT	I	Data block number
RETVAL	INT	Q	–1
IN_MSR	BOOL	I/Q	Start Read

Parameter types: I = input parameter, Q = output parameter,
I/Q = throughput parameter (initiation parameter)

Function description

The function works together with an AW-DB user data block. When the function is called, the DB number is forwarded in the DB_NO parameter.

Reading of the measured value is started by setting the IN_MSR parameter to TRUE. When the function has executed, the block resets the parameter.

The IN_MSR parameter remains set while the function is executing. Transfer of the data is terminated is complete when the parameter is reset (IN_MSR = FALSE).

Error evaluation

Errors are flagged in the Binary Result (BR = 0) and by RETVAL < 0.

Possible errors are as follows:

Data transfer errors during transfers with SFB 52 “RDREC”. The error is made available in the user data block AW-DB, DBW98 (see Error List, Section 6.8).

6.4.6 Interface, user DBs (AW-DBs)

User DB 1 is the user data block in the default condition of the function blocks; it’s symbolic name is IFFM_ICTRL. This DB consists of the user data interface and the instance data of the POS_CTRL block (FB 1).

The user can access the signals and/or data on the interface using absolute or symbolic addresses if reference is made via the symbol table of the project.

When calling POS_CTRL, the interface is transferred as an instance. With the other standard function blocks, the interface is assigned to the appropriate channel/axis using the DB_NO input parameter.

Note

If the POS_CTRL block is called for several channels/axes, a new instance DB of the POS_CTRL block is to created for each individual channel/axis. To assign the datza to the relevant channel/axis, this instance is always to be transferred when calling the block.

The module address is part of the user data block (AW-DB). It is entered either by the POS_INIT block or manually via ”Parameterize FM 354“ using the ”Enter module address in the user data block“ in the overview diagram. The user data block (AW-DB) must exist.

Information about symbolic programming

Normally, the blocks are entered in the symbol table with the symbol name, address, and data type; the symbol table is part of the project and included in the “FM353_354” block library. If you change the block numbers in your project using the SIMATIC Manager, the numbering in the symbol table must also be changed. The block assignment is provided unambiguously via the symbol table.

Before writing and compiling your user program, you must enter the blocks (AW-DB, FCs, FBs) in the symbol table according to your particular configuration. The symbolic structure of the interface is stored in the supplied instance data block DB 1 (IFFM_ICTRL). The symbolic reference is provided via your STEP 7 project – the symbol table.

The Annex shows the user interface with the appropriate symbols and with the absolute address.

Example symbol table:

Symbol	Address	Data type	Comments
IFFM_ICTRL	DB 1	FB 1	CPU / FM interface, instance DB to the POS_CTRL block
POS_INIT	FC 0	FC 0	Initialization
POS_CTRL	FB 1	FB 1	Operating modes, commands and data exchange

6.5 Interrupts

Interrupt processing

The FM 354 can generate process interrupts and diagnostic interrupts. You can process these interrupts only in an interrupt OB (OB 40 or OB 82). If an interrupt is generated without the associated OB having been loaded, the CPU goes to STOP (refer to the manual entitled *Programming with STEP 7*).

Interrupt servicing is enabled in the following stages:

1. General Interrupt Enable for the entire module:
   - Select the module in the hardware configuration.
   - Enable diagnostic and/or process interrupts with **Edit** > **Object Properties** > **Basic Parameters** (also see Figure 5.2).
   - Select the OB number for the process with **Edit** > **Object Properties** > **Addresses**.
   - Save and compile the hardware configuration.
   - Load the hardware configuration into the CPU.
2. Enable the events for the process interrupt in the machine data.

Evaluating a process interrupt

When the FM 354 generates a process interrupt, variable OB40_POINT_ADDR (or the corresponding variable in another process interrupt OB) contains the following information:

Table 6-10 Contents of doubleword OB40_POINT_ADDR

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0				in-process measuring	On-the-fly block chang		Length measurement terminated	Position reached
1								
2								
3								

The reason for the interrupt is made available in byte 0.

Lost process interrupts

If servicing of a process interrupt in the process interrupt OB has not yet been terminated, the module “makes a note” of all subsequent process interrupt events. If an event re-occurs before a process interrupt could be generated, the module generates the diagnostic interrupt “process interrupt lost”.

Evaluating a diagnostic interrupt

Following a diagnostic interrupt, the diagnostic information is made available in the local data area of OB 82 for quick analysis. Call the POS_DIAG block to ascertain the exact cause of error (see Section 6.3.4).

6.6 User data block (AW-DB)

Overview

The Table below describes the structure of the user data block.

Table 6-11 User data block (AW-DB)

AW-DB	FM 354							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
General addresses								
DBW0	Module address (data type INT)							
DBW2 to DBB12	Reserved							
DBB13	Position	Execution started						
Control signals								
DBB14					Acknowledge operator/traversing error		Switch to P bus Start-up	
DBB15	Drive enable	Block skip	Read-in enable	Acknowledge M function	Positive direction	Negative direction	Stop	Start
DBB16	Operating mode							
DBB17	Operating mode parameters							
DBB18	Override							
DBB19 to DBB21	Reserved							
Checkback signals								
DBB22	Channel initialized			Data error	Operator/ traversing error		Switch to P bus completed	
DBB23		Reverse prog. scan	Dwell in progress			Wait for external enable	Machining in progress	Start enable
DBB24	Active operating mode							
DBB25	Position reached. Stop.		On-the-fly setting of actual value completed	Servo enable status	Positive travel	Negative travel	End of measurement	Channel synchronized
DBB26	M function number							
DBB27				M function modification				

Table 6-11 User data block (AW-DB), continued

AW-DB				**FM 354**				
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD28	Actual position (with firmware version V3.7.6 and higher in conjunction with the blocks of the “FM353_354” library							
DBW32	Reserved							
Initiation signals								
Initiation signals for single settings (switches); transfer through Write request when change occurs								
DBB34	Simulation	Parking axis	Rotation monitoring				In-process measuring	Controller enable
DBB35		Software limit positions disabled		Enable input disabled	Retrigger ref. point	Length measurement		
Initiation signals for single commands; transfer through Write request when change occurs (signals are reset following transfer)								
DBB36	Reserved							
DBB37		Rescind setting of actual value	Restart		Automatic block return	Automatic block advance	Delete residual distance	Activate MD
Initiation signals for Write requests								
DBB38	Set actual value	Set reference point		On-the-fly MDI block	MDI block	Setpoint for incremental dimension	Frequency levels 1, 2	Speed levels 1, 2
DBB39	Teach-in	Request application data	Program selection	Digital outputs	Modify parameters / data		Zero offset	On-the-fly setting of actual value
DBB40 to DBB41	Reserved							
Initiation signals for Read requests								
DBB42			Operating error no.	Service data	Actual value block change	Next NC block	Active NC block	Basic operating data
DBB43	Read measured values	Application data	Additional operating data	Dig. inputs/ outputs	Parameter/ data			

Table 6-11 User data block (AW-DB), continued

AW-DB	FM 354							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Ready signals								
Status/checkbacksignals from POS_CTRL								
DBB44	Simulation	Parking axis	Rotation monitoring				In-process measuring	Controller enable
DBB45		Software limit positions disabled		Enable input disabled	Retrig. ref. point	Length measurement		
DBB46	Reserved							
DBB47		Rescind setting of actual value	Restart		Autom. block return	Autom. block advance	Delete residual distance	Activate MD
DBB48	Set actual value	Set reference point		On-the-fly MDI block	MDI block	Setpoint for incremental dimension	Frequency levels 1, 2	Speed levels 1, 2
DBB49	Teach-in	Request application data	Program selection	Digital outputs	Modify parameters/data		Zero offset	On-the-fly setting of actual value
DBB50 to DBB51	Reserved							
DBB52	Data error read	Operator/traversing error read	Operating error read	Service data	Actual value block change	Next NC block	Active NC block	Basic operating data
DBB53	Read measured values	Application data	Additional operating data	Dig. inputs/outputs	Parameter/data			

Table 6-11 User data block (AW-DB), continued

AW-DB				FM 354				
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error signals								
Error messages from POS_CTRL								
DBB54	Simulation	Parking axis	Rotation monitoring				In-process measuring	Controller enable
DBB55		Software limit positions disabled		Enable input disabled	Retrig. ref. point	Length measurement		
DBB56	Reserved							
DBB57		Rescind setting of actual value	Restart		Autom. block return	Autom. block advance	Delete residual distance	Activate MD
DBB58	Set actual value	Set reference point		On-the-fly MDI block	MDI block	Setpoint for incremental dimension	Frequency levels 1, 2	Speed levels 1, 2
DBB59	Teach-In	Request application data	Program selection	Digital outputs	Modify parameters/ data		Zero offset	On-the-fly setting of actual value
DBB60 to DBB61	Reserved							
DBB62	Data error read	Operator/ traversing error read	Operating error read	Service data	Actual value block change	Next NC block	Active NC block	Basic operating data
DBB63	Read measured values	Application data	Additional operating data	Dig. inputs/ outputs	Parameter/ data			
DBB64 bis DBB65	Reserved							
Processing status of POS_CTRL								
DBW66	Error code (communications error) of the last job request/transfer (data type: INT)							
DBB68					Read request not possible	Read job active	Write request not possible	Write job active
DBB69							Reset status/error	

Table 6-11 User data block (AW-DB), continued

AW-DB	FM 354							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Diagnostic data for the FM, read out with POS_DIAG								
DBB70		Module not initialized			Ext. chan. err. (DBB78)	External error	Int./HW err. (DBB 72, 73)	Module/ group fault
DBB71				Channel info available	Module type classes (08H)			
DBB72				Int. module supply volt. failed	Watchdog triggered		Comm. error (K bus)	
DBB73		Process int. lost			RAM error	FEPROM error		
DBB74	FM pos. ID (74H)							
DBB75	Length of diagnostic information (16)							
DBB76	Number of channels (1)							
DBB77								Channel error vector
DBB78	Operating error							
DBB79 to DBB85	Reserved							
Error code after error message “Operating error” (is read if ’operating error’ is set after calling the POS_DIAG block)								
DBB86	Error number (DS 164) – Detail event class							
DBB87	Error number (DS 164) – Detail event number							
DBB88 to DBB89	Reserved							
Error code following flagging of “Operator/traversing error”								
DBB90	Error number (DS 162) – Detail event class							
DBB91	Error number (DS 162) – Detail event number							
DBB92 to DBB93	Reserved							

Table 6-11 User data block (AW-DB), continued

AW-DB	**FM 354**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error code following flagging of “Data error”								
DBB94	Error number (DS163) – Detail event class							
DBB95	Error number (DS163) – Detail event number							
DBW96	Error code for the POS_DIAG block (return code SFC 51) (Data type: INT)							
DBW98	Error code for the POS_MSRM block (return code SFC 59) (Data type: INT)							
Data for the requests								
Zero offset								
DBD140	Data type: DINT							
Set actual value								
DBD144	Data type: DINT							
On-the-fly setting of actual value								
DBD148	Data type: DINT							
Set reference point								
DBD152	Data type: DINT							
Setpoint for incremental dimension								
DBD156								
Speed levels 1 and 2								
DBD160	Speed level 1							
DBD164	Speed level 2							
Frequency levels 1 and 2								
DBD168	Frequency level 1							
DBD172	Frequency level 2							

Table 6-11 User data block (AW-DB), continued

AW-DB				**FM 354**				
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
MDI block								
DBB176 to DBB177	Reserved							
DBB178				Position/ dwell			G function group 2	1
DBB179					M function group 3	2	1	Speed
DBB180	G function no. of group 1							
DBB181	G function no. of group 2							
DBB182 bis DBB183	Reserved							
DBD184	Value for position/dwell (data type DINT)							
DBD188	Value for speed (data type DINT)							
DBB192	M function no. of group 1							
DBB193	M function no. of group 2							
DBB194	M function no. of group 3							
DBB195	Reserved							
Modify parameter/data or request reading of relevant data								
DBB196	DB type							
DBB197	Number							
DBB198	Quantity							
DBB199	Request							
DBB200 to DBB219	Data array, structure/data type of Write data as per bytes 1 to 4 of this structure (e.g. a program block or max. 5 MD items)							
Digital inputs/outputs								
DBB220					Digital input 3	2	1	0
DBB221					Digital output 3	2	1	0
On-the-fly MDI block								
DBB222 to DBB223	Reserved							
DBB224				Position/ dwell			G function group 2	1
DBB225					M function group 3	2	1	Speed
DBB226	G function no. of group 1							

Table 6-11 User data block (AW-DB), continued

AW-DB	FM 354							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB227	G function no. of group 2							
DBB228 to DBB229	Reserved							
DBD230	Value for position/dwell (data type DINT)							
DBD234	Value for speed (data type DINT)							
DBB238	M function no. of group 1							
DBB239	M function no. of group 2							
DBB240	M function no. of group 3							
DBB241	Reserved							
Program selection								
DBB242	Program number							
DBB243	Block number							
DBB244	Direction of processing							
DBB245	Reserved							
Request for application data								
DBB246	Application data 1							
DBB247	Application data 2							
DBB248	Application data 3							
DBB249	Application data 4							
Teach-in								
DBB250	Program number							
DBB251	Block number							
DBB252 to DBB309	Reserved							
Data read as per request								
Basic operating data								
DBD310	Actual position (data type DINT)							
DBD314	Actual speed							
DBD318	Residual distance (data type DINT)							
DBD322	Setpoint position (data type DINT)							
DBD326	Sum of active coordinate offset, tool offset, zero offset (data type DINT)							
DBD330	Rotational speed							
DBD334 to DBD338	Reserved							

Table 6-11 User data block (AW-DB), continued

AW-DB				**FM 354**				
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Active NC block								
DBB342	Program number							
DBB343	Block number							
DBB344	Block skip	UP call	No. of UP calls	Position/ dwell		G function group		
						3	2	1
DBB345				Tool offset	M function group			Speed
					3	2	1	
DBB346	G function no. of group 1							
DBB347	G function no. of group 2							
DBB348	G function no. of group 3							
DBB349	Reserved							
DBD350	Value for position/dwell (data type DINT)							
DBD354	Value for speed (data type DINT)							
DBB358	M function no. of group 1							
DBB359	M function no. of group 2							
DBB360	M function no. of group 3							
DBB361	Tool offset no.							
Next NC block								
DBB362	Program number							
DBB363	Block number							
DBB364	Block skip	UP call	No. of UP calls	Position/ dwell		G function group		
						3	2	1
DBB365				Tool offset	M function group			Speed
					3	2	1	
DBB366	G function no. of group 1							
DBB367	G function no. of group 2							
DBB368	G function no. of group 3							
DBB369	Reserved							
DBD370	Value for position/dwell (data type DINT)							
DBD374	Value for speed (data type DINT)							
DBB378	M function no. of group 1							
DBB379	M function no. of group 2							
DBB380	M function no. of group 3							
DBB381	Tool offset no.							

Table 6-11 User data block (AW-DB), continued

AW-DB				FM 354				
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Application data								
DBD382	Application data 1 (data type: DINT)							
DBD386	Application data 2 (data type: DINT)							
DBD390	Application data 3 (data type: DINT)							
DBD394	Application data 4 (data type: DINT)							
Actual value block change								
DBD398	Data type DINT							
Service data								
DBD402	Frequency output value (data type DINT)							
DBD406	Pulse output counter (data type DINT)							
DBD410	Reserved							
DBD414	Reserved							
DBD418	Difference between setpoint and actual position (data type DINT)							
DBD422	Reserved							
DBD426	Switch adjustment (data type DINT)							
DBD430	Reserved							
Additional production data								
DBB434	Override							
DBB435	NC traversing program no.							
DBB436	NC block no.							
DBB437	UP call counter							
DBB438	Active G90/91							
DBB439	Active G60/64							
DBB440	Active G43/44							
DBB441	Active D number							
DBB442							Speed limit	
DBB443					Acceleration/deceleration value limited		Frequency value for acceleration limited	Start/stop frequency limited
DBB444 to DBB445	Reserved							
Parameter/data								
DBB446	DB type (MD, incremental dimension or traversing program)							
DBB447	Number							
DBB448	Quantity							
DBB449	Request							

Table 6-11 User data block (AW-DB), continued

AW-DB	FM 354							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB450 to DBB469	Array, structure/data type according to data, to be read as per bytes 1 to 4 of this structure (e.g. a program record or max. 5 MD items)							
DBB470 to DBB485	Reserved							
Measured values								
Measured values as per POS_MSRM call								
DBD486	Initial value or in-process measured value (data type DINT)							
DBD490	Final value (data type DINT)							
DBD494	Measured length value							
Array for operator control/monitoring								
Operator control and monitoring								
DBB498	Frequency levels transferred	Speed levels transferred	Increm. dim. transferred	Teach-in transferred	Prog. sel. transferred	MDI block transferred	Read MD	Write MD
DBB499	Operator/ traversing error	Data error	Diagnostic interrupt			Zero offset transferred	Set actual value transferred	MDI block transferred on-the-fly
DBW500	MD number							
DBD502	MD value (data type DINT)							
DBB506	Incremental dimension number							
DBB507	Reserved							
DBW508	Display number							
DBW510	Keyboard code							
DBW512	Reserved							
Operating mode selection								
DBB514		Jog mode	Auto mode	Auto/single block mode	MDI	Incremental mode (relative)	Approach to reference point	Open-loop control mode
DBB515	Acknowledge diagnostic interrupt	Acknowledge error						

Note

For the symbolic signal designations, please refer to the “FMSTSV_L” library, UDT 1 block, or to the “FM353_354” library, DB 1 (IFFM_CTRL).

6.7 Sample applications

Overview

This chapter provides information on the following:

- Basic example for setting the operating mode
- Example 1: Moving axes in “JOG mode” and “Approach to reference point mode”
- Example 2: Traversing an MD block
- Example 3: “Automatic” mode with program selection
- Example 4: Technology example for embedding OPs

General remarks

The installation of the FM 354 configuration package includes the following sample projects:

- **“zDt13_02_FM353_EX”** ([STEP7 directory]\EXAMPLES\zDt13_02)
  for application examples for the “FMSTSV_L” block library
- **“zDt13_03_FM353_EX”** ([STEP7 directory]\EXAMPLES\zDt13_03)
  for application examples for the “FM353_354” block library

The relevant technological functions (POS_CTRL, POS_DIAG, POS_INIT) are called in the OB 1, OB 82 and OB 100 blocks. DB 100 (DBEX) contains the relevant user signals/user data for all application examples.

Each example is programmed as a block (e.g., example 1 = FC 101, etc.). In The basic example (FC 100) is always necessary for the examples 1 to 3; it sets the relevant modes and copies the data between DB 1 and DB 100.

The examples 1 to 3 are interdependent. They are technologically simple examples which you can expand to suit your particular needs. To be able to use the functions provided by the examples 1 to 3, call the relevant examples in OB 1 analogous to example 1.

OB 1 contains an example after the call of POS_CTRL how the evaluation of the reported errors of the POS_CTRL could be programmed. You can expand this error evaluation accordingly if you want.

Example 4 requires the **OB_example4** source file specified in the source folder to be compiled. Since this is an application example for use of an OP, only example 4 should be called in OB 1, as not to overwrite data.

Note

In the examples, the axes do not traverse in simulation mode!

Because “DBEX” is a retentive DB, it is initialized in the start/restart routine (OB 100). If this is not required, simply delete the initialization section of OB 100 (network “DBEX Initialization”).

Basic example for setting the operating mode

This example is always required for sample applications 1 to 3.

Open the “**zDt13_02_FM353_EX**” or the **“zDt13_03_FM353_EX“** example project in the SIMATIC Manager with **File > Open... > Projects**. The block for this example is FC 100.

You will find the signals in “DBEX”.

This example must always be called. It sets the operating modes according to the user’s specifications, evaluates the mode status signals, and displays the current mode. The checkback signals required for the examples will be copied into “DBEX”.

In order to use the “Jog” or “Reference point approach” mode in Example 1, the user has to set the relevant mode code in byte MODE_IN of “DBEX” (01 for “Jog”, 03 for “Reference point approach). When “Jog” mode is selected, mode parameter 01 (MODE_TYPE) is additionally set for activating speed level 1 in “Jog” mode.

Mode	**Code**
Jog	01
Reference point approach	03
MDI	06
Auto	08

In Example 2, you must set “MDI” mode (mode coded in byte MODE_IN = 06).

In Example 3, you must set “Auto” mode (mode coded in byte MODE_IN = 08).

The active mode is displayed in byte MODE_OUT in the relevant code.

To restart the module (e.g. after diagnostic interrupt), bit RESET_AX must be set in “DBEX”. The example will then set bit RESET_AX in “AW-DB”. A restart will be initiated and bit RESET_AX reset in “DBEX”.

In order to work with the following examples, you must set the mode required for each.

Example 1

Open the example project. The block used for this example is FC 101.

The signals are in “DBEX”. The signals relevant for Example 1 only are in structure “EX1”.

The Drive Enable and the Controller Enable for the axis are set in “DBEX” (OB 100: DRV_EN = TRUE, SERVO_EN = TRUE) and are transferred to the interface (AW_DB) in Example 1.

In order for the example to function, you must first set either “Jog” mode (mode code 01) or “Reference point approach” mode (mode code 03) in byte MODE_IN of the “DBEX”. The respective mode checkback signal is flagged in byte MODE_OUT.

The traversing movements are shown in bits “GO_M” = TRUE (traverse -axis 1) or “GO_P” = TRUE (traverse + axis 1).

“Jog” mode active:

Once a mode has become active, the Write request “VLEV_EN” (AW-DB, transfer speed level 1, 2) is executed once. If you want to transfer it again, you must either reset the “VLEV_D” bit (status/checkback signal from the request) or set the “JOBRESET” (reset status/error) in DB_FM.

If you set bit “DIR_M” (minus direction) or bit “DIR_P” (plus direction) to TRUE in “DBEX”, the axis is moved in either a negative or positive direction.

“Reference point approach” mode active:

When you set the “START” bit to TRUE, the axis is moved in a negative or positive direction (depending on the machine data initialization) until the reference point is located. If the reference point approach was successful, the axis is synchronized (SYNC=TRUE).

If an operator or traverse error occurred, this is flagged by bit “OT_ERR” = TRUE. An error can be acknowledged by setting bit “OT_ERR_A” to TRUE.

Note:

Variable table 1, which contains all the relevant signals for monitoring and controlling Example 1 (“control and monitor variable” tool), is located in the “Blocks” directory.

Example 2

Open the example project. The block used for this example is FC 102.

The signals are in “DBEX”. The signals relevant for Example 2 only are in structure “EX2”.

The Drive Enable and the Controller Enable for the axis are set in “DBEX” (OB 100: DRV_EN = TRUE, SERVO_EN = TRUE), and are transferred to the interface (AW-DB) in Example 2.

In order for the example to function, you must set the “MDI” mode. Enter “MDI” mode (mode code 06) in the MODE_IN byte of “DBEX”. The relevant mode checkback signal is flagged in byte MODE_OUT.

Once the mode has been successfully set, a default MDI block is automatically transferred to the module (MDI network) when Write request “MDI_EN” has been set in “AW-DB” (transfer MDI block). This block can be changed in dependence on the system and the request. If it is to be retransferred, you either have to reset the “MDI_D” bit in “AW-DB” (status/checkback signal for request) or set bit “JOBRESET” (reset status/error).

Set the “START” bit in “DBEX” to TRUE in the “EX2” structure. The activated MDI block is started on the condition that the axis is synchronized and has a Start Enable. Then the “START” bit is reset. The MDI block cannot be restarted until the start enable is again available.

The block can be stopped by setting the “STOP” bit.
Only when the “STOP” bit has been reset to FALSE (and the “START” bit to TRUE) is a restart possible.

If an operator error or traversing error occurs, it is flagged by the “OT_ERR” bit (the bit is set to TRUE). The error can be acknowledged by setting the “OT_ERR_A” bit to TRUE.

Note:

Variable table 2, which contains all the relevant signals for monitoring and controlling Example 2 (“control and monitor variable” tool), is located in the “Blocks” directory.

Example 3

Open the example project. The block used for this example is FC 103.

The signals are in "DBEX". The signals relevant for Example 3 only are in structure "EX3".

The program to be selected in the Example has the program number "10". This program number is entered in Example 3.

The Read Enable, the Drive Enable, and the Controller Enable for the axis are set in "DBEX" (OB 100: READ_EN = TRUE, DRV_EN=TRUE, SERVO_EN=TRUE), and are transferred to the interface (AW-DB) in Example 3.

Prerequisite for successful program selection is the availability of that program in the FM.

In order for the example to function, you must set "Auto" mode. Set the "AUTO" mode (mode code 08) in byte MODE_IN of "DBEX". The relevant mode checkback signal is flagged in byte MODE-OUT.

Following successful mode selection, the program with the number "10" is automatically selected by setting Write request "PROGS_EN" in "AW-DB".

Set the "START" bit in "DBEX" to TRUE in structure "EX3". The selected program is started, assuming that the axis is synchronized and has a Start Enable. Then the "START" bit will be reset.

The program can be stopped by setting the "STOP" bit. It can be restarted by resetting the "STOP" bit to FALSE (and the "START" bit to TRUE).

If an operator error or traversing error occurs, it is flagged in the "OT_ERR" bit (the bit is then TRUE). The error can be acknowledged by setting bit "OT_ERR_A" to TRUE.

Note:

Variable table 3, which contains all the relevant signals for monitoring and controlling Example 3 ("control and monitor variable" tool), is located in the "Blocks" directory.

Example 4

Open the example project. The block used for this example is FC 104.

Note: If you call the example 4 in OB 1, then please do not call the examples 1 to 3 at the same time; otherwise, data would be overwritten.

In this example, the HMI interface signals for the data range from DBB 498 through DBB 515 are transferred to the interface area for the control signals, e.g. the modes (see Section 8.2).

You can trigger write and read jobs by assigning the data fields to be transferred the appropriate parameters and data.

For example, if you select the machine data screen PIC_763, you can write an MD using the ”set” softkey (SK) and read an MD using the ”read” softkey (SK).

Once you have selected the mode screen PIC_75 on the operator panel and selected the appropriate mode SK, the selected mode will be accepted into the control signals of the interface, and the appropriate mode will be set.

If you select the diagnostic screen PIC_77, you can acknowledge an error by pressing the ”Quit” SK or acknowledge a diagnostic alarm by pressing the ”Res” SK.

In this way, all interface signals which can be activated by the OP are requested. You can assign default values to all data fields and transmit trigger pulses for the jobs to be executed.

Structure of “DBEX” (DB 100)

```
DATA_BLOCK “DBEX”

STRUCT                             // *** General signals ***
   ERR_CODE_INIT   : INT;    // Error code for POS_INIT
   ERR_CODE_CTRL : INT;      // Error code for POS_CTRL
   ERR_CODE_DIAG  : INT;     // Error code for POS_DIAG
   OVERRIDE           : BYTE;   // Override
   MODE_IN             : BYTE;   // Mode setting (coded)
   MODE_OUT          : BYTE;   // Mode setting (coded)
   DRV_EN               : BOOL;   // Drive Enable
   SERVO_EN          : BOOL;   // Controller Enable
   OT_ERR_A            : BOOL;   // Acknowledgement for operator/traversing error
   RESET_AX            : BOOL;   // Restart
   DIAG_RD              : BOOL;   // Start of POS_DIAG
   PARA                    : BOOL;   // Initialized
   SYNC                    : BOOL;   // Synchronized
   START_EN            : BOOL;   // Start Enable
   POS_ROD             : BOOL;   // Position reached, Stop
   WORKING           : BOOL;   // Execution in progress
   GO_M                  : BOOL;   // Traverse in negative direction
   GO_P                   : BOOL;   // Traverse in positive direction
   OT_ERR              : BOOL;   // Operator-/traversing error
   DATA_ERR          : BOOL;   // Data error
   INIT_ERR            : BOOL;   // Error in POS_INIT
   DIAG_ERR          : BOOL;   // Error in POS_DIAG
   MINUS1              : BOOL;   // “MINUS1” error in POS_CTRL
   MINUS2              : BOOL;   // “MINUS2” error in POS_CTRL
   MINUS3              : BOOL;   // “MINUS3” error in POS_CTRL
   EX1: STRUCT                    // *** Signals for EXAMPLE 1 ***
      DIR_M              : BOOL;   // Negative direction
      DIR_P               : BOOL;   // Positive direction
      START              : BOOL;   // Start
      STOP                : BOOL;   // Stop
   END_STRUCT;
   EX2: STRUCT                    // *** Signals for EXAMPLE 2 ***
      START              : BOOL;   // Start
      STOP                : BOOL;   // Stop
   END_STRUCT;
   EX3: STRUCT                    // *** Signals for EXAMPLE 3 ***
      START              : BOOL;   // Start
      STOP                : BOOL;   // Stop
      READ_EN          : BOOL;   // Read Enable
   END_STRUCT;
END_STRUCT
BEGIN
END_DATA_BLOCK
```

6.8 Error list, system messages (CPU)

The Table below lists some of the errors which occur during data transfer with the internal SFC/SFBs (RET_VAL in SFCs 51, 58 and 59 and status messages (bytes 2 and 3), SFB 52/53, system messages) (see Reference Manual *System Software for S7-300/400; System and Standard Functions*).

Table 6-12 Error list

Error Code (AW-DB, DBW66)			Description
HEX	DEC	INT	
0	0	0	No errors
8082	32898	–32638	SZL_ID invalid or not in CPU
8085	32901	–32635	Information currently unavailable (caused by system)
80A0	32928	–32608	Negative acknowledgement when reading from module. Module removed during Read operation or module defective.
80A1	32929	–32607	Negative acknowledgement while writing to module. Module removed during Write operation or module defective.
80A2	32930	–32606	DP protocol error in layer 2 (data transfer over Profibus-DP interrupted, e.g. due to wirebreak, missing terminator, parameterization error, etc.)
80A3	32931	–32605	• PROFIBUS: DP protocol error in layer 2 (data transfer over Profibus-DP interrupted, e.g. due to wirebreak, missing terminator, parameterization error, etc.) • PROFINET IO: General CM error
80A4	32932	–32604	Communication problem on K bus
80A7	32935	–63601	DP slave or module is busy
80A9	32937	–63599	This function is not supported by DP slave or by the module.
80AA to 80AF	32938 to 32943	–32598 to –32593	DP slave or module is reporting a manufacturer–specific error in the evaluation. For specifying the error, please refer to the Manufacturer Documentation of the DP slave or of the module.
80B0	32944	–32592	The module does not know this data record.
80B1	32945	–32591	Invalid length specification in the RECORD parameter
80B2	32946	–32590	The configured slot is not occupied.
80B3	32947	–32589	Actual module type not the same as setpoint module type
80B5	32949	–32587	DP slave or module not ready
80B6	32950	–32586	Access to DP slave or module denied
80C0	32960	–32576	Module does not yet have the data to be read available
80C1	32961	–32575	Data from an identical Write job have not yet been processed on the module
80C2	32962	–32574	Module is currently servicing the maximum possible number of requests
80C3	32963	–32573	Needed resources (such as memory, etc.) are currently in use

Table 6-12 Error list, continued

Error Code (AW-DB, DBW66)			Description
HEX	DEC	INT	
80C4	32964	–32572	Communication error; repeat the job.
80C5	32965	–32571	Distributed I/O not available
80C7	32966	–32570	Job aborted due to restart (warm restart) or cold restart of DP master
8522	34082	–31454	DB too short; the data cannot be read out of the DB (Write request)
8532	34098	–31438	DB no. out of user area
853A	34106	–31430	DB does not exist (Write request)
8544	34116	–31420	Error on the n-th (n > 1) attempt to read a DB following the occurrence of an error (Write request)
8723	34595	–30941	DB too short. The data cannot be written to the DB (Read request).
8730	34608	–30928	DB write-protected in the CPU. Data cannot be written to the DB (Read request).
8732	34610	–30926	DB no. out of user area
873A	34618	–30918	DB does not exist (Read request)
8745	34629	–30907	Error on the n-th attempt (n > 1) to write to a DB following the occurrence of an error (Read request)
The errors 80A2 to 80A7, as well as 80Cx are temporary, i.e. it is possible that they are eliminated without your intervention after a certain time of waiting. Messages in the form 7xxx indicate temporary communication states.			

6.9 Technical specifications

Memory allocation

The Table below provides an overview of the memory allocation for the blocks and the user data block (AW-DB). All values are rounded.

Table 6-13 Memory requirements for the blocks and user data block

No.	Block	Block in bytes Loading buffer	MC7 code in bytes	Local data in bytes
0	POS_INIT	250	142	4
1	POS_CTRL	3,394	2,964	22
2	POS_DIAG	310	186	46
3	POS_MSRM	286	176	20
4	AW-DB	1,884	516	–

Table 6-14 Memory mapping of the blocks and of the user data block AW-DB, “FM353_354” library

No.	Block	Block in bytes Loading buffer	MC7 code in bytes	Local data in bytes
0	POS_INIT	250	142	4
1	POS_CTRL	4,966	3,198	34
2	POS_DIAG	310	186	46
3	POS_MSRM	380	242	32
4	AW-DB	2,012	576	–

Processing times for the blocks taking the following example system

The specified times are rounded.

Structure: CPU 315-2DP, FM 354 in the simulation mode

User cycle time: approx. 5 ms

FM cycle: 2 ms

Table 6-15 Processing times for the blocks

Block	Transfer	Cycle 1	Cycle 2	Cycle 3
POS_CTRL	Write control/status signals without data	1.3 ms	–	–
	Write control/status signals with data	1.5 ms	3.2 ms	1.5 ms
	Read control/status signals with data	3.6 ms	–	–

Table 6-15 Processing times for the blocks, continued

Block	Transfer	Cycle 1	Cycle 2	Cycle 3
POS_DIAG	Read process and diagnostic data	3.2 ms	–	–
POS_MSRM	Read measured values	3.6 ms		

Processing times for the blocks when used in a distributed configuration PROFIBUS DP), taking the following example

The specified times are rounded.

Structure: CPU 315-2DP, IM 153-2, FM 354 in the simulation mode

User cycle time: 5 ms

FM cycle: 2 ms

Baud rate: 1.5 MB

Table 6-16 Processing times for distributed blocks

No.	Block	System data transfer	Transfer time in the range from ... to	Average transfer time
0	POS_CTRL	Control/checkback signals	1.1...1.5 ms	1.3 ms
1	POS_CTRL	Write request	30...40 ms	35 ms
2	POS_CTRL	Read request	10...20 ms	15 ms
3	POS_DIAG	Read	2.1...2.7 ms	2.4 ms

Several cycles are required to read or write data via the relevant requests.

Response time for start signals to a distributed FM (PROFIBUS DP)

Table 6-17 Response times for start signals

From the view of the CPU: from calling the block	to	Response time in the range from ... to
Start “Jog”	Response at the output of the FM (axis traverses, incl. FM response time)	8...13 ms
Start “MDI”		10...15 ms

Starting up 7

Chapter Overview

Overview

This Chapter introduces you to the user interface for testing and start-up, and provides check lists for starting up the positioning module. The checklists will help you:

- Check all steps until the module is running.
- Prevent malfunctions of the module once it is in operation.

You are guided through start-up of the machine axes.

7.1 Installation and wiring

Installation information

You can find information about how to install your module:

- In Chapter 3 of this manual
- Inthe manual *S7-300 Programmable Controller, Hardware and Installation*

Wiring information

You can find information about how to wire your module:

- In Chapter 4 of this manual
- In the manual *S7-300 Programmable Controller, Hardware and Installation*

Checklist

The checklist below will help you check important steps in the installation and parameterization of the FM 354 positioning module.

Table 7-1 Installation and wiring checklist

Step	Check	What to do:	OK
1	Slots	Plug the module into one of the suitable slots.	
2	Shielding	Check the shielding of the FM 354 positioning module: • To ensure proper shielding, the module must be screwed down firmly on the rail. • The shielding for shielded lines for digital I/O modules must be connected to the shielding terminal element. • The shielding for the setpoint cable should not be grounded on the drive-unit end.	
3	Limit switches	Check the start/stop limit switches. The limit-switch connections must be connected to the power section. The start/stop limit switches should not be connected to the digital inputs.	
4	Parameterize	Make sure the FM 354 positioning module setup is consistent with the parameterization. Check in particular that: • The attached encoder matches the machine data. • The wiring of the digital I/O modules matches the machine data.	

7.2 Initial values for testing and optimization

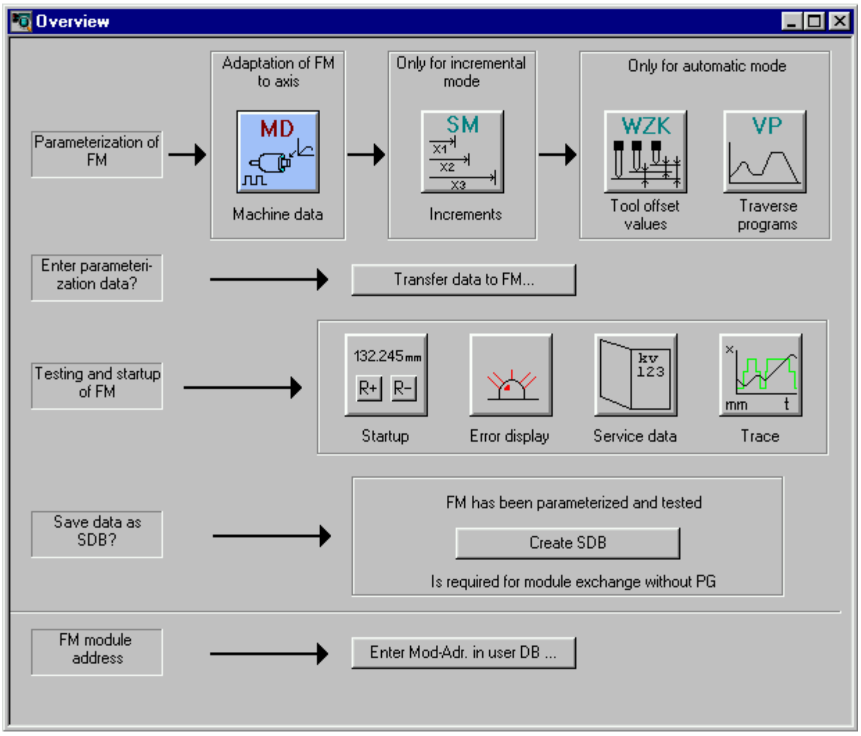
Parameterization information

You can find information about parameterization:

- In Chapter 5 of this manual
- In the on-line help in “Parameterize FM 354”

Overview

The following opening display appears in the “Parameterize FM 354” tool:



FM has been parameterized and tested

Is required for module exchange without PG

Fig. 7-1 Overview display for parameterization and start-up

You can return to this display at any point during parameterization by selecting the menu **View** > **Overview.**

As it is written to the FM 354, the DB-MD is checked for the input limits of the individual values and their interdependencies. It is then stored only if all values are allowed. Otherwise data error messages are displayed by way of the MPI. A defective DB will not be retained when the power is turned off.

Checklist

Despite the “acceptance” testing just mentioned, the ultimate responsibility for the accuracy of all machine data lies with the module user. So it is highly advisable to perform startup using the following checklist.

Table 7-2 Parameterization checklist

Step	Check	What to do:	OK
1	Machine data	**Set initial machine data contents** As shown in Table 5-4 machine data are subdivided into configuration data (**K**) and setting data (**E**). K data indicates how the FM 354 is connected to the machine axis or CPU user program, and must therefore already be fully set up before startup begins. E data is intended for changes during startup, and serves to optimize FM 354 response for the technological process of positioning. The values in Table 7-3 are recommended, and sometimes necessary, as initial settings.	
2	Increments	Increments are only needed for the “Relative incremental” mode. For the next part of the startup procedure it is helpful to set up an “Increments” data block (DB-SM) with the following values: Value 1 1 MSR Value 2 10 MSR Value 3 100 MSR Value 4 1,000 MSR Value 5 10,000 MSR with rotary axes: Value 6 1 rotary-axis cycle [MSR] MSR = measurement-system grid	
3	Tool offset data	Tool offset data is needed only for the “Automatic” mode and is not necessary for the startup described here. Generally, it is not needed until you start up the user program on the S7-300 CPU.	
4	Traversing programs	Traversing programs are needed only for the “Automatic” mode and are not necessary for the startup described here. Generally, it is not needed until you start up the user program on the S7-300 CPU.	
5	Create SDB ≥ 1 000	When you have completed **all start-up actions** on the FM 354 and your plant, create, save and load SDB ≥ 1 000 into the CPU/onto the memory card of the CPU. All the parameter data (DBs) of the FM 354 are stored in SDB ≥ 1 000. This SDB allows you to replace the FM 354 module in the event of a fault, and to download the parameters without a programming device/PC.	

Note

The measurement system (MD7) must match the measurement system specified in the other DBs.

The measurement system raster (MSR) is the smallest distance unit in the active system of measurement.

If at some point you have failed to take this precaution:

1. Delete all data blocks (which do not match the measurement system) or clear the memory of the FM 354 completely.
2. Modify the other data blocks on the PG.
3. Reload the data blocks to the FM 354.

Initial contents of MD

The table below shows you what initial contents are recommended or required for the E machine data at startup of the machine axis.

Enter the machine data in the tab windows or in table format in accordance with the following table.

Table 7-3 Initial contents of machine data

MD (E)	Value	Explanation
5	0	FM 354 triggers no process interrupts
16	$-10^9 ... +10^9$ [MSR]	Specified reference-point coordinates
17	0	Absolute encoder readjustment value (absolute encoders only). Entered automatically when the reference point of the FM 354 is set.
21/22	$-10^9/+10^9$ [MSR]	Software limit switches inactive
23[1)]	$v_{max} = 10 ... 5 \cdot 10^8$ [MSR/min]	Specified maximum axis speed
24	1 000 [MSR]	Large PEH target range
25	0	PEH time monitoring switched off
26	10 000 [MSR]	Zero speed range monitoring set to default value (If the “zero speed range” monitoring error occurs during the start-up, as described in Section 7.3, please enter the maximum value of 1 000 000 deviating from the default. You then define the value required for your true drive axis in Section 7.3.6)
27	0	Reference-point shift (incremental encoders only) should be entered for numerical readjustment of the reference point
28	$0.2 \cdot v_{max}$	20% of the maximum speed is the recommended initial value
29	$0.1 \cdot v_{max}$	10% of the maximum speed is the recommended initial value
30/31	0/0	Backlash compensation inactive
38	1 000 [MSR/min/MSR]	Generally applicable position control loop gain

Table 7-3 Initial contents of machine data, continued

MD (E)	Value	Explanation
39	0	Following-error monitoring inactive
40/41	1 000/1 000[10^3MSR/s^2]	Very low acceleration values
42	0	Jolt filter switched off
43[1)]	U_{max} = 1,000...10,000 [mV]	Specified maximum amount of drive setpoint
44	0	Offset value for drive setpoint
45	0	Voltage ramp inactive

1) This pair of values corresponds to the speed category of the drive. It serves as a basis for calculating the K_v factor in the servo, and must therefore be entered correctly.
Recommendation: So far as possible, U_{max} should be set in the range between 8 and 9 V.

7.3 Testing and optimization

Testing and optimization information

Once you have installed, wired and parameterized the unit, you can test and optimize your FM 354 positioning module. Testing and optimization can be performed with the aid of the testing and start-up interface with or without the user program.

You can also test individual modes and their traversing programs, and view and debug them during execution.

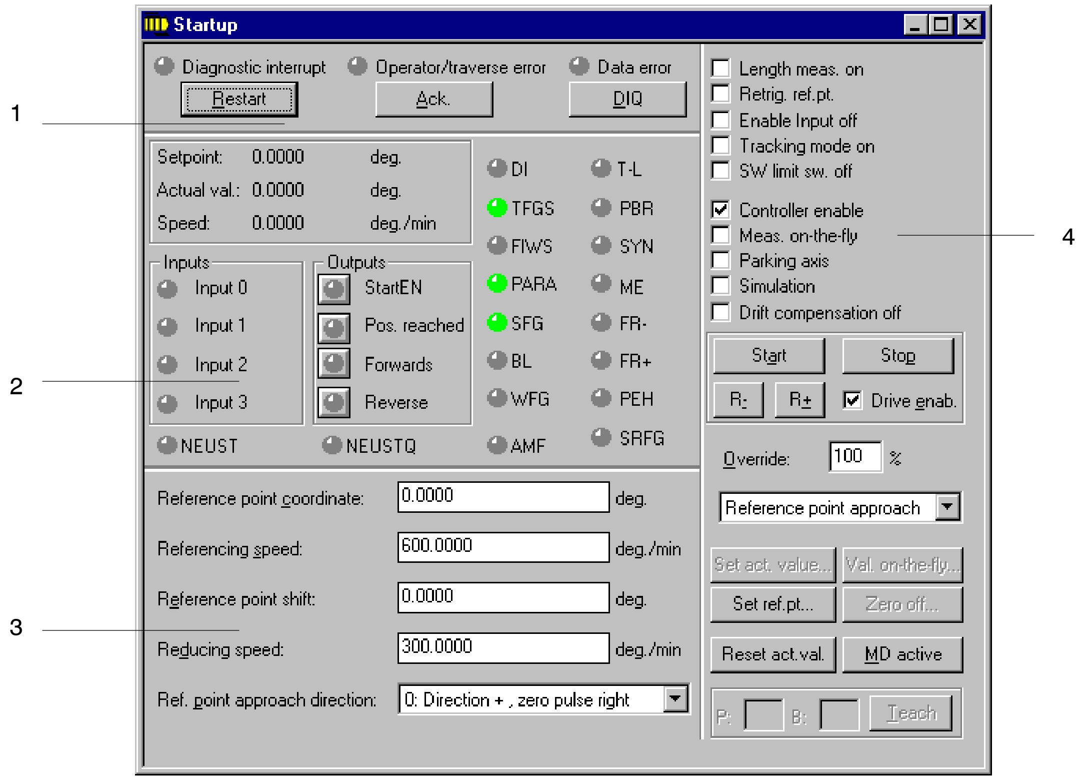
There are two ways of operating the FM:

- **CPU is in “STOP”, test without user program**
- **CPU is in “RUN”, test with user program**

  You can monitor the interface between the FM and the user program. You can also control the program from the start-up user interface when control signal [TFB] (TEST_EN) is enabled in the user program.

This interface is installed with “Parameterize FM 354”. Once the FM 354 has been parameterized, you can call it up by selecting the menu **Test > Startup** or by selecting from the overview display.

When you call up this menu the following screen appears:



Setpoint: 0.0000 deg.
Actual val.: 0.0000 deg.
Speed: 0.0000 deg./min

□ Length meas. on
□ Retrig. ref.pt.
□ Enable Input off
□ Tracking mode on
□ SW limit sw. off
☑ Controller enable
□ Meas. on-the-fly
□ Parking axis
□ Simulation
□ Drift compensation off
☑ Drive enab.

Override: 100 %

Reference point approach

Reference point coordinate: 0.0000 deg.
Referencing speed: 600.0000 deg./min
Reference point shift: 0.0000 deg.
Reducing speed: 300.0000 deg./min
Ref. point approach direction: 0: Direction + , zero pulse right

1 – Error field
2 – Status field (e.g. actual values, check-back signals)
3 – Field for mode-specific inputs
4 – Field for input of values/settings/commands and start/stop for movement

The abbreviations for the checkback signals are described in Table 9-2.

Fig. 7-2 Startup interface (e.g. for “Reference-point approach” mode)

Note

To start a movement, we recommend the following input sequence:

- Select a mode
- Turn simulation on (if you want an operating case)
- Servo enable
- Drive enable
- Override 1...100%

You can operate the “R+” and “R–” buttons in the “jogging” mode as follows:

1. Select “R+” or “R–” with the mouse
2. Press the space bar

You can operate “Start” and “Stop” with the mouse, or with the space bar if you have already selected the button.

The digital outputs are not set in the “Stop” status of the CPU.

When you operate the following buttons, you will get dialog windows:

- Set actual value...
- Set actual value on-the-fly
- Set reference point
- Zero offset

Warning

If you move the axis directly (without simulation), for safety’s sake make sure you can switch off the hardware if a hazard arises

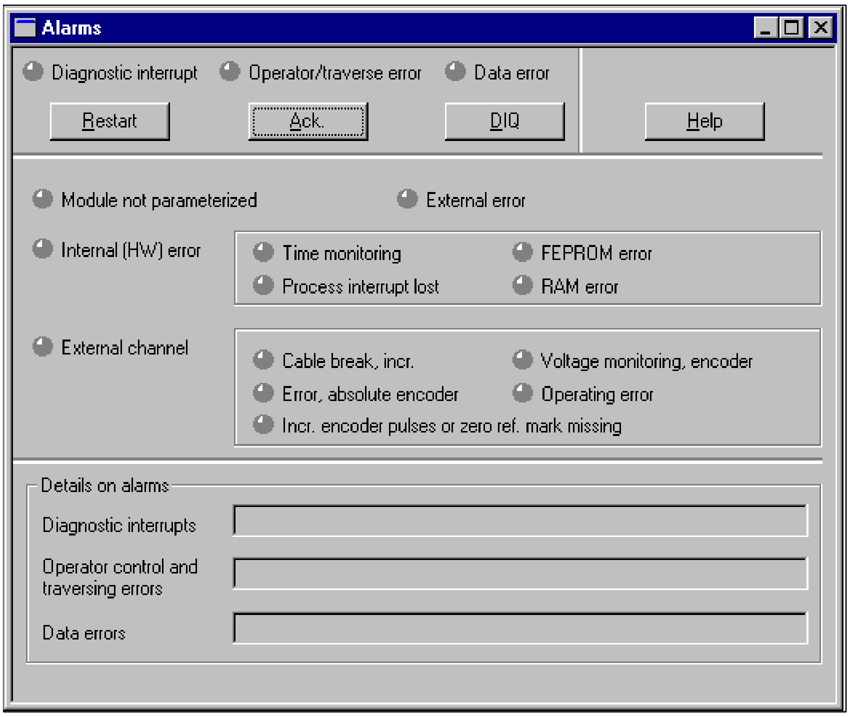
Note

If you use the start-up user interface to operate the FM 354 when the CPU is in “STOP”, and then switch the CPU to “RUN” and then immediately switch to the start-up interface in your user program by means of the [TFB] (TEST_EN) signals (e.g. if example application 3 is included in the user program), please note the following:

You must select the mode again from the start-up interface, or close the start-up interface and call it up again.

You can also call up the following screens:

The following display appears when you select **Test** > **Alarms**:



Alarms

○ Diagnostic interrupt ○ Operator/traverse error ○ Data error

Restart | Ack. | DIO | Help

○ Module not parameterized ○ External error

○ Internal (HW) error
- ○ Time monitoring
- ○ FEPROM error
- ○ Process interrupt lost
- ○ RAM error

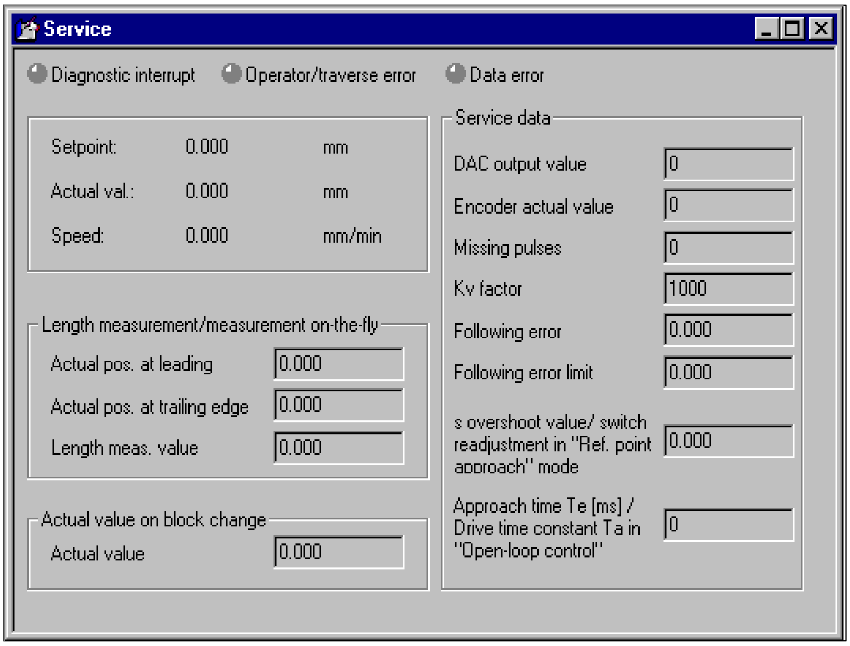
○ External channel
- ○ Cable break, incr.
- ○ Voltage monitoring, encoder
- ○ Error, absolute encoder
- ○ Operating error
- ○ Incr. encoder pulses or zero ref. mark missing

Details on alarms

Diagnostic interrupts	
Operator control and traversing errors	
Data errors	

Fig. 7-3 Troubleshooting

The following display appears when you select **Test** > **Service data**:



Service

○ Diagnostic interrupt ○ Operator/traverse error ○ Data error

Setpoint:	0.000	mm
Actual val.:	0.000	mm
Speed:	0.000	mm/min

Length measurement/measurement on-the-fly

Actual pos. at leading	0.000
Actual pos. at trailing edge	0.000
Length meas. value	0.000

Actual value on block change

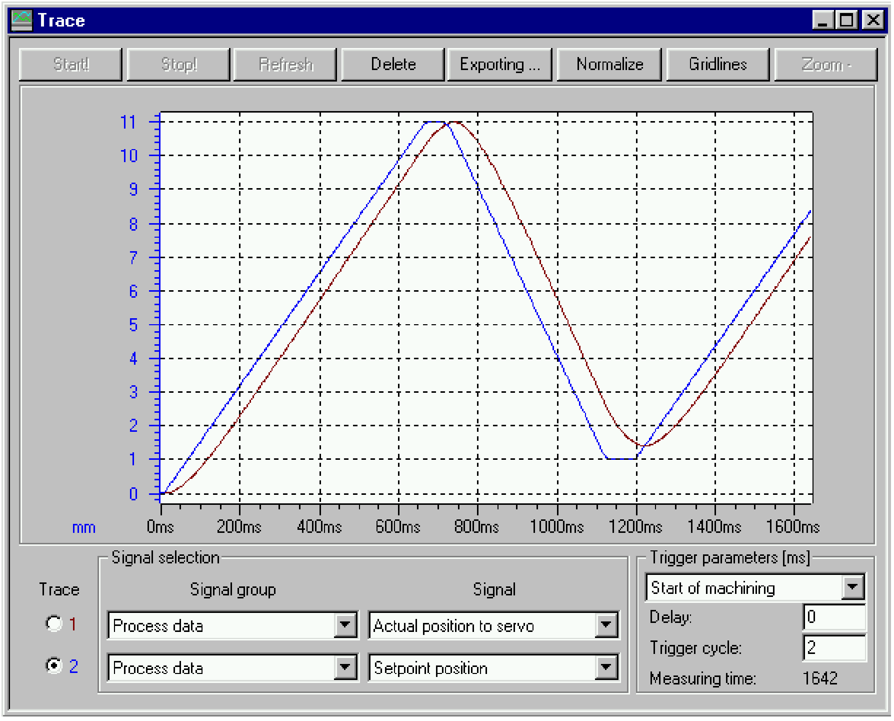
Actual value	0.000

Service data

DAC output value	0
Encoder actual value	0
Missing pulses	0
Kv factor	1000
Following error	0.000
Following error limit	0.000
s overshoot value/ switch readjustment in "Ref. point approach" mode	0.000
Approach time Te [ms] / Drive time constant Ta in "Open-loop control"	0

Fig. 7-4 Service data

The following display appears when you select **Test** > **Trace**:



Signal selection

Trace	Signal group	Signal
○ 1	Process data	Actual position to servo
● 2	Process data	Setpoint position

Trigger parameters [ms]

Start of machining

Delay:	0
Trigger cycle:	2
Measuring time:	1642

Fig. 7-5 Trace

Checklist

When starting up the machine axis, it is important to perform the following steps in the indicated sequence. Steps 1 to 5 are always necessary; the rest are optional, depending on your own application.

Table 7-4 Checklist - Startup of machine axis

Step	Check	What to do:	Page	OK
1	Activation of machine data	See Section 7.3.1	7-12	
2	Monitoring of drive and encoder switching	See Section 7.3.2	7-13	
3	Basic startup of position controller	See Section 7.3.3	7-17	
4	Optimization of position controller	See Section 7.3.4	7-19	
5	Readjustment of reference-point coordinates	See Section 7.3.5	7-23	
6	Activation of position controller diagnostics	See Section 7.3.6	7-24	
7	Activation of software limit switches, drift compensation and backlash compensation	See Section 7.3.7	7-26	

Note

In order for an axis to start, the start enable checkback signal must have been set.

If there is no start enable, this may be because:

- “Drive enable” is not set
- “Stop” is set
- “Operation in progress” is active

7.3.1 Activating the machine data

Overview

The checkback signal PARA notifies you that a DB-MD has been retained. This machine data is automatically activated at power-up. The module's positioning functions are ready to operate.

If no DB-MD is present as yet on the FM 354 when the control is switched on, the module can only communicate by way of the MPI interface. The control signals are not processed by the FM 354. Once an error-free DB-MD has been transferred, the machine data is automatically activated, PARA is set and the control signals are processed.

If the FM 354 is working with activated machine data, you can transfer a new data block or individual parameters in modified form to the module and, if the entire DB-MD is error-free, put the new or modified data into effect by means of the ”Activate machine data” function provided that an operating mode is active (parameterizing tool ”Start-up display”).

The following approaches are possible:

- If only E data have been modified in the machine data record since the last activation, the equipment is activated with module status “Operation in progress” = 0, without interrupting the servo cycle. “SYN” is retained.
- If K data has also been modified in the machine data record since the last activation, activation takes place with module status “Operation in progress” = 0 by way of a cold restart of the servo, just as occurs for a power-up of the module. The instantaneous actual position is still displayed, but encoder pulses from incremental encoders might go unrecorded. “SYN” is reset.
- If the machine data record contains erroneous data at activation time, the function is rejected, with the “Machine data cannot be activated” error message (see Table 11-5, Class 2, No. 21).

7.3.2 Checking the drive and encoder circuits

Overview

The following startup action allows you to monitor the proper actuation of the drive unit and the encoder:

Note

Always be sure to put MD modifications into effect with “Activate machine data.”

Caution

Before triggering any traversing movement, be sure to check that there is enough space for the axis to move in the desired direction.

Drive actuation

You can use the following flow chart to check the actuation of the drive.

Starting the axis

Start the axis with the “space bar” after preselecting the direction via R+ or R–.

The “Windows properties” allow for successive execution of the command (repeated starting/stopping of the axis) when the “space bar” is depressed in quick succession.

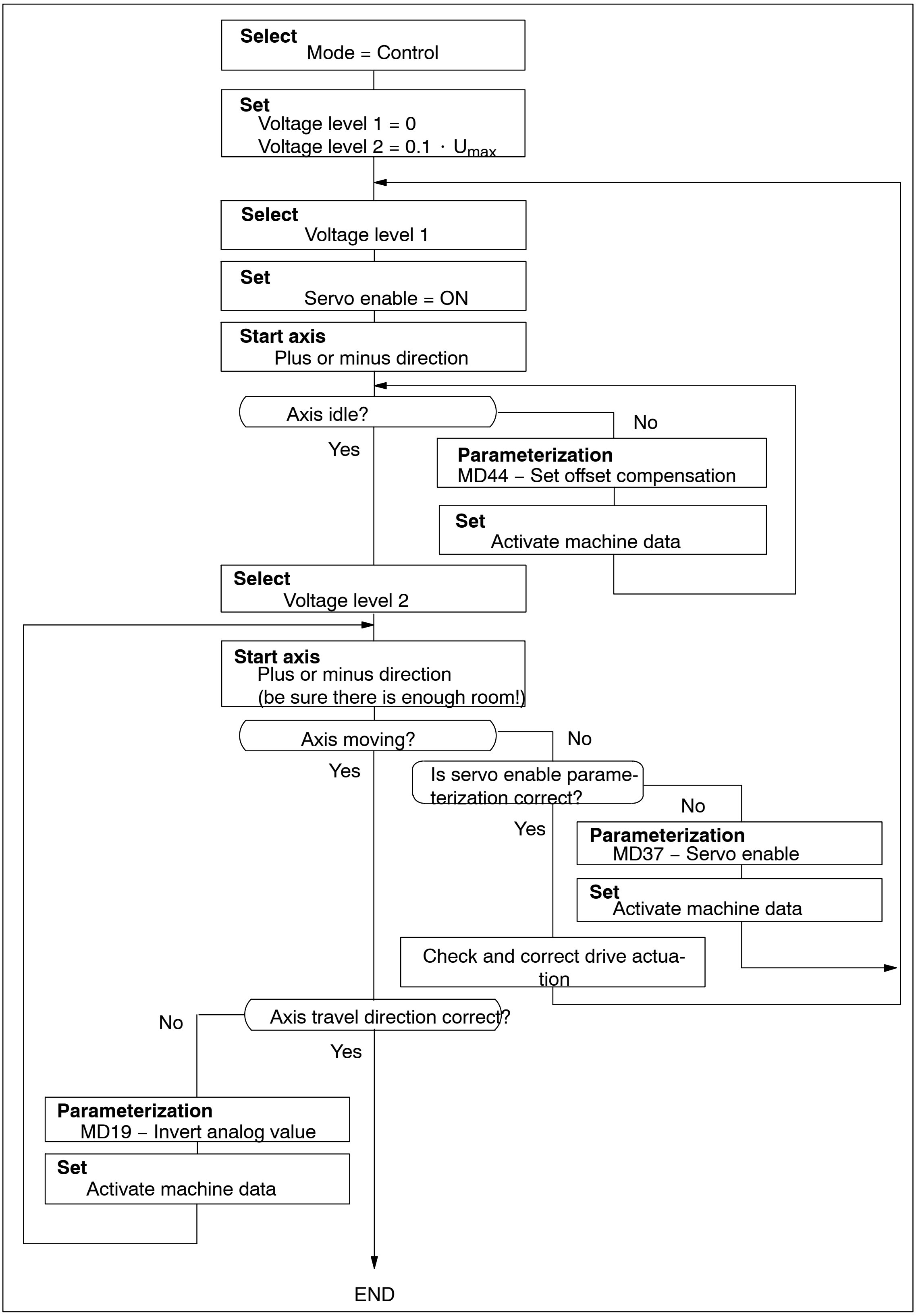


Fig. 7-6 Drive actuation

Encoder actuation and traversing speed

You can use the following flow chart to check the encoder actuation and traversing speed.

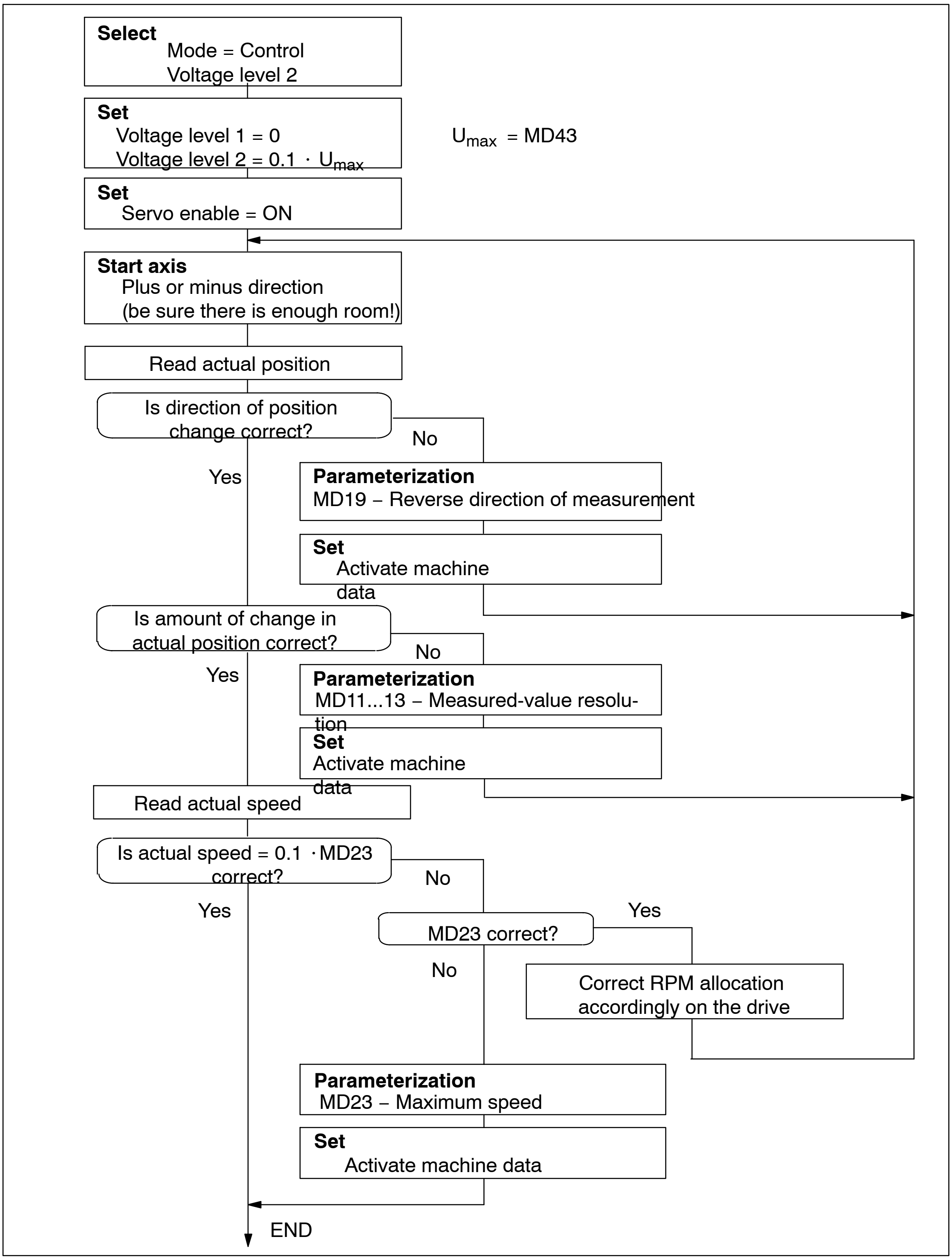


Fig. 7-7 Encoder actuation and traversing speed

Drive transition time and maximum voltage rise

For the following position-controller optimization it is important to know the drive time constant (transition time). In open-loop control mode and on errors with the response “Everything Off” (see Section 11) the voltage value is fed to the drive by way of a ramp defined in MD45. A variety of drives, as well as certain mechanical or technological situations, may require a limitation on the voltage rise. If you do not have a specific value in hand and wish to find a suitable rise value by trial and error, please use the following procedure:

Note

A voltage rise setting will obviously make the axis stop more slowly if an “Everything Off” error response occurs.

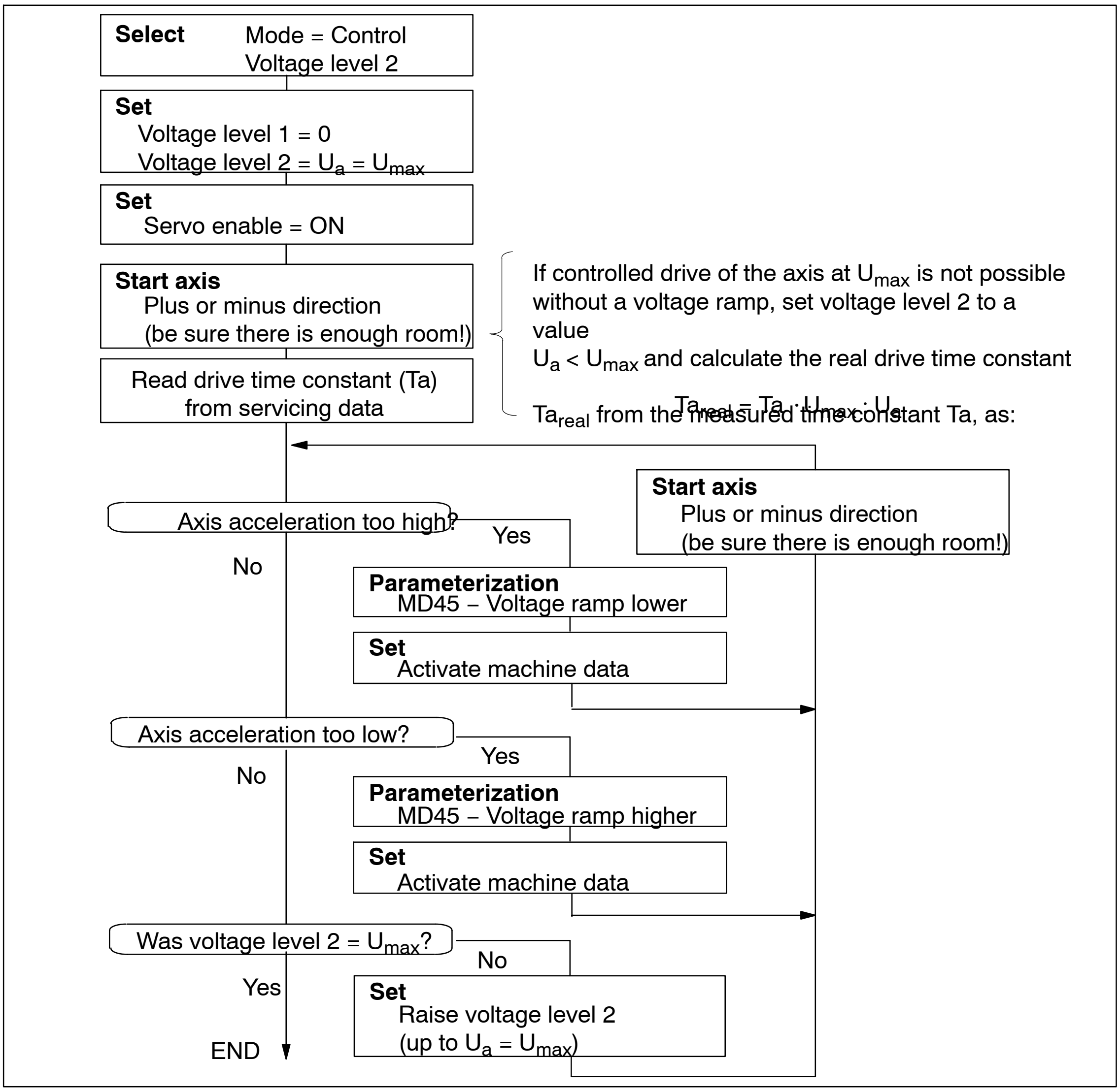

Select Mode = Control
Voltage level 2
Set
Voltage level 1 = 0
Voltage level 2 = $U_a = U_{max}$
Set
Servo enable = ON
Start axis
Plus or minus direction
(be sure there is enough room!)
Read drive time constant (Ta)
from servicing data
If controlled drive of the axis at U_{max} is not possible without a voltage ramp, set voltage level 2 to a value
$U_a < U_{max}$ and calculate the real drive time constant
Ta_{real} from the measured time constant Ta, as:
$Ta_{real} = Ta \cdot U_{max} : U_a$
Start axis
Plus or minus direction
(be sure there is enough room!)
Axis acceleration too high?
Yes
No
Parameterization
MD45 – Voltage ramp lower
Set
Activate machine data
Axis acceleration too low?
Yes
No
Parameterization
MD45 – Voltage ramp higher
Set
Activate machine data
Was voltage level 2 = U_{max}?
No
Yes
Set
Raise voltage level 2
(up to $U_a = U_{max}$)
END

Fig. 7-8 Drive transition time and maximum voltage rise

The axis is now ready for startup of the position controller.

7.3.3 Basic startup of the position controller

Overview

The following startup action allows you to monitor the basic function of the servo - i.e. position control of the axis:

Note

Always be sure to put MD modifications into effect with “Activate machine data.”

Non-release control

You can use the following flow chart to check the non-release control.

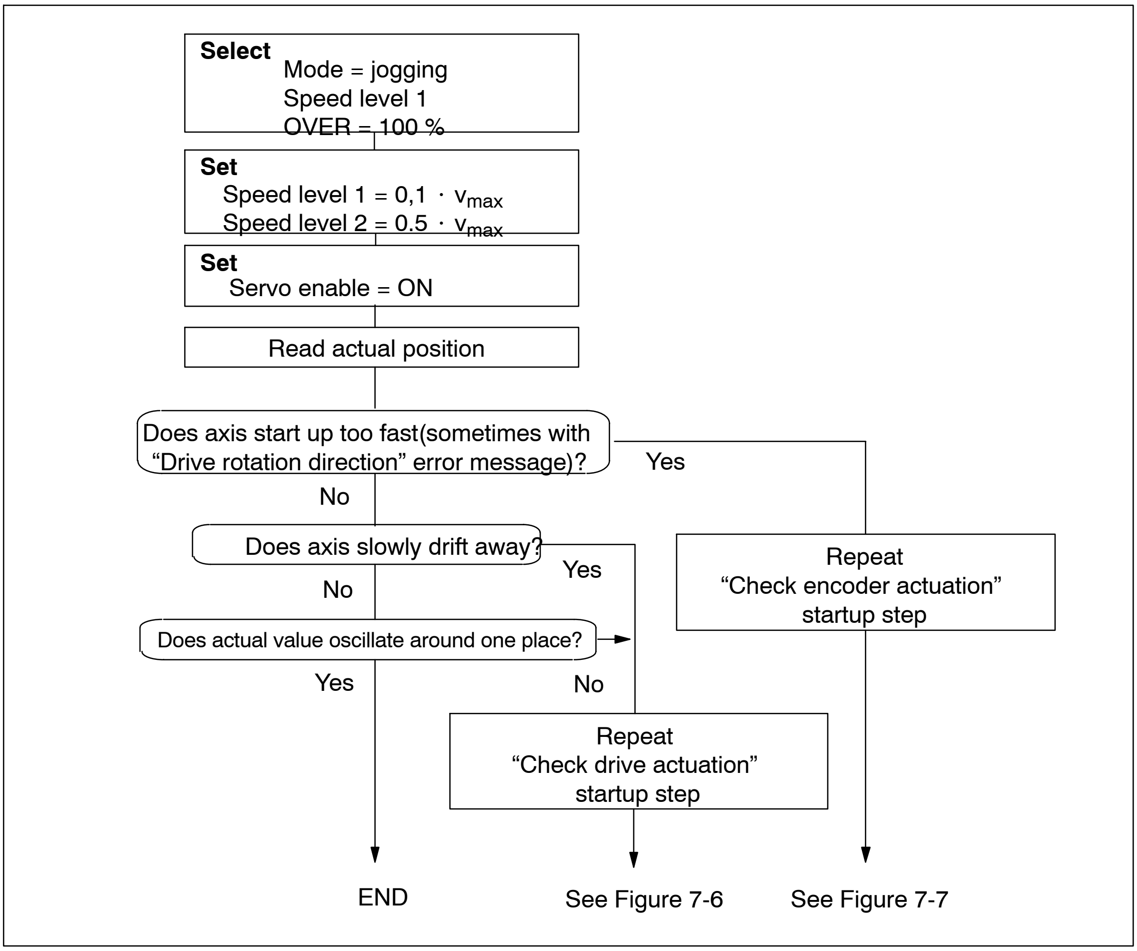

Select
Mode = jogging
Speed level 1
OVER = 100 %
Set
Speed level 1 = 0,1 · v_{max}
Speed level 2 = 0.5 · v_{max}
Set
Servo enable = ON
Read actual position
Does axis start up too fast(sometimes with “Drive rotation direction” error message)?
Yes
No
Does axis slowly drift away?
Yes
No
Does actual value oscillate around one place?
Yes
No
Repeat “Check encoder actuation” startup step
Repeat “Check drive actuation” startup step
END
See Figure 7-6
See Figure 7-7

Fig. 7-9 Non-release control

Positioning

Use the following flow chart to check axis travel to a target position.

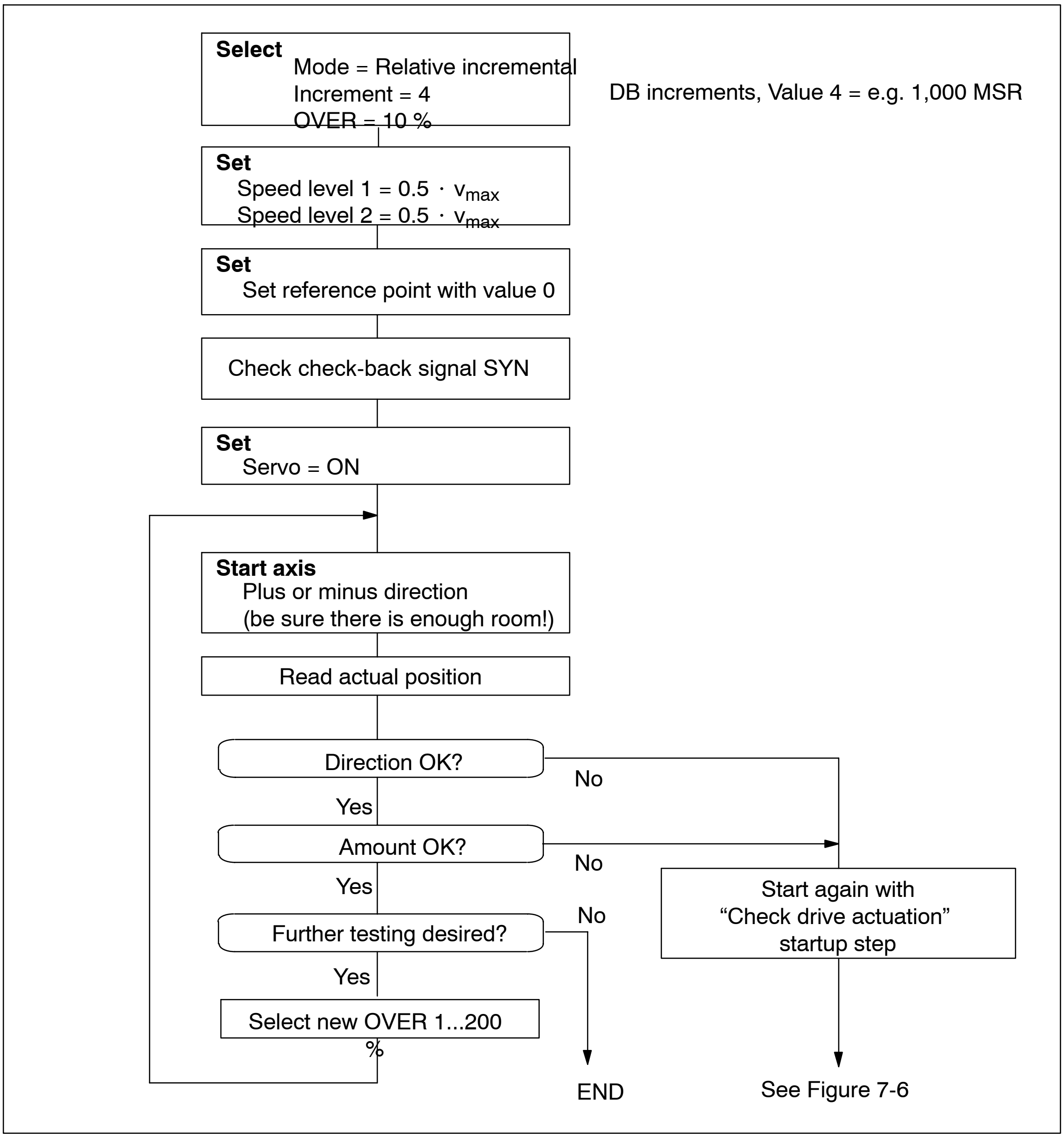

Select
Mode = Relative incremental
Increment = 4
OVER = 10 %
DB increments, Value 4 = e.g. 1,000 MSR
Set
Speed level 1 = 0.5 · v_{max}
Speed level 2 = 0.5 · v_{max}
Set
Set reference point with value 0
Check check-back signal SYN
Set
Servo = ON
Start axis
Plus or minus direction
(be sure there is enough room!)
Read actual position
Direction OK?
No
Yes
Amount OK?
No
Yes
Start again with
“Check drive actuation”
startup step
Further testing desired?
No
Yes
Select new OVER 1...200 %
END
See Figure 7-6

Fig. 7-10 Positioning

7.3.4 Optimizing the position controller

Overview

In principle, the dynamic response of an axis is essentially determined by the dynamic response of the variable-speed drive; there is not sufficient space to discuss this topic here. But this latter dynamic response, in turn, is influenced by the design characteristics of the machinery, such as friction, backlash, torsion and the like. By feeding back the measured displacement, a position controller generally closes the outermost loop of a controller cascade with the following structure:

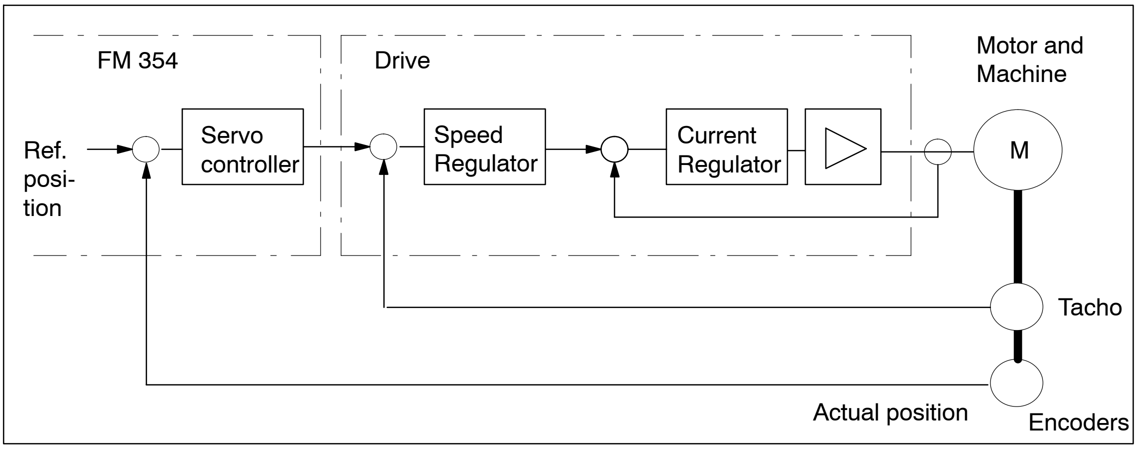


Fig. 7-11 Position control circuit

Procedure

The following instructions are intended as an aid for practical situations.

Position controllers must meet a variety of requirements for various technological applications.

Assessment criteria for the quality of the positioning process can include:

- Good uniformity of traversing movement
- Little or no overshoot at the target point for positioning
- Short positioning time
- A continuous acceleration (soft travel).

In most applications, several of these criteria will be important, so that most of the time the dynamic response of the controller can be optimized only with a number of compromises.

Execute test movements as in Figure 7-12 during the optimization steps described below.

To trigger test movements

You can trigger test movements as follows as you perform optimization:

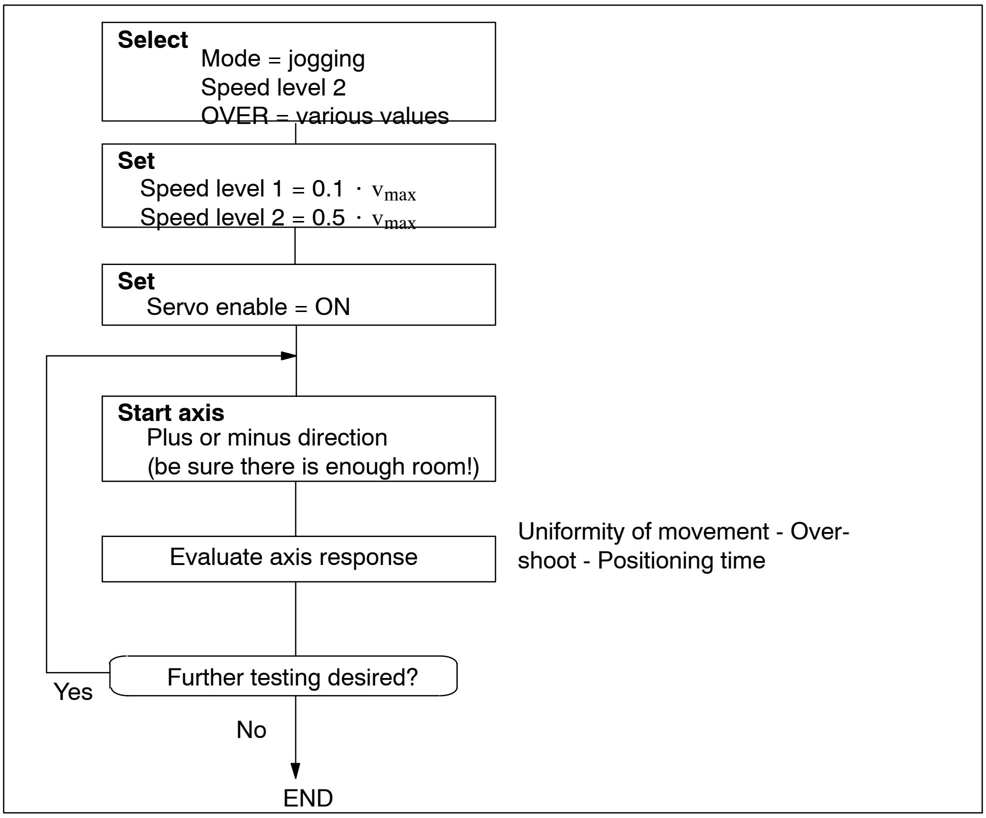


Fig. 7-12 Test movements for optimizing the servo control system

Selecting initial values of response-defining MD

Set the following machine data in accordance with the drive time constant Ta (Ta_{real}) determined in Section 7.3.2 to the initial values for the optimization steps below, e.g. for an axis in MSR 10^{-3} mm:

- Acceleration, delay

  MD40 = MD41 [mm/s^2] = 30 · MD23 [mm/min] : Ta [ms]

- Jolt time

  MD42 (ms) = 0

- Positioning loop amplification

  MD38 (1/min) = 100,000 : Ta (ms)

The acceleration value that actually acts on the system is reduced by the time response of the position control circuit - i.e. as a function of the K_v value. The maximum acceleration (a) in this setting can be attuned to the drive time constant, and can be estimated as follows:

a_{max} [mm/s^2] = 16 · MD23 [mm/min] : Ta [ms]

The qualitative effect of the parameters on the positioning process appears in the following table:

Table 7-5 Effect of machine data that defines response

	MD38	MD40/41	MD42
Quiet running	small	–	–
Noise immunity	great	–	–
Soft movement reversal	small	great	great
Positioning without overshooting	small	great	great
Fast positioning	great	small	small

Optimization of dynamic response

The following startup actions allow you to optimize the position controller to your requirements. Check all speed ranges, and if applicable give the greatest weight in evaluating the results to the speed that is the most significant for your technology.

Optimization for uniformity of movement

You can make optimization of the position controller considerably easier by analyzing the actuating signal or drive speed (tachometer voltage) with a storage oscillograph. The resulting oscillograms for the transition functions U(t) and v(t), i.e. the oscillation pattern, can be interpreted more easily (see Figure 7-13).

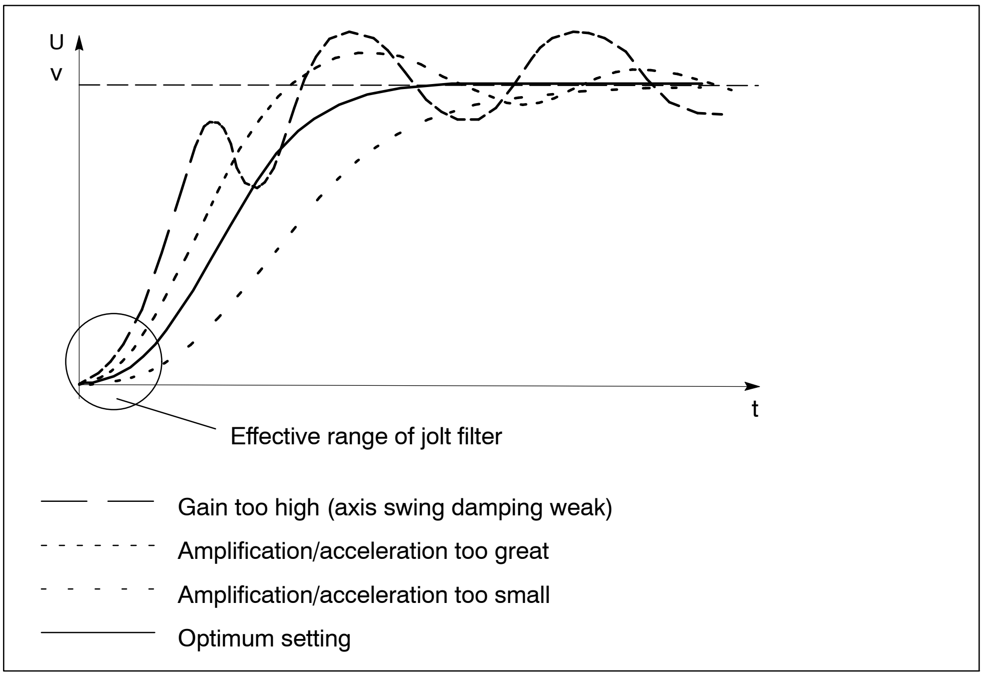


Fig. 7-13 Transition function of the position-control circuit

Optimization for overshoot

Evaluate the overshoot in the target position (s-overshoot in the servicing data).

For suitable machine data changes, see Table 7-5.

Optimization for positioning time

Evaluate the approach time to the target position (approach time Te in the servicing data).

For suitable machine data changes, see Table 7-5.

Optimization for especially soft travel (super-soft)

For particular applications, especially soft travel response of the axis is desirable. By choosing the following output values for the machine data affecting the dynamic response you can produce a very soft movement where the acceleration is controlled exclusively by the jolt filter. The effective maximum acceleration in movement-reversal processes responds proportionally to the difference in speed, and reaches its maximum in the transition from v = 0 to maximum speed (see Figure 7-14).

- Acceleration, delay

  MD40 = MD41 [mm/s^2] = 0

- Jolt time

  MD42 [ms] = 0.5 · Ta [ms]

- Positioning loop amplification

  MD38 [1/min] = 100,000 : Ta [ms]

The maximum value of the actual effective acceleration can be estimated as follows:

a_{max} [mm/s^2] = 16 · MD23 [mm/min] : Ta [ms]

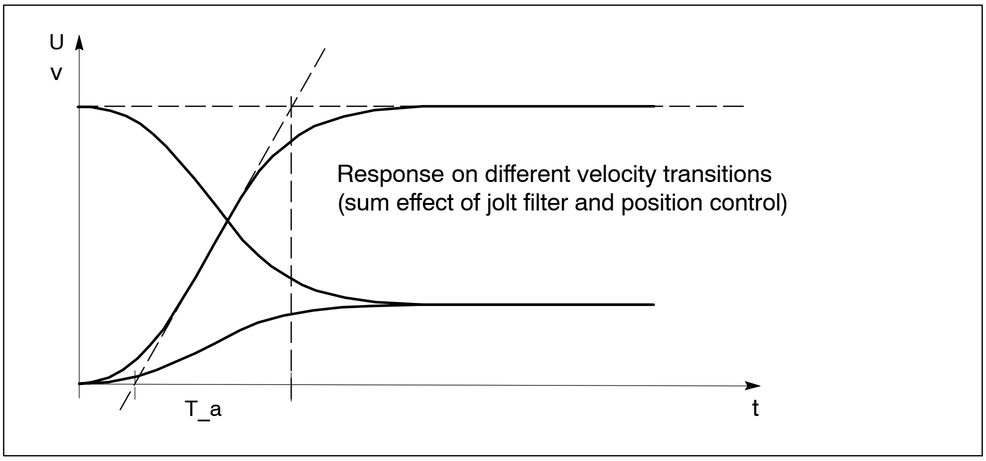


Fig. 7-14 Response on different velocity transitions (sum effect of jolt filter and position control)

Compromise optimization

When optimizing for several of the above criteria, you can determine the machine data from the results of the individual optimizations by a variety of methods:

- Guarantee of all partial results
  - Least determined value of MD38
  - Greatest value for each of MD40, MD41 and MD42
- Prioritization of one optimization criterion

  Set MD38 and MD40-MD42 to the values that match the highest-priority optimization criterion for your application, and again evaluate response as to the remaining criteria.
- Taking the mean of partial results

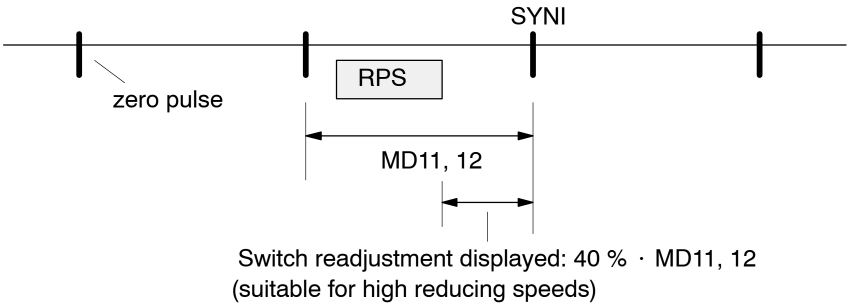
  Set MD38 and MD40-MD42 to the means of the individual partial results, and again evaluate response as to all criteria.

7.3.5 Realigning the reference point coordinates

Axis with incremental encoder

To ensure distinct reproducibility of reference recordings, it is necessary for the synchronizing zero pulse (SYNI) to be a distinct distance away from the reference point switch (RPS). At low reducing speeds, we recommend a distance of from 10% to 90% of the distance of one encoder revolution. For high reducing speeds we recommend 30% to 60%. Check this value in the servicing data report after executing a reference point approach (switch alignment value) and if you find nonconformity to the required value range, make a corresponding adjustment in the relative position allocation between the encoder and the reference point switch.

Example: Positive search direction


SYNI
zero pulse
RPS
MD11, 12
Switch readjustment displayed: 40 % · MD11, 12
(suitable for high reducing speeds)

Set the referencing velocity to the highest value compatible with your requirements. It is important to be able to decelerate to the reducing velocity across the length of the reference-point switch. If this is not the case, an additional repositioning to the RPS occurs before the search phase of the synchronizing zero pulse begins. Compare the cycle of the executed traversing movements with Section 9.2.3 and optimize the referencing speed (MD28).

Then readjust the reference-point coordinates proper by entering the necessary reference-point shift in the machine data. After the machine data is activated, the new reference-point shift takes effect with the next search for reference.

Axis with absolute encoder (SSI)

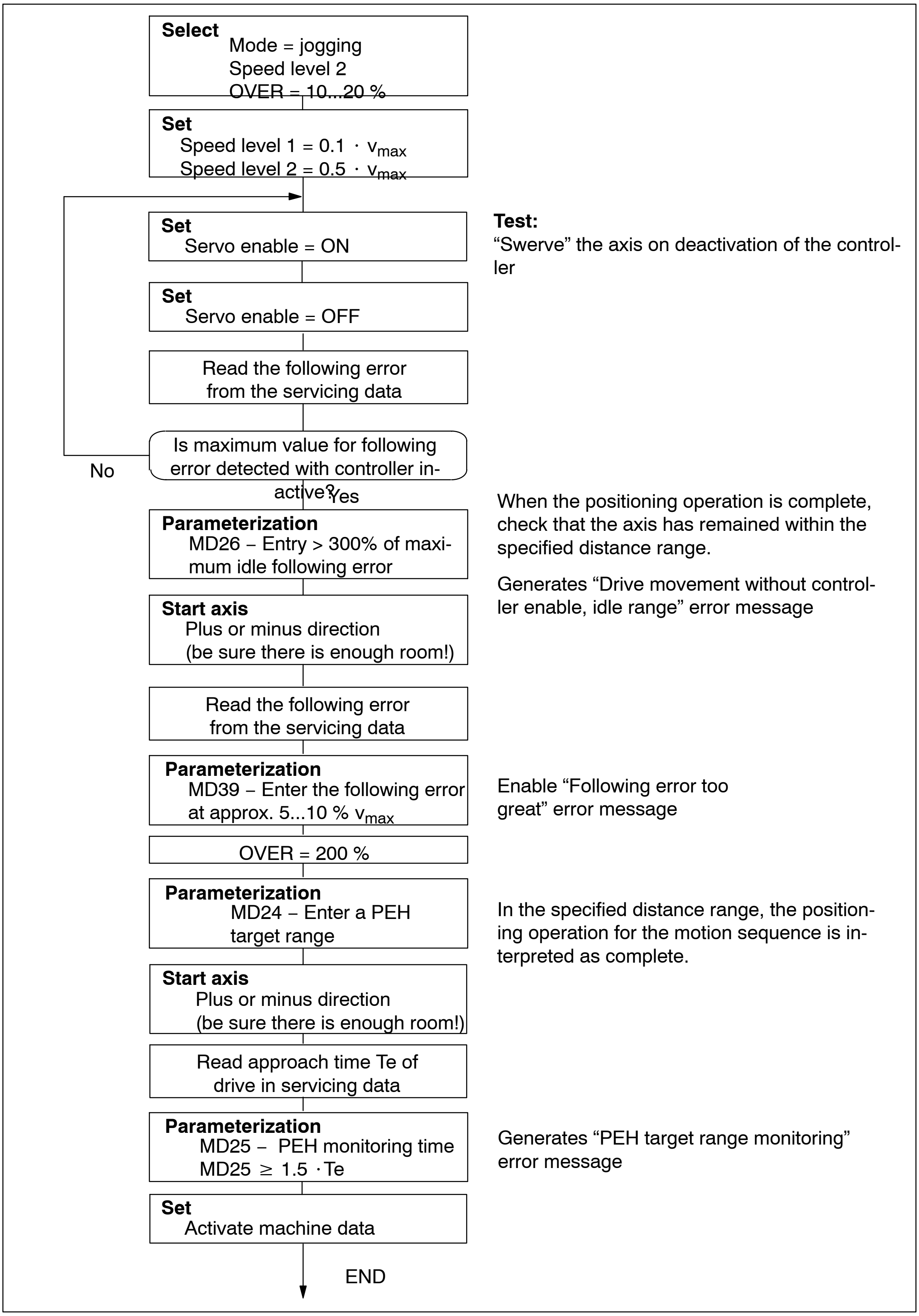
In a suitable mode (“jogging”, “incremental relative”) move to a known point on the axis and execute the Set reference point function with the known position value. The set position and actual position will immediately be set to this value, and the allocation of an absolute value to the absolute encoder (SSI) will be entered in the machine data record (MD17). If you want to archive this value externally, apart from the module’s own data memory, perform a readout of the machine data DB and save it to a floppy disk or to the hard disk of your PG.

7.3.6 Activating position controller diagnostics

Overview

Once the position controller has been optimized, activate the position controller diagnostics. If position control is performing improperly or the axis is responding abnormally, this function will trigger error messages.

You can use the following flow chart to start the position controller diagnostics:


Select
Mode = jogging
Speed level 2
OVER = 10...20 %
Set
Speed level 1 = 0.1 · v_{max}
Speed level 2 = 0.5 · v_{max}
Set
Servo enable = ON
Set
Servo enable = OFF
Read the following error from the servicing data
Is maximum value for following error detected with controller inactive?
No
Yes
Parameterization
MD26 – Entry > 300% of maximum idle following error
Start axis
Plus or minus direction
(be sure there is enough room!)
Read the following error from the servicing data
Parameterization
MD39 – Enter the following error at approx. 5...10 % v_{max}
OVER = 200 %
Parameterization
MD24 – Enter a PEH target range
Start axis
Plus or minus direction
(be sure there is enough room!)
Read approach time Te of drive in servicing data
Parameterization
MD25 – PEH monitoring time
MD25 ≥ 1.5 ·Te
Set
Activate machine data
END

Test:
“Swerve” the axis on deactivation of the controller

When the positioning operation is complete, check that the axis has remained within the specified distance range.

Generates “Drive movement without controller enable, idle range” error message

Enable “Following error too great” error message

In the specified distance range, the positioning operation for the motion sequence is interpreted as complete.

Generates “PEH target range monitoring” error message

Fig. 7-15 Activation of position controller diagnostics

7.3.7 Activating the software limit switches, drift compensation and backlash compensation

Software limit switches

Carefully run the axis up to its extreme limits as specified for normal operation of the machine. Enter these position actual values into the machine data MD21/MD22 as software limit switches, and activate them.

Note

If you change the reference-point coordinate later or use Set reference point for the absolute encoder, you must redefine the positioning values of the software limit switches.

If you do not need the software limit switches, the input limits -10^9 and 10^9 [MSR] must be entered in MD21/MD22 (for default values, see Table 5-4).

Drift compensation

If you want to use the drift compensation function in addition to the offset compensation already described in Section 7.3.2, activate it in the machine data (please see the function description in Section 9.7, Position control).

Backlash compensation

With indirect position measurement (for example, with an encoder on the motor) the free play of mechanical transmission elements during positioning may cause a position deviation of a machine part (such as a lathe saddle) that is to be positioned but does not lie in the measured-value feedback loop. As a rule, a piece of the distance will be “missing” after a reversal of direction. This backlash amount can be determined as a mean at various axis positions, and entered in the machine data MD30 and MD31.

You can use the following flow chart to determine backlash and activate backlash compensation.

Also please refer to the information on “backlash compensation” presented in Section 9.7 !

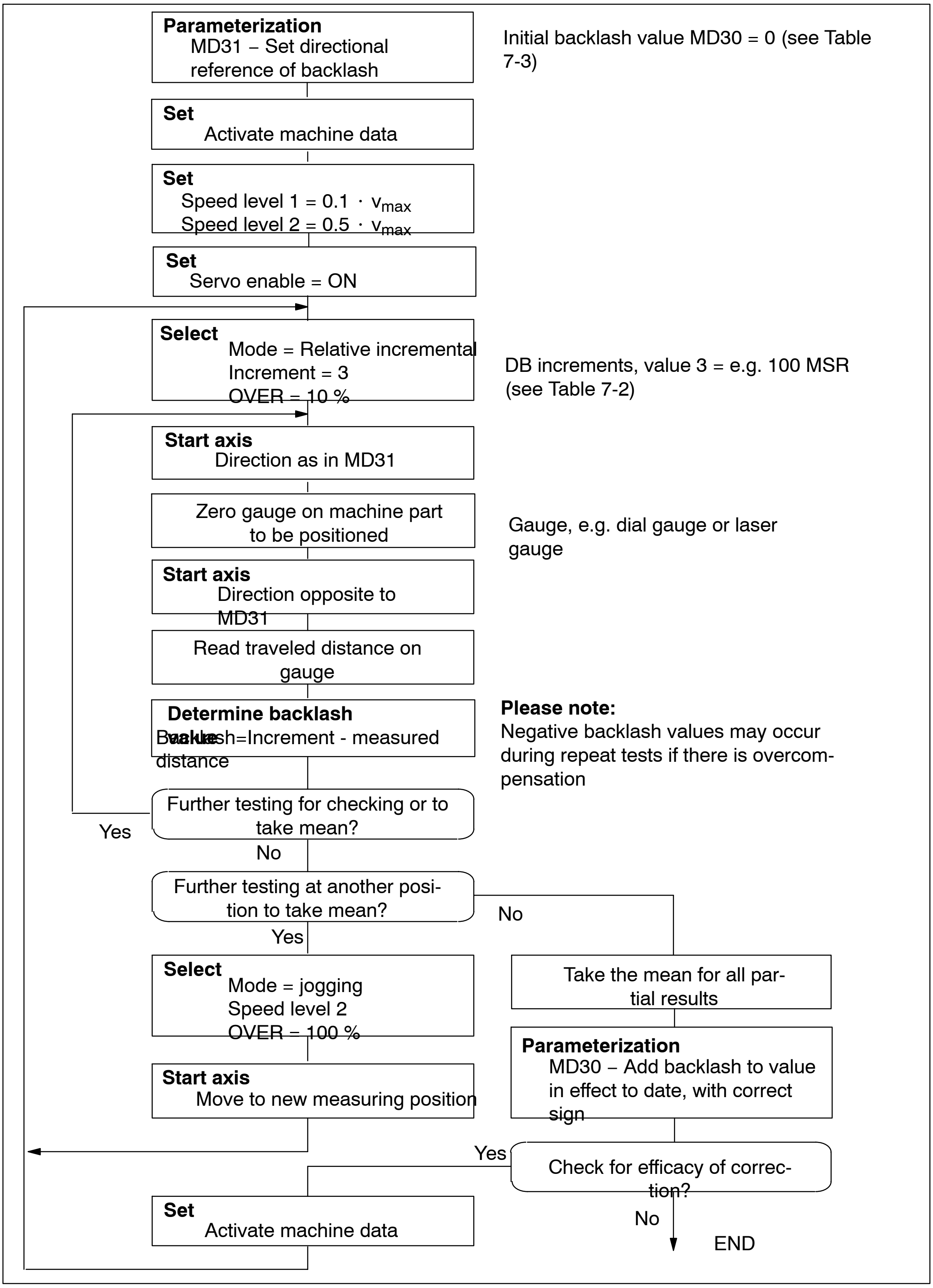

Parameterization
MD31 – Set directional reference of backlash
Initial backlash value MD30 = 0 (see Table 7-3)
Set
Activate machine data
Set
Speed level 1 = 0.1 · vmax
Speed level 2 = 0.5 · vmax
Set
Servo enable = ON
Select
Mode = Relative incremental
Increment = 3
OVER = 10 %
DB increments, value 3 = e.g. 100 MSR (see Table 7-2)
Start axis
Direction as in MD31
Zero gauge on machine part to be positioned
Gauge, e.g. dial gauge or laser gauge
Start axis
Direction opposite to MD31
Read traveled distance on gauge
Determine backlash value
Backlash=Increment - measured distance
Please note:
Negative backlash values may occur during repeat tests if there is overcompensation
Further testing for checking or to take mean?
Yes
No
Further testing at another position to take mean?
No
Yes
Select
Mode = jogging
Speed level 2
OVER = 100 %
Take the mean for all partial results
Parameterization
MD30 – Add backlash to value in effect to date, with correct sign
Start axis
Move to new measuring position
Yes
Check for efficacy of correction?
No
END
Set
Activate machine data

Fig. 7-16 Determination of backlash and activation of backlash compensation

7.3.8 Parameterizable zero-speed monitoring

Warning

The parameterizable zero-speed monitoring is only required in exceptions. The parameterization lies completely in the user's own responsibility and should be carried out with utmost care.

In the uncontrolled mode of the axis, a zero-speed monitoring of the axis motion is provided by the FM 354 via the velocity evaluation.

These are special cases in which the MD45 voltage ramp is active:

- In the "Controlling" mode:
  Deceleration of an axis motion
- In all the other modes:
  - Cancellation of drive enable during an axis motion
  - Restart during an axis motion
  - In case of all errors with error reaction "All OFF"
  - In case of some errors with error reaction "Feed STOP" (see Section 11, Error handling)

The evaluation is provided using an automatic mechanism implemented in the software, which is intended to:

- quit the "Machining running" status
- quit the position control with servo enable activated

For extraordinary exceptional cases of technologically required sequence of motions or extraordinary dynamic behaviors of an axis, for which this automatic mechanism cannot determine an axis standstill reliably, new machine data will be introduced with which exact criteria can be parameterized to detect the end of the motion.

MD54	Zero speed (see Table 5-4)
MD55	TimeOut time for zero-speed monitoring (see Table 5-4)

Note

MD54 and MD55 are zero (0) by default. Thus, the automatic zero-speed monitoring is enabled – principle of functioning as in earlier software versions.

MD54 and MD55 act independently of each other, i.e. can be parameterized in 4 possible combinations.

The TimeOut time is started after the setpoint 0 has been reached at the setpoint output of the FM 354, i.e. if MD45 is not equal to 0 after the voltage ramp has been processed.

7.3.9 Standard diagnosis for the position controller with parameterizable response time when overriding the actuating signal

Overview

The position controller of the FM 354 provides a standard diagnostic function which issues an error message when an output value of 10 V is reached as the maximum actuating signal for the drive.

Note

This diagnostic function cannot be deactivated.

It is deemed to be indispensable and serves for safety of the installation and protection of the operating personnel, in particular during commissioning of the machine.

Reaching of the 10 V output value can have the following causes and result in the following effects:

Table 7-6 Causes for reaching the 10 V output value

Effect	Cause	Response
U1 drive stopped:	Error ”Drive is not moving”	”Feedrate STOP”
U2 drive is moving in the opposite direction:	Diagnostic alarm ”Direction of rotation of the drive”	”All OFF”
U3 drive is moving in the setpoint direction:	No error message[1)]	None

[1)] Diagnosis possible by status evaluation of DS34[8].2)

The causes U1 and U2 which constitute the basis for the error messages above and result from a commissioned and optimized axis do not occur any longer in normal operation. Therefore, the appropriate error messages are already issued by the position controller when the 10 V output value occurs once.

In practice, there were several applications with inappropriately dimensioned axes. The encoder resolution in proportion to the parameterized maximum speed selected so was so bad that the distance corresponding to approximately one increment of the measuring system (”MR”) is covered when a setpoint of 10 V is output to the drive. In such an application, the axis status always represents itself as a standstill for the position controller, since also an axis being in position control can move by +/–1 increment (MR). The response criterion of the standard diagnostic function is:

- In case ”No drive motion” case: +/–2 MR/TZ
- In the ”Direction of rotation of the drive” case: v actual value sign opposite to the v setpoint

Calculating the required response time

The error message at the 10 V limit can be delayed by an appropriate time value by parameterizing a response time. If an axis motion by at least one increment (MR) in the programmed traversing direction is recognized, the response time is restarted. The error message is then issued according to the current axis status after expiry of the parameterized time.

It is recommended to proceed as follows:

- Calculate the maximum speed (MD23) into MR/TZ; (TZ = 2 ms):

  VMR = MD23 / TASTPROMIN / MWFAKTOR

  Calculate MWFAKTOR as per Section 5.3.1, Subsection ”Dependencies”.

  The number of sampling cycles per minute (TASTPROMIN) is 30,000.

- Calculate the theoretical minimum value for the required response time:

  TOV_min = 1 / VMR

- Define the practically required minimum value
  TOV = (2...3) · TOV_min. (This value, however, must be by a multiple higher in case of a bad starting behavior of the axis. When testing the axis, the value can be corrected starting from an initial value until a stable and safe positioning behavior is achieved.)

Enter the TOV value in MD56.

To avoid an undefined value in the existing MD56 automatically resulting in a new behavior when upgrading the firmware, the function must be re–enabled in MD60.

MD56	Standard diagnosis (see Table 5-4)
MD60	Function enable for the response time in MD56 (see Table 5-4)

Note

MD56 is zero (0) by default. Thus, the standard diagnosis is enabled without delay ∠ principle of functioning as in earlier software versions.

The standard diagnosis cannot be deactivated, only its effect can be delayed.

MD56 is in effect immediately when activated and entered online (category E).

While the response time is running, the override status is signaled in DS34[8].2 (this also pertains to cause U3).

Warning

Inappropriate input values can result in hazards to the installation when the parameterized response time is activated for the standard diagnosis. Therefore, the use of MD56 shall be the user’s solely responsibility. Siemens will not assume any liability in case of damage to the installation.

Parameterization examples

Example 1:

Machine data:

MD10 = 1　MD11 = 2,000,000 MSR　MD12 = 0

MD13 = 1,000　MD23 = 1,000,000 MSR/min

Calculating MD56:　MWFAKTOR = (MD11 + MD12 · 2^{-32}) / (4 · MD13)

MWFAKTOR = (2,000,000 + 0 · 2^{-32}) / (4 · 1,000) = 500

VMR = 1000000 / 30000 / 500 = 0.066...

TOV_min = 1 / 0.066... = 15

MD56 = 30 ... 45

Example 2:

Machine data:

MD10 = 1　MD11 = 20000 MSR　MD12 = 0

MD13 = 1024　MD23 = 30000000 MSR/min

Calculating MD56:　MWFAKTOR = (MD11 + MD12 · 2^{-32}) / (4 · MD13)

MWFAKTOR = (20000 + 0 · 2^{-32}) / (4 · 1024) = 4,8828125

VMR = 30000000 / 30000 / 4,8828125 = 204,8

TOV_min = 1 / 204,8 = 0,004

MD56 = 0

This case where TOV_min < 1 is not an example for an inappropriate ratio of the encoder resolution to the parameterized maximum speed. Probably, the cause lies in an FM354 error message from the mechanics or in the axis optimization.

Human-machine Interface 8

Chapter Overview

Summary

In this chapter you'll find an overview of the operator control and monitoring capabilities offered by the FM 354.

For operator control and monitoring of the FM 354, an operator panel can be connected to the CPU via the MPI interface (see Figure 1-2).

The FM can service no more than three nodes simultaneously.

The module uses the SIMATIC interface (backplane bus) to communicate with the control panel.

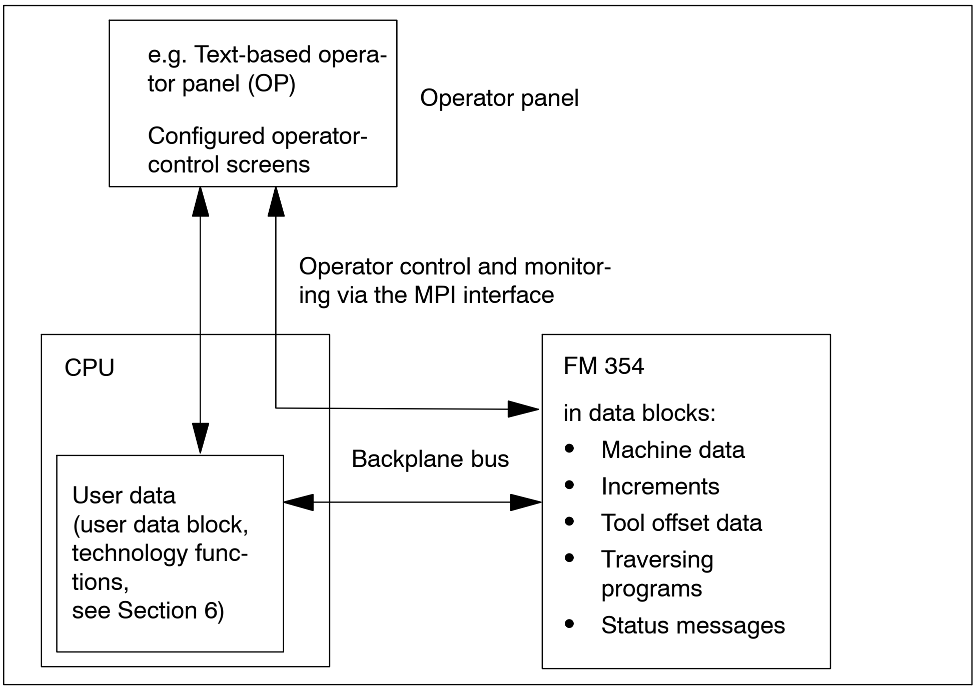

e.g. Text-based operator panel (OP)
Configured operator-control screens
Operator panel
Operator control and monitoring via the MPI interface
CPU
User data (user data block, technology functions, see Section 6)
Backplane bus
FM 354
in data blocks:
• Machine data
• Increments
• Tool offset data
• Traversing programs
• Status messages

Fig. 8-1 Operator control and monitoring for the FM 354

Operator control and monitoring of FM data/signals on the CPU 314

The data and signals that can be controlled and monitored at the control panel are listed in the user data block. These data or signals must be processed by the user program (for data and signals see Chapter 6 and Section 8.1).

What can I control on the FM 354?

Using the keyboard of the operator panel, you can change the data/signals in the data blocks:

- Machine data DB No. 1200
- Increment sizes DB No. 1230
- Tool offset data DB No. 1220
- Traversing programs DB No. 1001...1199

What can I monitor on the FM 354?

The following data and signals can be displayed on the operator panel display:

- Machine data DB No. 1200
- Increment sizes DB No. 1230
- Tool offset data DB No. 1220
- Traversing programs DB No. 1001...1199
- Status messages DB No. 1000 (DB-SS) etc.
  - Operating data, such as actual values
  - Active NC blocks
  - Linear measurements
  - Actual value block change
  - Check-back signals and error conditions
  - Service data

The configuration package includes a pre-configured interface for the COROS OP 07 and OP 17 operator panels.

8.1 Standard HMI (human-machine interface) for the OP 07 and the OP 17

Overview

This Section describes a preconfigured user interface, which you will need to change according to your project (e. g. FM addresses, DB no.), for the following COROS equipment (operator panels):

- OP 07
- OP 17

The tool to be used for this is the ”ProTool/Lite” configuring tool. You can use it to modify, add or delete graphics.

The user interface is addressed to user DB No.1 in the CPU (target system 1, address = 2) and to the DB-SS of the FM 354 (target system 2, address = 3).

The text field “FM user name” represented in the images can be renamed to a text of your choice.

You can print out the entire configuration using ”ProTool/Lite”. This provides you with detailed graphics descriptions.

You will find the preconfigured user interfaces **op07_354.pdb** und **op17_354.pdb** in the following directory:

STEP7 Directory\EXAMPLES\FM354\zEn14_02_FM354_OP_EX

DB-SS

This data block for status messages (DB 1000) contains the control/checkback signals, as well as the system data of the FM 354. The data of the DB-SS can only be read.

Monitoring

The data for monitoring can be read and displayed directly in the DB-SS as well as in the corresponding parameterized DBs of the FM 354.

The advantage of reading directly from the FM is that the values/signals do not have to be read via the user program first.

Operator control

For operator control, the data and signals (including memory bits and values) are written to the user DB of the user program.

User program

User data bock AW−DB is the interface for the OP.

When control signals, single functions and single commands are set in the user data block via the OP, they are immediately transferred to the FM by the POS_CTRL block.

The signals written in “Operator Control and Monitoring” (requests to transfer the data with Write as per Table 8-3) must be evaluated accordingly by the user program (taking into account any user−specific interlocks) before setting the Write or Read requests.

8.1.1 Standard user interface for the OP 07

User interface of the OP 07

The following illustration provides you with an overview of the OP 07 user interface (menu tree).

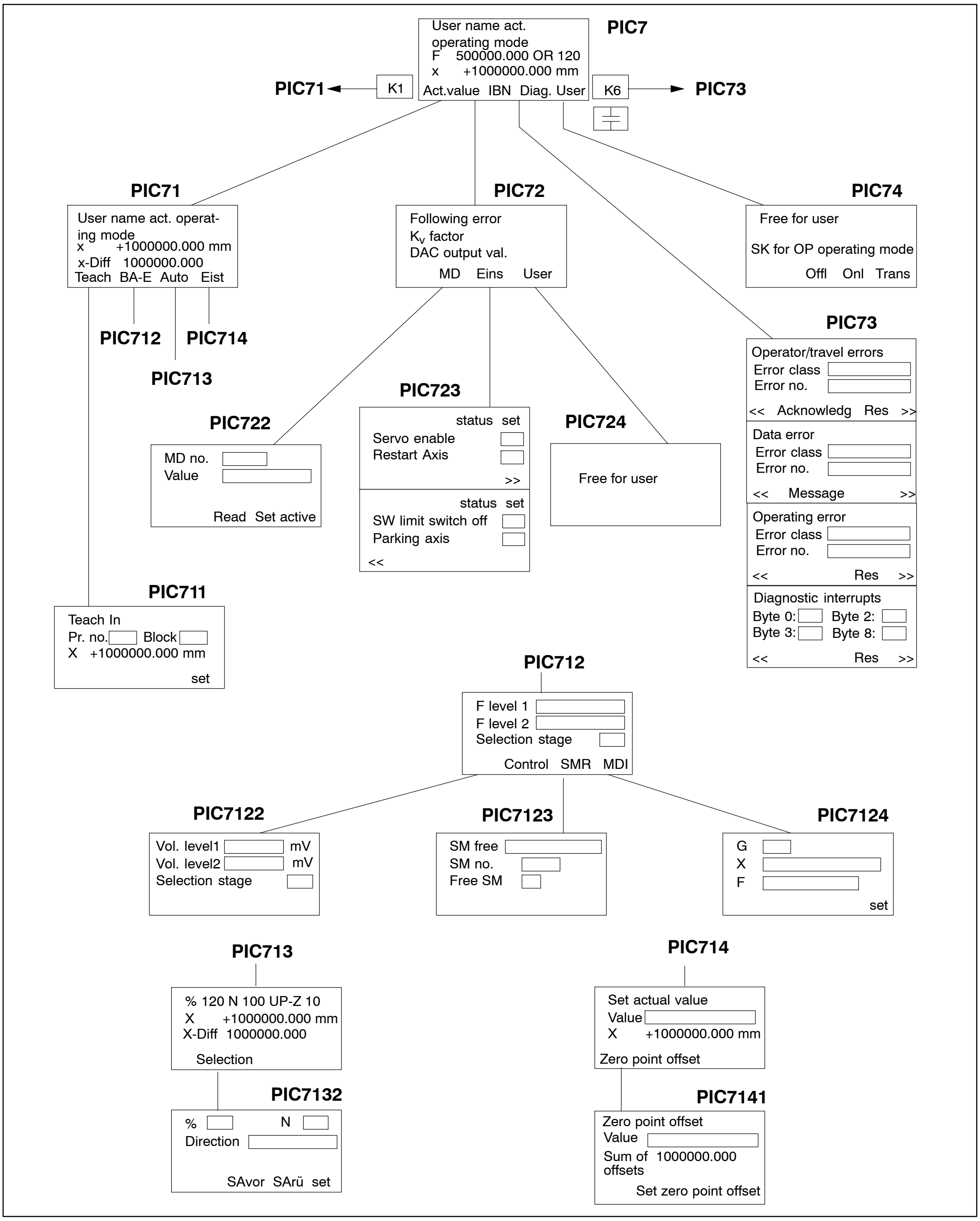


Fig. 8-2 Menu tree of the OP 07 user interface

Figure 8-2 describes the functions of the global function keys for the user interface of the OP 07.

┼ **ESC key** You can use this key to call up the previous screen of the higher levels.

>> << **Soft keys** You can use these keys to call up the next inset screen within the same screen (same screen number).

K1 **Function key** You can use this key to skip from any point in the menu tree to the actual value display main screen (PIC71).

K6 **Function key** You can use this key to skip from any point in the menu tree to the diagnostic main screen (PIC73).

Note

The screens of the user interface (see Figure 8-2 and description of each screen, Table 8-1) contain display fields and input/output fields. These fields contain values of configured variables.

- The display fields are addressed to the DB-SS (“Control 2,” DB1000) and are read directly from the FM 354.
- The input/output fields are addressed to the user DB No.:1 (Control 1-CPU).
  - Transmission of these values occurs from the OP 07 to the CPU into the user DB. These values (if needed) must be transmitted to the FM 354 by the user program.
  - If certain values or control signals can be written only under the right conditions (e.g. if axis is in HOLD mode or selection of a certain operating mode is required), then the user program must ensure, by analyzing the response signals, that these conditions are met.

Table 8-1 below describes the individual screens of the user interface.

Table 8-1 Description of the screens in the user interface

Screen name	Screen No.	Description
Main screen	PIC7	This screen is displayed to you after the OP 07 is enabled. The FM354 values are display values. You can use the soft keys to call up the following main screens: • Softkey “Istw” → PIC71 • Softkey “IBN” → PIC72 • Softkey “Diag” →PIC73 • Softkey “Anw” → PIC74
Main screen, actual value display	PIC71	The values on the screen are display values. You can use the soft keys to call up the following screens: • Softkey “Teach” → PIC711 • Softkey “BA-E” → PIC712 • Softkey “Auto” → PIC713 • Softkey “Eist” → PIC714
Main screen, startup	PIC72	The FM 354 values (service data) are display values. You can use the soft keys to call up the following screens: • Softkey “MD” →PIC722 • Softkey “Eins” → PIC723 • Softkey “Anw” → PIC724
Main screen, diagnostics	PIC73	This screen shows you the diagnostic interrupts and error messages for the FM 354 in four lower-level screens. The following are displayed: • Error class and error number for operator control and guidance errors • The bit numbers of the single diagnostic interrupt bytes You can use the soft keys “Quit” and “Res” to acknowledge the errors. See Section 11.3 for diagnostic interrupts and error messages
User screens	PIC74 and PIC724	You can use these screens to configure your own contents.
Setting data for operating mode	PIC712	This screen contains input/output fields for the velocity levels or a text field for selecting velocity level (selections 1 or 2 possible). You can use the soft keys to call up the following screens: • Softkey “Steu” → PIC7122 • Softkey “SMR” → PIC7123 • Softkey “MDI”→ PIC7124
Setting data for the “Automatic” operating mode	PIC713	This screen contains only display fields. You can use the “Anwahl” soft key to call up screen PIC 7132.
Set actual value	PIC714	The actual value display is a display field. The value for set an actual value appears in an input/output field. You can use the soft key “NPVer” to call up the screen PIC7141. You can use the soft keys “Iwrü” and “IWset” to execute functions.

Table 8-1 Description of the screens in the user interface, continued

Screen name	Screen No.	Description
Machine data	PIC722	This screen contains input/output fields. Input of values is password protected. The values entered are in the user DB. You can use the soft keys to set bits in the user DB: • Soft key “lesen” – You can use this soft key to insert a memory bit (in the user DB) which causes the user program to read the item of machine data whose number has been entered. • Soft key “set” – You can use this soft key to set a bit which causes the user program transfer to the FM 354 the number entered under value for the item of machine data under MD No. • Soft key “aktiv” – You can use this soft key (activate MD) to set a memory bit (“activate MD” bit in the user DB) which is transmitted from the user program to the FM 354. Each bit in the user DB which you set in this screen must be reset by the user program following execution of the function.
Settings for startup	PIC723	The fields of this screen which are identified by an “x” are display fields. If the bit is set, you will see an x. If the bit is not set, the field will be blank. The other fields are text entry fields in which you can toggle between “x” and “ ”. You can use the soft keys “<<” and “>>” to change between the inset screens of this screen.
Set data for the “Control” operating mode	PIC7122	This screen contains input/output fields for the control levels or a text field for control level selection (selections 1 or 2 possible).
Set data for the “incremental mode, relative” operating mode	PIC7123	This screen contains input/output fields. The fields “SM-No.” and “free SM” are addressed to the same address in the user DB. You can use the SM No. to enter the values 1 to 100. The “free SM” field is a text field and you can toggle back and forth between “x” and “”. If you select “x”, this corresponds to the value 254 for free SM. The value of the free SM is found in the user DB.
Set data for the “MDI” operating mode	PIC7124	This screen contains input/output fields. The MDI data record must be preassigned by the user program with the corresponding bits (G, X, F bits are set) and possibly, with values for G, X and F. The entry field behind G is a text field. There, you can select between the values 90 and 91. Using the soft key “set” you can set a memory bit (in the user DB), which must be analyzed by the user program. With modification of this bit, the MDI block must be transmitted from the user DB to the FM 354, and the bit must be reset.

Table 8-1 Description of the screens in the user interface, continued

Screen name	Screen No.	Description
Program selection	PIC132	This screen contains input/output fields. This field for direction is a text field. It is upward and downward selectable. You can use the soft keys to set bits in the user DB: • Soft key “SAvor” - You can also use this soft key (automatic block search forward) or “SArü” (automatic block search backward) to set the corresponding bits in the user DB. • Soft key “set” - you can use this soft key to set a memory bit (in the user DB).
Teach In	PIC711	This screen contains input/output fields. You can use the soft key “set” to set a bit in the user DB. The actual value display (X) is a display field.
Zero offset	PIC7141	This screen features an input/output field. The sum of the offsets (Sumn Ver) is a display field. You can use the soft key “NPVset” to execute a function.

8.1.2 Standard user interface for the OP 17

User interface of the OP 17

The following illustration provides you with an overview of the OP 17 user interface (menu tree).

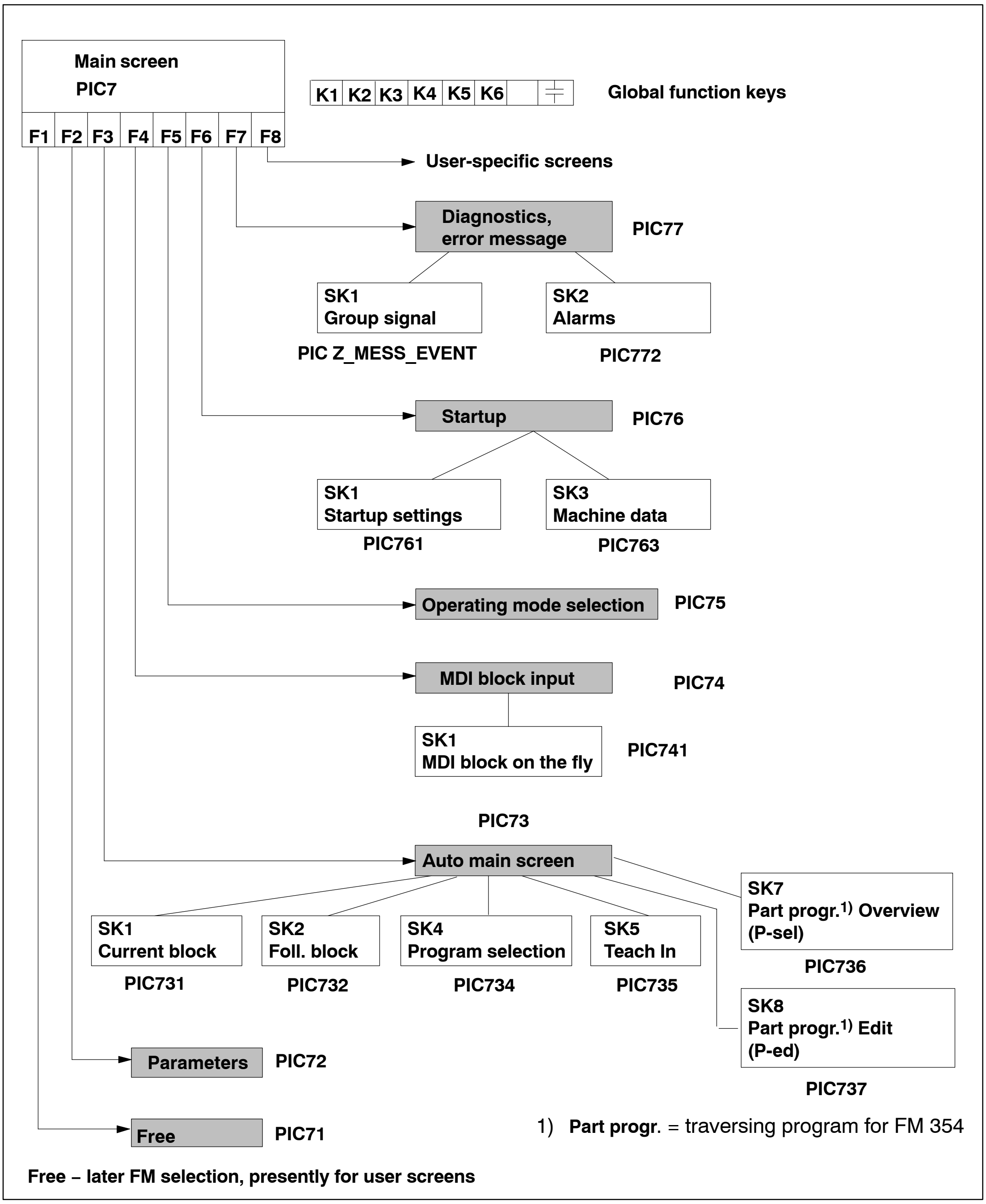


Fig. 8-3 Menu tree of the OP 17 user interface

Figure 8-3 describes the functions of the global function keys for the user interface of the OP 17.

	ESC key	You can use this key to call up the previous screen of the higher level (the table of contents in the main screen).
K1	**Function key**	You can use this key to jump from any point on the menu tree to the main screen (PIC7).
K2	**Function key**	You can use this key to jump from any point on the menu tree to the diagnostics, error message screen (PIC77).
K3	**Function key**	You can use this key to jump from any point on the menu tree to the operating mode selection screen (PIC75).
K4	**Function key**	OP 17 “Offline” operating mode selection
K5	**Function key**	OP17 “Online” (normal) operating mode selection
K6	**Function key**	OP 17 “Transfer” operating mode selection
F1 ... F8		F1 to F8 (local soft keys)

Note

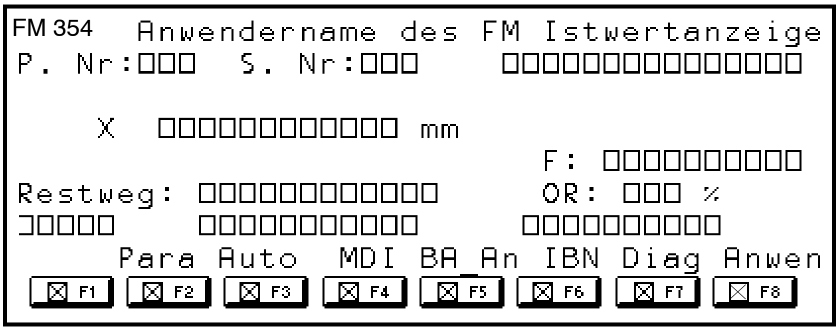
The screens of the user interface (see Figure 8-3 and description of the individual screens) contain display fields and input/output fields. These fields contain values of configured variables.

- The display fields are addressed to the DB-SS (“Control 2,” DB1000) and are read directly from the FM 354.
- The input/output fields are addressed to user DB No.: 1 (control 1-CPU).
  - Transmission of these values occurs from the OP 17 to the CPU into the user DB. These values (if needed) must be transmitted to the FM 354 by the user program.
  - If certain values or control signals can be written only under the right conditions (e.g. if axis is in HOLD mode or selection of a certain operating mode is required), then the user program must ensure, by analyzing the response signals, that these conditions are met.

The pending errors are displayed in the errors line. More detailed error information is provided on the screens “Diagnostics, Troubleshooting” and “Interrupt messages.”

Description of the individual screens

The following illustration shows the screen layout of the preconfigured interface.

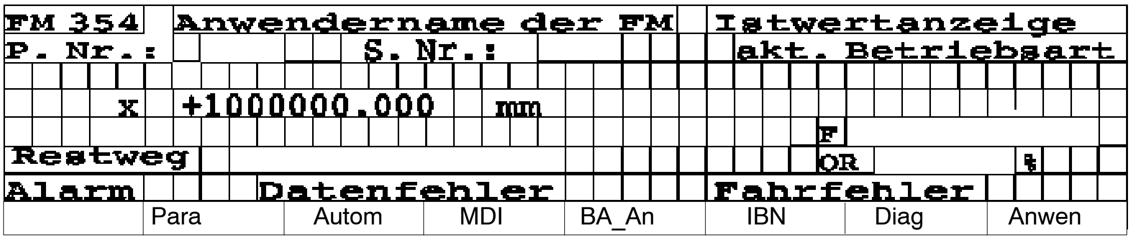


```
FM 354  Anwendername des FM Istwertanzeige
P. Nr:□□□  S. Nr:□□□    □□□□□□□□□□□□□□□□

    X  □□□□□□□□□□□□ mm
                              F: □□□□□□□□□□
Restweg: □□□□□□□□□□□□     OR: □□□ %
□□□□□    □□□□□□□□□□□     □□□□□□□□□□
     Para Auto  MDI BA_An IBN Diag Anwen
 ☒ F1  ☒ F2  ☒ F3  ☒ F4  ☒ F5  ☒ F6  ☒ F7  ☒ F8
```

Fig. 8-4 Screen layout of the preconfigured interface

The illustrations shown below (Fig. 8-5 to Fig. 8-19) provide you with the screen content of the configured screens.

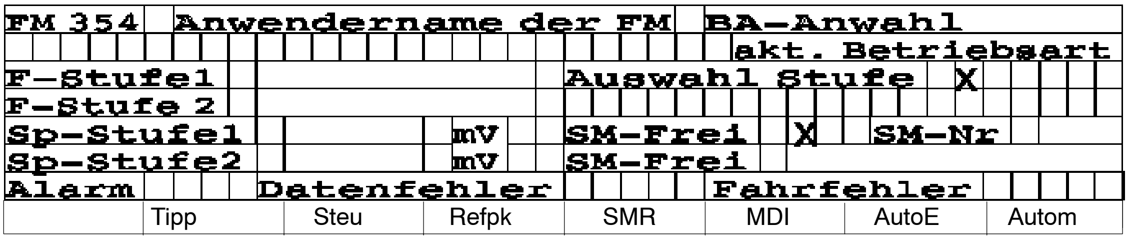


```
FM 354  Anwendername der FM  Istwertanzeige
P. Nr.:          S. Nr.:      akt. Betriebsart

     X  +1000000.000  mm
                                        F
Restweg                                 OR      %
Alarm      Datenfehler       Fahrfehler
     Para   Autom   MDI   BA_An   IBN   Diag   Anwen
```

Fig. 8-5 Main screen, PIC7

This screen is displayed to you after the OP 17 is enabled. The FM354 values are display values. The soft keys (F1 - F8) are used to request the continuation screens. The first and last soft keys (User) are kept free for the user to insert his/her own screens (e.g. including additional FMs).



```
FM 354  Anwendername der FM  BA-Anwahl
                              akt. Betriebsart
F-Stufe1                Auswahl Stufe  X
F-Stufe 2
Sp-Stufe1       mV      SM-Frei   X   SM-Nr
Sp-Stufe2       mV      SM-Frei
Alarm      Datenfehler       Fahrfehler
     Tipp   Steu   Refpk   SMR   MDI   AutoE   Autom
```

Fig. 8-6 Operating mode selection, PIC75

This screen permits you to enter the operating mode, velocity levels or control levels or the incremental value.

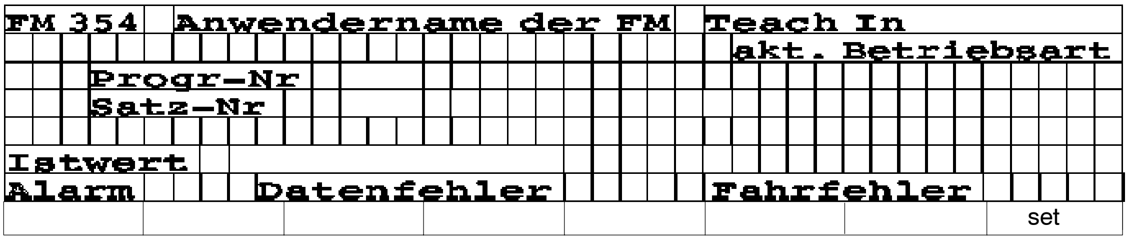


Fig. 8-7 Teach In PIC735

This illustration displays the actual value for you. You can enter the values for Teach In.

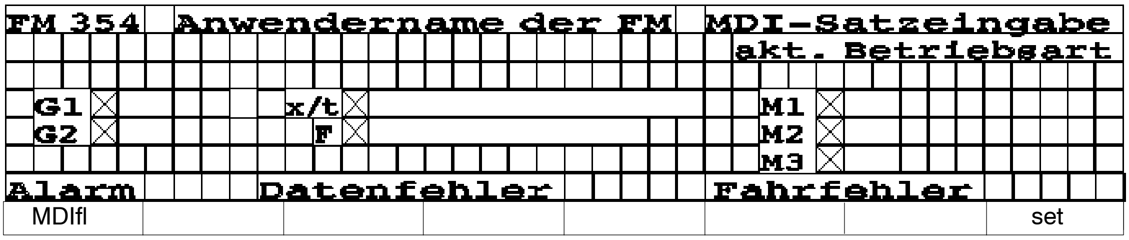


Fig. 8-8 MDI block entry PIC74

The fields identified by an X are text fields and can be toggled back and forth between “X” and “ ”. You can enter the values of the MDI block.

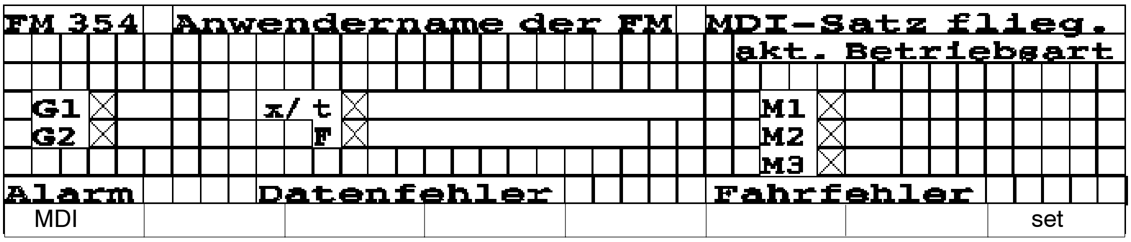
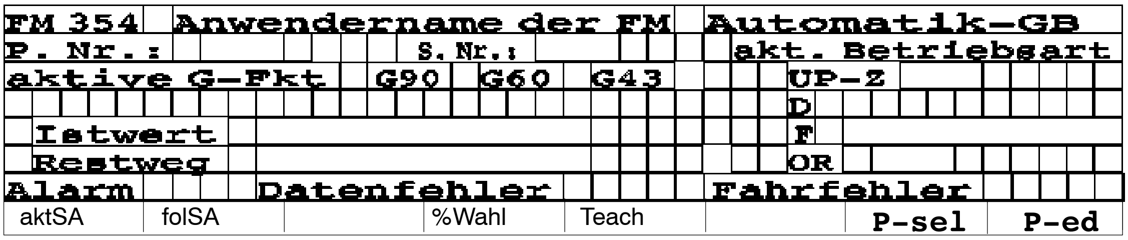


Fig. 8-9 MDI block on the fly PIC741

Handling of the MDI block on the fly corresponds to that of the MDI block.

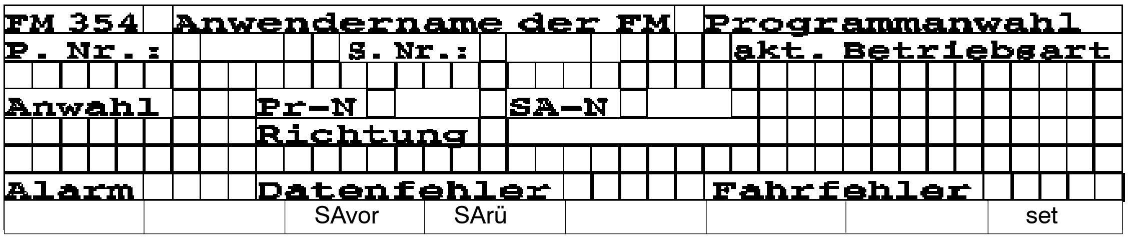


```
FM 354  Anwendername der FM  Automatik-GB
P. Nr.:            S. Nr.:   akt. Betriebsart
aktive G-Fkt  G90  G60  G43  UP-Z
                             D
 Istwert                     F
 Restweg                     OR
Alarm   Datenfehler          Fahrfehler
aktSA  folSA  %Wahl  Teach   P-sel  P-ed
```

Fig. 8-10 Automatic main screen PIC73

This screen contains only display fields.

In the screens PIC736 ”Part program overview (P-sel)” and PIC737 ”Part program Edit (P-ed)”, you can select programs and read/write traversing program blocks.



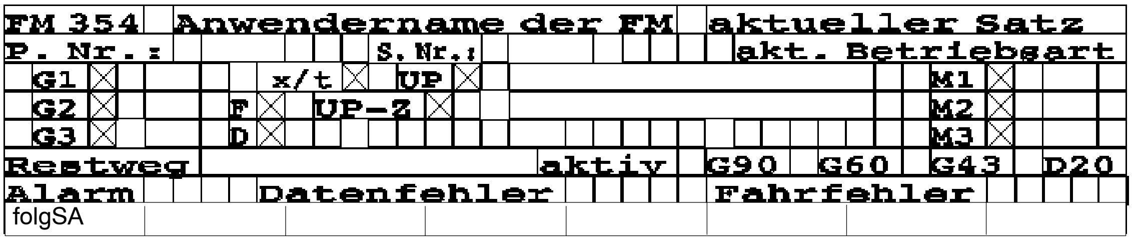
```
FM 354  Anwendername der FM  Programmanwahl
P. Nr.:          S. Nr.:     akt. Betriebsart

Anwahl     Pr-N        SA-N
           Richtung

Alarm      Datenfehler       Fahrfehler
           SAvor  SArü                      set
```

Fig. 8-11 Program selection PIC734

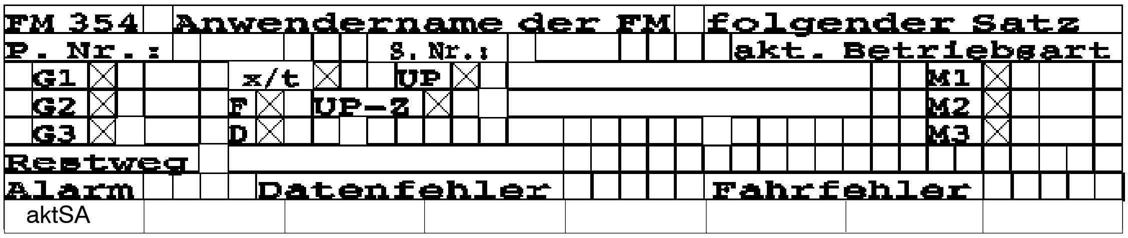
This screen features input/output fields. It is upward and downward selectable.



```
FM 354  Anwendername der FM  aktueller Satz
P. Nr.:          S. Nr.:     akt. Betriebsart
 G1 ☒      x/t ☒  UP ☒                 M1 ☒
 G2 ☒    F ☒  UP-Z ☒                   M2 ☒
 G3 ☒    D ☒                           M3 ☒
Restweg           aktiv  G90  G60  G43  D20
Alarm      Datenfehler       Fahrfehler
folgSA
```

Fig. 8-12 Current block PIC731

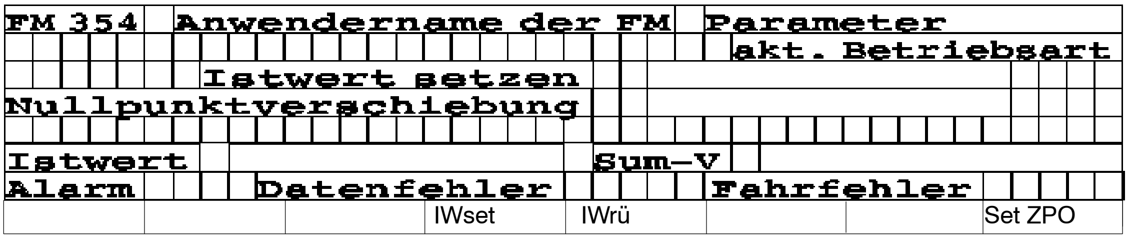
This screen contains only display fields.



```
FM 354  Anwendername der FM  folgender Satz
P. Nr.:          S. Nr.:     akt. Betriebsart
 G1 ☒     x/t ☒  UP ☒                  M1 ☒
 G2 ☒    F ☒  UP-Z ☒                   M2 ☒
 G3 ☒    D ☒                           M3 ☒
Restweg
Alarm      Datenfehler       Fahrfehler
aktSA
```

Fig. 8-13 Continuation block PIC 732

This screen contains only display fields.



FM 354 Anwendername der FM Parameter

akt. Betriebsart

Istwert setzen

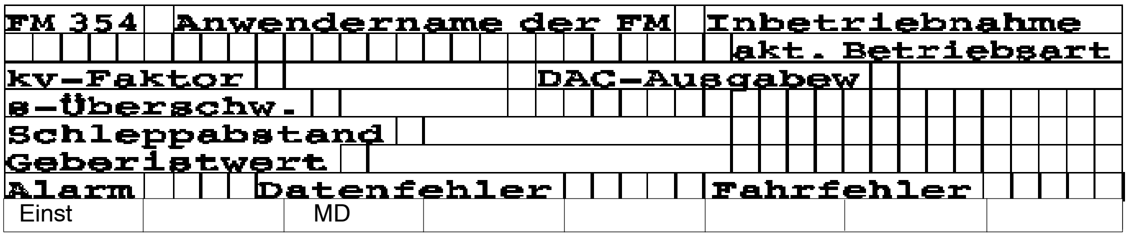
Nullpunktverschiebung

Istwert Sum-V

Alarm Datenfehler Fahrfehler

Fig. 8-14 Parameters, PIC72

The sum of the offsets and the actual value display are display fields.



FM 354 Anwendername der FM Inbetriebnahme

akt. Betriebsart

kv-Faktor DAC-Ausgabew

s-Überschw.

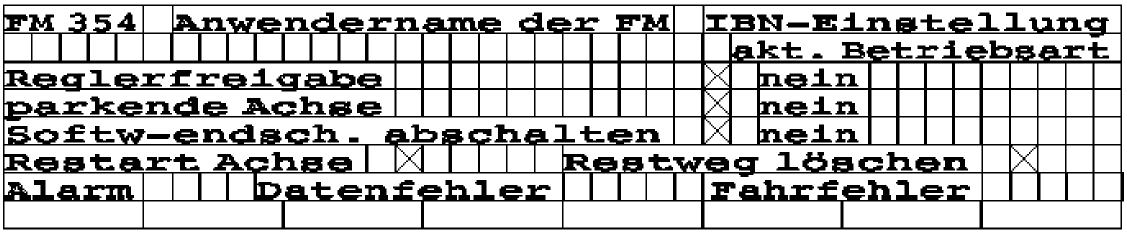
Schleppabstand

Geberistwert

Alarm Datenfehler Fahrfehler

Fig. 8-15 Startup PIC76

The FM 354 values (service data) are display values.



FM 354 Anwendername der FM IBN-Einstellung

akt. Betriebsart

Reglerfreigabe ☒ nein

parkende Achse ☒ nein

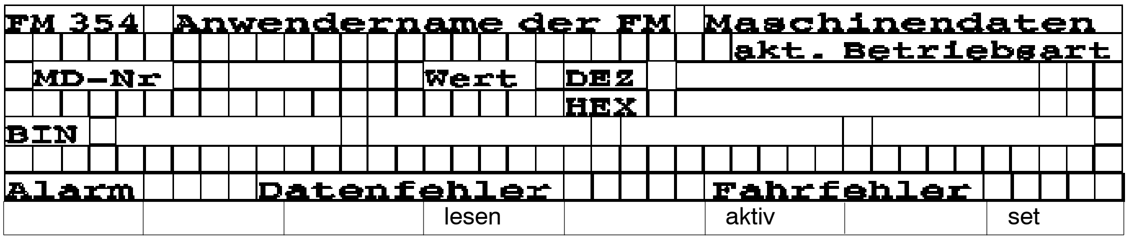
Softw-endsch. abschalten ☒ nein

Restart Achse ☒ Restweg löschen ☒

Alarm Datenfehler Fahrfehler

Fig. 8-16 Startup settings PIC761

The fields in which an “X” is entered are display fields.



FM 354 Anwendername der FM Maschinendaten

akt. Betriebsart

MD-Nr Wert DEZ

HEX

BIN

Alarm Datenfehler Fahrfehler

Fig. 8-17 Machine data PIC763

Value input is password protected.

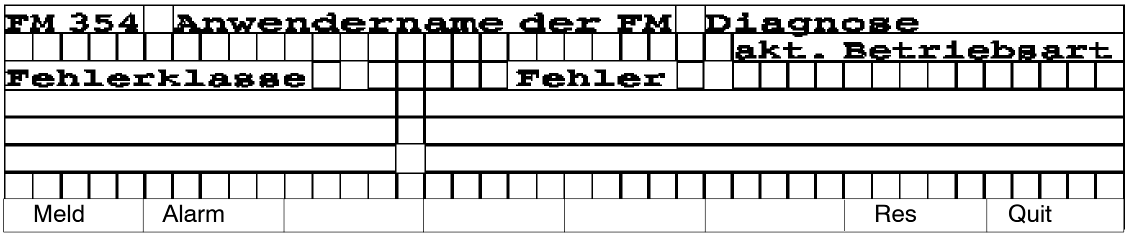


Fig. 8-18 Diagnostics, error message PIC77

This screen displays the FM 354 errors. The fields are display fields.

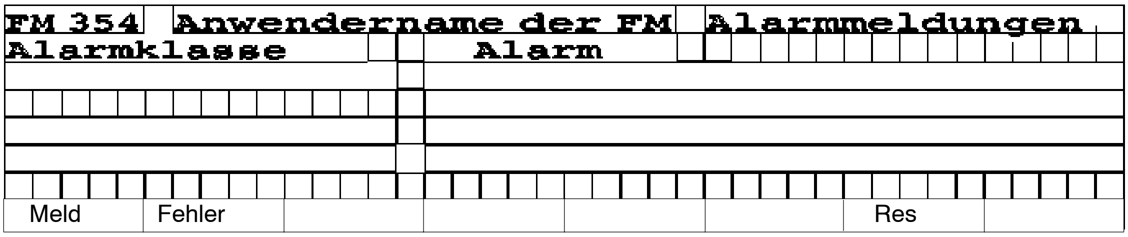


Fig. 8-19 Interrupt messages PIC772

This screen displays the FM 354 errors. The fields are display fields.

The screen “PICZ_MESS_EVENT” contains factory-supplied default “ProTool/Lite” screens for the OP17.

8.2 Analysis of the user DB by the user program for operator control

Overview

The Table below tells you which Write requests must be submitted by the user program and which signals are written directly to the FM.

Table 8-2 Analysis of the user DB by the user program

OP 07/17		User program		see PIC	
User DB, DBX	Triggered by	Function	User DB, DBX	OP 07	OP 17
499.5 499.6 499.7	FM 354	Diagnostic interrupt Data error Operator control/traversing error		–	7
499.1 = 1	SK ”IWset”	Transfer data for “Set actual value” from the user DB to the FM	38.7	714	72
499.2 = 1	SK ”NPVset”	Transfer data for “Set zero point offset” from user DB to the FM	39.1	7141	
37.6 = 1	SK ”IWrü”	Transfer “Cancel setting of actual value” to the FM		714	
37.2 = 1	SK ”SAvor”	Transfer “Block advance” to the FM		7132	734
37.3 = 1	SK ”SArü”	Transfer “Block return” to the FM			
498.3 = 1	SK ”set”	Transfer data for “Program selection” from the user DB to the FM	39.5		
498.4 = 1	SK ”set”	Transfer data for “Teach-in” from the user DB to the FM	39.7	711	735
498.2 = 1	SK ”set”	Transfer data for “MDI block entry” from the user DB to the FM	38.3	7124	74
499.0 = 1	SK ”set”	Transfer data for “MDI block on-the-fly” from the user DB to the FM	38.4	–	741
34.0	TF ”Servo enable”	To change “Servo enable”, transfer yes/no to the FM		723	761
34.6	TF ”park. axis”	To change “Parking axis”, transfer yes/no to the FM			

SK = Softkey, TF = Text field

1) The relevant code for the operating mode (BA) is to be entered in DBB16 of the user DB

2) Code = 254 in DBB17 of the user DB

3) User DB, DBB196 = 1, DBB197 from DBW500+1, DBB198 = 1, DBB199 = 1

4) User DB, DBB196 = 1, DBB197 from DBW500+1, DBB198 = 1, DBB199 = 4, DBD200 of DBD502

5) User DB, DBBX499.5 must be acknowledged with DBX515.7, DBX399.6 and DBX399.7 must be acknowledged with DBX515.6

Table 8-2 Analysis of the user DB by the user program, continued

OP 07/17	Triggered by	User program		see PIC	
User DB, DBX		Function	User DB, DBX	OP 07	OP 17
514.6 = 1	SK ”Jog”	Transfer data for “Jog” mode and “Jog” mode[1] to the FM	38.0	–	75
514.0 = 1	SK ”Control”	Transfer data for “Control” mode and “Control” mode[1] to the FM	38.1		
514.1 = 1	SK ”Refpt”	Transfer “Reference point approach” mode[1] to the FM			
514.2 = 1	SK ”SMR”	Transfer data for “Incremental relative” mode and “Incremental relative” mode[1] to the FM	38.2[2]		
514.3 = 1	SK ”MDI”	Transfer “MDI” mode[1] to the FM			
514.4 = 1	SK ”AutoE”	Transfer “Automatic single block” mode[1] to the FM			
514.5 = 1	SK ”Autom”	Transfer “Automatic” mode[1] to the FM			
35.6	TF ”Software limit switch off”	To change “Disable software limit switch”, transfer yes/no to the FM		723	761
37.5 = 1	TF ”Restart axis”	Transfer “Restart axis” to the FM		723	
37.1 = 1	TF ”Delete distance to go”	Transfer “Delete distance to go” to the FM		–	
498.1 = 1	SK ”read”	Read MD No. from the user DB, retrieve its value from the FM, and enter that value in the user DB	[3] 39.3 43.3	722	763
37.0 = 1	SK ”active”	Transfer “Active MD” to the FM			
498.0 = 1	SK ”set”	Transfer MD No. and its value from the user DB to the FM	[4] 39.3		
515.7 = 1	SK ”Res”	Error acknowledgement “Res” in the FM 354 (diagnostic interrupt)	[5]	73	77
515.6 = 1	SK ”Ack”	Error acknowledgement “Ack” in the FM 354 (data error, operator control/traversing error)			

SK = Softkey, TF = Text field

[1] The relevant code for the operating mode (BA) is to be entered in DBB16 of the user DB

[2] Code = 254 in DBB17 of the user DB

[3] User DB, DBB196 = 1, DBB197 from DBW500+1, DBB198 = 1, DBB199 = 1

[4] User DB, DBB196 = 1, DBB197 from DBW500+1, DBB198 = 1, DBB199 = 4, DBD200 of DBD502

[5] User DB, DBBX499.5 must be acknowledged with DBX515.7, DBX399.6 and DBX399.7 must be acknowledged with DBX515.6

Variables in the user DB

The Table below contains the signals/data that must be entered in the user DB via the OP (FM interface).

For a detailed description of the structure of the user DB, see Section 6.6.

Table 8-3 Variables for the user DB

User-DB	Variable type	Description	User-DB
DBB17	BYTE	Velocity or voltage level 1, 2 [BP]	–
	BOOL	Single functions	Internal Write request
DBX34.0		Servo enable	
DBX34.6		Parking axis	
DBX35.6		Disable software limit monitoring	
	BOOL	Single commands	Internal Write request
DBX37.0		Activate machine data	
DBX37.1		Delete distance to go	
DBX37.2		Automatic block advance	
DBX37.3		Automatic block return	
DBX37.5		Restart	
DBX37.6		Undo set actual value	
DBD140	DINT	Zero offset	DBX39.1
DBD144	DINT	Set actual value	DBX38.7
DBD156	DWORD	Setpoint for incremental dimension	DBX38.2
DBD160	DWORD	Speed level 1	DBX38.0
DBD164	DWORD	Speed level 2	
DBD168	DWORD	Voltage/frequency level 1	DBX38.1
DBD172	DWORD	Voltage/frequency level 2	
DBB176 to DBB195	STRUCT	MDI block	DBX38.3
DBB222 to DBB241	STRUCT	MDI block on-the-fly	DBX38.4
DBB242	BYTE	Program selection – program number	DBX39.5
DBB243	BYTE	Program selection – block number	
DBB244	BYTE	Program selection – direction	
DBB250	BYTE	Teach in – program number	DBX39.7
DBB251	BYTE	Teach in – block number	
DBW500	WORD	MD No.	–
DBD502	DINT/ relev. MD	MD value	–
DBB506	BYTE	SM No.	–

8.3 Data block for status messages (DB-SS)

Overview

The following table lists the parameters/data that can be read out during operation.

Table 8-4 Parameters/data of the DB–SS (DB No. 1000)

Byte	Variable type	Value	Significance of the variables	Comment
0...35			DB header	
36...59			Internal header information	
Off-set[1]	**Variable type**	**Value**	**Significance of the variables**	**Comment**
24	8 x BOOL		Control signals	Byte 0
25	8 x BOOL		Control signals	Byte 1
26	2 x BYTE		Control signals	Byte 2, 3
28	2 x BYTE		Control signals	Byte 4, 5
30	8 x BOOL		Checkback signals	Byte 0
31	8 x BOOL		Checkback signals	Byte 1
32	BYTE		Checkback signals	Byte 2
33	8 x BOOL		Checkback signals	Byte 3
34	BYTE		Checkback signals	Byte 4
35	8 x BOOL		Checkback signals	Byte 5
36	12 x BYTE		Reserved	
48	DWORD		Velocity level 1	
52	DWORD		Velocity level 2	
56	DWORD		Control level 1	
60	DWORD		Control level 2	
64	DWORD		Setpoint for incremental value	
68	STRUCT	MDI block structure	MDI block	
88	16 x BOOL		Single functions	
90	16 x BOOL		Single commands	
92	DINT		Zero offset	
96	DINT		Set actual value	
100	DINT		Set actual value on the fly	
104	16 x BOOL		Digital inputs/outputs	
106	STRUCT	MDI block structure	MDI block on the fly	
126	BYTE		Program selection	Program number
127	BYTE		Program selection	Block number

[1] A variable in the S7 protocol is addressed by the DB No. and, depending on data format, by the DBB, DBW and DBD No. (offset in DB), as well.

Table 8-4 Parameters/data of the DB–SS (DB No. 1000), continued

Offset[1]	Variable type	Value	Significance of the variables	Comment
128	2 x BYTE		Program selection	Direction, free
130	4 x BYTE		Request application data	Application data 1-4
134	BYTE		Teach In	Prog. no.
135	BYTE		Teach In	Block number
136	DINT		Reference coordinate	
140	4 x DINT		Free	
156	DINT		Actual position	Basic operating data
160	DINT		Actual velocity	Basic operating data
164	DINT		Residual travel	Basic operating data
168	DINT		Target position	Basic operating data
172	DINT		Sum of current tool offset	Basic operating data
176	DINT		Rotational speed, rotary axis	Basic operating data
180	DINT		Free	
184	DINT		Free	
188	STRUCT	NC block struct.	Active NC block	
208	STRUCT	NC block struct.	Next NC block	
228	DINT		Code application 1	Application data
232	DINT		Code application 2	Application data
236	DINT		Code application 3	Application data
240	DINT		Code application 4	Application data
244	DINT		Actual position on leading edge	Length measurement/ inprocess measurement
248	DINT		Actual position on trailing edge	Length measurement
252	DINT		Length measurement value	Length measurement
256	DINT		Act. value at external block change	
260	DINT		DAC value/frequency	Servicing data
264	DINT		Encoder actual value	Servicing data
268	DINT		Error pulses	Servicing data
272	DINT		K_V-factor	Servicing data
276	DINT		Following error	Servicing data
280	DINT		Following error limit	Servicing data

[1] A variable in the S7 protocol is addressed by the DB No. and, depending on data format, by the DBB, DBW and DBD No. (offset in DB), as well.

Table 8-4 Parameters/data of the DB−SS (DB No. 1000), continued

Off-set[1)]	Variable type	Value	Significance of the variables	Comment
284	DINT		s Overshoot/Switch readjustment in Reference Point Approach mode	Servicing data
288	DINT		Approach time Te (ms)/drive constant Ta (ms) in OL Control mode	Servicing data
292	8 x DINT		Free	
324	BYTE		Override	Additional operating data
325	BYTE		Part program no.	
326	BYTE		Part program block no.	Additional operating data
327	BYTE		No. of callup subroutine loops	Additional operating data
328	BYTE		G90/91 Active	Additional operat. data
329	BYTE		G60/64 Active	Additional operat. data
330	BYTE		G43/44 Active	Additional operat. data
331	BYTE		Active D No.	Additional operat. data
332 332.1 332.2 332.3	8 x BOOL		Status messages 1 • Speed limitation to limit value from MD • Limitation to ± 10 V • Limitation of min. acceleration or min. deceleration in effect	Additional operating data
333	8 x BOOL		Free	
334	2 x BYTE		Free	
336	4 x 8 x BOOL		Diagnostics, system-specific	
340	4 x BYTE		Diagnostics, channel-specific	Identifier
344	2 x 8 x BOOL		Diagnostics, channel-specific	Channel error
346	4 x 8 x BOOL		Free	
350	2 x BYTE		Error number DS 162	Operator/travel error
352	BYTE		Free	
353	BYTE		Free	
354	2 x BYTE		Error number DS 163	Data error
356	BYTE		Free	
357	BYTE		Free	
358	2 x BYTE		Error number DS 164	Operator control error
360	BYTE		Free	
361	BYTE		Free	
362	32 x BOOL		Process interrupt	

1) A variable in the S7 protocol is addressed by the DB No. and, depending on data format, by the DBB, DBW and DBD No. (offset in DB), as well.

The control and checkback signals in Table 8-4 can be the following signals:

Bit / Byte	7	6	5	4	3	2	1	0
Control signals:								
24					BFQ/ FSQ		TFB	
25	AF	SA	EFG	QMF	R+	R–	STP	ST
26	BA							
27	BP							
28	OVERR							
29								
Checkback signals:								
30	PARA			DF	BF/FS		TFGS	
31		PBR	T-L			WFG	BL	SFG
32	BAR							
33	PEH		FIWS	SRFG	FR+	FR–	ME	SYN
34	MNR							
35				AMF				

The following table describes the control and checkback signals in German and English.

Table 8-5 Control and checkback signals

German	English	Description
Control signals		
TFB	TEST_EN	Switch P bus interface to “Start-up”
BFQ/FSQ	OT_ERR_A	Acknowledge operator control and traversing errors
ST	START	Start
STP	STOP	Stop
R–	DIR_M	Negative direction
R+	DIR_P	Positive direction
QMF	ACK_MF	Acknowledgement for M function
EFG	READ_EN	Read enable
SA	SKIP_BLK	Skip block
AF	DRV_EN	Drive enable

Table 8-5 Control and checkback signals, continued

German	English	Description
BA	MODE_IN	**Mode** **Code** Jog 01 Control 02 Reference point approach 03 Incremental relative 04 MDI 06 Automatic 08 Automatic single block 09
BP	MODE_TYPE	**Operating parameters** **Code** Speed levels 1 and 2 Voltage levels 1 and 2 Incremental dimension selection 1...100, 254
OVERR	OVERRIDE	Override
Checkback signals		
TFGS	TST_STAT	Switching of P bus interface completed
BF/FS	OT_ERR	Operator control/traversing errors
DF	DATA_ERR	Data error
PARA	PARA	Channel initialized
SFG	ST_ENBLD	Start enable
BL	WORKING	Execution in progress
WFG	WAIT_EI	Waiting for external enable
T-L	DT_RUN	Dwell time running
PBR	PR_BACK	Reverse program scanning
BAR	MODE_OUT	Active mode
SYN	SYNC	Channel synchronized
ME	MSR_DONE	Measurement completed
FR–	GO_M	Negative travel
FR+	GO_P	Positive travel
SRFG	ST_SERVO	Servo enable status
FIWS	FVAL_DONE	On-the-fly setting of actual value completed
PEH	POS_RCD	Position reached. Stop.
MNR	NUM_MF	M function number
AMF	STR_MF	M function modification

Description of Functions

9

Chapter Overview

Overview

This chapter describes the functions of the FM 354.

You can activate these functions via the user program by calling the appropriate standard function blocks or technology functions.

9.1 Control and checkback signals

Overview

The **POS_CTRL** block transfers the control signals from the user DB to the module and transfers the checkback signals from the module to the user DB.

Bit Byte	7	6	5	4	3	2	1	0
Control signals:								
14					BFQ/ FSQ		TFB	
15	AF	SA	EFG	QMF	R+	R–	STP	ST
16	BA							
17	BP							
18	OVERR							
19								
Checkback signals:								
22	PARA			DF	BF/FS		TFGS	
23		PBR	T-L			WFG	BL	SFG
24	BAR							
25	PEH		FIWS	SRFG	FR+	FR–	ME	SYN
26	MNR							
27				AMF				
28	ACT_POS [1]							
29								
30								
31								

[1] With FM firmware version V3.7.6 in conjunction with the blocks of the “FM353_354” library.

9.1.1 Control signals

Overview

The axis is operated and controlled by means of control signals.

Table 9-1 describes the control signals and their functions.

Table 9-1 Control signals

Symbol		Name	Function
English	German		
TEST_EN	TFB	Sw./over P-bus interface	Interrupts communication with the user program, and switches over the P bus interface for operation with the start-up user interface.
OT_ERR_A	BFQ/ FSQ	Acknowl. operator/ travel error	... resets an error message. Before acknowledging the error, correct its cause.
START	ST	Start	... starts movement in "Automatic", "MDI" and "Reference-point approach" modes.
STOP	STP	Stop	... interrupts movement or processing of the program ... cancels reference point approach.
DIR_M	R–	Direction minus	... moves axis in negative direction. • In Jogging and Control modes, moves axis in negative direction (level-dependent). • Starts movement in negative direction in Incremental relative and Reference-point approach modes. • Specifies direction of movement for rotary axes in MDI and Automatic modes.
DIR_P	R+	Direction plus	... moves axis in positive direction. • In Jogging and Control modes, moves axis in positive direction (level-dependent). • Starts movement in positive direction in Incremental relative and Reference-point approach modes. • Specifies direction of movement for rotary axes in MDI and Automatic modes.
ACK_MF	QMF	Acknowledge M function	... only "acknowledge-driven" during M function output (see machine data list in Table 5-4, MD32). ... acknowledges receipt of M function. Program sequence can be continued.
READ_EN	EFG	Read-in enable	... prevents read-in (processing) of the next block ... has effect only in Automatic mode. The read-in enable is required in order to read in the next traversing block during program execution.
SKIP_BLK	SA	Skip block	... skips identified blocks in the program ... has effect only in Automatic mode.

Table 9-1 Control signals, continued

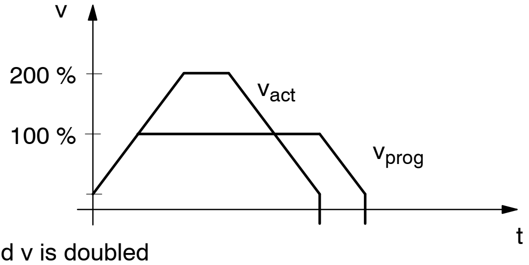
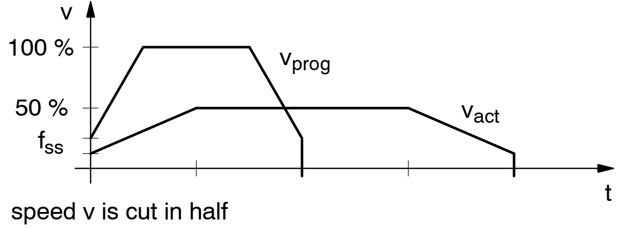
Symbol		Name	Function
English	German		
DRV_EN	AF	Drive enable	... enables movement. When the signal is reset, a **deceleration** of the movement takes place. If MD 37.15 = 0, program execution or the movement is aborted and the distance to go deleted. If MD 37.15 = 1, the following applies (machining resumed after emergency stop): • Rapid deceleration of the movement. • On axis standstill FR+ or FR− = 0; BL = 1 the drive remains activated and Servo Enable active • If an error occurs in this state (e.g. the user starts the movement without a Start Enable, etc.), the relevant error response is initiated, e.g. the distance to go is deleted; BL = 0 (a new path must be specified).
MODE_IN	BA	Operating mode	Operating mode (see Section 9.2) Code Jogging 01 Open-loop control 02 Reference point approach 03 Incremental relative 04 MDI 06 Automatic 08 Automatic single block 09
MODE_ TYPE	BP	Mode parameter	... selects speed levels in Jogging mode ... selects frequency levels in Control mode ... selects increment in Incremental relative mode (value 1 – 100 or 254).
OVERRIDE	OVERR	Override	... affects response of traversing movement. Range: 0-255% ... override has no effect in Control mode • Velocity override Range: 0-255% Speed adjusted by percentage **Example:** Override doubled from 100% to 200%  – speed v is doubled – acceleration and deceleration values are not affected $v_{act} = \frac{v_{prog} \cdot Override}{100}$ The positioning time is not cut in half.

Table 9-1 Control signals, continued

Symbol		Name	Function
English	German		
OVERRIDE	OVERR	Override	• Time override If you parameterize the “time override” function in MD37, there are two ranges: – range 100-255%: speed override operates as described above – range 0-100%: time override operative Speed, acceleration and deceleration are changed in such a way that the time necessary for the traversing movement is directly correlated with the override value. **Example:** Cut override in half, from 100% to 50%  – speed v is cut in half – acceleration and deceleration are quartered – Cut override in Start/Stop frequency $v_{act} = \frac{v_{prog} \cdot \text{Override}}{100}$ $a_{act} = \frac{a \cdot \text{Override}^2}{100^2}$ $t_{act} = \frac{t \cdot 100}{\text{Override}}$ Positioning time is doubled. Taking the override into account as a time override presupposes the following additional condition: If a traversing movement consists of multiple positioning blocks with block change on-the-fly (the axis does not stop between blocks), changing the override value affects only the speed. Acceleration and deceleration are additionally affected only after the axis comes to a stop (e.g., reversal of direction). **Note:** Time override has effect only in the MDI and Automatic modes.

Note

For further functions, **settings and commands** concerning open-loop control, see Section 9.3.2 and Section 9.3.3.

9.1.2 Checkback signals

Overview

The checkback signals indicate the processing status of the axis and report it to the user program.

Table 9-2 describes the checkback signals and their functions.

Table 9-2 Checkback signals

Symbol		Significance	Function
English	German		
TST_STAT	TFGS	Sw./over P bus interface complete	Communication with the user program is not possible, since the P bus interface has been switched over for operation with the start-up tool.
OT_ERR	BF/FS	Operator/ travel errors	... signaled to the user if an operator-control error or travel error is pending (e.g., unallowed control signal has been set, (R+) and (R–) set simultaneously) An error message causes the movement to be canceled. See Chapter 11
DATA_ERR	DF	Data error	... is reported to the user when a data error occurs. see Chapter 11
PARA	PARA	Parameterize	... module parameterized. All machine data applicable for control of an axis are present on the module.
ST_ENBLD	SFG	Start enable	... signals that the FM 354 is ready for positioning and output. • “Start enable” is set: – if no static stop or error is pending and the drive enable is pending – if the mode setting and mode checkback match (after mode change) – if no axis functions (including M output, dwell time) are active, or after functions have been completed – for further processing of a function interrupted with unprogrammed stop – in Automatic mode, after program has been selected (one program active) and after M0, M2, M30, or at end of block with Automatic single-block • “Start enable” is deleted: – if a function has been started and is active, or – if a start condition is active (stat.) – if there is an error and an unprogrammed stop • Without Enable Start, none of the functions that can be operated with Travel Plus, Travel Minus and Start can be executed.

Table 9-2 Checkback signals, continued

Symbol		Significance	Function
English	German		
WORKING	BL	Processing in progress	... indicates that a function has been started with Start or Travel Plus/Minus, and is active. • "Processing in progress" is set with: – "Jogging", "Control" mode during the movement up to standstill after cancelation of R+, R– – Reference-point approach mode, during approach until reference point is reached – MDI, Incremental relative mode, during the positioning process or while functions of the MDI block are being processed – Automatic mode, during processing of a traversing program until the end of the program. • "Processing in progress" is deleted: – by errors and restarts – by mode changes. – after axis standstill
WAIT_EI	WFG	Wait for external enable	... takes effect only if a digital input has been parameterized by means of MD34 (see Section 9.8.1). Set: if the enable input has not yet been set or has been reset when a movement has been activated.
DT_RUN	T-L	Dwell time in progress	... only active in Automatic and MDI mode. As soon as a traversing block with a dwell time has been processed, (T-L) is output during the programmed time period.
PR_BACK	PBR	Program processing in reverse	... is set after a Start in Automatic mode if a program is being processed in reverse.
MODE_OUT	BAR	Active mode	The selected mode is not fed back until it is internally active. For a mode change, for example, a movement must be stopped before another mode can become active (does not apply to switching between Automatic and Automatic single-block modes).
SYNC	SYN	Synchronism	... module is synchronized (see Section 9.8) Required for axis motion in modes: • Incremental Relative • MDI • Automatic
MSR_DONE	ME	End measurement	... signals an executed measurement (see Section 9.3.10)

Table 9-2 Checkback signals, continued

Symbol		Significance	Function
English	German		
GO_P GO_M	FR+ FR–	Travel plus Go_minus	... means the axis is traveling in the direction of increasing actual values or in the direction of frequency output “+” in Control mode ... means the axis is traveling in the direction of decreasing actual values or in the direction of frequency output “–” in Control mode. • As soon as an active traversing movement is pending, the messages (FR+) or (FR–) are output depending on the traversing direction. They can only be pending as alternatives. • “Travel Plus” or “Travel Minus” is actuated at the start of the acceleration phase and remains active until the axis comes to a standstill or the POS_ROD target area has been reached.
ST_SERVO	SRFG	Servo enable status	• Feedback of servo enable status after single function has been activated • See also Section 9.3.2 ”Servo enable” • See also Section 11.1, error reaction ”Everything OFF”
FVAL_ DONE	FIWS	Set actual value on-the-fly complete	... set Actual value on-the-fly is executed. The signal is reset when “Set actual value on-the-fly” is activated (see Section 9.3.6).
NUM_MF	MNR	M function number	M command 0 – 99
STR_MF	AMF	Change M function	... is indicated simultaneously with the M function number. • If M functions are programmed in a traversing block, their output is signaled by setting “Change M function.” • “Change M function” remains pending until: – the specified time has expired, for time-controlled M functions – the user has acknowledged, for acknowledgment-controlled M functions.

Table 9-2 Checkback signals, continued

Symbol		Significance	Function
English	**German**		
POS_RCD	PEH	Position reached, Stop (“PEH”)	• When the preset target position is reached correctly, (PEH) is actuated, and remains in effect until the next axis movement. • (PEH) is actuated only in the following modes and cases: – Reference-point approach: If the reference point has been reached in full (including reference-point shift). – MDI, Incremental relative: If the preset position has been reached. – Automatic: If a traversing block has been positioned in full and the axis remains motionless until the next traversing movement. • It is not set if no synchronization is available yet.
ACT_POS	ACT_POS	Actual position	The firmware version V3.7.6 of the FM 354 positioning module in conjunction with the blocks of the “FM353_354” block library provides direct access to the actual value. This is stored in the checkback signals during the module cycle (2 ms) and read by the POS_CTRL block. In addition, as previously, the actual value is included as a DINT variable in the basic operating data read request.

9.1.3 General handling information

Overview

Before data/settings can be transferred to the FM 354, an operating mode must be active (e.g. “Jogging” mode = 1 and MODE = 1). That means that communication with the FM 354 has been initiated and the FM 354 has access to valid machine data.

Operating modes (codes)	Relevant control signals	Relevant checkback signals	Required data/ settings
Jogging (01)	[R+], [R–], [STP], [AF], [OVERR], [BP] = 1 or 2	[BL], [SFG], [FR+], [FR–], [SYN], [WFG]	Speed levels 1, 2 (User DB, DBX38.0) Servo Enable (User DB, DBX34.0)
Control (02)	[R+], [R–], [STP], [AF], [BP] = 1 or 2	[BL], [SFG], [FR+], [FR–], [WFG]	Frequency levels 1, 2 (user DB, DBX38.1)
Reference point approach (03)	[R+], [R–], [ST], [STP], [AF], [OVERR]	[BL], [SFG], [FR+], [FR–], [WFG], [SYN], [PEH]	Servo Enable (User DB, DBX34.0)
Incremental relative (04)	[R+], [R–], [STP], [AF], [OVERR], [BP] = 1...100 for increment table or 254	[BL], [SFG], [FR+], [FR–], [WFG], [SYN], [PEH]	Speed levels 1, 2 (user DB, DBX38.0) Servo Enable (user DB, DBX34.0) Setpoint for incremental dimension (user DB, DBX34.2) (only if BP = 254; if BP is = 1 to 100, the relevant incremental dimensions must be parameterized)
MDI (06)	[ST], [STP], [AF], [QMF], [OVERR]	[BL], [SFG], [FR+], [FR–], [WFG], [SYN], [PEH], [AMF], [MNR], [T-L]	MDI block (user DB, DBX38.3) Servo Enable (user DB, DBX34.0)
Automatic (08) Automatic single block (09)	[ST], [SA], [EFG], [STP], [AF], [QMF], [OVERR]	[BL], [SFG], [FR+], [FR–], [WFG], [SYN], [PEH], [AMF], [T-L], [PBR], [MNR]	Program selection (user DB, DBX 39.5) (assuming that the relevant traversing program was parameterized) Servo Enable (user DB, DBX34.0)

Error condition:

- Reported via BF/FS – acknowledgement with BFQ/FSQ
- Reported via DF – acknowledgement with next correct data transfer
- Reported via diagnostic interrupt – acknowledgement with “Restart” (user DB, DBX37.5)

Hints to the user

Here are a few hints for starting a movement and about the response of the FM 354 to a change of the status of the S7-300 CPU:

It is assumed that the FM 354 has been parameterized correctly.

- First a mode must be set. The servo enable must subsequently be set in order to prevent the axis from “running away,”.
- Before starting a movement in any mode, the relevant data (e.g. speed levels) must be transferred and the override must be > 0.
- It is only possible to start the movement when the start enable is set and the enable input is set (if parameterized).

  Enable Start is set if:

  - No error occurred
  - Mode is active
  - No Stop is called
  - Drive enable is set
- A static Stop signal prevents all movements or block processing.
- Response of the FM 354 to transition of the S7-300 CPU from RUN to STOP state:
  - As described for Restart (see Section 9.3.3)
  - The digital outputs are switched off
  - Interface to the user program is switched off
- Response of the FM 354 to transition of the S7-300 CPU from STOP to RUN state:

  A cold restart of the module is executed.

Module control

The following table lists the control signals used to start a movement.

Prerequisite: Drive enable [AF] = 1, Stop [STP] = 0,
Start enable [SFG] = 1

Mode (BA)	Parameters	Command / Signal state	Activation of movement

Jogging (mode = 01)	Velocity level
	BP = 1 = level 1
	BP = 2 = level 2
		R+, R– / Level	R+ or R– with “Level” = 1
			(R+ and R– simultaneously → error)

Control (mode = 02)	Frequency level
	BP = 1 = level 1
	BP = 2 = level 2
		R+, R– / Level	R+ or R– with “Level” = 1
			(R+ and R– simultaneously → error)

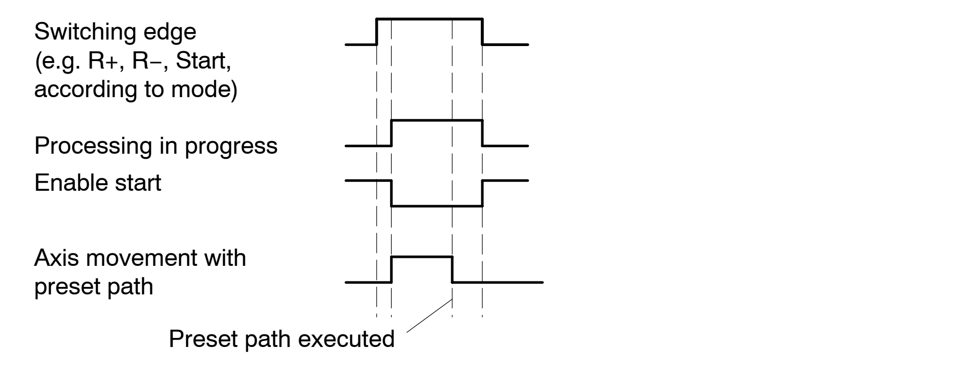
Reference point approach (mode = 03)	–	Start, R+, R– / Edge	Direction as in MD R+ or R– = 0/1 or Start = 0/1 (speed as in MD)

Incremental relative (mode = 04)	BP = 1...100
	BP = 254
		R+, R– / Edge	R+ = 0/1 or R = 0/1
			(speed level 1)

MDI (mode = 06)	–	Start / Edge	Start = 0/1 (R+, R– relevant only for rotary axis with absolute measure specified for direction selection)
Automatic (mode = 08)	–	Start / Edge	Start = 0/1 (according to program presetting)
Automatic single block (mode = 09)	–	Start / Edge	Start = 0/1

Stat. pending start condition

“Processing in progress” remains active after the end of machining and there is no start enable as long as the start condition is not reset.


Switching edge
(e.g. R+, R–, Start, according to mode)
Processing in progress
Enable start
Axis movement with preset path
Preset path executed

The following table lists the control signals used to interrupt/terminate a movement.

Mode (BA)	Interrupt movement	Continue movement	Interrupt/ end movement, Stop
Jogging (mode = 01)	Stop = 1 or Enable input[1] = 0	Stop = 0 or Enable input[1] = 1	R+ or R– with “Level” = 0 or mode change Drive enable = 0[2]
Control (mode = 02)	Stop = 1 or Enable input[1] = 0	Stop = 0 or Enable input[1] = 1	R+ or R– with “Level” = 0 or mode change Drive enable = 0[2]
Reference point approach (mode = 03)	–	–	Stop = 0/1 or ref. received or mode change or enable input[1] = 0 Drive enable = 0[2]
Incremental relative (mode = 04)	Stop = 1 or Enable input[1] = 0	Stop = 0 or Enable input[1] = 1, with R+ or R–	Position reached or mode change Drive enable = 0[2]
MDI (mode = 06)	Stop = 1 or Enable input[1] = 0	Stop = 0 or Enable input[1] = 1, with Start = 0/1	Position reached or “block” processed or mode change Drive enable = 0[2]
Automatic (mode = 08)	Stop = 1 or Enable input[1] = 0	Stop = 0 or Enable input[1] = 1, with Start = 0/1	Program end or mode change New program selected after stop Drive enable = 0[2]
Automatic single block (mode = 09)	Stop = 1 or Enable input[1] = 0	Stop = 0 or Enable input[1] = 1, with Start = 0/1	Program end or mode change New program selected after stop Drive enable = 0[2]

[1] **Prerequisite:** Digital input defined in MD34; see Section 9.8.1

[2] if MD37.15 not defined, see Table 9-1 Control signal [AF]

9.2 Operating modes

Overview

The following operating modes are implemented on the FM 354:

• Jogging (ER)	Code 01
• Open-loop control (STE)	Code 02
• Reference point approach (REF)	Code 03
• Incremental relative (SMR)	Code 04
• MDI (Manual Data Input)	Code 06
• Automatic (A)	Code 08
• Automatic single block (AE)	Code 09

Selecting the mode

By calling POS_CTRL, the mode (coding) entered by the user program in the user data block is called to transfer the operating mode (code) to the FM 354.

The axis is controlled by enabling and disabling appropriate control signals.

Checkback signal for mode

When the specification is allowed, the FM 354 feeds back the specified mode to the user program. If this checkback mode matches the specified one, the mode is active.

Changing modes

Changing modes triggers an internal stop.

If a mode change is attempted while a traversing movement is in progress, the modes are not switched until the axis comes to a stop. The mode checkback is performed after the movement in the old mode is completed.

This does not apply to changes between Automatic and Automatic Single-Block mode.

9.2.1 Jogging

Overview

In Jogging mode, axis traversing movements are specified by way of the direction keys (R+ or R–) and by speed.

Velocity

Before the axis can be moved, speeds (velocities) 1 and 2 must first be transferred to the FM 354 (user DB, DBX38.0).

You can choose between two mutually independent velocities (level 1 and level 2) with the mode parameter (BP).

The velocity can also be controlled using the override, and can be changed during the movement.

Name	Lower input limit	Upper input limit	Unit
Speed	10	500 000 000	MSR/min

MSR stands for measurement system raster (see Section 5.3.1)

Handling by the user

The table below gives you an overview of how to handle this mode.

Triggering of movement, Direction (R)	level selection	Speed
R+ or R– “level-controlled”	BP = 1	Value for speed level 1
	BP = 2	Value for speed level 2

Note

Please see also Section 9.1.3!

Control actions

Preconditions:

- The FM 354 has been initialized.
- The mode has been selected and acknowledged.
- Drive Enable [AF] = 1 (control signal in user DB, DBX15.7)
- Stop [STP] = 0 (control signal in user DB, DBX15.1)
- Servo Enable (RF) = 1 (user DB, DBX34.0)
- Speed levels have been transferred.

Table 9-3 Control actions for “Jogging” mode (examples)

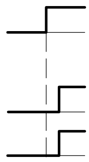
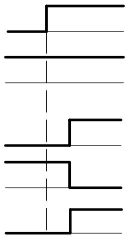
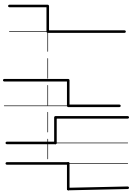
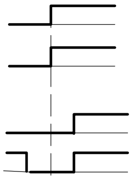
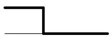
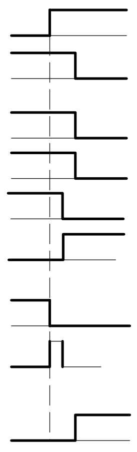
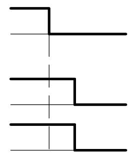
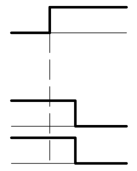
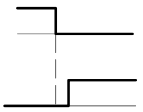
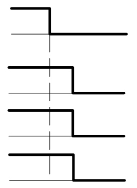
Signal name	Level	Explanation
Control action 1, enable “Jogging” mode		
Control signal: Mode [BA] **Checkback signals:** Active mode [BAR] Start enable [SFG]		The user initiates a [BA] command. The module returns [BAR] and [SFG].
Control action 2, move axis – positive direction		
Control signals: Direction plus [R+] Drive enable [AF] **Checkback signals:** Travel plus [FR+] Start enable [SFG] Processing in progress [BL]		When [SFG] and [AF] are active, [R+] is actuated. The axis cancels the [SFG] and outputs messages [BL] and [FR+]
Control action 3, deactivate axis – positive direction		
Control signal: Direction plus [R+] **Checkback signals:** Travel plus [FR+] Start enable [SFG] Processing in progress [BL]		[R+] is canceled When the axis has come to a standstill, the [BL] and [FR+] messages are canceled and [SFG] is activated. Before the axis comes to a standstill, it is possible to define a new direction “through start”.
Control action 4, move axis – negative direction		
Control signals: Direction minus [R–] Velocity level [BP] **Checkback signals:** Travel minus [FR–] Processing in progress [BL]		[R–] is actuated in combination with velocity level 2. The axis travels at velocity level 2, and returns [BL] and [FR–]. The [SFG] signal is canceled.
Control action 5, switch over set-up velocity		
Control signal: Velocity level [BP]		A switchover from [level 2 to level 1] causes a dynamic transition between velocity levels 1 and 2.

Table 9-3 Control actions for “Jogging” mode (examples), continued

Signal name	Level	Explanation
Control action 6, ambiguous direction command (special situation)		
Control signals: Direction plus [R+] Direction minus [R−] **Checkback signals:** Travel minus [FR−] Processing in progr. [BL] Start enable [SFG] Operator/travel error [BF/ FS] **Control signals:** Direction minus [R+] Error acknowledgement [BFQ/FSQ] **Checkback signal:** Start enable [SFG]		[R+] is actuated while the axis is traversing with [R−]. The ambiguous direction command causes the axis to stop and [BF/FS] to be output. [FR−] and [BL] are reset. Only when [R+] is canceled and the error is acknowledged [BFQ/FSQ] is [SFG] actuated again and a new direction command can be initiated.
Control action 7, cancel drive enable (special situation)		
Control signal: Drive enable [AF] **Checkback signals:** Travel minus [FR−] Processing in progr. [BL]		[AF] is deactivated during the traversing movement. The axis is stopped immediately. [FR−] and [BL] are canceled.
Control action 8, reset during axis motion (special situation)		
Single command “Restart “, (DBX37.5) **Checkback signals:** Travel plus [FR+] Processing in progr. [BL]		Restart is defined during the traversing movement. The axis is stopped immediately. [FR+] and [BL] are cleared.
Control action 9, change direction		
Control signal: Direction plus [R+] **Checkback signal:** Start enable [SFG]		Only when [R+] is canceled is [SFG] reactivated.
Control action 10, change mode		
Control signal: Mode [BA] **Checkback signal:** Active mode [BAR] Travel plus [FR+] Processing in progress [BL]		A new [BA] 1 is preselected during the traversing movement. The axis is stopped. [FR+] and [BL] are deleted.

9.2.2 Open-loop control

Overview

In the “Control” mode, a frequency with selectable magnitude is specified and then used to perform a controlled movement. The direction of movement is determined by way of direction keys (R+ or R–).

The actual value of the axis is updated at the same time.

Note

A closed-loop controller, which may have been activated by a servo enable, is interrupted while the frequency is being output. After the Jogging signals R+ or R– have died off, servo control is referred to the new actual value, and reinstated after the axis comes to a stop, if the servo controller enable is still active when the axis stops.

Frequency values

The frequency is specified in the user DB, DBX38.1.

You can choose between two mutually independent frequency levels (level 1 and level 2) with the mode parameter (BP).

Name	Lower input limit	Upper input limit	Unit
Frequency	0	200 000	Hz

The values of the frequency levels can be altered during movement.

Handling by the user

The table below gives you an overview of how to handle this mode.

Triggering of movement, Direction (R)	Level selection	Speed
R+ or R– “level-controlled”	BP = 1	Value for frequency level 1
	BP = 2	Value for frequency level 2

Note

Please see also Section 9.1.3!

Control actions

The control and checkback signals are handled in the same way as in “Jogging” mode.

9.2.3 Reference point approach

Overview

In Reference-point approach mode, the direction keys (R+ or R–) or Start are used to position the axis to a point (reference-point coordinate MD16) specified in the machine data.

The axis is thus synchronized (see Section 9.6.3).

The override is set at 100% for the reducing speed.

An active zero offset or Set actual value is reset.

Machine data

The following table lists the machine data that is of significance for reference-point approach:

MD	Designation	Value/Meaning	Comments/ Unit
16	Reference-point coordinate	–1,000,000,000 – +1,000,000,000	[MSR]
18	Type of reference-point approach	0 = direction +, current-sourcing pattern zero/ zero pulse external, to right of reference point switch 1 = direction +, current-sourcing pattern zero/ zero pulse external, to left of reference point switch 2 = direction – , current-sourcing pattern zero/ zero pulse external, to right of reference point switch 3 = direction – , current-sourcing pattern zero/ zero pulse external, to left of reference point switch 4 = direction +, reference point switch center 5 = direction – , reference point switch center 8 = direction + , reference point switch edge 9 = direction – , reference point switch edge	
27	Reference-point shift	–1,000,000,000 – +1,000,000,000	[MSR]
28	Referencing speed	10 – v_{max} (MD23)	[MSR/min]
29	Reducing speed	10 – x (see Machine data list, Table 5-4)	[MSR/min]
34	Inputs	5 = reference point switch for reference point approach 6 = reversing switch for reference point approach	Assigned depending on input

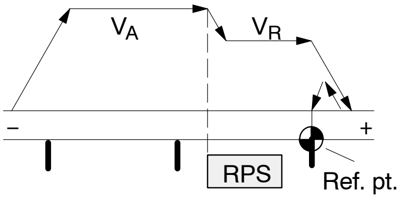
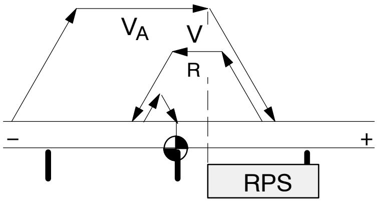
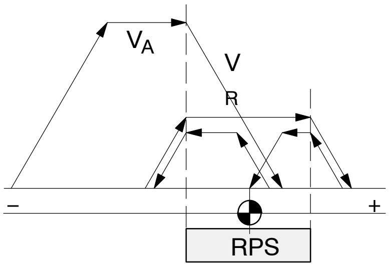
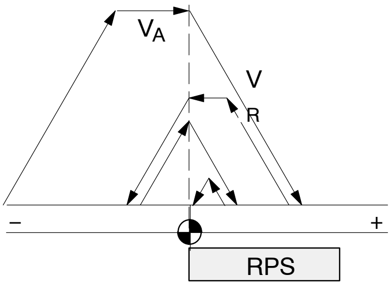
MSR stands for measurement system raster (see Section 5.3.1)

Handling by the user

- with connected reference-point switch (RPS).
- without connected reference-point switch (RPS).

With reference point switch (RPS)

It is necessary to connect the reference point switch (RPS) to a digital input and parameterize it in MD34.

Triggering of movement, direction for synchronization (R)	**Type of reference-point approach**	**Sequence of motions (reference point offset = 0)** **V_A – referencing velocity** **V_R – reducing velocity**
R+ (“edge-controlled”) or Start	1st situation Current-sourcing pattern zero or zero pulse external, to right of RPS	
	2nd situation Current-sourcing pattern zero or zero pulse external, to left of RPS	
	3rd situation RPS centered (no zero pulse necessary)	
	4th situation RPS edge (no zero pulse necessary)	
R– (“edge-controlled”) or Start	1st situation see above	equals R+ 2nd situation mirrored
	2nd situation see above	equals R+ 1st situation mirrored
	3rd situation RPS centered (no zero pulse necessary)	equals R+ 3rd situation mirrored
	4th situation RPS centered (no zero pulse necessary)	equals R+ 4th situation mirrored

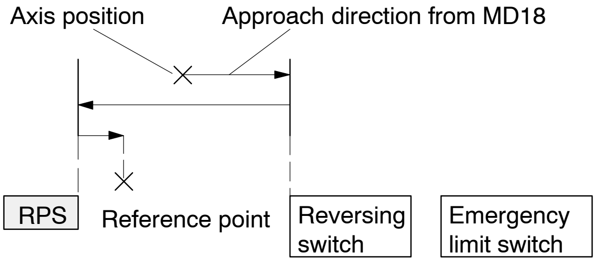
When crossing the RPS, a signal length of $\Delta t \geq 2 \cdot \mathrm{FM}$ cycle must be assured!

Using a reversing switch

If it is possible for the axis to be “behind” the reference point switch when you start reference point approach, a reversing switch can be installed at the end of the axis in the approach direction, in order to reverse the axis in the direction of the reference point switch.

On axis movements with referencing feed, a signal length of $\Delta t \geq 2 \cdot \mathrm{FM}$ cycle must be assured for the reversing switch!

Example



The value for the reference-point shift (MD27) is traveled after the synchronization point is reached.

Without reference-point switch (RPS)

The following table describes how a reference can be recorded without a reference-point switch.

Recording of synchronization	Sequence of movements
R+, R– or Start	1. Instantaneous position is defined as reference point (reference-point coordinate). 2. Value for reference-point shift is traveled.

Note

Please see also Section 9.1.3!

Control actions

Preconditions:

- The FM 354 has been initialized.
- The mode has been selected and acknowledged.
- Drive Enable [AF] is = 1 (control signal in the user DB, DBX15.7)
- Stop [STP] = 0 (control signal in the user DB, DBX15.1)
- Servo Enable (RF) is = 1 (user DB, DBX34.0)

Table 9-4 Control actions for “Reference point approach” mode (examples)

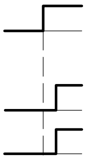
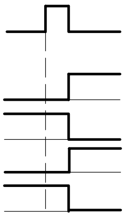
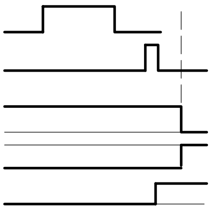
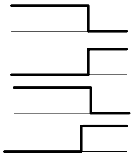
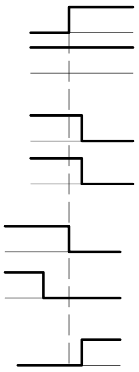
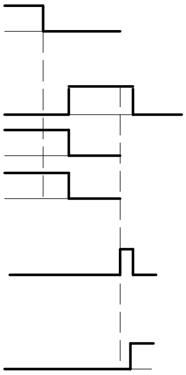
Signal name	Level	Explanation
Control action 1, enable “Reference point approach” mode		
Control signal: Mode [BA] **Checkback signals:** Active mode [BAR] Start enable [SFG]		The user initiates a [BA] command. The module returns [BAR] and [SFG].
Control action 2, move axis – positive direction		
Control signal: Direction plus [R+] **Checkback signals:** Travel plus [FR+] Start enable [SFG] Processing in progr. [BL] Synchronization [SYN]		When [SFG] is active, [R+] or [Start] are actuated, for example. The axis cancels [SFG], outputs the [BL] and [FR+] messages and travels here in the positive direction (defined in MD). An existing synchronization is reset.
Control action 3, reference point switch (RPS) reached		
RPS Encoder zero marker **Checkback signals:** Travel plus [FR+] Travel minus [FR–] Synchronized [SYN]		When the RPS is reached, the velocity is reduced. The encoder is synchronized when the zero marker is detected. The axis is positioned by traversing through the reference point offset to the reference point (the direction is reversed if necessary).
Control action 4, approach reference point		
Checkback signals: Travel minus [FR–] Position reached, stop [PEH] Processing in progress [BL] Start enable [SFG]		When reference point is reached. [FR–] is removed. [PEH] is set. [BL] is also removed. [SFG] is set.

Table 9-4 Control actions for “Reference point approach” mode (examples), continued

Signal name	Level	Explanation
Control action 5, ambiguous direction command (special situation)		
Control signals:		
Direction plus [R+]		[R+] is defined although [R–] is active.
Direction minus [R–]		
Checkback signals:		
Travel minus [FR–]		The ambiguous direction command causes the axis to stop. [FR–] and [BL] are canceled, and an error is output.
Processing in progress [BL]		
Control signals:		
Direction plus [R+]		The [SFG] does not reappear until [R+] and [R–] have been canceled].
Direction minus [R–]		
Checkback signal:		
Start enable [SFG]		
Control action 6, cancel servo enable (special situation)		
Single function ”Servo Enable” (DBX34.0)		The “servo enable” is deactivated during the traversing movement.
Checkback signals:		
Oper./travel error [BF/FS]		
Travel minus [FR–]		The axis is stopped immediately and outputs an error. [FR–] and [BL] are canceled.
Processing in progr. [BL]		
Control signal:		
Acknowledge operator/ travel error [BFQ/FSQ]		When the error is acknowledged, the error message is canceled and the start enable is activated.
Checkback signals:		
Start enable [SFG]		

9.2.4 Incremental relative

Overview

In the Incremental Relative mode it is possible to execute single positionings over relative distances using user-definable increments.

The traversing movement is triggered with the direction keys (R+ and R–).

Defining the position

The options available for defining the increment with the mode parameter are:

- Via the user program by defining a position for incremental mode (user DB, DBX38.2)

  The setpoint for the incremental dimension must be entered in the AW-DB, DBD156, before the write order is initiated.

- Using the increment table; see Section 5.3.2

Speed level 1 (user DB, DBX38.0, see Section 9.2.1) is used as speed setpoint, and can be modified while the movement is in progress.

On-the-fly position changes (e.g. changing of the position setpoint during the course of a movement) is **not** possible.

Handling by the user

The table below gives you an overview of how to handle this mode.

Triggering of movement, Direction (R)	Increment selection	Position, Distance to be traveled
R+ or R–	BP = 254	in accordance with the setpoint for incremental dimension (user DB, DBX38.2)
	BP = 1...100	as in SM table (DB-SM)

Position setting

Name	Lower input limit	Upper input limit	Unit
Increment	0	1 000 000 000	MSR

MSR stands for measurement system raster (see Section 5.3.1)

Note the following when interrupting a movement with “Stop”:

- To continue movement in the same direction - the residual distance is processed with the appropriate direction key.
- To continue movement with ”delete distance to go” (user DB, DBX37.1), the remaining distance to go is deleted and the incremental dimension is again traveled (provided the incremental dimension was not changed).
- To position in the opposite direction - the residual distance is deleted automatically.

Note

Please see also Section 9.1.3!

Control actions

Prerequisites:

- The FM 354 has been initialized.
- The mode has been selected and acknowledged.
- The Drive Enable [AF] is = 1 (control signal in the user DB, DBX15.7)
- Stop [STP] = 0 (control signal in the user DB, DBX15.1)
- The Servo Enable (RF) is = 1 (user DB, DBX34.0)
- Speed levels have been transferred.
- The axis is synchronized.

Table 9-5 Control actions for “Incremental relative” mode (examples)

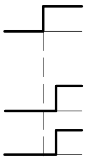
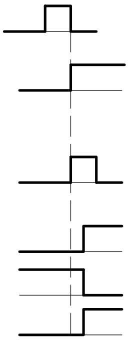
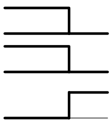
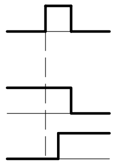
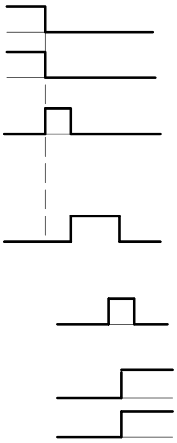
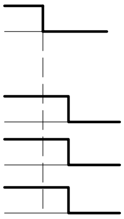
Signal name	Level	Explanation
Control action 1, enable “Incremental relative” mode		
Control signal: Mode [BA] **Checkback signals:** Active mode [BAR] Start enable [SFG]		The user initiates a [BA] command. The module returns [BAR] and [SFG].
Control action 2, define position		
Transfer incremental dimension (DBX38.2) Select increment (254) **Control signal:** Direction plus [R+] **Checkback signals:** e.g. Travel plus [FR+] Start enable [SFG] Processing in progress [BL]		When the incremental dimension has been transferred and selected, [R+] can be specified. The axis cancels the [SFG] and outputs messages [BL] and [FR+]
e.g. Travel plus [FR+] Processing in progress [BL] Position reached, stop [PEH]		When the defined position has been reached, the axis enables [PEH] and checkback signals [FR+] and [BL] are reset. In the ”Incremental mode relative”, [SFG] is already set when the interpolator is ready (setpoint coincidence reached). [BL] continues to be present if a new Start ([R+] or [R–] has been provided before [PEH] is reached.

Table 9-5 Control actions for “Incremental relative” mode (examples), continued

Signal name	Level	Explanation
Control action 3, stop during positioning		
Control signal: Stop [STP] **Checkback signals:** Travel minus [FR–] Start enable [SFG]		If Stop is enabled during positioning, the axis stops. [FR–] is reset, and [SFG] is activated. [PEH] is not output, since positioning is not complete. Before the axis comes to a standstill, it is possible to define a new direction “through start”.
Control action 4, error during traversing movement		
Checkback signals: Travel plus [FR+] Processing in progress [BL] **Control signal:** Acknowledge operator control/travel error [BFQ/FSQ] **Checkback signals:** Start enable [SFG] **Control signal:** Direction plus [R+] **Checkback signals:** Travel plus [FR+] Processing in progress [BL]		The axis moves. An error is output during the traversing movement. [FR+] and [BL] are canceled, and [BFQ/FSQ] is enabled. When the error has been acknowledged, the start enable is activated. The movement can be restarted with [R+]. [FR+] and [BL] are activated. [SFG] is canceled.
Control action 5, change mode		
Control signal: Mode [BA] **Checkback signals:** Active mode [BAR] Travel plus [FR+] Processing in progress [BL]		[BA] is deactivated during the traversing movement. The axis is stopped by way of the deceleration ramp. [FR+] and [BL] are reset.

9.2.5 Incremental relative

Overview

In the Incremental Relative mode it is possible to execute single positionings over relative distances using user-definable increments.

The traversing movement is triggered with the direction keys (R+ and R–).

9.2.6 MDI (Manual Data Input)

Overview

In the MDI mode it is possible to execute single positionings by way of traversing blocks. These traversing blocks are provided by the user program.

The MDI block and MD block on-the-fly have an identical block structure.

MDI block

The structure of the MDI block is identical to that of the traversing blocks (see Chapter 10 resp. Section 9.3.12) except that it has neither a program number nor a block number.

The user program passes the “MDI block” (user DB, DBX38.3) to the FM 354, and the block can then be executed. The block can be executed repeatedly, since it is stored internally. The feedrate is override-dependent.

The MDI block remains in effect until it is overwritten with a new MDI block. A new block can be transmitted while another block is being processed.

Table 9-6 MDI block (see Chapter 10 for command structure)

Name	Lower input limit	Upper input limit	Unit
Position X / Dwell time t	–1,000,000,000 2	+ 1,000,000,000 100,000	MSR from MD7 ms
Speed F	10	500 000 000	MSR from MD7/min
G function group 1	G04 G90 G91	Dwell time Absolute dimensions Incremental dimensions	–
G function group 2	G30 100% G31 10% to G39 90%	Override Acceleration/ Deceleration	–
M function group 1, 2, 3	M1 – 17 M19 – 96 M99 M97, 98 M2, M30	User functions Change signal programmed as digital output Not allowed	–

MSR stands for measurement system raster (see Section 5.3.1)

For rotary axes with absolute programming, the commands [R+], [R–] are defined as direction commands. They must be available before positioning starts.

MDI block on-the-fly

The MDI block currently being processed is canceled when the user program outputs an “MDI block on-the-fly” (user DB, DBX38.4).

Transfer of “MDI block on-the-fly” interrupts the active “MDI block”. The new block is executed immediately without “Start”.

The MDI block on-the-fly is **not** saved in the FM 354.

With software version 3.6 of the FM 354 and higher, the error ”Data cannot be accepted at the time of transfer” (Class 4/No.1) is signaled when ”MDI block flying” is provided and [BL] has already been reset or [PEH] =1.

Block structure

The following table shows the block structure of the MDI block.

X/t	Position/dwell time programmed (fills in value 1)
G1...G2	G function group 1...2
M1...M3	M function group 1...3
F	Speed programmed (fills in value 2)

Example[1]	**Byte**	**Data format**	**Bit**							
			7	6	5	4	3	2	1	0
0	0	Byte	0							
0	1	Byte	0							
Bits 0, 1, 4 set	2	8 bits	0	0	0	X/t	0	0	G2	G1
Bit 0 set	3	8 bits	0	0	0	0	M3	M2	M1	F
90	4	Byte	G-function1							
30	5	Byte	G-function2							
0	6	Byte	0							
0	7	Byte	0							
100,000	8	DINT	32-bit value 1							
5,000	12	DINT	32-bit value 2							
0	16	Byte	M-function 1							
0	17	Byte	M-function 2							
0	18	Byte	M-function 3							
0	19	Byte	0							

[1] Traversing block with absolute dimensions (G90), an end position of 100,000 DSG as per MD7, and a speed of 5,000 DSG/min.

Note:

When the assignment bit (byte 2 and byte 3) is not set, the associated values must be deleted.

Handling by the user

The table below gives you an overview of how to handle this mode.

Triggering of movement	Type of movement
Start	as defined by “MDI block” (user DB, DBX38.3)
“MDI block on-the-fly” transmitted to the FM 354	as defined by “MDI block on-the-fly” (user DB, DBX38.4)

Note

Please see also Section 9.1.3!

Control actions

Prerequisites:

- The FM 354 has been initialized.
- The mode has been selected and acknowledged.
- Drive Enable [AF] is = 1 (control signal in user DB, DBX15.7)
- Stop [STP] is = 0 (control signal in user DB, DBX15.1)
- Servo Enable (RF) is = 1 (user DB, DBX34.0)
- Axis is synchronized.

Table 9-7 Control actions for “MDI” mode (examples)

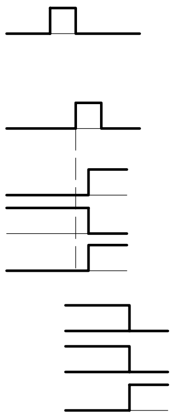
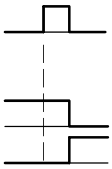
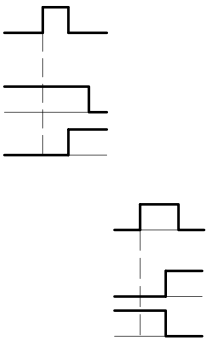
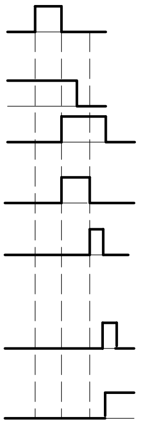
Signal name	Level	Explanation
Control action 1, define position		
Transfer MDI block (DBX38.3) **Control signal:** Start [ST] **Checkback signals:** e.g. Travel plus [FR+] Start enable [SFG] Processing in progr. [BL] e.g. Travel plus [FR+] Processing in progr. [BL] Position reached, stop [PEH]		When the MDI block has been transferred, [ST] can be initiated. The axis cancels the [SFG] and outputs messages [BL] and [FR+] When the defined position has been reached, the axis enables [PEH]; [SFG] and checkback signals [FR+] and [BL] are reset.
Control action 2, change position during positioning		
Transfer MDI block on-the-fly (DBX38.4) **Checkback signals:** Travel plus [FR+] Travel minus [FR–]		If a new “MDI block on-the-fly” is transferred during positioning, the current positioning operation is canceled immediately, and the new positioning operation is started on-the-fly. In this case, for example, this causes the direction to be changed from [FR+] to [FR–].

Table 9-7 Control actions for “MDI” mode (examples), continued

Signal name	Level	Explanation
Control action 3, stop during positioning with new start signal for resumed positioning		
Control signal: Stop [STP] **Checkback signals:** Travel minus [FR–] Start enable [SFG] **Control signal:** Start [ST] **Checkback signals:** Travel minus [FR–] Start enable [SFG]		If Stop is enabled during positioning, the axis stops. [FR–] is reset, and [SFG] is activated. [BL] remains active and [PEH] is not output, since positioning is not complete. If [ST] is initiated again, [FR–] and [SFG] are reset and positioning is completed. Before the axis comes to a standstill, it is possible to define a new direction “through start”.
Control action 4, stop during positioning with new start signal and new MDI block		
Control signal: Stop [STP] **Checkback signals:** Travel plus [FR+] Start enable [SFG] Transfer MDI block (DBX38.3) Transfer “delete residual path” (DBX37.1) **Control signal:** Start [ST] **Checkback signals:** Travel minus [FR–]		If Stop is enabled during positioning, the axis stops. [FR+] is reset, and [SFG] is activated. When a new MDI block has been transferred, [ST] is enabled again. “Delete residual path” is also enabled. The axis deletes the residual path of the old positioning operation, and starts executing the new traversing block. [FR–] is enabled, and [SFG] is reset. **Note:** If no new “MDI block” is transferred, execution of the current “MDI block” is repeated from the start. Without “delete residual path”, the interrupted positioning operation would be continued (see control action 3)

9.2.7 Automatic

Overview

In the Automatic mode (following-block mode), the FM 354 processes traversing programs autonomously. These programs are created with “Parameterize FM 354” (see Chapter 5, Section 5.3.4) and stored as a data block. The traversing programs contain information about movement sequences and outputs (see Chapter 10).

Program selection

Programs are selected (user DB, DBX39.5) by way of the user program, by specifying a program number and an optional block number, as well as the direction of machining. A program can be selected only when other programs have been interrupted or terminated or at the start of a program.

A selected program remains active until it is inactivated by selecting program number = 0, or overwritten by selecting another program.

If modifications are made to a preselected program, including the subprogram, preselection of the program is canceled. You must then select the program again. A modification can be made to a program when BL = 0 (start of program/end of program) and on Stop.

Triggering of movement	Select program		Type of movement (according to programmed blocks)
	Block No.	Processing direction	
Start	0	forward	Start at beginning of program, process by ascending block number
	0	reverse	Start at end of program, process by descending block number
	e.g. 30	forward	Block search forward to block No. 30, by ascending block number
	e.g. 30	reverse	Block search in reverse to block No. 30, by descending block number
Start with automatic block search forward		forward	1. Automatic block search forward to interruption point 2. Positioning to interruption point (if a movement was performed in another mode) 3. Process the interrupted block and continue the program
Start with automatic block search in reverse		reverse	1. Automatic block search in reverse to interruption point 2. Positioning to interruption point (if a movement was performed in another mode) 3. Process the interrupted block and continue the program

User DB allocation

Data format	Significance
Byte 0	Program number
Byte 1	Block number
Byte 2	Direction of machining: 0 = process forward 1 = process in reverse

Forward processing

The program processes the block numbers in ascending order.

Processing begins at Start, with the first block (specified block number = 0).

If processing is to begin at some other point of the traversing program, specify the desired block number.

Processing will take place by searching forward to this block, then processing forward until the program end command is recognized.

Backward processing

The program processes the block numbers in descending order.

Processing begins at Start, with the last block (specified block number = 0).

If processing is to begin at some other point of the traversing program, specify the desired block number. Processing will take place by searching back to this block, then processing in reverse until the program beginning is recognized.

Note

If reverse processing is to execute the same sequence of movements as the forward movement, the effects of the corresponding commands must be taken into account in the programming. For example:

- M outputs should be written separately in a block;
  note M output (MD32) and G60/G64.
- Note change between G60/G64 and G90/G91.
- Note start and end of tool offsets.
- M18 is not executed.
- M02 and M30 at the end of the program are not processed.

Block advance

The program is prepared up to the end point of the target block; the tool offset, M commands and dwell times are output; and the traversing movements are suppressed.

When processing traversing programs with a forward block search, there are a number of special cases:

- The external forward block search (G50) is not executed.
- Continuous travel with functions to set (G88, 89) or delete (G87) an actual value on-the-fly is not executed.
- The blocks under G50, G87, G88, G89 (in the processing direction) should contain a path in absolute coordinates.

Block return

Similar to block advance

Automatic block advance/return

Automatic block advance/return means that, after the interruption of an active automatic program (by an operating mode change), you can continue execution from this point of interruption in the appropriate direction of processing.

With forward block search, the interrupted program must previously have been going in the forward direction.

With block search in reverse, the interrupted program must previously have been going in the reverse direction.

The command for automatic forward or reverse block search is evaluated in the FM 354 at Start, and a forward or reverse block search to the interruption point is executed in the ”Jog” mode at the velocity of stage 1. Positioning to the interruption point takes place (if positioning has taken place previously in some other mode), and then the interrupted block is processed, including any required output.

Control actions

Prerequisites:

- The FM 354 has been inititalized.
- The mode has been selected and acknowledged.
- Drive Enable [AF] is = 1 (control signal in user DB, DBX15.7)
- Stop [STP] is = 0 (control signal in user DB, DBX15.1)
- Servo Enable (RF) is = 1 (user DB, DBX34.0)
- Axis is synchronized.

Table 9-8 Control actions for “Automatic” mode (examples)

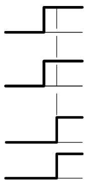
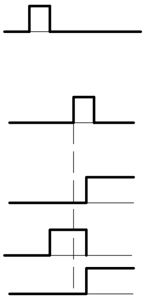
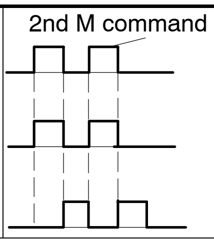
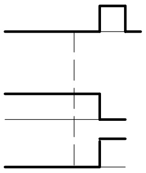
Signal name	Level	Explanation
Control action 1, Automatic/Automatic single block mode		
Control signals: Mode [BA] Read-in enable [EFG] **Checkback signals:** Active mode [BAR] Start enable [SFG]		The user initiates [BA] and [EFG]. The module returns [BAR] and [SFG].
Control action 2, positioning by program selection		
Program selection (DBX39.5) **Control signal:** Start [ST] **Checkback signals:** Travel plus [FR+] or Travel minus [FR–] Start enable [SFG] Processing in progress [BL]		When [SFG] appears, the program can be activated by [ST] when [EFG] is active. Processing commences, e.g. with a positioning operation. [FR+] or [FR–] and [BL] are activated. [SFG] is reset.
Control action 3, M function output		
Checkback signals: Change M function [AMF] M function number [MNR] **Control signal:** Acknowledge M function [QMF]		If M function output is acknowledgement-driven, for example, the user program can continue to process the [MNR] when [AMF] appears. M function output is complete. [QMF] acknowledges the M function, and [AMF] and [MNR] disappear.
Control action 4, M function output and positioning		
Control signal: Acknowledge M function [QMF] **Checkback signals:** Position reached, stop [PEH] Travel plus [FR+] or Travel minus [FR–]		Block with M output (same as control action 3) and position is started. The program is resumed on completion of the M function output. [FR+] and [FR–] are activated and [PEH] is reset.

Table 9-8 Control actions for “Automatic” mode (examples), continued

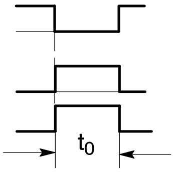
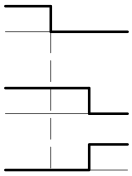
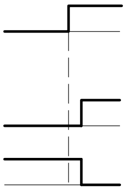
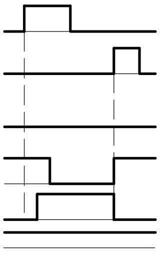
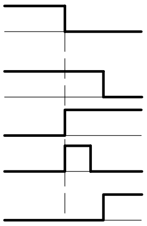
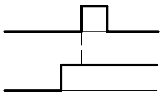
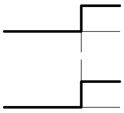
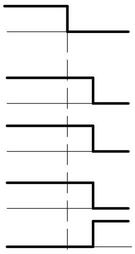
Signal name	Level	Explanation
Control action 5, traversing block with dwell		
Checkback signals: Travel plus [FR+] or Travel minus [FR–] Dwell time running [T-L] Position reached, stop [PEH]		During processing of a traversing block with dwell, the dwell time t_0 [T-L] and [PEH] are output.
Control action 6, cancelation of the read-in enable during program execution (special situationl)		
Control signal: Read-in enable [EFG] **Checkback signals:** Travel plus [FR+] or Travel minus [FR–] Position reached, stop [PEH]		If [EFG] is canceled during program execution, the current block is processed up to the end, and program execution is then suspended. [FR+] and [FR–] are reset. [PEH] is actuated.
Control action 7, resume program execution after read-in enable (special situation)		
Control signal: Read-in enable [EFG] **Checkback signals:** Travel plus [FR+] or Travel minus [FR–] Position reached, stop [PEH]		The program resumes on [EFG]. [FR+] and [FR–] are reset. [PEH] is reset.
Control action 8, stop during positioning with new start signal for resumed positioning (special situation)		
Control signals: Stop [STP] Start [ST] **Checkback signals:** Position reached, stop [PEH] Travel plus [FR+] Start enable [SFG] Processing in progress [BL]		Interrupt with Stop [FR+] is cleared when the axis comes to a standstill, and [SFG] is enabled (if Stop is not active). [PEH] remains cleared, since the defined position has not yet been reached. Start clears [SFG] and enables [FR+] again. [BL] remains enabled. Before the axis comes to a standstill, it is possible to define a new direction “through start”.
Control action 9, end of program reached		
Checkback signals: Travel plus [FR+] or Travel minus [FR–] Processing in progr. [BL] Position reached, stop [PEH] M function number [MNR] Start enable [SFG]		The end of the program is indicated by the enabling of [PEH], output of M2, M30 and resetting of [BL].

Table 9-8 Control actions for “Automatic” mode (examples), continued

Signal name	Level	Explanation
Control action 10, delete start signal and residual path (special situation)		
Control signal: Start [ST] Transfer “delete residual path” (DBX37.1)		If “delete residual path” is also preselected on [ST], the block interrupted by Stop is not executed up to the end, but the next block is started immediately.
Control action 11, positioning for rotary axis (special situation)		
Control signals: Direction plus [R+] or direction minus [R–] Start [ST]		If the axis is operated as a rotary axis, the FM always attempts to select the shortest path during positioning. This direction preference can be suppressed by specifying [R+] or [R–].
Control action 12, deactivate operating mode during program execution (special situation)		
Control signal: Mode [BA] **Checkback signals:** Old mode [BAR] Travel plus [FR+] or Travel minus [FR–] Processing in progress [BL] New mode [BAR]		If a new operating mode is selected during active program execution, the axis is stopped. [FR+] or [FR–] and [BL] are reset.

9.2.8 Automatic single block

Overview

Functions, same as “Automatic” mode

Whereas in “Automatic” mode the FM 354 automatically starts processing the next block after completing a given block, in “Automatic single-block” mode the axis waits for a new Start signal after processing each block that contains a traversing path, dwell time or M command (except for blocks with G50, G88 or G89).

You can change between Automatic single-block and Automatic mode at any time, without stopping the movement or interrupting the output.

9.3 System data

Overview

This chapter describes settings and functions that apply in multiple modes, and that are likewise necessary in order to control and operate the FM 354, and data of the FM available for checkback messages.

- Change parameters/data (Write request in user DB, DBX39.3), page 9-38
- Single functions (user DB, DBB34 and 35), page 9-42
- Single functions (user DB, DBB36 and 37), page 9-45
- Zero offset (Write request in the user DB, DBX39.1), page 9-47
- Set actual value (Write request in the user DB, DBX38.7), page 9-49
- Set actual value on-the-fly (Write request in the user DB, DBX39.0), page 9-50
- Request application data (Write request in the user DB, DBX39.6), page 9-51
- Teach in (Write request in the user DB, DBX39.7), page 9-52
- Set reference point (Write request in the user DB, DBX38.6), page 9-52
- Measured values, page 9-53
- Basic operating data (Read request in the user DB, DBX42.0), page 9-56
- Active NC block (Read request in the user DB, DBX42.1),
  next NC block (Read request in the user DB, DBX42.2), page 9-57
- Application data (Read request in the user DB, DBX43.6), page 9-58
- Actual value block change (Read request in the user DB, DBX42.3), page 9-58
- Servicing data (Read request in the user DB, DBX42.4), page 9-58
- Additional operating data (Read request in the user DB, DBX43.5), page 9-59
- Parameters/data (Read request in the user DB, DBX43.3), page 9-59

9.3.1 Change parameters/data (Write request in user DB, DBX39.3)

Overview

You can use this function to modify parameters and data in the FM 354 data blocks or to define parameters and data which can then be read out with a Read request (user DB, DBX43.3) (see Section 9.3.17).

Structure of Write request

The following table shows which parameters or data can be changed or read by setting the indicated codes.

Addr. in user DB	Data format	Symbol	Description				
196	Byte	DB type	type	1 = MD	2 = SM	3 = TO	4 = NC (traversing program)
197	Byte	Number	Info 1	MD No. (5 – 45)	SM No. (1 – 100)	TO No. (1 – 20)	Progr. No. (1 – 199)
198	Byte	Number	Info 2	Number of MDs, consecutive (1 – 5)	Number of SMs, consecutive (1 – 5)	0 = Tool offset complete 1 = Tool length only 2 = Wear value abs. only 3 = Wear value add. only	Block No. (1 – 255)
199	Byte	Job	1 = Read job parameters 2 = Write parameters 4 = Write parameters and save retentively[1)]				
200 – 219	depends on type	Data field	• MD: Data format of machine data see Table 5-4 or • SM: Data format of increments see Table 5-5 (DWORD) or • WK: Data format of tool offset see Table 5-6 (DINT) or • NC: Block format see Section 9.3.12 ”Active NC block”				

1) not with cyclic operation under 10 s

Example 1

The software limit switches (MD21, MD22) for the axis are to be set to the values 100 mm and 50,000 mm. These values are to remain in effect only until the unit is shut down.

DB type	= 1
Data number	= 21
Number of data	= 2
Job type	= 2
Data array	
Byte 200 –203	= 100,000 (MD21, data format DINT)
Byte 204 – 207	= 50,000,000 (MD22, data format DINT)
Bytes from 208 on	= 0

For activation of the machine data, see Section 9.3.3

Notes

Please note the following when changing the parameter data:

- **Machine data**

  Machine data can always be modified. Once you have modified the machine data, the machine data have to be reactivated (for single command, see Section 9.3.3).

- **Increments**

  Modifications can be made in all operating modes (even in “Incremental relative” mode) during movement. The modifications of the increments must always be complete before a new movement is started in “Incremental relative” mode. If this is not the case, the error message “incremental dimensions do not exist” is output Cl. 2/No. 13.

- **Tool offset data**

  Modifications can be made in all operating modes and during movement. If modifications are made during starting or at block transitions when the tool compensation is active (internal access to offset values), the error message “tool offset value does not exist” is output Cl.3/No.35.

- **Traversing programs**

  - Programs which are not selected can always be modified.
  - If modifications are made to a preselected program, including the subprogram, preselection of the program is canceled. You must then select the program again. A modification can be made to a program when BL = 0 (start of program/end of program) and on Stop.

Delete block: Specify the program no. and the block no. in the “data field”.
The other data/bits must not be assigned.

Insert block: The block number does not exist in the selected program.
The contents should be entered in accordance with
the “block format”.

Modify block: The block with the corresponding block number is
overwritten with the contents in accordance with “block format”.

Example 2

Procedure for changing the actual value and the velocity in block 10 of traversing program 1.

1. Submit job request to read the block. Enter the following values in the user DB:

   DB type = 4
   Number = 1
   Number of data = 10
   Job type = 1
   Data array = 0
2. Set Write request (user DB, DBX39.3).
3. Following termination of the Write request (one cycle later), read out the block by setting the Read request (user DB, DBX43.3).
4. Save the data read out from DBB 446 to 469 of the user DB in DBB196 to 219.
5. Enter the actual value in data array DBB208 to 211 in the user DB (data type DINT).
6. Enter the velocity in DBB212 to 215 of the user DB (data type DINT).
7. Enter a 4 in DBB199 of the user DB (request) (retentive storing of the block).
8. Set Write request (user DB, DBX39.3).

Retentive storage of parameter data

Please note the following when using the function “Write parameters with retentive storage” (byte 4, job type 4):

Retentive writing must only occur on demand (not cyclically)!

Retentive data are stored on FEPROM (maintenance-free, no battery required). This memory has a physical limit for the possible number of delete/reprogram cycles: minimum 10^5, typically 10^6. The possible number of delete/reprogram cycles can be multiplied, from the user’s viewpoint, by providing a larger retentive memory capacity (much larger than the parameter data memory) and organizing the memory accordingly.

$$\text{Number of delete/ reprogram cycles} = \frac{64\,000 \cdot 10^6 \text{ (typical)}}{\text{Block size (in bytes), in which parameter data are modified}}$$

Block sizes:

DB machine data	284 bytes
DB increments	468 bytes
DB tool offset data	308 bytes
DB traversing programs	108 + (20 x no. of traversing blocks) bytes

Example:

Assuming a service life of 10 years and 24-hour operation, a typical limit = 10^6.

Parameterization data	DB size	Number of possible delete/ reprogram cycles	Number of possible delete/ reprogram cycles per minute
MD	284 bytes	$237.04 \cdot 10^6$	46
Traversing programs (20 blocks)	508 bytes	$125.49 \cdot 10^6$	24

Note

SDB ≥ 1 000 (system data block, created for module replacement), contains parameter data which were valid at the time of start-up. If data/parameters are modified during operation and stored retentively on the FM, these data are not contained in SDB ≥ 1 000. These modifications are lost when the module is replaced, and should be traceable in the user program.

9.3.2 Single functions (user DB, DBB34 and 35)

Overview

You can use this function to transfer single settings to the FM 354 and activate the corresponding functions. These settings are:

- Length measurement
- Inprocess measurement
- Retrigger reference point
- Switch off enable input
- Follow-up mode
- Switch off software end position monitoring
- Switch off automatic drift compensation
- Servo enable
- Parking axis
- Simulation

Callup of single settings

The individual functions remain activated until they are reset.

Length measurement, inprocess measurement

Since both functions use the same digital input on the FM 354, only one function can be executed at a time. In double activations, both functions are switched to inactive. An error message is issued.

For function description, see Section 9.3.10

Retrigger reference point

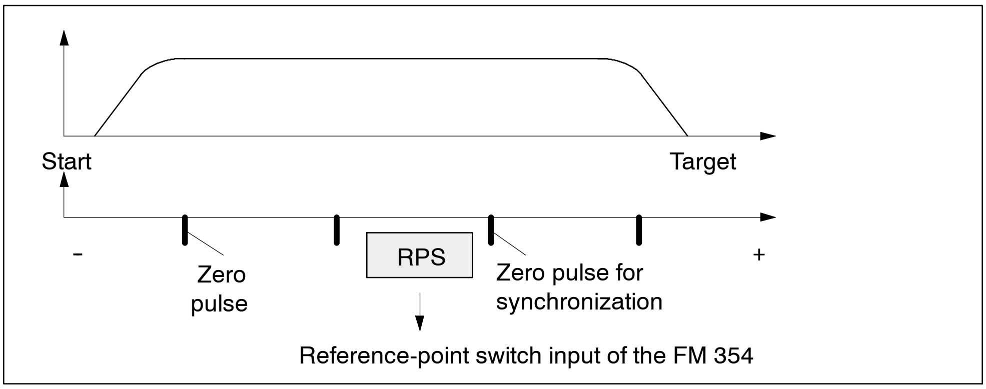
A precondition for retrigger reference point is that the axis has been synchronized by reference point approach.

With this setting, the axis resynchronizes upon overrunning the synchronization point when the direction of travel is the same as the direction of reference point approach. Regardless of the current speed, the reference point coordinates are allocated to the current actual position, taking into account any active shift.

Any resulting change in the actual value causes no internal change of the target.

When crossing the RPS, a signal length of $\Delta t \geq 2 \cdot$ FM-cycle must be assured!

When a Set Actual Value On-the-Fly is pending, activation of Retrigger Reference Point is interlocked.


Start
Target
–
+
Zero pulse
RPS
Zero pulse for synchronization
Reference-point switch input of the FM 354

Hint to the user:

You can use Retrigger Reference Point, for example, to compensate for slippage of the trolley in a high-bay warehouse during operation, without having to resynchronize the axis with the Reference-Point Approach mode. When retriggering in reference point approach with a zero pulse, be careful that the total slippage between the reference point switch and the stepper motor does not become so great that the synchronizing zero pulse migrates to an “adjacent” zero pulse!

Switch off enable input

With the “switch off enable input” function, you can switch off evaluation of the enable input (see Section 9.8.1).

Follow-up mode

The “ollow-up mode“ function is used to cancel closed-loop control of the axis.

- For external movement of the axis, the actual value is tracked.
- This setting can be switched on or off only if “Processing in progress” = 0. (does not apply to MD37.15 “Resume after Emergency Stop”)

Deactivate end position monitoring

You can use this function to deactivate monitoring of the software limit switches (see Section 9.9).

It can be switched on or off only if “Processing in progress” = 0.

Deactivate automatic drift compensation

This function can be used to switch off the automatic drift compensation.

Automatic drift compensation means:

The drift is balanced to zero by an automatic matching of the analog actuating signal.

- The setting can be switched on or off if the axis is not in motion.
- Automatic drift compensation has no effect:
  - in Control mode
  - in the Follow-up Mode setting
  - if there is no servo enable
  - in the absence of a controller ready signal (if parameterized)
  - if the axis is in motion.

Servo enable

You can use this function to:

- activate **position control** (the prerequisite for closed-loop-controlled operation of the FM 354)
- to switch the signal through to the drive as indicated in MD37
- to provide control of stops between movements in the Control mode.

Parking axis

This function can be used to change over the measurement system while the complete system is running.

With this setting:

- Encoder synchronization (SYN = 0) is deleted
- Pending error messages are deleted and no new ones are initiated (including diagnostic interrupts) by the encoder when ”Parking Axis” is disabled.
- Digital outputs are inactive; analog voltage value: 0 V.

The setting can be switched on or off if “Processing in progress” = 0.

Simulation

You can use this function to:

- Test function sequences without the drive and measuring system.

  A drive, if any connected, must be switched off.
- Evaluate all digital inputs (**Caution**, if you are going to simulate sequences that use such signals, they should be connected to the inputs of the FM 354 - e.g. for Reference-Point Approach).
- The servo simulates a controlled system; “Controller Ready” is not necessary.
- The setting can be switched on or off if “Processing in progress” = 0.
- All internal function sequences behave as in normal operation.

When the function is deactivated, the axis is reset internally (see Restart, Section 9.3.3).

9.3.3 Single commands (user DB, DBB36 and 37)

Overview

You can use this function to transfer single commands to the FM 354. These commands are:

- Activate machine data
- Delete residual distance
- Automatic block search in reverse
- Automatic block advance
- Restart
- Undo set actual value

Callup of single commands

The single commands are activated when the corresponding data record is transmitted to the FM 354.

The commands are deleted in the FM 354 after execution.

Activate machine data

Once you have downloaded the machine data (MD) or the MD block (from the programming device), the machine data have to be activated. At the first parameterization, the machine data is transferred automatically. In terms of effects, the FM 354 distinguishes between "K" and "E" machine data.

MD category	Effect in FM 354 after activation
"K"	"Reset" of the FM • As long as "Reset" is in progress, it is not possible to transfer other data • For internal response, see Restart
"E"	FM operating condition is maintained

For machine data, see Section 5.3.1.

This command is possible only when the axis is not in motion ("Processing in progress" = 0). An operating mode must be selected.

An MD block is likewise activated by switching on or off.

Delete residual path

You can use this command to delete a residual distance that remains after a job has been canceled.

- Effective only in "Incremental Relative", "MDI", and "Automatic" modes after a stop (SFG = 1 and BL = 1). If processing is not interrupted with a stop, the "Delete Distance to Go" request is suspended in the FM 354.
- On starting after a Delete Residual Distance in MDI mode, the active MDI block is processed from the start.
- On starting after a Delete Residual Distance in Incremental Relative and Automatic modes, processing continues with the following block.

Automatic block advance

This command is described in Section 9.2.7.

Automatic block return

This command is described in Section 9.2.7.

Restart

You can use this command to reset the axis.

- Pulse output is interrupted.
- The instantaneous processing status is canceled, and synchronization in incremental encoders is deleted.
- Active compensation values are deleted.
- An acknowledge signal is issued for all errors.

Undo set actual value

You can use this command to reset coordinates modified with the functions “Set actual value” and “Set actual value on-the-fly” to their original value (if the axis is not in motion).

9.3.4 Zero offset (Write request in the user DB, DBX39.1)

Function of zero offset

A selection, change or cancellation of a zero offset takes effect with the next positioning action. With a zero offset, the instantaneous shift of the coordinate system is canceled, provided that a zero offset was already active and the specified shift change ($ZONPV_{new} - ZO_{old}$) was executed with the first positioning. All coordinates and software limit switches, the reference point and the actual value are updated accordingly.

The reference of the workpiece coordinate system is provided via the zero offset. The actual values read from the FM 354 refer to the machine coordinate system. The position specification in a traversing block is referred to the workpiece coordinate system.

Example of a zero offset:

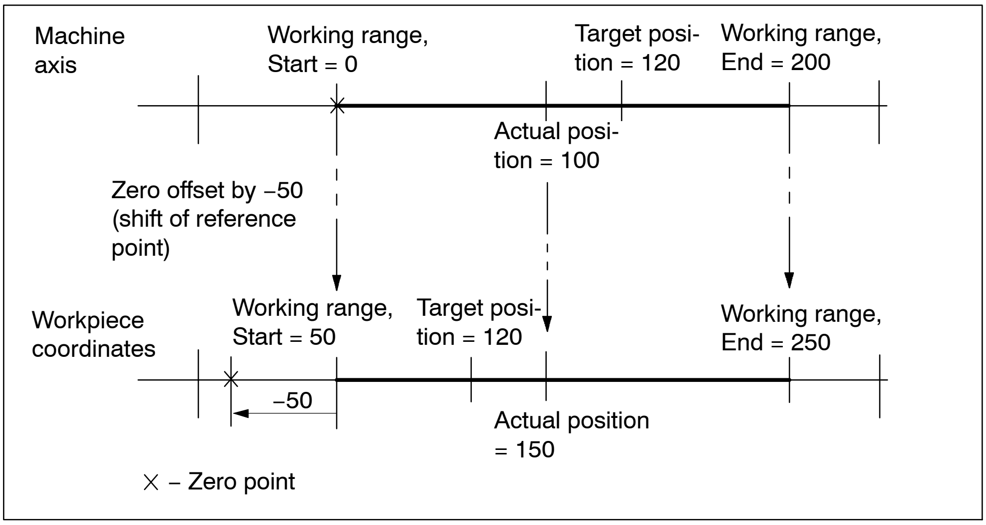


Fig. 9-1 Zero offset

The zero offset can be deleted by:

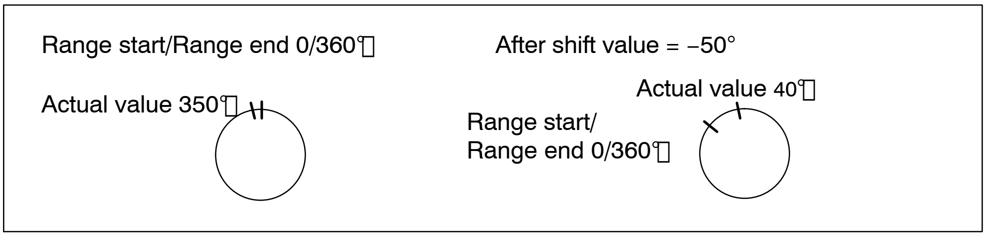
- Transmitting shift value = 0
- Starting Reference-Point Approach mode
- Set reference point
- Eliminating axis synchronization (e.g. by a restart).

Rotary axis

The following restriction applies to a rotary axis:

Zero offset < Rotary-axis range. The actual value is normalized.

Example:



The start and end of the range are shifted −50°.

Exceptions:

In the Incremental Relative, MDI and Automatic modes, a zero offset is not possible until the block has been processed (position reached, programmed stop set), i.e. it is not possible when the axis is stationary after execution is interrupted with an abnormal stop.

9.3.5 Set actual value (Write request in the user DB, DBX38.7)

Overview

You can use this function to assign a new value to the current actual value.

Function of set actual value

By transmitting the coordinates, the actual value is set to this value when the axis is not in motion (after selecting “Processing in progress” = 0). The coordinates of the software limit switches remain unchanged.

Example of setting actual value:

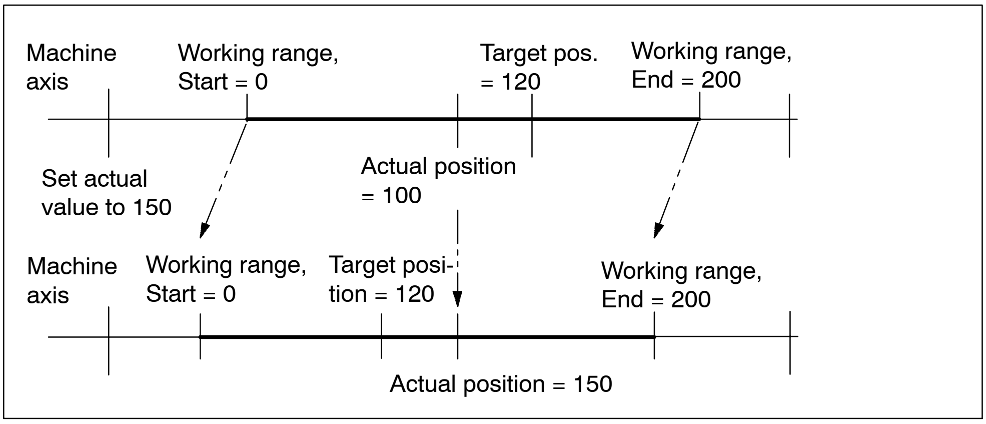


Fig. 9-2 Set actual value

The coordinates can be reset to their original value by:

- Including synchronization in Reference Point Approach mode
- Set reference point
- Remove setting actual value
- Restart

9.3.6 Set actual value on-the-fly (Write request in the user DB, DBX39.0)

Overview

You can use this function to assign a new value to the actual value by means of an external event.

Function of set actual value on-the-fly

By transmitting the coordinates (new actual value), set actual value on-the-fly is activated.

However, the Set Actual Value function is not triggered via the appropriate digital input until “Processing in progress” = 1.

Set actual value on-the-fly can be activated again by transmitting Set actual value on-the-fly again.

The coordinates can be reset to their original value by:

- Including synchronization in Reference Point Approach mode
- Set reference point
- Remove setting actual value
- Restart

Note:

For “set actual value on-the-fly” in “Automatic” mode, see Section 9.2.7

9.3.7 Request application data (Write request in the user DB, DBX39.6)

Overview

A selection of up to four display data items whose values can be read out with “read application data” (see Section 9.3.13).

Code table:

Code	Significance
0	No parameter request
1	Actual position
2	Actual speed
3	Residual distance
4	Set position
5	Total current coordinate shift
6	Rotational speed
16	Frequency output value
17	Pulse output counter (0...2^{16}–1) [pulse]
20	Difference between target and actual position [MSR]
22	Switch adjustment in “Reference point approach” mode

The code should be entered in CODE_AP1 – AP4.

These values are always updated in the FM cycle.

The selection is stored on the FM, i.e. you need make the selection only once to read the corresponding values cyclically afterwards (user DB, DBX43.6).

9.3.8 Teach in (Write request in the user DB, DBX39.7)

Overview

In a program block selected with the program number and block number, the current actual position is entered as a position setpoint (**Caution:** This is an absolute position).

The Teach-in facility is possible only in the following modes:

- Jogging
- Incremental Relative
- MDI

and while the axis is not in motion.

The program and the appropriate program block must be present on the FM 354 (see Parameterization, Chapter 5).

9.3.9 Set reference point (Write request in the user DB, DBX38.6)

Overview

You can use this function to synchronize the axis without reference point approach.

Function

With Set Reference Point, a position value at the instantaneous position of the axis, indicated as a parameter, is accepted as an actual value.

”Set reference point” is not possible during machining ([BL] = 1) and will only come into effect after machining ([BL] = 0).

This function is not effective in “Automatic” mode (mode 08) or in “Automatic Single block” mode (mode 09).

9.3.10 Measured values

Prerequisites

The following prerequisites must be fulfilled to execute the “Measuring” function:

1. Connect a bounce-free switching-signal encoder (sensing probe) to a digital output of the FM 354.
2. Parameterize “Measurement” for this input in MD34.

Activating the measuring function

The “Individual settings” function (AW-DB, DBB34 and 35) can be used to activate a “length measurement“ or “in–process measuring”.

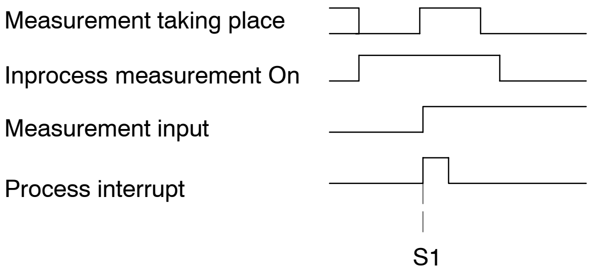
Because both functions use the same digital input of the FM 354, only one function can be performed at a time. If both functions are activated at the same time, both functions are deactivated, and an error message is output.

Function description

Measurement functions can be executed in all modes. An executed measurement is signaled by the checkback signal “ME” and optionally also by a process interrupt.

Inprocess measurement

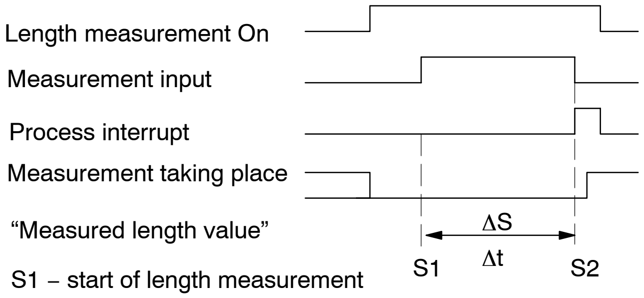
The present actual position is captured at each rising edge of the touch probe. At the same time, the axis movement is interrupted (servo-controlled braking).



S1 – execution of measurement

Length measurement

The current actual position value is acquired at both the rising and the subsequent falling edge of the sensing probe. In addition, the distance (absolute value) actually traversed between the two edges is determined.



S1 – start of length measurement
S2 – end of length measurement
ΔS – Measured length value
Δt – minimum signal length at the digital input: ≥ 2 · FM cycle

The measured length is determined as follows under certain supplementary conditions:

- Retrigger the reference point:
  The measured length constitutes the difference of the edge positions.
- Set the actual value: The measured length constitutes the distance actually traversed.
- Zero offset:
  The "Zero offset" function does not change the actual position of the axis and is thus not relevant for evaluations regarding the "Length measurement" function.
- Rotary axis:
  The measured length is the path actually traversed over several rotations of the rotary axis.
- Direction reversal:
  The axis can change the traversing direction within the length to be measured as often as you want.
  - Case 1:
    The axis passes the falling edge in the traversing direction of the rising edge: The measured length is the difference of the edge positions.
  - Case 2:
    The axis passes the falling edge in the opposite direction to the rising edge: End of the length measurement with output of an error message.

Error messages

The following table lists the errors that can occur in the execution of the Measurement function.

Error	Significance
Travel errors	The “Digital input not parameterized” error is signaled when a measurement function is selected without a digital input having been parameterized (see Troubleshooting, Table 11-7, Class 3 No. 30).
Operator contr. errors	The “Measurement function undefined” error is signaled when both measuring functions are selected (see Troubleshooting, Table 11-6, Class 2 No. 16).
Measuring error	An erroneous length measurement is indicated by signaling back the length “–1.” Possible causes include: • Resynchronization in the Reference-Point Approach mode while a measurement is in progress • Execution of the Set Reference Point function while a measurement is in progress • Direction of travel at the falling edge is opposite to the direction of travel of the previous rising edge

Measurement–value checkback signals

The checkback signal ME (see Section 9.1) signals the status of function execution, as follows:

“ME”	Inprocess measurement	Length measurement
0	• the Length Measurement and Inprocess Measurement functions are inactive • with Start after a prior measurement	• the Length Measurement and Inprocess Measurement functions are inactive • with front edge of the touch probe signal after a prior measurement
1	With the front edge of the touch probe signal (= Inprocess Measurem. is in progr.)	With the back edge of the touch probe signal (= Length Measurement is in progress)

Reading out the measured values

After performing the “Length measurement” and “In–process measurement” functins you can read out the measured values from the FM 354 by calling the **POS_MSRM** block using the AW-DB read job, DBB43.7 (see Section 6.3.5).

The measured values read out are valid for the performed measuring process starting from the “ME” checkback signal until the edge of the subsequent measuring job arives.

No.	Value 0	Value “–1”	All other positive values	All other negative values
1	the Length Measurement and Inprocess Measurement functions are inactive	Actual position for rising touch-probe edge in Inprocess Measurement and Length Measurement functions		
2	• the Length Measurement and Inprocess Measurement functions are inactive • always with the Inprocess Measurement function	Actual position for falling touch-probe edge in Length Measurement function		
3	• the Length Measurement and Inprocess Measurement functions are inactive • always with the Inprocess Measurement function • measured length 0 is actually possible, because touch probe has been connected while axis is stationary	Erroneous length measurement	Measured length	Nonexistent

9.3.11 Basic operating data (Read request in the user DB, DBX42.0)

Overview

The following display data are basic operating data:

- Actual position [MSR]
- Actual speed [MSR/min]
- Distance-to-go [MSR]
- Set position [MSR]
- Total of active coordinate shifts for tool offset, zero offset [MSR]
- Rotational speed (rotary axis only) [rpm]

9.3.12 Active NC block (Read request in the user DB, DBX42.1), next NC block (Read request in the user DB, DBX42.2)

Active NC block

... are display data in “Automatic” mode

/	Skipped block
L	Subprogram callup (fills in UP number)
P	Number of runs for subprogram (fills in UP number of runs)
X/t	Position/dwell time programmed (fills in value 1)
G1-G3	G function group 1-3
D	Tool offset value number
M1-M3	M function group 1-3
F	Speed programmed (fills in value 2)

Byte	Data for-mat	Bit							
		7	6	5	4	3	2	1	0
0	Byte	NC program number							
1	Byte	NC block number							
2	8-bit	/	L	P	X/t	0	G3	G2	G1
3	8-bit	0	0	0	D	M3	M2	M1	F
4	Byte	G function 1							
5	Byte	G function 2							
6	Byte	G function 3							
7	Byte	0							
8	DINT	32-bit value 1 (UP number, bytes)							
12	DINT	32-bit value 2 (UP number of runs, bytes)							
16	Byte	M function 1							
17	Byte	M function 2							
18	Byte	M function 2							
19	Byte	D function							

Next NC block

as described in “active NC block”

9.3.13 Application data (Read request in the user DB, DBX43.6)

Overview

The values requested with ”Request application data” (user DB, DBX42.3, see Section 9.3.7) are returned by the FM 354 when the user submits the Read request (user DB, DBX43.6).

9.3.14 Actual value block change (Read request in the user DB, DBX42.3)

Overview

The “actual value block change” function is described in Section 10.1, G50, G88, G89.

9.3.15 Servicing data (Read request in the user DB, DBX42.4)

Overview

The following display data of the measuring circuit are servicing data:

- DAC output value [mV]
- Actual encoder value [MSR]
  - Incremental encoder: (0...2^{16} – 1) [encoder increments]
  - Absolute encoder: Absolute value + encoder
- Missing pulses [pulses]
- K_v factor (position control loop gain)
- Following error [MSR]
- Following-error limit [MSR]
- s overshoot value/switch alignment in “Reference point approach” mode [MSR]
- Approach time Te [ms]/drive time constant Ta [ms] in “Open-loop control” mode

9.3.16 Additional operating data (Read request in the user DB, DBX43.5)

Overview

The following display data are additional operating data:

- Override [%]
- NC traverse program no.
- NC block no.
- Callup SR loop no. counter
- G90/91 active, see Section 10.1
- G60/64 active, see Section 10.1
- G43/44 active, see Section 10.1
- D No. active, see Section 10.1
- Status messages 1 (data type: BOOL):
  - Speed limitation to limit value from MD23
  - Limitation to ± 10 V
  - Limitation of minimum acceleration or minimum deceleration in effect
- Status messages 2 (data type: BOOL): not in use

9.3.17 Parameters/data (Read request in the user DB, DBX43.3)

Overview

This request is used to read the parameters and data modified by means of the ”Change parameters/data” function (Write request in the user DB, DBX39.3, see Section 9.3.1).

9.4 System of measurement

Overview

At the start of parameterization, you must fill in the basic machine data item **system of measurement** (MD7). This item governs the input of values.

Variants of the system of measurement

You can set the system of measurement for the following three units:

- mm
- inches
- degrees

Input of machine data

All value inputs and all value ranges refer to the setting in the system of measurement.

Internal processing of values

In “Parameterize FM 354” and in the FM 354 itself, values are processed in the following base units:

- 0.001 0.001 mm
- 0.0001 inch
- 0.0001 degree

Examples

The sample values in the table below illustrate the relation between the system of measurement and internal values:

System of measurement	Internal values	Input at interface	
mm	10^{-3} mm	10 995 · 10^{-3} mm	10.995 mm
inches	10^{-4} inch	10,995 · 10^{-4} inch	1.0995 inch
degrees	10^{-4} degree 10^{-2} degree	3,600,000 · 10^{-4} degree 36,000 · 10^{-2} degree	360.0000 degree 360.00 degree

Note

The measurement system (MD7) must match the measurement system specified in the other DBs.

The measurement system raster (MSR) is the smallest distance unit in the active system of measurement.

If at some point you have failed to take this precaution:

1. Delete all data blocks (which do not match the measurement system) or clear the memory of the FM 354 completely.
2. Modify the other data blocks on the PG.
3. Reload the data blocks to the FM 354.

9.5 Axis type

Overview

You can select the axis type with machine data item MD8. Choose either of the following types:

- Linear axis
- Rotary axis

Linear axis

A linear axis moves between two range limits (traversing range min -10^9, max 10^9). The range limits may be bounded by software limit switches (MD21, MD22) to form the "working range."

Linear axes have a limited traversing range. The limits are imposed by the:

- Resolution of the numerical scale
- The range covered by an absolute encoder.

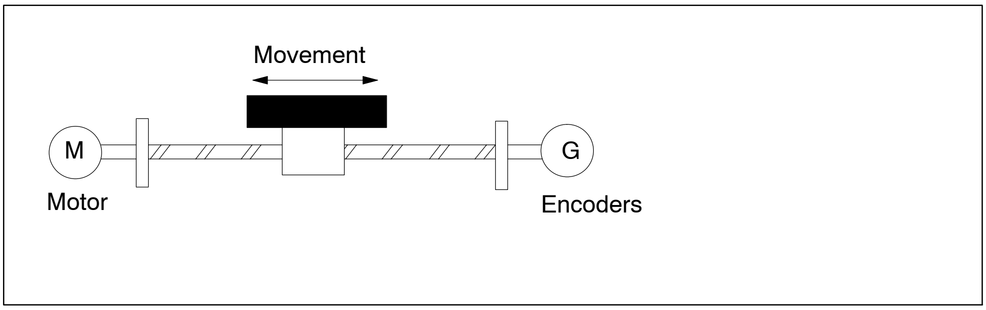


Fig. 9-3 Linear axis

Rotary axis

With rotary axes, the actual value is reset to “0” after one revolution. Rotary axes thus have an infinite traversing range.

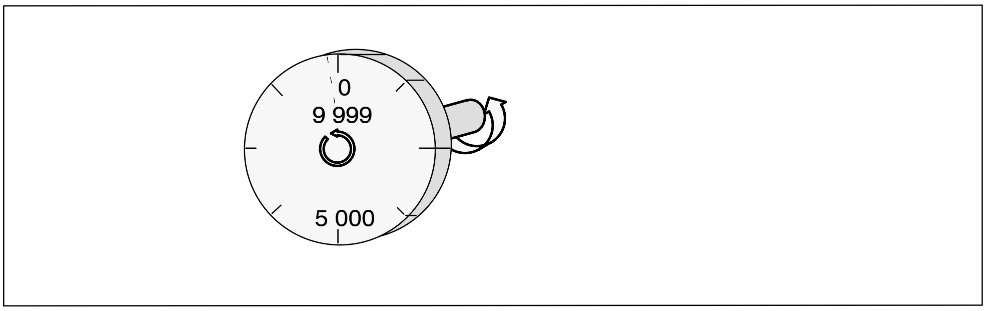


Fig. 9-4 Rotary axis

Rotary axis end

Machine data item MD9 defines the value by which the FM 354 recognizes the end of the rotary axis.

This value is the theoretical maximum that the actual value can reach. At this value, display of the actual value switches back to 0.

The theoretical maximum, however, is never displayed, because it is physically located in the same position as the start of the rotary axis (i.e. 0).

Example:

The following example in Figure 9-4 illustrates the behavior of the axis.

Assume you specify a value of 10,000 for the end of the rotary axis.

The value 10,000 will never be displayed. The display always rolls over from 9,999 to 0.

If the direction of rotation is negative, the display rolls over from 0 to 9,999.

Encoders on rotary axes

Rotary axes are subject to certain restrictions in the choice of encoder/gearbox/motor, as shown in Figure 9-5. These restrictions arise from the need to reproduce the actual position accurately over several revolutions when referencing (with incremental encoders or on POWER OFF/ON with absolute encoders). See also “Dependencies”, Section 5.3.1).

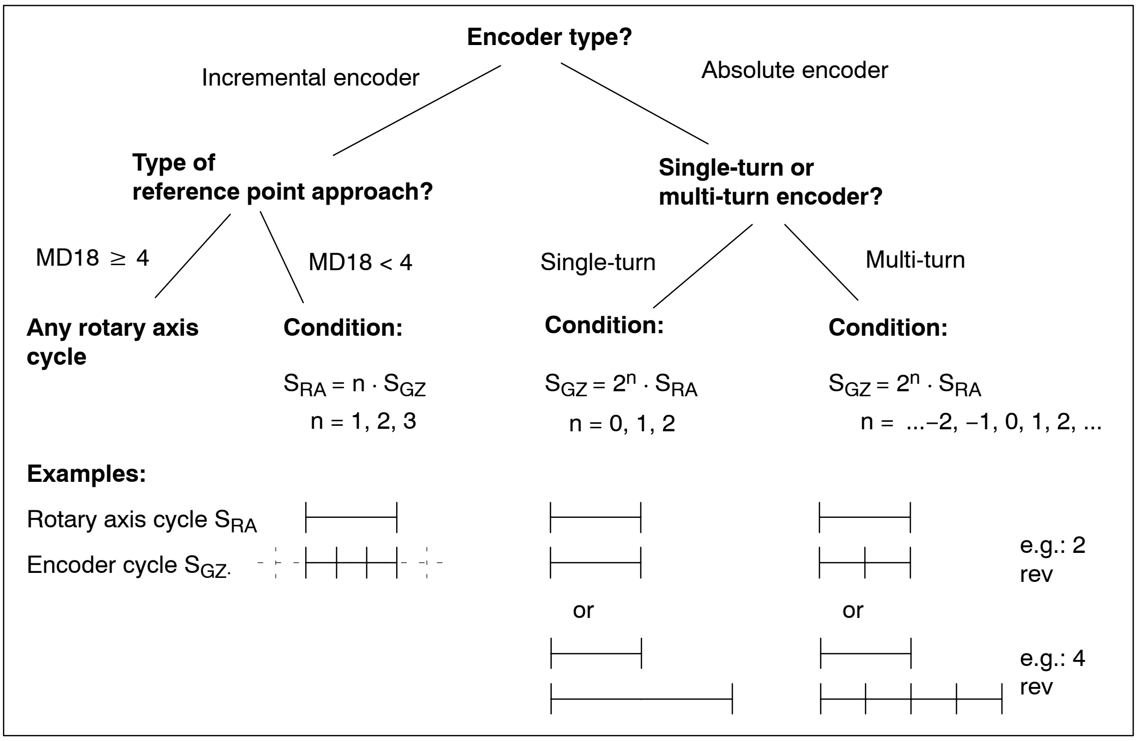


Fig. 9-5 Encoders on rotary axes

9.6 Encoders

Overview

One of the following encoders must be connected to the measuring-system interface of the FM 354 (see Fig. 1-4).

- Incremental encoder
- Absolute encoder (SSI)

Paths and positions are displayed in 10^{-3} mm, 10^{-4} inches or 10^{-4} degrees, as selected with machine data item MD7.

The path resolution of the machine axis obtained by the encoder is calculated within the FM from machine data MD11 to MD13.

Encoder selection

The prerequisite for achieving a given positioning accuracy is an n-fold improvement in path resolution by the encoder.

Recommended values for n		
Minimum	**Optimum**	**Maximum**
2	4	10

For that reason, when configuring a given specific application, select an encoder that meets the desired positioning accuracy requirements.

The known design data for the machine axis and the desired measurement resolution A:

$$A = \frac{1}{n} \cdot \text{Positioning accuracy} \qquad \text{[mm], [inches], [degrees]}$$

yield a calculation of the necessary pulse number per encoder revolution according to the following relationship (taking a metric measuring system as an example):

Incremental encoder	**Absolute encoder (SSI)**
$I_G = \dfrac{S\ [\text{mm}]}{4 \cdot i_{GS} \cdot A\ [\text{mm}]}$	$S_G = \dfrac{S\ [\text{mm}]}{i_{GS} \cdot A\ [\text{mm}]}$

The table below gives you an overview of the data used in this calculation and their meaning. You will find the machine data (MD) assignments under “Function parameters”.

Symbol	Significance
I_G	Increments per encoder rotation (incremental encoder)
S_G	Number of steps per encoder revolution (absolute encoder)
S	Distance per spindle or rotary table revolution [mm/rev], [inches/rev], [degrees/rev]
A	Required resolution [mm], [inches], [degrees]
4	Pulse multiplication (constant)
i_{GS}	Ratio between encoder and mechanism - Number of encoder revolutions $\left[\frac{\text{number of encoder revs}}{\text{spindle revolution}}\right]$ or $\left[\frac{\text{number of encoder revs}}{\text{or rotary table revs}}\right]$

If unusual numbers of pulses or steps result, the encoder with the next-higher number of pulses or steps should be selected.

9.6.1 Incremental encoders

Overview

Incremental encoders serve to detect position values, supplying pulses that the FM 354 adds up to form an absolute value. After the FM 354 is switched on, there is an offset, which cannot be determined in advance, between the internal position value and the mechanical position of the axis. In order to establish the position reference, the internal value must therefore be set to a predefined value at a specific axis position. This value is stored in the machine data (MD) as a reference point coordinate (see Section 9.2.3).

Incremental encoders

The following variant applications are possible:

- **Rotary incremental encoder on linear axes**

  Encoders with one zero pulse per revolution may be used. The number of encoder pulses must be a multiple of ten or a power of two.

- **Rotary incremental encoder on rotary axes**

  Encoders with one zero pulse per revolution may be used. The number of encoder pulses must be a multiple of ten or a power of two. With indirect encoder mounting and reference point approach with a zero pulse (MD18 < 4), you must ensure that the revolution of the rotary axis is divisible without remainder by the cyclical zero pulse (see “Dependencies” Sections 5.3.1. and 9.5).

- **Linear scales on linear axes**

  Scales may be used with at least one reference zero pulse, or with a cyclic zero pulse.

  In comparison to rotary incremental encoders, instead of the encoder revolution a period of division is used as a basis here, corresponding for example to the segment between two zero-mark pulses.

Function parameters

Table 9-9 shows you how to adapt the selected encoder to the FM 354.

Table 9-9 Function parameters – Incremental encoders

MD	Designation	Value/Meaning	Comments/Unit
10	Encoder type	**1 = Incremental encoder**	(Code number)
11	Displacement per encoder revolution (division period)	1...1 000 000 000 see Section 5.3.1, Dependencies	[MSR] (integer portion)
12	Residual distance per encoder revolution (division period)	0...2^{32}–1 see Section 5.3.1, Dependencies	[2^{-32}MSR] (fractional portion)
13	Increments per encoder revolution (division period)	2^1...2^{25} see Section 5.3.1, Dependencies	Entry according to encoder rating plate
19.0	Direction adjustment	**1 = invert measured value direction**	–
20 20.0 20.2 20.3	Hardware monitoring	 **1 = Cable break** **1 = Pulse monitoring** **1 = Voltage monitoring**	Entry for monitoring to be switched on

MSR stands for measurement system raster (see Section 5.3.1)

Sample encoder adjustment

Encoder: Number of increments per revolution (MD13) = 2,500

(The FM 354 works by the principle of quadruple evaluation. This yields an FM-internal number of increments per revolution = 10,000.)

Machine design:

- Motor with 50:30 gear ratio on spindle with 10 mm pitch = 10,000 MSR
- Encoder on motor.

From this one can calculate the following traversing distance per encoder revolution:

$$\text{Gear ratio:} \quad i = \frac{\text{50 spindle revolutions}}{\text{30 motor revolutions}} = 1.666666...$$

$$\text{Displacement per encoder revolution} = i \cdot 10{,}000 \text{ MSR} = 16{,}666.666... \text{ MSR}$$

The following values are entered:

MD	Value	Unit
11	16 666	[10^{-3} mm]
12	0.666... · 2^{32} = 2,863,311,530	[$2^{-32} \cdot 10^{-3}$ mm]
13	2 500	[pulse/rev]

Monitoring/error diagnostics

If MD20 = 0 is input, all monitoring functions are active.

Individual monitoring functions can be inactivated by entering 0 in the designated bit of MD20.

You can deactivate the error messages using the single function “parking axis” (see Section 9.3.2).

Table 9-10 Error diagnostics – Incremental encoder

Diagnosis	Cause	Error message
Cable break monitoring	Signals of one track pair (A, $\overline{A}$ / B, $\overline{B}$ / N, $\overline{N}$) do not behave as negations of one another.	The FM 354 responds with a diagnostic interrupt, external channel error (see Troubleshooting, Table 11-4)
Pulse monitoring	• In the case of decimal encoders (e.g. 1000 incr. per rotation), the pulse rate between two zero pulses is not divisible mod 10 • In the case of binary encoders (e.g. 1024 incr. per rotation), the pulse rate between two zero pulses is not divisible mod 16	• If the contents of the missing–pulse memory (sum of all mod values) exceed the value 7, the diagnostic interrupt ’external channel error’ (see Troubleshooting, Table 11-4) is generated. • The control signal ”Restart” clears the missing-pulse memory.
	In Reference-Point Approach mode, no zero pulse is recorded after leaving the reference-point switch within the path as defined in MD11, 12.[1]	**Effect:** • Encoder cannot be synchronized. • On leaving the reference-point switch in Reference-Point Approach mode, the FM 354 will travel no more than the distance of one encoder revolution (MD11), and needs the deceleration distance from the reducing speed.
Voltage monitoring	Encoder power failure	Diagnostic interrupt, external channel error (see Troubleshooting, Table 11-4)

[1] The monitoring can be disabled using MD20

Warning

Hardware monitoring functions should be skipped only for test purposes, since positioning errors may destroy the machine.

Exception:

Pulse monitoring for encoders with non-cyclic zero pulse.

Encoder connection

See Section 4.5.

9.6.2 Absolute encoders (SSI)

Overview

Absolute encoders (SSI) have several significant advantages over incremental encoders:

- Longer cable lengths
- Reliable data capture by using a single-step GRAY code
- No encoder synchronization needed.

Absolute encoders (SSI)

You can use 13-bit single-turn encoders or 25-bit multi-turn encoders with the SSI protocol (left-justified).

- **Absolute encoder (SSI) on linear axes**

  Make sure the value range of the encoder is at least equal to the traversing distance of the axis.

- **Absolute encoder on rotary axes**

  Make sure that the absolute value range captured by the encoder corresponds to a ratio of 2^x or 2^{-x} to one revolution of the rotary axis, and that it encompasses at least one rotary axis revolution (see “Dependencies” Section 5.3.1 and Figure 9-5).

Function parameters

Table 9-11 shows you how to adapt the selected encoder to the FM 354.

Table 9-11 Function parameters – Absolute encoders (SSI)

MD	Designation	Value/Meaning	Comments/Unit
10	Encoder type	**3 = Absolute encoder (SSI 13 bit)** **4 = Absolute encoder (SSI 25 bit)** **13 = Absolute encoder (SSI 13 bit)** **14 = Absolute encoder (SSI 25 bit)**	GRAY Code GRAY Code Binary Code Binary Code
11	Displacement per encoder revolution (division period)	1...1 000 000 000 see Section 5.3.1, Dependencies	[MSR] (integer portion)
12	Residual distance per encoder revolution	0...2^{32}–1 see Section 5.3.1, Dependencies	[2^{-32} MSR] (fractional portion)
13	Increments per encoder revolution (division period)	2^1...2^{25} see Section 5.3.1, Dependencies	Entry according to encoder rating plate

Table 9-11 Function parameters – Absolute encoders (SSI), continued

MD	Designation	Value/Meaning	Comments/Unit
14	Number of revolutions of SSI encoder	0/1 = Single-turn encoder $2^1 ... 2^{12}$ for multi-turn encoder see Section 5.3.1, Dependencies	Only powers of 2 allowed
15	SSI baud rate	1 = 78,000 Baud 2 = 156,000 Baud 3 = 312,000 Baud 4 = 625,000 Baud 5 = 1,250,000 Baud	(Code number) The baud rate depends on the cable length between FM 354 and encoder
19.0	Direction adjustment	**1 = invert measured value direction**	–
20	Hardware monitoring		Entry for monitoring to be switched on
20.1		**1 = Error in absolute encoder**	
20.3		**1 = Voltage monitoring**	

MSR stands for measurement system raster (see Section 5.3.1)

Sample encoder adjustment

Encoder: Number of increments per revolution (MD13) = 4096 = 2^{12}

Number of revolutions (MD14) = 256 = 2^8

Machine axis design:

- Motor with 50:30 gear ratio on spindle with 10 mm pitch = 10,000 MSR
- Encoder on motor.

From this one can calculate the following traversing distance per encoder revolution:

Gear ratio: $i = \frac{\text{50 spindle revolutions}}{\text{30 motor revolutions}} = 1.666666...$

Displacement per encoder revolution = $i \cdot 10{,}000$ MSR = 16,666.666... MSR

The following values are entered:

MD	Value	Unit
11	16 666	$[10^{-3}$ mm]
12	$0.666... \cdot 2^{32}$ = 2,863,311,530	$[2^{-32} \cdot 10{-}3$ mm]
13	4096	[puls/rev]
14	256	[rev]

Note

The encoder covers an absolute traversing distance of 256 · 16,666.666... MSR. In the 10^{-3} mm system of measurement this corresponds to a maximum axis traversing distance of 4,266.666... mm.

Monitoring/error diagnostics

If MD20 = 0 is input, all monitoring functions are active.

Individual monitoring functions can be inactivated by entering 0 in the designated bit of MD20.

You can deactivate the error messages using the single function “parking axis” (see Section 9.3.2).

Table 9-12 Error diagnostics – Absolute encoder

Diagnosis	Cause	Error message
Voltage monitoring	Encoder power failure	Diagnostic interrupt, external channel error (see Troubleshooting, 11-4)
Error in absolute encoder	Error in protocol for data transfer between absolute encoder and FM 354	Diagnostic interrupt, external channel error (see Troubleshooting, 11-4)

⚠

Warning

Hardware monitoring functions should be skipped only for test purposes, since positioning errors may destroy the machine.

When voltage monitoring is deactivated and the power supply to the encoders or the FM is switched off, an immediate failure in absolute value signaling can cause drive movements if:

- an operating mode other than open-loop control is active
- and follow-up mode is deactivated
- and the servo enable is activated or not parameterized.

Encoder connection

See Section 4.5.

9.6.3 Synchronizing the encoders

Overview

When using incremental encoders, at switch-on there is an offset, which cannot be determined in advance, between the internal position value in the FM and the mechanical position of the axis. To establish the position reference, the value internal to the FM must be synchronized with the real position value of the axis. Synchronization is performed by taking over a position value at a known point of the axis.

When using absolute encoders (SSI), at switch-on there is already a defined relationship between the position value internal to the FM and the mechanical position of the axis. This reference can be adjusted by setting an absolute encoder alignment value (see Section 9.3.9, Setting the reference point).

Absolute encoder alignment

is the compensation value for numerical alignment of the internal FM position value.

Reference point approach

is an operating mode used to position the axis at the reference point.

Reference point

is a fixed point on the axis. It is:

- The target coordinate in the Reference-Point Approach mode
- Removed from the synchronization point by the amount of the reference-point shift, in axes with incremental encoders.

Reference point offset

Difference in distance between the synchronization point and the reference point.

The reference-point shift serves:

- for numerical measuring-system readjustment when an encoder is changed
- as a displacement reserve to brake the drive if the synchronization point is overshot.

Reference point switch (RPS)

The reference point switch selects the synchronizing (encoder) zero marker on the traversing path of the axis.

- It is also the signal encoder for a speed reduction before the synchronization point is reached.
- It is connected to a digital input of the FM 354.

Synchronization point

is a defined point on the traversing path of the axis. It is defined by the mechanical position of a reference-point switch or in association with a cyclic zero mark of an incremental encoder.

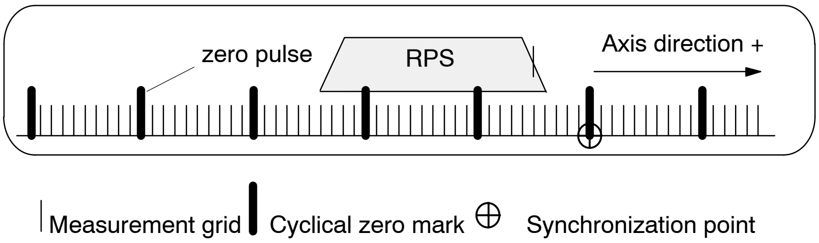
Synchronization

Creating the position reference between the internal FM position value and mechanical position of the axis.

Measured value synchronization with incremental encoders

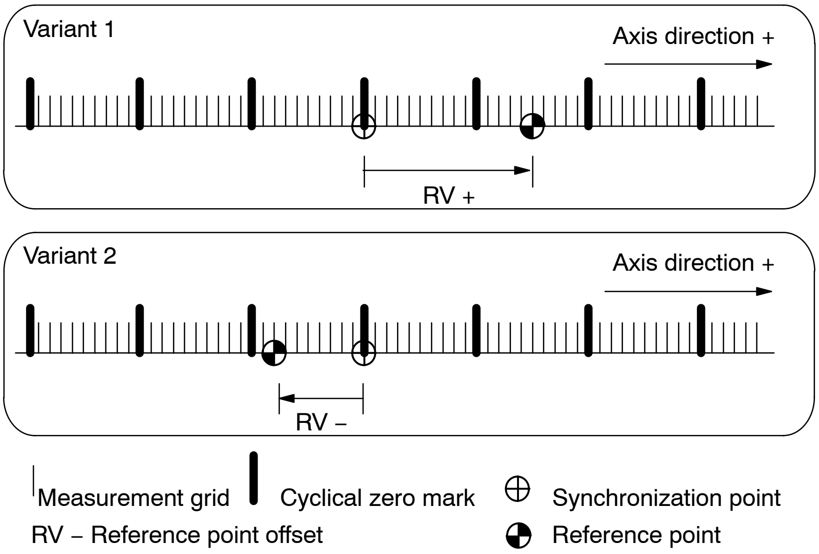
Irrespective of the approach direction, the synchronization point can be located on the side of the lower or the higher actual position values in relation to the reference point switch position. It is demarcated by the rising edge of a zero pulse or by the reference point switch. This selection is made by the MD18 (together with the approach direction).

Example



With reference to the synchronization point, the reference point can be located on the side of the lower or the higher position actual values. In the “reference point approach” operating mode the machine axis additionally traverses this distance, during its last phase of motion, once the synchronization point has been found. Consequently, the axis halts the motion, in each instance, exactly on the reference point.

Example



9.7 Position controller

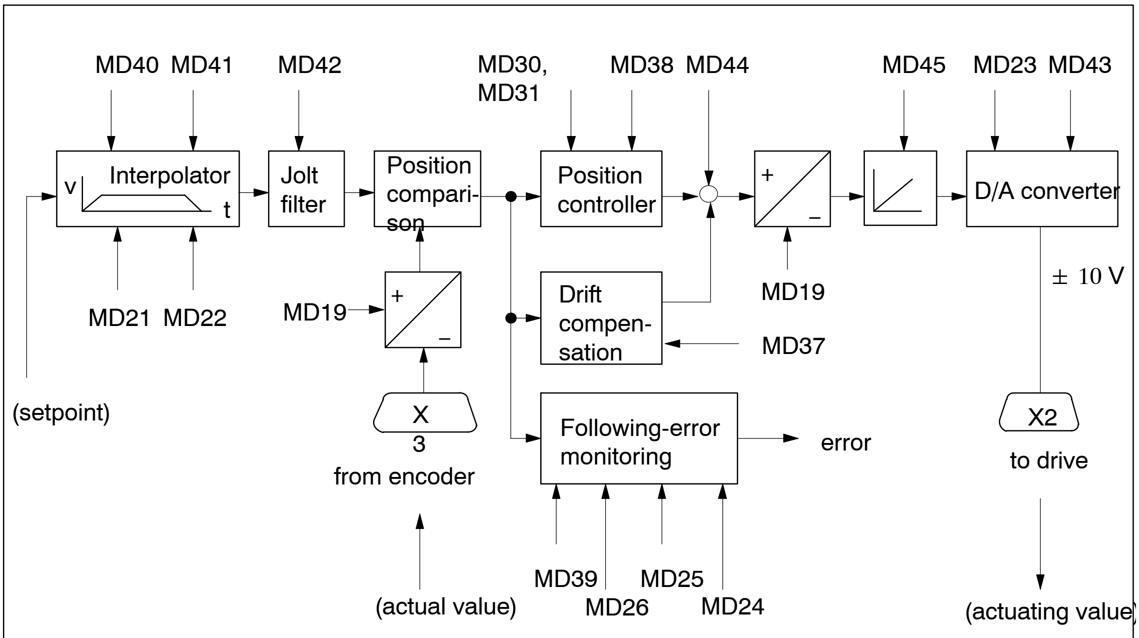
Overview

The closed-loop position controller performs the following tasks:

- guidance of the drive at the proper speed during execution of a movement
- precise approach of the axis to the programmed target position
- maintenance of the axis in position in the face of interfering factors.

The position controller is configured as a proportional-action controller. In its environment are a number of function units that provide support by performing special tasks within the complex of movement control, and that can be adapted to axis conditions by means of a variety of machine data.

The diagram 9-6 gives you an overview of the function units and the machine data acting upon them.

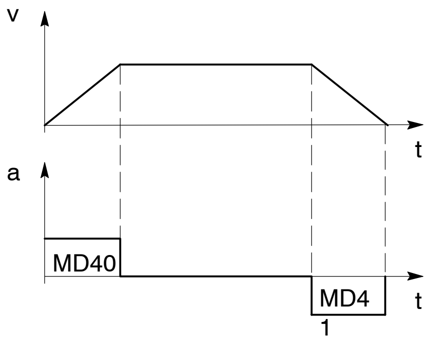


MD19 –	Direction adjustment	MD37 –	Control signal Servo Bit 16
MD21 –	Software limit switch - start	MD38 –	Positioning loop amplification
MD22 –	Software limit switch - stop	MD39 –	Min. following error, dynamic
MD23 –	Maximum speed	MD40 –	Acceleration
MD24 –	Target range; position reached, stop (PEH)	MD41 –	Deceleration
MD25 –	Monitoring time	MD42 –	Jolt time
MD26 –	Idle range	MD43 –	Set voltage, max.
MD30 –	Backlash compensation	MD44 –	Offset compensation
MD31 –	Directional reference, backlash	MD45 –	Voltage ramp

Fig. 9-6 Overview of position controller

Interpolator

The machine data for acceleration (MD40) and deceleration (MD41) can be used to adapt the transition response of the command variable defined by the interpolator to the transition response of the controlled system.



v – speed

a – acceleration

t – time

MD	Designation	Value/Meaning	Unit
40	Acceleration	0 = without ramp 1...100,000	$[10^3$ MSR/s$^2]$
41	Deceleration		

For settings of MD40/41 see Section 7.3.4

Software limit switches

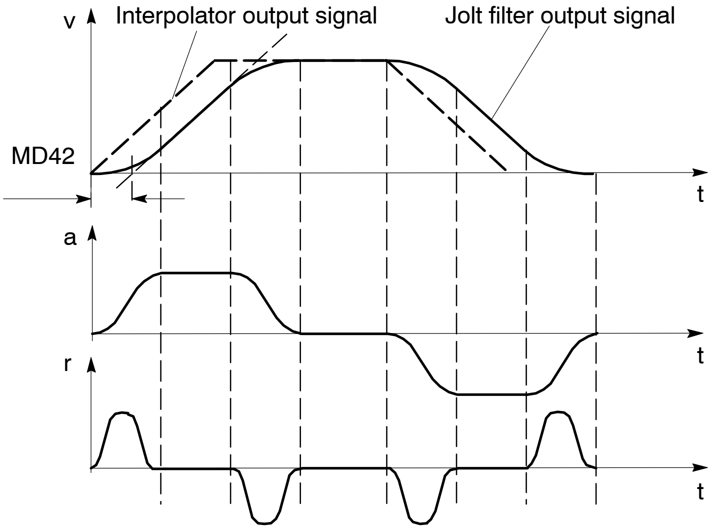
Software limit switches MD21 and MD22 (see Section 9.9) are used to limit the working area.

MD	Designation	Value/Meaning	Unit
21	Software limit switch, start	–1 000 000 000...< +1 000 000 000 see Section 5.3.1, Dependencies	[MSR]
22	Software limit switch, stop		

Jolt filter

Without jolt limitation the acceleration and deceleration act as abrupt variables. Jolt limitation allows the break points of a ramp-like speed curve to be smoothed out for both acceleration and deceleration. This yields particularly “soft” (jolt-free) acceleration and braking for certain positioning tasks, such as conveying of fluids.

Jolt time can be set in MD42 as the parameter for jolt limitation.



v – speed
a – acceleration
r – jolt
t – time

MD	Designation	Value/Meaning	Unit
42	Jolt time	0...10,000	[ms]

Basic diagnostics in the position control loop

In servo-controlled mode, the manipulated variable is compared periodically with the possible maximum values (⩾ 10 V or maximum frequency). A violation of the maximum limit is interpreted as follows:

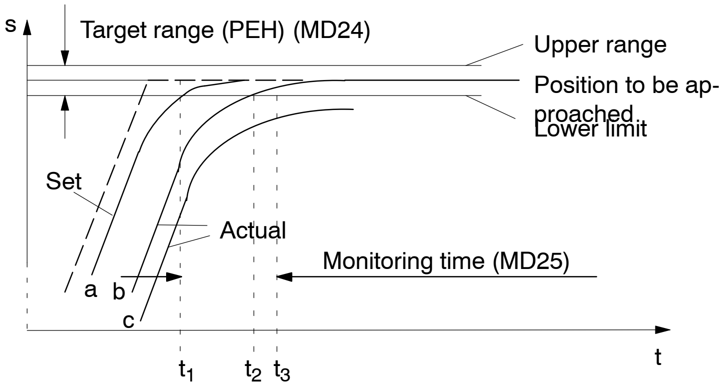
No axis movement:	"No drive movement" error message (see Table 11-7, Class 3/No. 65)
Traverse in the opposite direction:	"Direction of drive rotation" error message (see Table 11-5, Class 1/No. 11)
Correct travel direction:	Oversteer message in status message 1 (see Section 9.3.16, Additional operating data)

In all operating modes except Control mode, the "servo enable" signal is required for the duration of every traversing movement, irrespective of the parameter definitions. If the servo enable is not detected or is deactivated during the movement, the "servo enable missing" message is triggered (see Table 11-5, Class 3/No. 61).

In all operating modes except Open-loop control mode, the "servo ready" signal is required for the duration of every traversing movement when the parameter is active (MD37.2). If the servo ready is not detected or is deactivated during the movement, the "servo not ready" message is triggered (see Table 11-5, Class 3/No. 62).

Position approach

The following diagram shows position approach.



PEH - Position reached, stop
s – path
t – time

On approach to a position, the monitoring time is activated:

Time	Position monitoring
t_1 (a)	After the interpolator reaches the target position, the monitoring time (MD25) for reaching the target range is started in the CL controller, after the overtravel in the jolt filter dies down to the target range value (PEH on setpoint side).
t_2 (b)	Before the monitoring time expires, the actual position reaches the target range. Positioning is completed. A PEH is signaled, and exact matching is performed by the CL controller.
t_3 (c)	After the monitoring time expires, the actual position has not reached the target range (PEH). **Error message:** “PEH Target Range Monitoring” (see Troubleshooting, Table 11-7, Class 3/No. 64)

MD	Designation	Value/Meaning	Unit
24	Target range	0...1 000 000	[MSR]
25	Monitoring time	0 = no monitoring 1...65 534	[ms], rounded to 2-ms steps

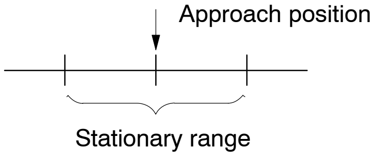
Following error monitoring

Axis standstill

A message is output on an axis standstill setpoint or deactivated servo enable if disturbances cause the axis to move out of position.

MD	Designation	Value/Meaning	Unit
26	Stationary range	0...1 000 000	[MSR]

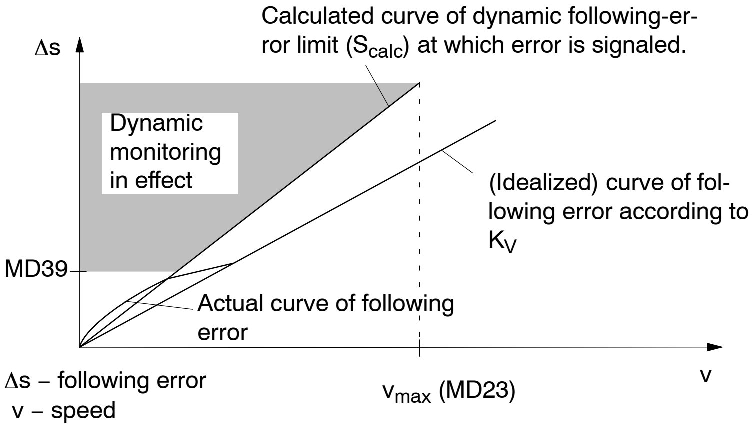
The standstill zone is located symmetrically around the target approach position.



When the tolerance window for idle is exceeded, the FM 354 signals a “Stationary Range” error (see Troubleshooting, Table 11-5, Class 1/No. 12).

Axis moving

To monitor following error during movement, the FM 354 calculates the allowable following error for the instantaneous traveling speed from the parameterized positioning loop amplification (MD38). Above the parameterized “Minimum following error (dynamic),” a comparison is performed with the actual value for the following error.



MD	Designation	Value/Meaning	Unit
39	Minimum following error (dynamic)	0 = no monitoring 1...1 000 000	[MSR]

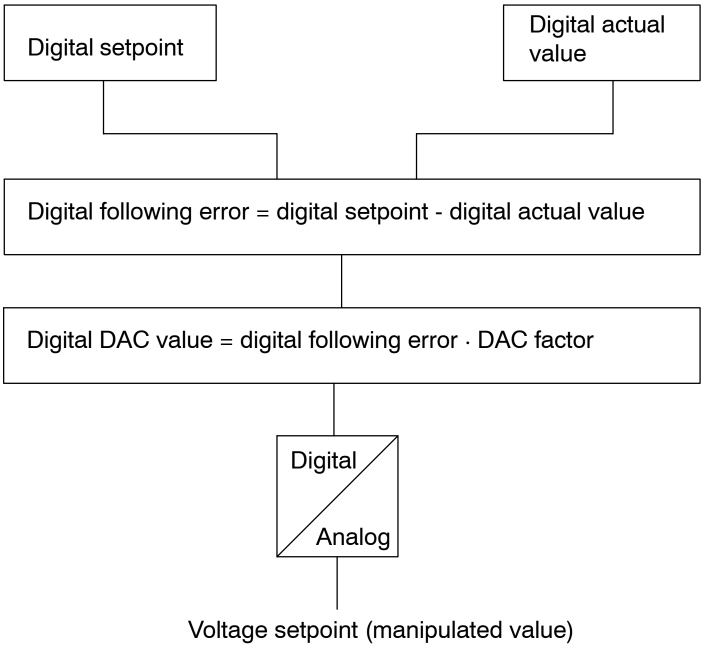
When the calculated dynamic following-error limit is exceeded (1.5fold of the idealized following error), the FM 354 signals the error ”Following error too large” (see Troubleshooting, Table 11-7, Class 3/No. 66).

Exception:

If an axis standstill occurs above the “minimum dynamic following error”, the error message described under Basic diagnostics “no drive movement” is output (see Table 11-7, Class 3/No. 65).

Position controller

The following error is calculated by periodical comparison of the set position defined by the interpolator with the actual position of the axis detected by the encoder. From this error, the position controller calculates the actuating signal needed for the drive in order to balance to a following error value of zero. The actuating signal is provided via a D/A converter (DAC) as an analog signal at the FM 354 output.


Digital setpoint
Digital actual value
Digital following error = digital setpoint - digital actual value
Digital DAC value = digital following error · DAC factor
Digital
Analog
Voltage setpoint (manipulated value)

Position control loop gain

The positioning loop amplification (K_V factor) specifies at what speed of axis travel a given following error sets in. The mathematical (proportional) relationship is:

$$K_V = \frac{\text{Velocity}}{\text{Following error}} = \frac{v\ [10^3\ \text{MSR/min}]}{\Delta s\ [\text{MSR}]}$$

Although the magnitude of the following error plays no dominant role for a single axis, the K_V factor still affects the following important characteristics of the axis:

- Positioning accuracy and stopping control
- Uniformity of movement
- Positioning time

The following relationship applies for these characteristics:

The better the axis design, the greater the achievable K_v factor, and the better the axis parameters from the technological viewpoint. The size of the K_v factor is especially affected by the time constants, backlash and spring components in the controlled system. In real applications the K_v factor moves within the following bandwidth:

- K_v = 0.2...0.5 poor-quality axis
- K_v = 0.5...1.5 good axis (normal case)
- K_v = 1.5...2.5 high-quality axis

The MD38 value is input with a resolution of 10^3, so that the following input value results:

$$MD38 = 10^3 \cdot K_v = 10^3 \cdot \frac{\text{Velocity}}{\text{Following error}} = 10^3 \cdot \frac{v\ [10^3\ \text{MSR/min}]}{\Delta s\ [\text{MSR}]}$$

MD	Designation	Value/Meaning	Unit
38	Positioning loop amplification	0...10,000	[(MSR/min)/MSR]

Drift compensation

Thermal conditions will shift the zero error in the control loop during operation. This effect is called drift. In a closed control loop with a proportional-action controller, this results in a temperature-dependent positioning error. You can activate automatic drift compensation with MD37, under which continuous balancing takes place in the positioning control loop.

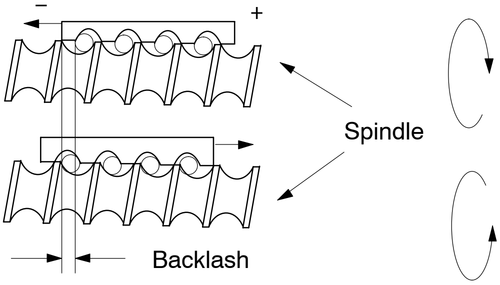
A basic compensation of the zero point error by means of the offet is required for the optimum effect of the drift compensation
(see MD44, offset compensation).

MD	Designation	Value/Meaning	Unit
37	Servo control signal	16 = automatic drift compensation active	–

Backlash compensation

Mechanical drive components as a rule have a certain amount of backlash (free play).

Mechanical reversing backlash can be compensated with MD30. In an indirect measuring system (with the encoder on the motor), the mechanical backlash is traveled at each change of direction before any axis movement occurs. The result is positioning errors.



When the position encoder is situated on the machine part to be positioned (e.g. on a saddle - direct arrangement), backlash adversely affects the achievable K_V factor. On the other hand, if the position encoder is attached to the drive motor (indirect arrangement), a high K_V factor can be achieved, but at the cost of position deviations that cannot be detected by the position controller. A backlash amount entered in MD30 is applied as a correction by the position controller as a function of the traveling direction at a given moment, thus achieving an approximate compensation for backlash in positioning.

MD31 is used to label the “backlash-free” or “accurate-measurand” traveling direction of the axis. If MD31 = 0, the “backlash-free” direction is the one that matches the direction of axis movement when synchronization is recorded. Depending on MD18, this will correspond to the following association:

MD18 = 0, 2, 4, 8: Plus direction is backlash-free
MD18 = 1, 3, 5, 9: Minus direction is backlash-free

MD	Designation	Value/Meaning	Unit
30	Backlash compensation	0 to 1 000 000	[MSR]
31	Directional reference of backlash	0 = as in reference point approach (not for absolute encoders) 1 = positive 2 = negative	–

The backlash is traversed at a speed of 1% of the maximum possible speed (MD23).

Other function options

The particulars of backlash compensation can be defined with MD52 and MD53.

MD	Function	Value/Description	Unit
52	Speed for backlash compensation	0 Backlash traversing speed 1 % of MD23 1 to 1 00 Backlash traversing speed in % of MD23	[%]
53	Mode for backlash compensation	0 = Backlash compensation prior to positioning 1 = Backlash compensation during positioning The backlash traversing speed is added to the programmed speed for the traversing job affected by override.	–

Note the following carefully:

- Backlash travel is not affected by override, and is uninterruptible.
- If the interdependencies of the MDs are violated, an error is reported with ”range violation” as its cause.
- Backlash compensation goes into effect prior to reference point approach.
- A change in MD30 (backlash) takes effect only after a traversing movement in a backlash-free direction (MD31).
- In ”Open-loop control” mode, backlash is traversed with the preset voltage rating.

Offset compensation

The analog modules in the positioning control loop (D/A converter of the FM354 and closed-loop controller module of the drive) cause a zero error because of operating-voltage and component tolerances. The result is that at an internal digital rotational-speed specification of zero in the FM 354, the drive motor will already be running undesirably. As a rule, drive controllers have adjustment capabilities for balancing. But by setting a voltage offset via MD44 the analog system can be balanced at startup from the FM side.

MD	Designation	Value/Meaning	Unit
44	Offset compensation	–1,000...+1,000	[mV]

For calculation of the offset value, see Section 7.3.2, Drive interface.

Direction alignment

MD19 allows you to align the direction by defining an assignment between the voltage sign of the manipulated signal and the axis movement.

MD	Designation	Value/Meaning	Unit
19.1	Direction adjustment	1 = invert analog value	–

Voltage ramp

A ramp–shaped voltage rise/drop can be defined in MD45 for the voltage output to the drive when the position controller is inactive. This serves to limit acceleration and thus power for the drive controller, and is preferable to setting options that may be available on the drive, since it has no adverse effects on active position control.

The voltage ramp is active in the following situations:

- Continuously in Control mode
- Deceleration on cancelation of the drive enable [AF] (see Section 9.1.1)
- Deceleration on transition of CPU from RUN to STOP
- Deceleration on error response “Everything Off” (see Sections 11.1, Tab. 11-4 and 11-5)
- Class 3 errors/Nos. 62, 65 and 66 (see Table 11-7)

MD	Designation	Value/Meaning	Unit
45	Voltage ramp	0...10,000,000	[mV/s]

D/A converter

Output of analog voltage setpoint

Velocity assignment and max. voltage setpoint

The manipulated signal calculated by the position controller is available internally on the FM as a velocity setpoint (see position loop gain). To convert this value to the analog actuating signal, a conversion factor (DAC factor) within the FM is necessary. This factor is formed as the quotient of MD43 and MD23. MD23 contains the configured maximum speed of the machine axis, and MD43 contains the voltage setpoint of the actuating signal to be output by the FM 354 for this purpose; as a compromise between the highest possible resolution and adequate close-loop control reserve, this voltage should lie between 8 V and 9.5 V.

Warning

This assignment MUST be identical with the setting on the drive!

MD	Designation	Value/Meaning	Unit
23	Maximum speed	10...500,000,000	[MSR/min]
43	Set voltage, max.	1,000...10,000	[mV]

9.8 Digital inputs/outputs (Read request in the user DB, DBX43.4)

Overview

Four digital inputs and four digital outputs of the FM 354 can be used specifically to a given application.

The conventions and parameterization for this purpose are defined in the machine data MD34 to MD36.

The signals are processed in the FM cycle.

The signal status of the digital inputs and outputs can be recognized by readback (user DB, DBX43.4).

Function parameters

Table 9-13 shows you the functions assigned to each digital I/O.

Table 9-13 Function parameters for digital I/Os

MD	Designation	Data type, bit array/meaning				
34	Digital inputs[1]	**I0**	**I1**	**I2**	**I3**	
		0	8	16	24 =	External start[2]
		1	9	17	25 =	Enable input
		2	10	18	26 =	External block change
		3	11	19	27 =	Set actual value on-the-fly
		4	12	20	28 =	Measurement (inprocess measurement, length measurement[2])
		5	13	21	29 =	Reference point switch for reference point approach[2]
		6	14	22	30 =	Reversing switch for reference point approach[2]
35	Digital outputs[1]	**Q0**	**Q1**	**Q2**	**Q3**	
		0	8	16	24 =	Position reached, stop
		1	9	17	25 =	Aaxis movement forwards
		2	10	18	26 =	Axis movement backwards
		3	11	19	27 =	Change M97
		4	12	20	28 =	Change M98
		5	13	21	29 =	Start enable
		7	15	23	31 =	Direct output

[1] see Section 5.3.1, Dependencies

[2] Signal length ≥ 2 · FM cycle

Level adjustment

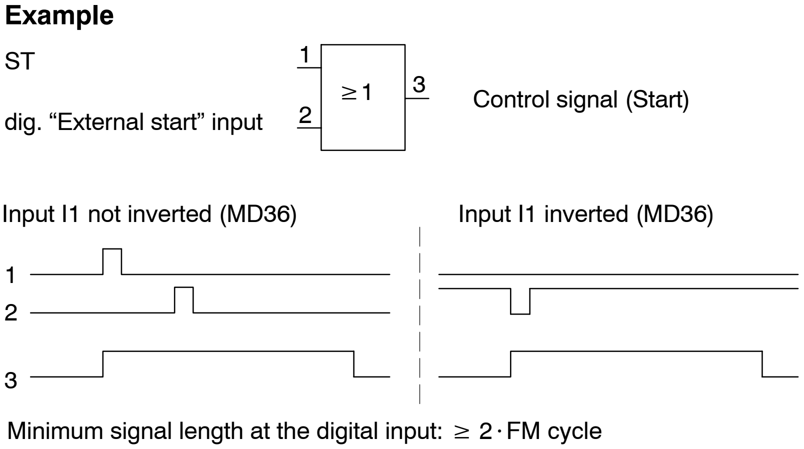
MD	Designation	Value/Meaning	Comments
36	Input adjustment	8 = I0 inverted 9 = I1 inverted 10 = I2 inverted 11 = I3 inverted	Front edge always activates the function

9.8.1 Function description of digital inputs

External start

The control signals of the axis include the start signal which triggers a positioning operation in “Reference point approach”, “MDI” and “Automatic” modes. A logical OR is established with the “External Start” digital input and the control signal (ST).

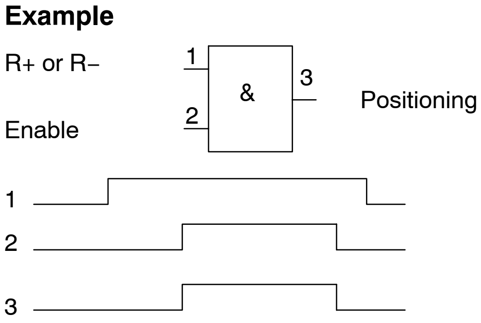
External start is connected to digital input I1.


Example
ST
dig. “External start” input
1
2
≥1
3
Control signal (Start)
Input I1 not inverted (MD36)
Input I1 inverted (MD36)
1
2
3
Minimum signal length at the digital input: ≥ 2 · FM cycle

Enable input

The enable input signal must be set, if defined in MD34, for a positioning operation/movement/output of the axis to take place. A reset stops the movement (external movement enable).

- In the Jogging and Control modes, the movement of the axis proceeds as long as the AND link continues between the control signal (R+/R−) and the enable input.



- In the other modes, note the following:

  If the enable input is still not enabled after a start edge, the start edge is stored internally and “waiting for enable” is indicated by the checkback signals. When the input is set, movement begins and the stored Start edge is deleted (a Stop likewise deletes the stored Start edge).

External block change

see Chapter 10

Set actual value on-the-fly

see Chapter 10 and Section 9.3.6

Measurement

see Section 9.3.10

Reference point switch for reference point approach

see Section 9.2.3

Reversal switch for reference point approach

see Section 9.2.3

9.8.2 Function description of digital outputs (Write request in the user DB, DBX39.4)

Output of PEH, FR+, FR–, SFG

The following checkback signals: position reached, stop (PEH), axis movement forward (FR+), axis movement in reverse (FR–), and enable Start (SFG), are additionally output via digital outputs. The output assignment is parameterized by way of MD35.

Output of change M97 or M98

The change M-function (AMF) checkback signal for the M functions M97 and M98 is output as a digital output. It allows these M functions (switching signals) to be applied without being delayed by the user cycle time.

Direct output

Outputs Q0...Q3 (D_OUT1...D_OUT4), which are defined in MD35 as “direct output”, can be used directly by the user program (user DB, DBX39.4) and can also be controlled by the FM 354.

Since the same memory is used in the user DB for Write request (user DB, DBX39.4) and Read request (user DB, DBX43.4), the jobs cannot be used simultaneously in the cycle.

Note

The outputs are subject to deactivation on module errors of error classes with the response “Everything Off”.

9.9 Software limit switches

Overview

To limit the working range, entries in the machine data (MD21 and MD22) specify the start and stop limit switches. These limit switches are active at synchronization of the axis.

If the limit switches are not needed, values lying outside the possible working range should be entered in the machine data (MD21 and M22), or monitoring should be switched off via the user program.

Warning

The software limit switches do not replace the hardware limit switches for EMERGENCY STOP responses.

Effect of software limit switches in modes

“Jogging mode”

At the limit switch the traveling movement is stopped in the limit-switch position, and an error is signaled.

“Control mode”

If the actual value is beyond the end position, the traveling movement is stopped and an error is signaled. The limit-switch position is overshot by the amount of the necessary deceleration distance.

“Reference-point approach” mode

No effect.

“Incremental relative”, “MDI”, “Automatic mode”

Movement is stopped, or not even started, as soon as read-in of the set position reveals that the position lies outside the working range. An error is signaled.

The following special cases exist:

- Continuous travel (–) for set actual value on-the-fly (G88 see Chapter 10)
- Continuous travel (+) for set actual value on-the-fly (G89 see Chapter 10)

Effect of software limit switches in tracking mode

If the actual value is beyond the end position, an error is signaled.

Response after error

Leaving end position or traveling into working range after error

1. Acknowledge the error message!
2. Travel to the working range with the “Jogging”, “Control”, “Incremental Relative” or “MDI” mode.

Rotary axis

The end position of MD_{start} may be greater than $MD_{stop.}$

When traveling into the working range (e.g. end position was previously switched off), the shortest path is always chosen.

If both default values are parameterized the software limit switches are inactive.

9.10 Process interrupts

Overview

Process interrupts are interrupts that quickly signal states in the current process to the user program.

The appropriate setting in the machine data (MD5) specifies which signals are to be quickly communicated to the user program.

Process-interrupt generation

The process interrupt is generated by way of machine data item MD5:

MD	Designation	Significance
5	Process-interrupt generation (data type - bit field)	0 = Position reached 1 = Length measurement completed 3 = Change block on-the-fly 4 = Inprocess Measurement

Hint to the user

You must program the interrupt processing routine in OB40.

The prerequisite is that process-interrupt signaling must have been activated as part of the environment definition (see Chapter 5).

■

Writing Traversing Programs 10

Chapter Overview

Overview

To execute the desired operations of the machine axis (sequence, position, etc.) in "Automatic" mode, the FM 354 needs certain information. This information is programmed with "Parameterize FM 354" (traversing program creation) in the form of a traversing program, based in principle on DIN 66025.

Traversing programs

Each traversing program is stored under a program number.

A traversing program consists of not more than 100 traversing blocks.

The program number and traversing blocks are converted to an internal format (see Section 9.3.12), are stored in the appropriate data block, and are transferred to the module. Where they are administered.

The possible number of programs depends on the amount of memory available (max. 16 Kbytes) and on the length of the individual programs.

Program length in bytes: 108 + (20 x no. of traversing blocks)

Program name

Any program can be assigned a name (optional).

The program name may have up to 18 characters, and is saved with the program.

Program number

Programs may be numbered from % 1 to % 199.

Traversing block

A traversing block contains all the data required to perform a machining step.

Program structure

A program consists of several blocks. Each block number occurs only once, and numbers are arranged in ascending order.

A sample program structure follows:

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D	L	P	
	5	90			500 000	100 000	10						Start of program = lowest block number
	6	91			...	...							
	7	...											
	:												
	45												
	46						2						End of program = M2 or M30

10.1 Traversing blocks

Block structure

The following Figure gives you an overview of the structure of traversing blocks.

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D	L	P

/ – Identifier for skipped block
N – Block number
G1 – G function of first function group
G2 – G function of second function group } see Table 10-1
G3 – G function of third function group
X/t – Position/dwell time
F – Speed
M1 – M function of first function group
M2 – M function of second function group } see Table 10-2
M3 – M function of third function group
D – Tool offset number
L – Call a program as a subprogram
P – Number of subprogram runs

Skip block /

Program blocks which are not to be executed every time the program runs can be identified as skippable blocks by an oblique "/ ". When the program is being processed, the "Skip block" control signal can be used to decide whether skippable blocks are to be skipped. The last block may not be skipped.

Block number N

The program is executed in ascending order of block numbers, from 1 to 255, or in descending order if executed in reverse.

G function group 1...3

In each traversing block only one G function may be entered from each G function group.

The following figure shows an example.

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D
	10	90	34	43	100 000	400 00				10

G functions

Table 10-1 lists the possible G functions and the individual G function groups.

Table 10-1 G functions

G No.	G Function	G Function group
04[1]	Dwell time	1
87	Turn off measuring system shift for Set Actual Value On-the-Fly	
88[1]	Continuous travel for (–) for Set Actual Value On-the-Fly	
89[1]	Continuous travel for (+) for Set Actual Value On-the-Fly	
90	Absolute measure	
91	Increment size	
30	100% override on acceleration/deceleration	2
31	10% override on acceleration/deceleration	
32	20% override on acceleration/deceleration	
..	..	
39	90% override on acceleration/deceleration	

[1] These G functions take effect only on a block-by-block basis. The other G functions remain active until canceled explicitly.

Table 10-1 G functions, continued

G No.	G Function	G Function group
43	Tool offset (+)	3
44	Tool offset (–)	
50[1]	External block change	
60	Block change – exact positioning	
64	Set actual value on-the-fly, continuous-path mode	

[1] These G functions take effect only on a block-by-block basis. The other G functions remain active until canceled explicitly.

G30, G90 and G64 are the **initial settings** after the start of the program.

Dwell G04

A traversing block with dwell can only contain M functions and the time parameter apart from this G function.

The following applies for dwell time:

Name	Lower input limit	Upper input limit	Unit
Dwell time	2	100 000	ms

Input values are rounded upwards in FM cycle steps (2ms). Dwell times take effect block after block only.

Block change G60, G64 (approach conditions)

With G60, the exact programmed position is approached and the feed movement is stopped (exact stop block change).

G64 causes the next block to be processed immediately as soon as the point of deceleration is reached (change block on-the-fly).

G60 and G64 are mutually exclusive and self-maintaining.

M commands have no effect on G64 operation.
(For a detailed description, see Section 10.3).

External block change (G50) with delete residual path

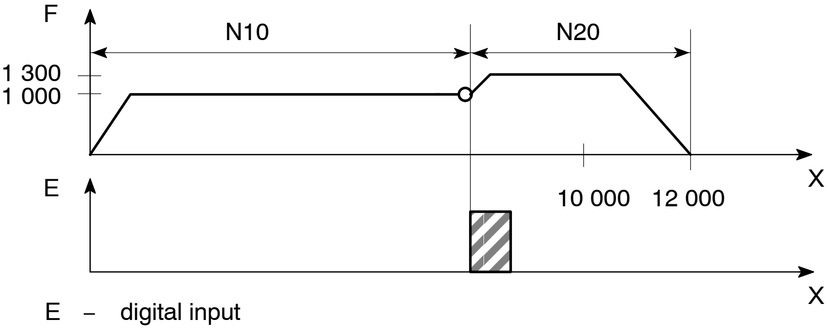
The “external block change” function causes a block change on-the-fly triggered by a digital input. The fast input must be parameterized with the “External block change” function by way of machine data item MD34.

The function takes effect only on a block-by-block basis (no effect on G60 and G64).

Example of external block change

The following figures show the program structure and program flow of an example of “External block change.”

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D
	10			50	10 000	1 000				
	20				12 000	1 300				



Notes to the example of external block change

The axis travels until a signal change from 0 to 1 takes place at the digital input. This triggers two reactions:

- A block change on-the-fly, and thus immediate processing of block N20.
- Storage of the actual position at the time of this signal change to “Actual value block change.” This position is also the starting position for any subsequent chain-measure programming.

Depending on the situation, N20 is processed as follows:

- If the block position in N20 is less than the actual position at the time when the digital input is received (reversal of direction), the equipment is stopped so that the position can then be approached in the opposite direction.
- If no position is programmed in block N20, movement is braked, the functions programmed in N20 are executed, and processing then moves on to the next block (except if the block contains M0, M2 or M30).
- If the programmed path in block N20 is less than the deceleration distance, the programmed position is overshot and then positioned by a reversal of direction.

If no signal change occurs at the digital input, the target position of N10 is approached, with the following additional response:

When the target position is reached, the error message “Digital input not actuated” is output (see Table 11-5, Class 2 No. 15).

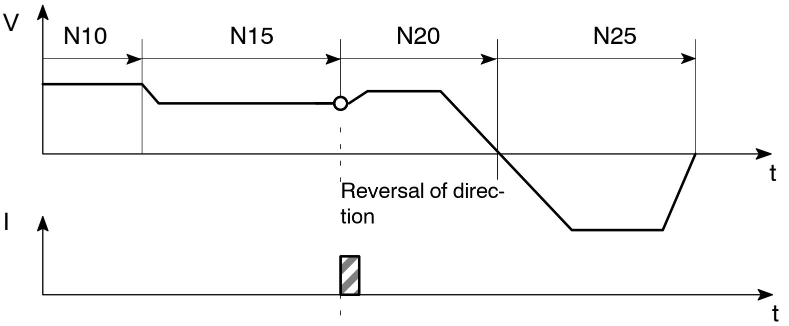
Set actual value on-the-fly G87, G88, G89

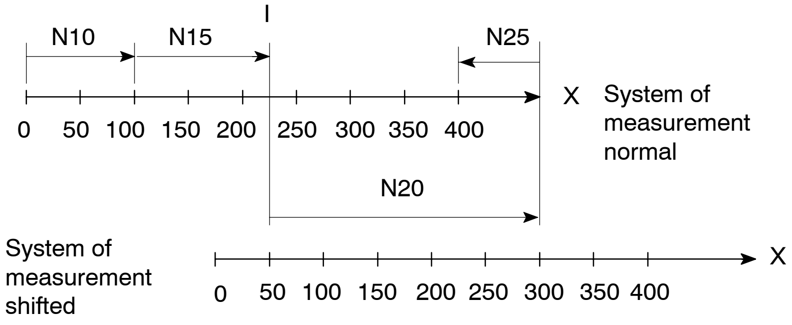
The “Set actual value on-the-fly” function is programmed and triggered by a digital input; the block change occurs on the fly and the actual value is set to a new dimension (programmed coordinate) at the same time. The digital input must be parameterized with the “Set actual value on-the-fly” function by way of machine data item MD34.

Example of set actual value on-the-fly

The following figures show the program structure, program flow and actual-value curve for an example of “Set actual value on-the-fly.”

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D
	10	90			100	400 000				
	15	89 (88)			50	200 000				
	20	90			300	400 000				
	25	87			400	400 000				





I – digital input

Notes to the example of set actual value on-the-fly

This changes blocks on-the-fly from N10 to N15, with G89 causing movement in a positive direction and G88 causing movement in a negative direction at the speed programmed in N15.

The axis now travels in the specified direction until a positive edge change occurs at the digital input. This triggers the following responses:

- Block change on-the-fly and immediate processing of block N20
- Set actual value on-the-fly to the block position from N15 (50 in the example), and resulting shift of the coordinate system
- Save current actual value.

The programmed position in block N20 refers to the shifted coordinate system.

At the block change from N20 to N25, G87 cancels the shift of the coordinate system and causes reference-measure programming to the block position of N25.

The saved actual value can be read out with “Actual value block change.”

The shift of the coordinate system is maintained until it is canceled by G87 or by a mode change. It is possible to use the existing shift of the coordinate system in different programs. The coordinate system can be shifted again without previously canceling an existing coordinate system shift.

G88, G89 can be programmed multiple times. The shift in each case refers to the original state. The software limit switches are always shifted concurrently.

If the signal change of the digital input does not occur, the axis runs until it reaches the limit switch.

Note

The G functions G87, G88 and G89 take effect only on a block-by-block basis and must be reselected if necessary.

Dimensions G90, G91

The traversing movement at a specific point can be described by

- Reference-measure input (absolute measure input) G90 or
- Incremental input (relative measure input) G91

You can switch back and forth at will between reference-measure and incremental input.

The status at startup is reference-measure programming, G90.

G90 and G91 are modal.

Absolute dimensioning G90

Absolute dimensioning is the method used to specify dimensions that generally refer to the coordinate system.

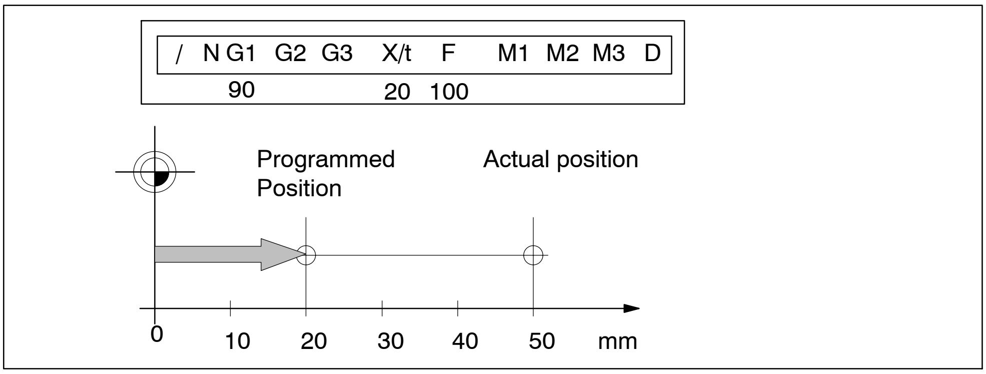


Fig. 10-1 Reference-measure input G90

Note

To ensure precise reproduction of the program, the first block should contain reference-measure programming.

Incremental dimensioning G91

Incremental dimensioning is the method used to specify incremental dimensions that refer to the last actual position.

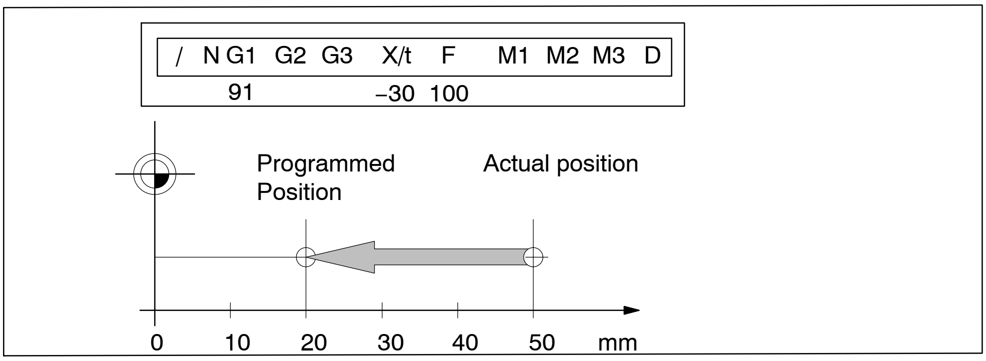


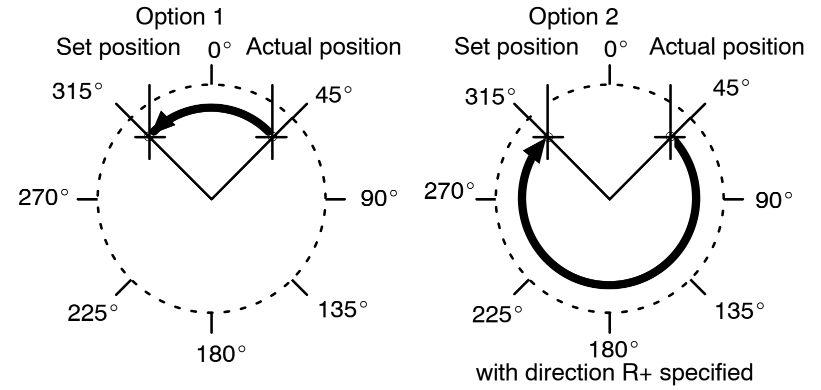
Fig. 10-2 Incremental input G91

Axis as rotary axis

If the axis is operated as a rotary axis, the measuring system must be adjusted in such a way that the measurement scale refers to the full circle (e.g. 0° and 360°).

- Reference-measure input G90

  In a full circle with 360°, reference-measure programming (G90) has the peculiarity that there are always two options for reaching the set position.



/	N	G1	G2	G3	X/t	F	M1	M2	M3	D
	10	90			315	100				

Fig. 10-3 Rotary axis

Option 1:

With G90, the axis autonomously always takes the shortest path to reach the set position of 45°, going via 0° to 315°.

Option 2:

The control signals (R+) or (R–) force the respective direction of the axis - in this example 45° via 180° to 315°. (R+) or (R–) must already be pending when positioning is activated (START).

Note

The direction (R+) or (R–) must be specified sufficiently in advance. A traversing direction **cannot** be forced on a traversing block that is already active, or on the traversing blocks (up to 4) that have already been calculated in advance in G64 operation.

Operation with option 1 or option 2 is at the user's discretion.

- Incremental input G91

  With incremental programming G91, the direction of rotation of the rotary axis is defined by the sign of the position setpoint. Multiple rotations can be programmed by setting a value > 360° as the position setpoint.

Acceleration override G30...G39

The acceleration override is used to control acceleration and deceleration during positioning movements. The acceleration and deceleration values are set by machine data. G30 through G39 in the traversing block can be used to achieve a percentage reduction in both values. These functions are modal.

G	Function
30	100% override for acceleration/deceleration
31	10% override for acceleration/deceleration
to	
39	90% override for acceleration/deceleration

Changing the acceleration override in the program prevents block change on-the-fly. Consequently G60 response is forced in the preceding block.

The acceleration override is turned off by:

- Mode changes
- Resetting the axis with a Restart (single command)
- Changing or ending the program.

Tool compensation G43, G44

Tool compensation allows you to continue using an existing machining program, even when the tool dimensions have changed.

Tool offset is selected with G43 or G44, as applicable, and the tool offset number D1...D20. Tool offset is turned off with G43 or G44, as applicable, and the tool offset number D0.

A total of 20 tool offset storage areas and tool wear storage areas are available. The values are loaded to the module with the “Tool offset data” data block and are saved permanently. When selected, changed or turned off, the tool offset is not taken into account until the next positioning action.

A selected tool offset is maintained in effect until it is either turned off or replaced with a new one. Likewise a mode or program change, or the end of a program, will turn tool offset off.

Variants in tool offset

Tool offset is made up of two correction-value components:

- Tool length offset

  The tool length offset is the actual tool length from tool zero to the tool tip.

- Tool length wear value

  The tool length wear value allows the change in tool length due to wear to be compensated in two ways:

  Absolutely: by specifying a fixed wear value

  Additively: by adding an “offset value” to the current tool length wear value contents.

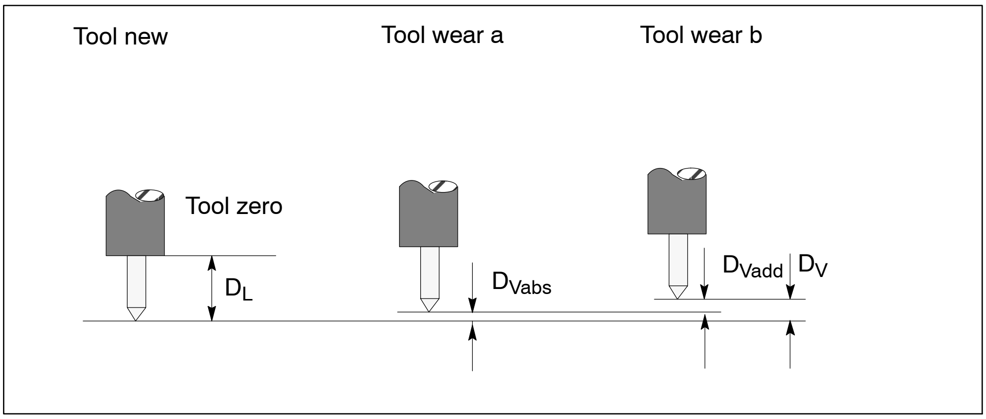


Fig. 10-4 Tool offset

Notes to the figure:

The tool offset thus consists of the tool length offset and the tool length wear value:

$D = D_L - D_V$

$D_V = D_{Vabs} + D_{Vadd}$

D – Tool offset
D_L – Tool length offset (positive or negative)
DV – Tool length wear value (positive or negative)
D_{Vabs} – Wear, absolute (positive or negative)
D_{Vadd} – Wear, additive (positive or negative)D_{Vadd}

If the additive wear is modified on–line, the FM computes the new absolute wear and resets the additive wear to zero.

Direction of tool offset

The functions G44 (–) and G43 (+) correct the position value in such a way that the tool tip reaches the programmed set position.

- **Negative tool offset G44**

  As a rule, the tool points to the workpiece in a negative direction. With the infeed adjustment, the positioning value (traversing path) becomes smaller.

  Referred to the measuring system, the following position is thus approached:

  $X_{ms} = X_{set} + (D)$

  X_{ms} – Position of measuring system
  X_{set} – Programmed set position
  D – Tool offset

- **Positive tool offset G43**

  The positioning value (traversing path) becomes greater with the infeed adjustment. The position value is corrected by:

  $X_{ms} = X_{set} - (D)$

To program a tool offset in the traversing block, at least the tool length offset must be input. If no correction is to be applied even when the function has been selected, the tool length offset and tool length wear value must be preset to 0.

A tool length wear value can be deleted by an absolute input of 0.

Position X

Positions may be input with a negative or positive sign. The plus sign on positive values may be omitted.

Name	Lower input limit	Upper input limit	Unit
Position	– 1,000,000,000	+ 1,000,000,000	MSR from MD7

Speed F

The input speed is calculated against the override. If the speed value is numerically greater than the maximum allowed speed, it is limited to the magnitude of the machine data item. Speeds are self-maintaining and need to be re-input only when changed.

Name	Lower input limit	Upper input limit	Unit
Speed	10	500 000 000	MSR from MD7/min

M functions

Up to three M functions can be programmed in one traversing block, with any assignment of M1, M2 and M3. The output sequence of the M functions is always M1→M2→M3 (for information about output see Section 9.1).

The following figure shows an example.

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D
	10	90	34	43	100 000	400 00	10	11	12	1

Table 10-2 M functions

M No.	M Function	M Function group
0	Stop at end of block	1, 2, 3
2, 30	End of program	
1, 3...17	User functions	
18	Endless loop (skip back to start of program)	
19...29, 31...96	User functions	
97, 98	Change signal programmable as digital output	
99	User functions	

M0, M2, M18 and M30 are always output at the end of the traversing movement.

M0, M2, M18 and M30 are mutually exclusive within a single block.

Stop at end of block M0

If M no. 0 is programmed in a traversing block, the program stops at the end of the traversing block and M0 is output. Only a new START edge causes the traversing program to be continued.

End of program M2, M30

If M2 or M30 is programmed in a block, then after positioning is complete the M function is output with a subsequent programmed stop and a jump back to the start of the program. The Start edge can restart the program. M2 or M30 is always the last output in the block.

If the program is called up as a subprogram, the action skips to the main program. In this case M2 or M30 is not output.

Infinite loop M18

M18 is always output as the last M function in the block.

Two cases are distinguished:

- M function M18 is output like any other M function. Only after the block has been processed all the way to the end (including M18) does the axis skip back to the start of the program.
- If M function M18 is programmed alone in the last block of a traversing program, the M function is not output, and the axis immediately skips back to the start of the program.

Change signal programmable as digital output M97, M98

If M97 or M98 is programmed in a block, the M function output proceeds via the digital outputs as defined in machine data item MD35, in the same way as the checkback signals.

Tool offset number D

Twenty tool offset numbers (D1 – D20) are available. D0 in conjunction with G43 or G44 causes the tool offset to be switched off. The offset values must previously have been loaded to the module. Nonstandard offset values have a value 0.

Subprogram call P, L

A block with a subprogram call (P is the “number of runs”, L is the “program number”) cannot contain any further information.

A maximum of 19 subroutines can be called up in one program. Nesting is not possible.

Name	Lower input limit	Upper input limit
P = Number of subprogram runs	1	250

10.2 Program execution and direction of machining

Forward processing

As a rule, programs are processed by ascending block number.

Reverse processing

If programs are processed in reverse, the effects of commands must be taken into account in the programming:

- Commands are self-maintaining (G90, G91, G60, G64, G30...G39)
- Active tool offset (G43, G44, D0...D20)
- Change of coordinate systems via G87, G88, G89.

For these reasons, a distinction can be made between forward processing and reverse processing, in terms of both geometry and block transition response.

10.3 Block transitions

Overview

This chapter describes the influence of certain commands on block transitions.

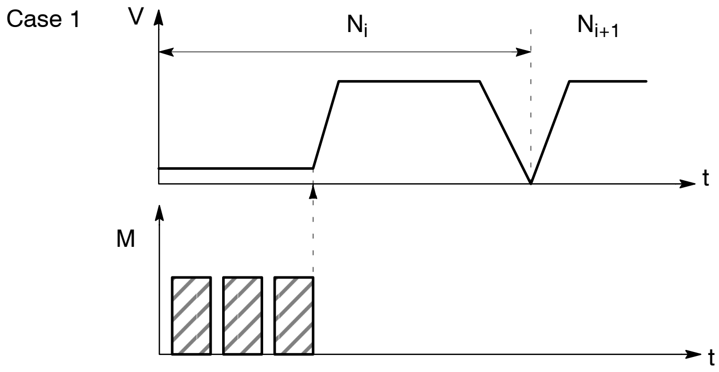
Exact positioning - G60

G60 mode is overlaid with G50 and G88 to G89 (force block change on-the-fly).

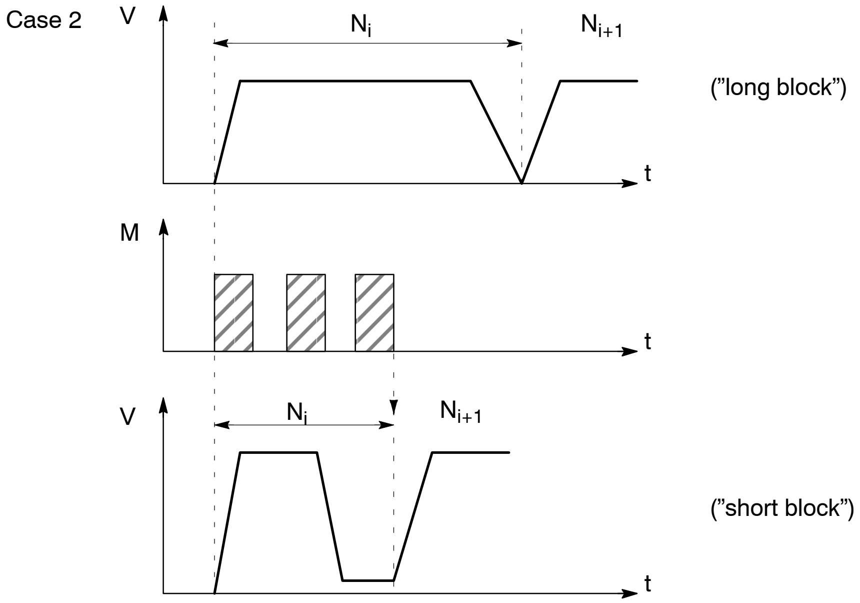
The program advances to the next block when the target range is reached.

The influence of M functions is as indicated in machine data item MD32.

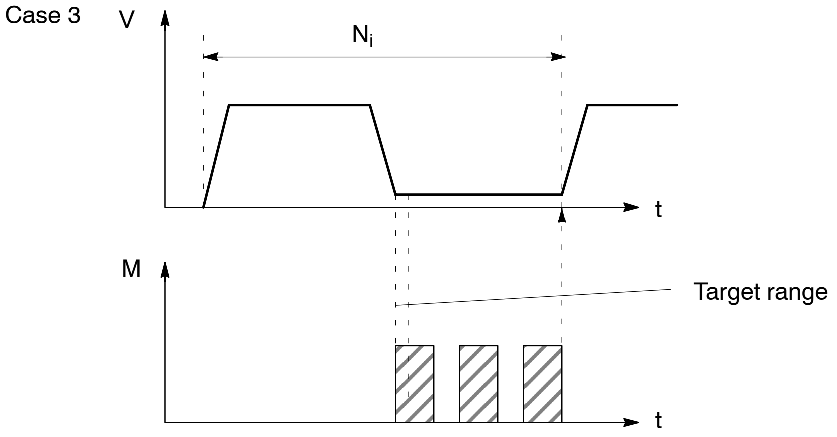
Output of M function before positioning



Output of M function during positioning



Output of M function after positioning



Change block on-the-fly - G64 (standard case)

Changing from one traversing block to the next proceeds without stopping the axis.

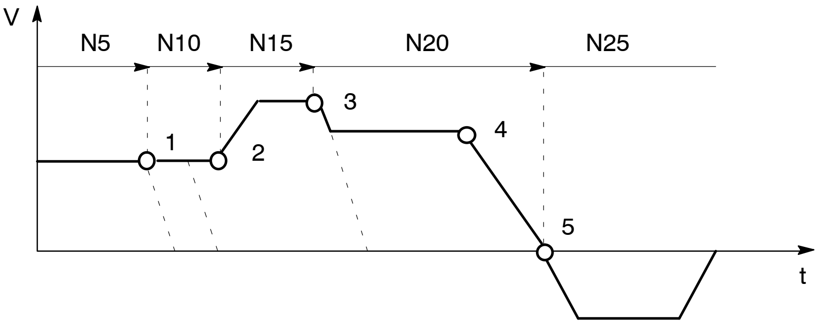
The acceleration and braking function is calculated for multiple blocks when the G64 function is programmed. The number of blocks processed in advance is three.

When the block changes, the feed rate is changed in such a way that a higher speed from a preceding block is never carried over into the next block, and a higher speed from a following block never goes into effect while a given block is still traversing its own path. This means that acceleration does not begin until the starting point of the block, and deceleration to a lower speed for a following block is initiated as with G60. When the speed of the following block is reached, the residual distance in the current block is processed at the feed rate of the following block.

Sample programming (standard case)

The following figure shows a sample program with the programming flow.

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D
	5	90		64	10 000	100 00				
	10				20 000					
	15				30 000	200 00				
	20				40 000	150 00				
	25			64	30 000	100 00				



1 – Block N10 is started at the point of deceleration of N5.

2 – N15 is started at the point of deceleration of N10. Acceleration to the higher traversing speed begins when the set position of N10is reached.

3 – N20 is started at a lower traversing speed at the braking point of N15.

4 – In a change of traversing direction, the axis brakes until it comes to a standstill and waits until the actual value of the encoder has reached the target range.

5 – When the target range is reached, the axis accelerates in the opposite direction up to the traversing speed of the new block.

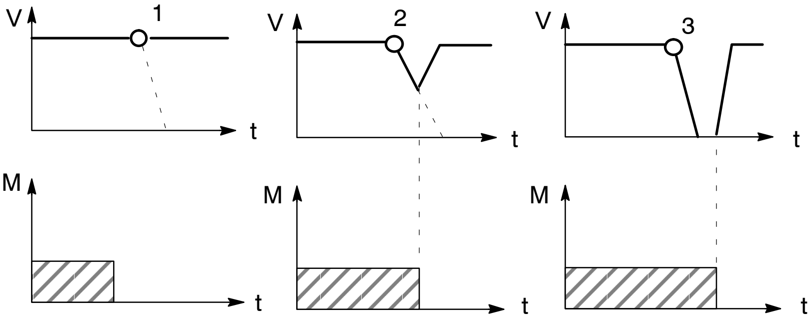
To be able to approach a position correctly, the axis must calculate the point of deceleration. The relevant parameters for this calculation are the residual traversing distance, the deceleration value and the current traversing speed.

The point of deceleration is also the earliest possible time for a block change.

Change block on-the-fly - G64 (deceleration)

There are a number of conditions that may delay or prevent a block change on-the-fly. Here a distinction is necessary between the case in which this type of block change is suppressed intentionally, and the case in which the selected function does not permit a block change on-the-fly.

- Block change on-the-fly is suppressed:
  - By removing the Enable read-in control signal - this stops program processing at the end of the current block. To continue the program, the enable must be re-input.
  - By output of the M function before or after positioning.
  - By M function M0 (stop at end of block). To continue the program, the START control signal must be reset.
  - By a block with a dwell time.
  - By processing a program in the Automatic/Single Block mode. Each block must be activated individually.
  - By a change in the acceleration override.
- Functions that themselves prevent block change on-the-fly:
  - M functions (during positioning).



1 – Since the M output is completed at the point of deceleration, a block change takes place on-the-fly.

2 – The M output is not yet complete at the point of deceleration. The axis begins to brake. At the end of the M output, the axis returns to speed (transition on-the-fly from deceleration ramp to acceleration ramp).

3 – Axis comes to a complete standstill and waits for the end of M output.

Influence of M function on block change on-the-fly

Machine data can specify the output time for M functions:

- M function is output before or after positioning with a block change

  M-function output and positioning proceed in alternation.

  – M function output before positioning causes exact-positioning response in the preceding block.

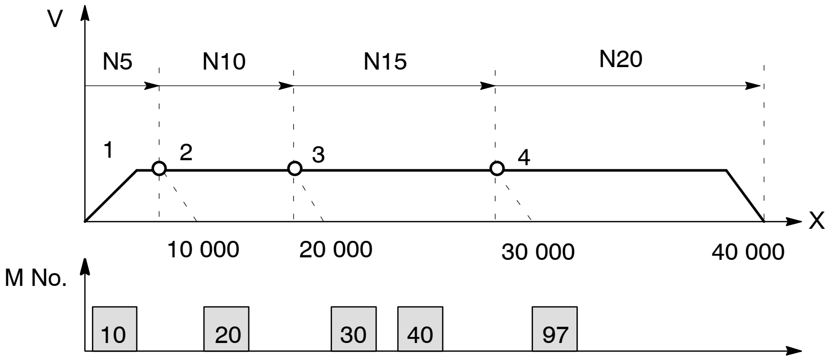
  – M function output after positioning causes exact-positioning response within the block.

- M function is output during positioning

  M-function output and positioning proceed simultaneously.

  The following figure shows a sample program with M function output “during positioning”.

/	N	G1	G2	G3	X/t	F	M1	M2	M3	D
	5	90			10 000	100 00	10			
	10				20 000		20			
	15				30 000		30	40		
	20			60	40 000				97	



1 – Output of M10 is **not** position-dependent, since no relevant position for a position-dependent M function is present.

2 – At the block change from N5 to N10, output is prepared. The axis the M function does not proceed until the actual position has reached the programmed position of N5.

3 – If two M functions are programmed in a traversing block, the first M function is output depending on position, followed by the second M function.

4 – The change signal for M97 or M98 is output with the G64 block transition (digital output) if the actual position has reached the programmed position of the block. The actual position runs behind the set position (difference = overtravel).

Troubleshooting

11

Chapter Overview

Overview

The FM 354 provides diagnostics for the following:

- I/Os
- Module processes

This chapter “Troubleshooting” describes the different types of errors, their cause, effect and elimination.

Error localization

The FM 354 distinguishes between

- Errors which trigger a diagnostic interrupt in the CPU, and
- Errors which the module reports by way of checkback messages.

In the event of diagnostic interrupt, STATUS LEDs light up in addition.

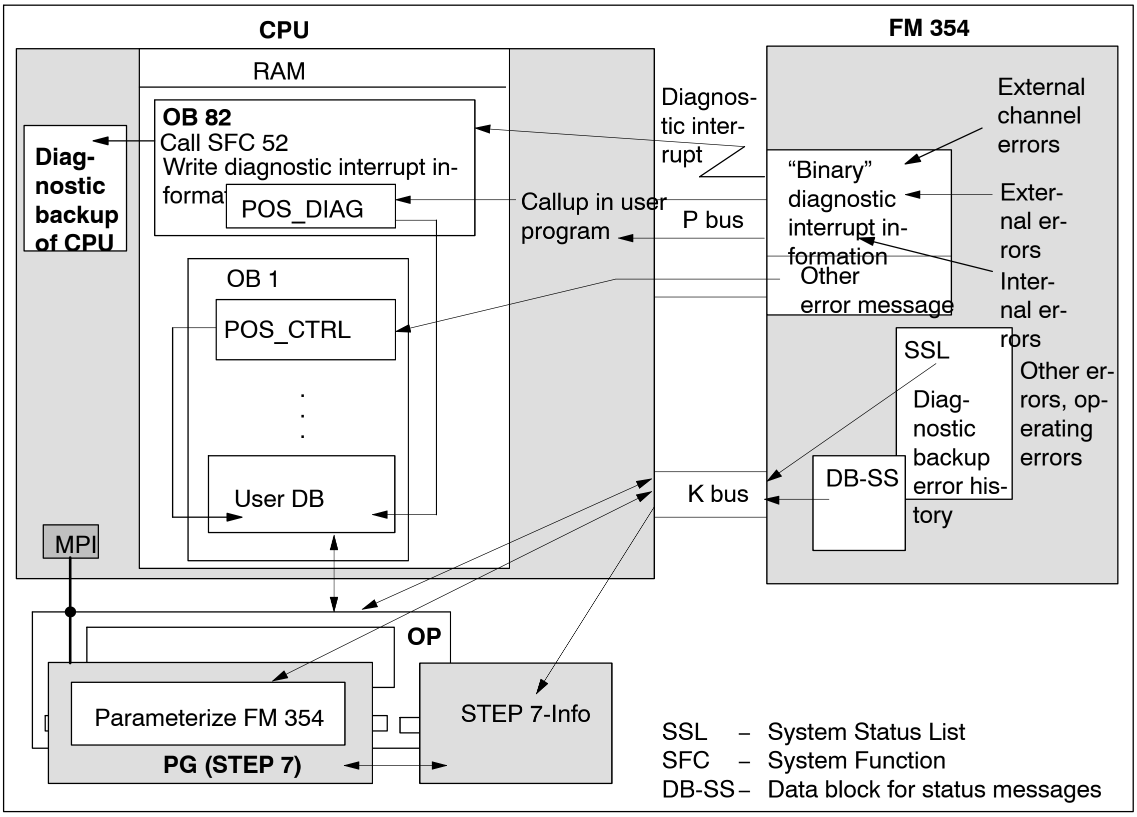


Fig. 11-1 Overview of diagnostics/errors

Error evaluation by software

The following manuals describe how to include diagnostics-capable modules in your user program, and how to evaluate the diagnostic messages:

- Programming manual *System Software for S7-300/400; Program Design* (OB Types, Diagnostic Interrupt OB 82)
- Reference manual *System Software for S7-300/400; System and Standard Functions*

A basic description of the diagnostic system of the S7-300 can be found in the user manual *Standard Software for S7 and M7, STEP 7*.

11.1 Error classes and module responses

Overview

The FM 354 contains monitoring circuits which are active during startup or during continuous operation. Errors occurring during those times are reported to the system and to the user program.

The table below lists the error classes and their meaning.

Table 11-1 Error classes, overview

Message	Error Class	Response	Significance
Diagnostic interrupt	Internal errors	Everything OFF	...are hardware faults in the module which are discovered by diagnostics routines (e.g. memory errors). (see Section 6.3.4 for diagnostic interrupt data and error list, Table 11-4)
	External errors		...are errors which can occur due to faulty module connection (e.g. parameters for initialization of module MP station numbers are missing or are incorrect). (see Section 6.3.4 for diagnostic interrupt data and error list, Table 11-4)
	External channel errors		...are measurement system errors or errors which can occur by connecting the digital outputs or during operation (operating errors) of the FM 354 (e.g. cable break, incremental value encoder). (see Section 6.3.4 for diagnostic interrupt data and error list, Table 11-4 and 11-5)
Checkback signals	Operator control and travel errors	Feed STOP	... are errors (operator/travel errors) which can occur during operator inputs to the FM 354 (e.g. direction signals R+ and R– set at the same time, see Error List, Tables 11-6 and 11-7).
	Data errors	Warning!	... are errors (data, machine data and traversing program errors) which are detected on interpretation of invalid data (see Error List, Table 11-8).

Error response

Each error message triggers an appropriate response.

Table 11-2 Overview of internal error responses

Error response	Significance
Everything OFF	• Stop movement by way of voltage ramp (MD45) • Disable digital outputs • Deactivate servo enable • SYN is cleared after the error has been acknowledged with Restart • No new travel jobs possible
Feed STOP	• Stop movement by controlled deceleration • Travel job is canceled and terminated. • Measured data acquisition and position control are continued. • No new travel jobs possible
Warning	• Message only • Movement and control of axes not affected

11.2 Error messages

Introduction

The following approaches to error localization are available for the FM 354:

- Error display by LEDs
- Error messages to the system and to the user program

11.2.1 Fault indication by LED

Status and error displays

The FM 354 features the following status and error displays:

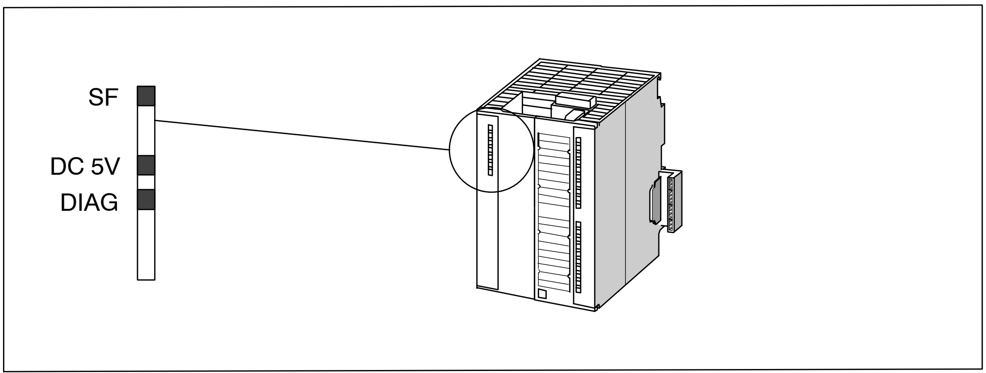


Fig. 11-2 Status and error displays of the FM 354

Significance of the status and error displays

The status and error displays are explained in the order in which they are arranged on the FM 354.

Table 11-3 Status and error displays

Display	Significance	Explanations
SF (red) LED-ON	Group errors	This LED indicates an error condition in the FM 354. Diagnostic interrupt (internal error, external error or external channel error) To eliminate the error see Error List, Table 11-4.
5 V DC (green)	Power supply switched on	This LED indicates that the hardware is ready for operation. If not illuminated, this may indicate one of the following conditions: • Your line voltage is not OK. • Load current supply is faulty. • Module is incorrectly connected. • Your control circuit is configured incorrectly (sum of rated and starting currents is too great), or • The module is faulty.
DIAG (yellow) LED-ON LED-blinking	Diagnostics	This LED indicates the various diagnostic conditions. Diagnostic interrupt (external channel error). To eliminate the error see Error List, Table 11-4. If this LED blinks when the LED “SF” is simultaneously activated, this indicates a system error. If this occurs, then please consult the appropriate sales department. The exact circumstances which resulted in the error are of major importance in this case.

11.2.2 Diagnostic interrupts

Overview

Internal errors, external errors and external channel errors are indicated to an interrupt-capable system by means of diagnostic interrupts (see diagnostic interrupt data in Tables 11-4, 11-5 and Section 6.3.4). This presupposes that the diagnostic interrupt message was activated at the time of configuration (see Chapter 5). If the system is not interrupt-capable, the diagnostic interrupt information must be read out cyclically with POS_DIAG.

Error Class	Coding	Message
Internal errors	Byte-.Bit-No. 0.1 Group error byte 2, 3	LED “SF”
External errors	Byte-.Bit-No. 0.2	LED “SF” and “DIAG”
External channel errors	Byte-.Bit-No. 0.2, 0.3 Group error byte 8	LED “SF” and “DIAG”

A diagnostic interrupt is reported by the FM 354 as ”incoming” or ”outgoing”.

<table>
<tr><th colspan="6">Diagnostic interrupt</th></tr>
<tr><td colspan="4">Message to the CPU (precondition: interrupt message activated (see Section 5.2)</td><td rowspan="3">Message in the “troubleshooting” display of “Parameterize FM 354”
Menu: <b>Test &gt; Troubleshooting</b></td><td rowspan="3">Entry in diagnostic buffer</td></tr>
<tr><td rowspan="2">No OB 82 exists → CPU switches to STOP</td><td colspan="2">OB 82</td><td>OB 1</td></tr>
<tr><td>Enters the diagnostic information in the diagnostic buffer of the CPU (4 bytes) and calls SFC 52</td><td>Enters the diagnostic information in the user DB starting at address 70 and calls POS_DIAG</td><td>Calls POS_DIAG</td></tr>
</table>

Interrupt acknowledgement

If processing is to continue after a diagnostic interrupt, then Restart after the error has been remedied (user DB, DBX37.5).

Internal errors cannot be acknowledged. External errors are self-acknowledging.

11.2.3 Error messages in checkback signals

Overview

Operator/travel errors [BF/FS] and data errors/machine data errors/traversing program errors [DF], are communicated to the user by way of checkback signals (call of the POS_CTRL block) and operating errors by means of diagnostic interrupts (see Section 6.3.4). The error-specification is stored in the form of an error number (see error list in Table 11-6...11-8).

Error acknowledgement

Set/clear control signal [BFQ/FSQ]
or
on message [DF] → write a new write job

Note

Invalid data are not accepted. The original data are retained.

Reading out the error number

The errors are identified by the detail event class (DEKL) and the detail event number (DENR).

Error technology class	DEKL	DENR	Message
Operating error	1	1...n	Diagnostic interrupt
Operator errors	2	1...n	Checkback signals
Travel errors	3	1...n	Checkback signals
Data error	4	1...n	Checkback signals or Data block
Machine data errors	5	1...n	
Traversing program errors	8	1...n	

11.2.4 Message in data block

Overview

Please note the following for direct access to DBs (e.g. using an OP).

If data errors/machine data errors/traversing program errors are detected when the parameters are written to the data block (e.g. in the parameterization tool), an error message is stored in the data block. The error-specification is stored in the form of an error number in the corresponding data block (see error list in Table 11-8). The error message occurs each time the data block is written to until the cause has been eliminated.

It is recommended to scan the error message after every write operation.

11.2.5 Viewing the diagnostic buffer (PG/PC)

Overview

The last five error messages are stored in the diagnostic buffer.

There are two ways of proceeding:

1. In the **S7 SIMATIC Manager** select the menu **File** > **Open** > **Accessible Nodes**.
2. In the **Accessible Nodes** window, select the MPI address of your module.
3. You can view the diagnostic buffer in the menu **Target system** > **Module status**.

or

1. Open your project in the **S7 SIMATIC Manager**.
2. Select the menu **View** > **Online.**
3. In the dialog box, select the FM 354 and the associated program.
4. You can view the diagnostic buffer in the menu **Target system** > **Module status**

11.3 Error lists

Note

In the following tables, please note:

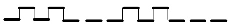
The module response described under “Effect” refers to the error-specific module response. The error response described in Table 11-2 occurs in addition.

11.3.1 Diagnostic interrupts

Overview

The diagnostic interrupts are listed according to error class in Tables 11-4, 11-5.

Table 11-4 Diagnostic interrupt

Byte. Bit	**Error message, error analysis and elimination**		**Message/ Display**
0.1	**Internal errors**	**Error response: “Everything Off”, as in Table 11-2**	
2.1 (8031)	**Communication disturbance**		SF ▨ DC5V ▨ DIAG □
	Cause	MPI/K-bus communication fault caused by unknown event	
	Effect		
	Elimination	• Check connection • Check programming device/CPU • Switch module on/off • Replace module	
2.3 (8033)	**Internal time monitoring circuit (Watchdog)**		
	Cause	• Pronounced noise conditions on the FM 354 • Errors in the FM 354	
	Effect	• Deactivate entire FM 354 • LED indicators: SF: On DIAG: Flashing cycle 	
	Elimination	• If this manual is observed, the errors should not occur However, should this still be the case, please consult the responsible sales department. When doing so, it is vitally important to also report the exact circumstances leading to the error. • Replace the FM 354	

Note: (xxxx) value = Hexadecimal notation in diagnostic buffer

Table 11-4 Diagnostic interrupt, continued

Byte. Bit	**Error message, error analysis and elimination**		**Message/ Display**
0.1	**Internal errors**	**Error response: “Everything Off”, as in Table 11-2**	
2.4 (8034)	**Internal module power supply failure**		SF □ DC5V □ DIAG □
	Cause	• Drastic voltage dip • FM 354 power supply faulty	
	Effect	Deactivate entire FM 354	
	Elimination	• Check FM 354 power connection • If FM 354 power supply defective, replace FM 354	
3.2 (8042)	**FEPROM error**		SF ▨ DC5V ▨ DIAG □
	Cause	Memory for firmware code faulty	
	Effect		
	Elimination	Replace the FM 354	
3.3 (8043)	**RAM errors**		
	Cause	• Faulty RAM data memory • Faulty flash-EPROM data memory	
	Effect		
	Elimination	Replace the FM 354	
3.6 (8046)	**Process interrupt lost**		
	Cause	• A process interrupt event was detected by the FM 354 and cannot be reported, because the same event has not yet been acknowledged by the user program/CPU. • Faults on backplane bus	
	Effect		
	Elimination	• Incorporate OB40 into user program • Check bus connection of the module • Deactivate using MD5 process interrupt • Switch module on/off	
0.2	**External errors**	**Error response: “Everything Off”, as in Table 11-2**	
0.6 (8006)	**Module not parameterized**		SF ▨ DC5V ▨ DIAG □
	Cause	No MPI parameterization data received from the CPU	
	Effect	MPI interface is initialized with default values. MPI address = 12	
	Elimination	Check parameterization, see S7-300 description	

Note: (xxxx) value = Hexadecimal notation in diagnostic buffer

Table 11-4 Diagnostic interrupt, continued

Byte. Bit	Error message, error analysis and elimination		Message/ Display
0.2, 0.3	**External channel errors Error response: "Everything Off", as in Table 11-2**		
8.0 (8090)	**Cable break, incremental encoder**		SF ▨ DC5V ▨ DIAG ▨
	Cause	• Measurement system cable not plugged in or sheared off • Encoder without internode signals • Incorrect pin connection • Cable too long	
	Effect		
	Elimination	• Check encoder and measurement system cable • Observe limit values • Using the MD20, monitoring can be temporarily skipped, at the responsibility of the owner/operator.	
8.1 (8091)	**Error, absolute encoder**		
	Cause	Telegram traffic between FM 354 and the absolute encoder (SSI) is faulty or is disrupted: • Measurement system cable not plugged in or sheared off • Unauthorized type of encoder (only allowable per MD10) • Encoder incorrectly set (programmable encoder) • Telegram length (MD13, MD14) incorrectly specified • Encoder delivers erroneous values • Noise interference on measurement system cable • Baud rate set too high (MD15)	
	Effect		
	Elimination	• Check encoder and measurement system cable • Check telegram traffic between encoder and FM 354 • Using the MD20, monitoring can be temporarily skipped, at the responsibility of the owner/operator.	

Note: (xxxx) value = Hexadecimal notation in diagnostic buffer

Table 11-4 Diagnostic interrupt, continued

Byte. Bit	**Error message, error analysis and elimination**		**Message/ Display**
0.2, 0.3	**External channel errors Error response: "Everything Off", as in Table 11-2**		
8.2 (8092)	**Erroneous pulses, incremental encoder or zero reference mark missing**		SF □ DC5V □ DIAG □
	Cause	The pulses generated by the encoder are monitored for missing pulses (Section 9.6.1). This monitoring feature does not include a comparison with the parameter specifications in MD 13. • Encoder monitoring circuit has discovered erroneous pulses • In "reference point approach" operating mode, no zero reference mark came within one encoder revolution after the reference point switch was passed. • Encoder faulty: does not deliver the specified number of pulses • Interference on the measurement system cable	
	Effect		
	Elimination	• Check encoder and measurement system cable • Observe limit values • Observe rules on shielding and grounding • Using the MD20, monitoring can be temporarily skipped, at the responsibility of the owner/operator.	
8.3 (8093)	**Voltage monitoring, encoder**		
	Cause	• Short-circuit in encoder supply cable (5 V incrementally, 24 V SSI) • Failure of module internal encoder supply unit	
	Effect		
	Elimination	• Check connections • Replace FM 354 if encoder cable ok • Using the MD20, monitoring can be temporarily skipped, at the responsibility of the owner/operator.	
8.7 (8097)	**For operating errors, see Table 11-5**		

Note: (xxxx) value = Hexadecimal notation in diagnostic buffer

Table 11-5 Operating errors

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Operating errors		**Error response: “Everything Off”, as in Table 11-2**		
1 (01)	**1 (01)**	**Software limit switch, beginning is passed**		Diagnostic interrupt
		Cause	Limit switch passed: in “Control” or “Correction” operating mode	
		Effect	• The limit switch position is passed by the necessary stopping distance. • Set actual value is not executed.	
		Elimination	• Following acknowledgment of the error, it is possible to traverse to the working range. • Alter value of software limit switch (MD21) • Switch off monitoring limit switch occasionally! ⚠ (With the limit switches (MD21/22) disabled, the travel range limits are established by the maximum allowable values for the limit switches).	
1 (01)	**2 (02)**	**Software limit switches, end is passed**		Diagnostic interrupt
		Cause	Limit switch passed: in “Control” or “Correction” operating mode	
		Effect	• The limit switch position is passed by the necessary stopping distance. • Set actual value is not executed.	
		Elimination	• Following acknowledgment of the error, it is possible to traverse to the working range. • Alter value of software limit switch (MD22) • Switch off monitoring limit switch occasionally! ⚠ (With the limit switches (MD21/22) disabled, the travel range limits are established by the maximum allowable values for the limit switches).	
1 (01)	**3 (03)**	**Beginning of traversing range passed**		Diagnostic interrupt
		Cause	When operating in “Control” operating mode with soft limits disabled, the traversing range beginning was passed.	
		Effect	The limit switch position is passed by the necessary stopping distance.	
		Elimination	Following acknowledgment of the error, it is possible to traverse to the working range.	

Cl. = Detail event class, No. = Detail event number

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-5 Operating errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Operating errors		**Error response: "Everything Off", as in Table 11-2**		
1 (01)	**4 (04)**	**Traversing range end passed**		Diagnostic interrupt
		Cause	When operating in "Control" operating mode with soft limits disabled, the traversing range beginning was passed.	
		Effect	The limit switch position is passed by the necessary stopping distance.	
		Elimination	Following acknowledgment of the error, it is possible to traverse to the working range.	
1 (01)	**11 (0B)**	**Drive, direction of rotation**		Diagnostic interrupt
		Cause	Drive turns in wrong direction (reported only when setpoint voltage is ± 10 V)	
		Effect		
		Elimination	• Check drive • Check or correct MD19 • Following "Restart" continue working using the user program	
1 (01)	**12 (0C)**	**Zero speed control range**		Diagnostic interrupt
		Cause	The zero speed control range was exited in the following instances: • Servo Enable deactivated • On an axis stillstand in the PEH target area • In "Open-loop control" mode with no traversing command • Other causes: Same as "Direction of rotation, drive", Class 1/No. 11	
		Effect		
		Elimination	• Check electrical and mechanical drive disable (terminals, connecting cables, control element functions) • Match MD26	
1 (01)	**90...99 (5A...63)**	**System errors**		Diagnostic interrupt LED "DIAG" blinking
		Cause	Internal errors in the module	
		Effect	Undefined effects possible	
		Elimination	If this manual is observed, the errors should not occur However, should this still be the case, please consult the responsible sales department. When doing so, it is vitally important to also report the exact circumstances leading to the error.	

Cl. = Detail event class, No. = Detail event number

Note: Value (xx) = Hexadecimal notation of the error number

11.3.2 Error messages

Overview

The errors are listed in Tables 11-6...11-8 according to error class.

Table 11-6 Operator errors

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Operator errors		**Error response : “Feed STOP” see Table 11-2**		
2 (02)	**1 (01)**	**Operating mode not allowed**		CBS
		Cause	The operating mode selected is not allowed.	
		Effect		
		Elimination	Select an allowed operating mode	
2 (02)	**4 (04)**	**Incorrect operating mode parameters**		CBS
		Cause	In the “Jogging” and “Control” operating modes, the selected velocity or control level is not 1 or 2. In incremental operation the set value number is not allowed (1 - 100, and 254 permitted).	
		Effect		
		Elimination	Set operating mode parameters to an allowable value.	
2 (02)	**5 (05)**	**Start enable missing**		CBS
		Cause	A travel command was given in the absence of a start enable (start, external start, R+/R−)	
		Effect		
		Elimination	Restore travel command and wait for start enable	
2 (02)	**9 (09)**	**Axis is not synchronized**		CBS
		Cause	Synchronization of the axes is necessary in the “Incremental, relative,” “MDI” and “Automatic” operating modes.	
		Effect		
		Elimination	Execute reference point approach	
2 (02)	**11 (0B)**	**Direction specification not allowed**		CBS
		Cause	In operating modes “Jog,” “Control” or “Incremental, relative” the direction settings R+/R- are active at the same time. With “Reference point operation” the direction setting no longer agrees with the startup direction specified in the MD.	
		Effect		
		Elimination	Correct the direction parameters	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-6 Operator errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Operator errors		**Error response : “Feed STOP” see Table 11-2**		
2 (02)	**12 (0C)**	**Axis movement not possible**		CBS
		Cause	With an unacknowledged error, no drive enable or stop, a traverse command was triggered.	
		Effect		
		Elimination	Restore traverse command and switch acknowledge error or Stop to inactive, or give drive enable.	
2 (02)	**13 (0D)**	**Incremental value not in place**		CBS
		Cause	The setpoints defined by the operating mode parameters are missing or a change in incremental dimensions occurred when the operating mode started.	
		Effect		
		Elimination	Parameterize and read in setpoint parameters	
2 (02)	**14 (0E)**	**No program preselected**		CBS
		Cause	No program preselected at “Start.”	
		Effect		
		Elimination	First preselect program then start.	
2 (02)	**15 (0F)**	**Digital input not activated**		CBS
		Cause	The programmed target was reached in a block with external block change (G50).	
		Effect		
		Elimination	Check programming (MD34) and connection of digital input.	
2 (02)	**16 (10)**	**Measurement function undefined**		CBS
		Cause	Length measurement and inprocess measurement selected simultaneously	
		Effect	No measurement function effective.	
		Elimination	Reselect one of the two measurement functions.	
2 (02)	**21 (15)**	**Activate machine data not allowed**		CBS
		Cause	“Machining in progress” is still active (see Section 7.3.1)	
		Effect	Activate machine data not executed	
		Elimination	Terminate processing, repeat activation	
2 (02)	**22 (16)**	**On-the-fly MDI block inoperative**		CBS
		Cause	MDI block inactive or already executed	
		Effect	On-the-fly MDI block is not being processed	
		Elimination	Delete error message and initiate execution as MDI block.	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-7 Travel errors

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Travel errors		**Error response : “Feed STOP” see Table 11-2**		
3 (03)	**1 (01)**	**Software limit switch, beginning**		CBS
		Cause	Limit switch approached: in “Jog” operating mode, in “Automatic” operating mode if G88/89, without switching signal from the corresponding digital input. The axis is located to the left of the software limit switch because of actual value set.	
		Effect	• Axis movement is stopped at the limit switch position. • Set actual value is not executed.	
		Elimination	• Following acknowledgment of the error, it is possible to traverse to the working range. • Alter value of software limit switch (MD21) • Switch off monitoring limit switch occasionally! ⚠ (With the limit switches (MD21/22) disabled, the travel range limits are established by the maximum allowable values for the limit switches).	
3 (03)	**2 (02)**	**Software limit switch, end**		CBS
		Cause	Limit switch approached: in “Jog” operating mode, in “Automatic” operating mode if G88/89, without switching signal from the corresponding digital input. The axis is located to the right of the software limit switch because of actual value set.	
		Effect	• Axis movement is stopped at the limit switch position. • Set actual value is not executed.	
		Elimination	• Following acknowledgment of the error, it is possible to traverse to the working range. • Alter value of software limit switch (MD22) • Switch off monitoring limit switch occasionally! ⚠ (With the limit switches (MD21/22) disabled, the travel range limits are established by the maximum allowable values for the limit switches).	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-7 Travel errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Travel errors		**Error response : “Feed STOP” see Table 11-2**		
3 (03)	**3 (03)**	**Traversing range beginning approached**		CBS
		Cause	• During traversing with soft limit switches disabled, the traversing range beginning was approached. • The axis is located to the left of the traversing range beginning because of actual value set. (Traversing range: $\pm 10^9$ or from range covered by absolute encoder)	
		Effect	• Axis movement is stopped at the traversing range limit. • Set actual value is not executed.	
		Elimination	Travel in the opposite direction	
3 (03)	**4 (04)**	**Traversing range end approached**		CBS
		Cause	• During traversing with soft limit switches disabled, the traversing range end was approached. • The axis is located to the left/right of the traversing range end because of actual value set. (Traversing range: $\pm 10^9$ or from range covered by absolute encoder)	
		Effect	• Axis movement is stopped at the traversing range limit. • Set actual value is not executed.	
		Elimination	Travel in the opposite direction	
3 (03)	**5 (05)**	**Target position not within traversing range**		CBS
		Cause	• The position to be approached is outside the working range limited by the software limit switches. • The rotary axis programming is specified as a reference value which does not fall within the positive complete circle.	
		Effect		
		Elimination	• Correct position to be approached. • Alter value of software limit switch (MD) • Switch off monitoring limit switch occasionally! ⚠ (With the limit switches (MD21/22) disabled, the travel range limits are established by the maximum allowable values for the limit switches).	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-7 Travel errors, continued

Cl.	**No.**	**Error message, error analysis and elimination**		**Message/ Display**
Travel errors		**Error response : “Feed STOP” see Table 11-2**		
3 (03)	**23 (17)**	**Target velocity zero**		CBS
		Cause	• Zero was entered as programmed velocity. • No feed was programmed for positioning.	
		Effect		
		Elimination	Input an allowable velocity value	
3 (03)	**28 (1C)**	**M2/M30 missing**		CBS
		Cause	• In the last program, block, no M2, M30 or M18 is programmed. • The last program, block is a skip block.	
		Effect		
		Elimination	Per causes	
3 (03)	**30 (1E)**	**Digital input not parameterized**		CBS
		Cause	For traversing with set actual value on the fly (G88, G89), external block change (G50) or measurement, no digital input necessary for that purpose is parameterized.	
		Effect	The functions are not started.	
		Elimination	Parameterize the digital inputs by way of MD34.	
3 (03)	**35 (23)**	**Tool offset value not in place**		CBS
		Cause	No tool offset values are available on the FM 354 or tool offsets are accessed and modified when an override is active.	
		Effect		
		Elimination	Parameterize and read in tool offset values	
3 (03)	**36 (24)**	**Set actual value on the fly, incorrect value**		CBS
		Cause	Value is no longer within the range $\pm 10^9$	
		Effect		
		Elimination	Input a correct value	
3 (03)	**37 (25)**	**MDI-block on the fly, incorrect syntax**		CBS
		Cause	Incorrect M or G commands or incorrect block structure	
		Effect		
		Elimination	Input a correct MDI block	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-7 Travel errors, continued

Cl.	No.	Error message, error analysis and elimination				Message/ Display
Travel errors		**Error response : "Feed STOP" see Table 11-2**				
3 (03)	**38 (26)**	**MDI block on the fly, incorrect velocity**				CBS
		Cause	Velocity not within the range between > 0 and max. allowable traverse velocity (500,000,000 MSR/min)			
		Effect				
		Elimination	Input a correct MDI block			
3 (03)	**39 (27)**	**MDI block on the fly, incorrect position or dwell time**				CBS
		Cause	Position or dwell time is outside allowable values. Position: ± 10^9 MSR Dwell time: > 100,000 ms			
		Effect				
		Elimin.	Input a correct MDI block			
3 (03)	**40 (28)**	**MDI block on the fly erroneous**				CBS
		Cause	Incorrect block syntax			
		Effect				
		Elimination	Input a correct MDI block			
3 (03)	**61 (3D)**	**Controller enable missing**				CBS
		Cause	Traverse command of the axis without controller enable (except for "Control" operating mode)	or	Removal of controller enable during "Processing in progress"	
		Effect	No axis movement	or	Axis is stopped via deceleration ramp MD41 (the Servo Enable is maintained until axis comes to a standstill)	
		Elimination	Set controller enable by way of user program			
3 (03)	**62 (3E)**	**Controller not ready for operation**				CBS
		Cause	Axis started without "Controller ready message"	or	"Controller ready message" canceled whilst "processing in progress".	
		Effect	No axis movement	or	Axis is stopped via voltage ramp (MD45) with actual value transfer after axis comes to rest (internally like follow-up mode)	
		Elimination	• Check drive/connecting cables • Analysis of the "Controller ready" message can be disabled by MD37!			

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-7 Travel errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Travel errors		**Error response : “Feed STOP” see Table 11-2**		
3 (03)	**64 (40)**	**PEH target area monitoring**		CBS
		Cause	Following conclusion of the setpoint value specification to the position controller, the target area is not reached within the specified time.	
		Effect		
		Elimination	• Check drive • Match MD24, MD25	
3 (03)	**65 (41)**	**No drive movement**		CBS
		Cause	• Axis standstill at maximum drive control signal (± 10 V) • on violation of the defined following error limit	
		Effect	• Deceleration of the drive via voltage ramp (MD45) • Actual value transfer (internally like ”follow-up mode”)	
		Elimination	• Check drive/connecting cables • Check controller enable signal between FM 354 and drive	
3 (03)	**66 (42)**	**Following error too great**		CBS
		Cause	Excessive following error during axis movement	
		Effect	Deceleration of the drive via voltage ramp (MD45)	
		Elimination	• Check drive • Check MD23, MD43	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors

Cl.	No.	Error message, error analysis and elimination		Message/ Display
General data errors		**Error response: “Warning” see Table 11-2**		
4 (04)	**1 (01)**	**Data at time of transmission unacceptable**		CBS or DB
		Cause	Data not transmitted in appropriate operating mode	
		Effect	Data not accepted	
		Elimination	Transmit data in appropriate operating mode	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
General data errors		**Error response: “Warning” see Table 11-2**		
4 (04)	**2 (02)**	**Velocity level 1 incorrect**		CBS or DB
		Cause	Velocity not within the range between > 0 and max. allowable traverse velocity (500,000,000 MSR/min)	
		Effect	Velocity does not become effective	
		Elimination	Input an allowed velocity value	
4 (04)	**3 (03)**	**Velocity level 2 incorrect**		CBS or DB
		Cause	Velocity not within the range between > 0 and max. allowable traverse velocity (500,000,000 MSR/min)	
		Effect	Velocity does not become effective	
		Elimination	Input an allowed velocity value	
4 (04)	**4 (04)**	**Voltage level 1 incorrect**		CBS or DB
		Cause	Voltage specified does not fall within the range of ± 10 V	
		Effect	Voltage does not become effective	
		Elimination	Input an allowed voltage value	
4 (04)	**5 (05)**	**Voltage level 2 incorrect**		CBS or DB
		Cause	Voltage specified does not fall within the range of ± 10 V	
		Effect	Voltage does not become effective	
		Elimination	Input an allowed voltage value	
4 (04)	**6 (06)**	**Preset incremental value too high**		CBS or DB
		Cause	Incremental value is greater than 10^9 MSR	
		Effect	Original incremental value is retained	
		Elimination	Input an allowable incremental value	
4 (04)	**7 (07)**	**MDI block, incorrect syntax**		CBS or DB
		Cause	Incorrect M or G commands or incorrect block structure	
		Effect	Original MDI block is retained	
		Elimination	Input a correct MDI block	
4 (04)	**8 (08)**	**MDI block, incorrect velocity**		CBS or DB
		Cause	Velocity not within the range between > 0 and max. allowable traverse velocity (500,000,000 MSR/min)	
		Effect	Original MDI block is retained	
		Elimination	Input a correct MDI block	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
General data errors		**Error response: “Warning” see Table 11-2**		
4 (04)	**9 (09)**	**MDI block, position or dwell time incorrect**		CBS or DB
		Cause	Position or dwell time falls outside the allowable values Position: $\pm 10^9$ MSR Dwell time: > 100,000 ms	
		Effect	Original MDI block is retained	
		Elimination	Input a correct MDI block	
4 (04)	**10 (0A)**	**Zero offset value, offset value incorrect**		CBS or DB
		Cause	Value falls outside the range $\pm 10^9$ MSR	
		Effect	Does not become effective	
		Elimination	Input a correct value	
4 (04)	**11 (0B)**	**Set actual value, actual value incorrect**		CBS or DB
		Cause	Actual value falls outside the software limit switches or outside the range $\pm 10^9$ MSR	
		Effect	Set actual value does not become effective	
		Elimination	Input a correct value	
4 (04)	**12 (0C)**	**Set reference point value, reference point incorrect**		CBS or DB
		Cause	Value falls outside the range $\pm 10^9$ MSR	
		Effect	Set reference point does not become effective	
		Elimin.	Input a correct value	
4 (04)	**13 (0D)**	**Digital output not possible**		CBS or DB
		Cause	Output not available for direct output of the user program	
		Effect	Output is not executed	
		Elimination	• Correct user program • Correct parameterization of the output assignment within the MD35 to the desired assignment	
4 (04)	**14 (0E)**	**Request application data incorrect**		CBS or DB
		Cause	Incorrect request code	
		Effect	Old application data are retained	
		Elimination	Request code 0-6, 16-23 and 25 possible	
4 (04)	**15 (0F)**	**Teach In, program number incorrect**		CBS or DB
		Cause	The program was not parameterized or read in.	
		Effect	Teach In is not executed	
		Elimination	Parameterize and read in program or correct program number	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
General data errors		**Error response: “Warning” see Table 11-2**		
4 (04)	**16 (10)**	**Teach In, block number incorrect**		CBS or DB
		Cause	The block number in the program selected is not in place.	
		Effect	Teach In is not executed	
		Elimination	Specify correct block number	
4 (04)	**17 (11)**	**Teach In, dwell time or subprogram-request in block**		CBS or DB
		Cause	The block number in the program selected is not in place or incorrect block number was selected.	
		Effect	Teach In is not executed	
		Elimination	Specify correct block number	
4 (04)	**18 (12)**	**Teach In, no axis stoppage**		CBS or DB
		Cause	Axis is still in motion	
		Effect	Teach In is not executed	
		Elimination	Stop axis and repeat task	
4 (04)	**40 (28)**	**Transmit non-relevant data**		CBS or DB
		Cause	The data (data blocks) transmitted are unknown to the FM 354	
		Effect	Data not accepted	
		Elimination	Correct user program	
4 (04)	**81 (51)** **82 (52)** **83 (53)** **84 (54)** **85 (55)**	**Programmable modules communication: unauthorized DB type** **Programmable modules communication: Info 1 incorrect** **Programmable modules communication: Info 2 incorrect** **Programmable modules communication: unauthorized task** **Programmable modules communication: data errors**		CBS or DB
		Cause	Incorrect data	
		Effect	Task is not executed	
		Elimination	Correct and retransmit	
4 (04)	**120 (78)**	**Measurement system grid deviates**		CBS or DB
		Cause	The measurement system in the DBs “NC, SM, TO” does not agree with MD7.	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
4 (04)	**121 (79)**	**Incorrect DB type in the module**		CBS or DB
		Cause	An incorrect type of DB has been transmitted into the FM 354	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Delete DB, correct and retransmit	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
General data errors		**Error response: “Warning” see Table 11-2**		
4 (04)	**122 (7A)**	**DB type or DB no. already exists**		CBS or DB
		Cause	DB type already exists	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Delete corresponding DB prior to transmission	
4 (04)	**123 (7B)**	**NC program number already exists**		CBS or DB
		Cause	NC program number already exists	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Prior to transmission, delete corresponding DB with the program number	
4 (04)	**124 (7C)**	**Parameter “Save” incorrect**		CBS or DB
		Cause	Coding not 0 or 1	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Coding not 0 or 1	
4 (04)	**125 (7D)**	**DB memory filed**		CBS or DB
		Cause	The available memory is assigned	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Delete unnecessary programs (DBs) or compress memory by way of parametering interface	
4 (04)	**126 (7E)**	**Allowable program length exceeded**		CBS or DB
		Cause	Number of blocks too high	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct program and retransmit	
4 (04)	**127 (7F)**	**Writing parameters/data is not possible**		CBS or DB
		Cause	Axis does not come to a stop	
		Effect	Parameters/data do not become effective	
		Elimination	Stop axis	
4 (04)	**128 (80)**	**Incorrect module identification**		CBS or DB
		Cause	DBs which do not belong to the module were transmitted (no identification 354)	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Transmit the DBs belonging to the FM 354	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
General data errors		**Error response: “Warning” see Table 11-2**		
4 (04)	**129 (81)**	**Incremental value, incorrect value**		CBS or DB
		Cause	Value range outside ± 10^9	
		Effect	Incremental value not effective	
		Elimination	Transmit correct value	
4 (04)	**130 (82)**	**Tool offset, incorrect value**		CBS or DB
		Cause	Value range outside ± 10^9	
		Effect	Tool offset not effective	
		Elimination	Transmit correct value	
4 (04)	**131 (83)**	**Not possible to insert block**		CBS or DB
		Cause	Memory full	
		Effect	Function is not executed	
		Elimination	Delete unnecessary DBs and repeat function	
4 (04)	**132 (84)**	**Not possible to delete block**		CBS or DB
		Cause	Block does not exist, no “assignment bits (bytes 2 and 3) enabled in block (when data available).	
		Effect	Function is not executed	
		Elimination	Check program and repeat function with correct block number	
4 (04)	**144(90)**	**SDB cannot be loeaded**		CBS or DB
		Cause	Module not at standstill	
		Effect	SDB is rejected	
		Elimination	Stop module and retry load operation.	
4 (04)	**145(91)**	**SDB user data error**		CBS or DB
		Cause	SDB contains value error	
		Effect	SDB is rejected	
		Elimination	Generate SDB with parameter assignment tool and retry load operation	
Machine data errors		**Error response: “Warning” see Table 11-2**		

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Machine data errors		**Error response: “Warning” see Table 11-2**		
5 (05)	**7 (07)**	**Measurement system**		CBS or DB
		Cause	The measurement system grid (MSR) entered does not agree with the MSR in the other DBs of the module.	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	• Check MSR and correct as necessary • When making correct input, delete the other DBs on the module before retransmitting.	
5 (05)	**8 (08)**	**Type of axis**		CBS or DB
		Cause	No linear or rotary axis parameterized	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**9 (09)**	**Rotary axis**		CBS or DB
		Cause	Impermissible value range or dependency violation (see Section 5.3.1)	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**10 (0A)**	**Encoder type**		CBS or DB
		Cause	Unacceptable type of encoder	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**11 (0B)** **12 (0C)** **13 (0D)** **14 (0E)**	**Travel per encoder revolution** **Distance to go per encoder revolution** **Increments per encoder revolution** **Number of revolutions, absolute encoder**		CBS or DB
		Cause	• Impermissible value range or • dependency violation on no. 11, 12, 13 (see Section 5.3.1)	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**15 (0F)**	**Baud rate, absolute encoder**		CBS or DB
		Cause	Unacceptable baud rate	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Machine data errors			**Error response: “Warning” see Table 11-2**	
5 (05)	16 (10) 17 (11)	**Reference point coordinates, absolute encoder adjustment**		CBS or DB
		Cause	Unacceptable value range	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	18 (12)	**Type of reference point travel**		CBS or DB
		Cause	Unacceptable type of reference point travel	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	19 (13)	**Direction matching undefined**		CBS or DB
		Cause	Direction matching undefined	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	20 (14)	**Disable hardware monitoring undefined**		CBS or DB
		Cause	Disable hardware monitoring undefined	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	21 (15) 22 (16) 23 (17) 24 (18) 25 (19) 26 (1A) 27 (1B) 28 (1C) 29 (1D) 30 (1E)	**Software limit switch, begin** **Software limit switch, end** **Maximum velocity** **Target range (PEH)** **Monitoring time** **Stoppage area** **Reference point offset** **Referencing velocity** **Reducing velocity** **Backlash compensation**		CBS or DB
		Cause	• Impermissible value range or • dependency violation on no. 21, 22, 28, 29 (see Section 5.3.1)	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	31 (1F)	**Backlash vector reference**		CBS or DB
		Cause	Backlash vector reference undefined	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Machine data errors			**Error response: "Warning" see Table 11-2**	
5 (05)	**32 (20)**	**Type of output, M-function**		CBS or DB
		Cause	Type of output, M-function not defined	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**33 (21)**	**Output time, M-function**		CBS or DB
		Cause	Unacceptable value range	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**34 (22)**	**Digital inputs**		CBS or DB
		Cause	Inputs undefined or defined more than once	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**35 (23)**	**Digital outputs**		CBS or DB
		Cause	Outputs undefined or defined more than once	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**36 (24)**	**Input adapter**		CBS or DB
		Cause	Input adapter undefined	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**38 (26)** **39 (27)** **40 (28)** **41 (29)** **42 (2A)** **43 (2B)** **44 (2C)** **45 (2D)**	**Positioning circuit amplification** **Minimum following error, dynamic** **Speed-up** **Slow-down** **Jerk time** **Set voltage, max.** **Offset compensation** **Voltage ramp**		CBS or DB
		Cause	Unacceptable value range	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Machine data errors		**Error response: “Warning” see Table 11-2**		
5 (05)	**52 (34)** **53 (35)** **54 (36)** **55 (37)** **56 (38)**	**Velocity for backlash compensation** **Backlash compensation mode** **Illegal standstill speed for controlling** **Illegal time-out time for enforced zero-speed detection** **Illegal response time for the standard diagnosis**		CBS or DB
		Cause	Impermissible value range or dependency violation (see Section 5.3.1).	
		Effect	DB does not become effective and is stored non-retentively.	
		Elimination	Correct and retransmit.	
5 (05)	**96 (60)**	**Software limit unacceptable**		CBS or DB
		Cause	With linear axes: Software begin limit switch greater than software limit switch end with rotary axes: Software begin/end limit switches not within rotary axis cycle and not at maximum input value.	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**97 (61)**	**Limitation, software limit with absolute encoder**		CBS or DB
		Cause	Travel distance between software limit switch begin and end is greater than the absolute value range of the encoder.	
		Effect	DB does not become effective and is stored non-retentively	
		Elimination	Correct and retransmit	
5 (05)	**99 (63)**	**Impermissible actual value evaluation factor**		CBS or DB
		Cause	Impermissible relationship in the assignments for distance per encoder revolution (MD11, 12) and increments per encoder revolution (MD13) (see Section 5.3.1).	
		Effect	DB does not become effective and is stored non-retentively.	
		Elimination	Correct and retransmit.	
5 (05)	**102 (66)**	**Limitation, software limit for linear axis**		CBS or DB
		Cause	For encoder resolutions < 1 MSR, the permissible traversing range in the ratio of MSR to increments is limited (e.g. for 0.5 μm per enc. pulse to $0.5 \cdot 10^9$ MSR) (see Section 5.3.1).	
		Effect	DB does not become effective and is stored non-retentively.	
		Elimination	Correct and retransmit.	
Traversing program errors		**Error response: “Warning” see Table 11-2**		

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Traversing program errors	**Error response: “Warning” see Table 11-2**			
8 (08)	**1 (01)**	**Program selection, subroutine number incorrect**		CBS or DB
		Cause	• The subroutine requested in the program is not in place on the FM 354. • The subprogram called in the program contains another subprogram call. Nesting is not possible.	
		Effect	Program selection is not executed	
		Elimination	• Parameterize and read in program, correct as necessary • Select another program	
8 (08)	**8 (08)**	**Program selection, program number not in place**		CBS or DB
		Cause	The program was not parameterized, not in place on the FM 354.	
		Effect	Program selection is not executed	
		Elimination	• Parameterize and read in program, correct as necessary • Select another program	
8 (08)	**9 (09)**	**Program selection, block number missing**		CBS or DB
		Cause	The block number is missing in the program selected	
		Effect	Program selection is not executed	
		Elimination	• Correct program • Select different block number	
8 (08)	**10 (0A)**	**Program, block number unacceptable**		CBS or DB
		Cause	Block number missing or outside of the number range	
		Effect	Program is not stored	
		Elimination	Correct program	
8 (08)	**11 (0B)**	**Program selection, direction specification incorrect**		CBS or DB
		Cause	Direction specification incorrect	
		Effect	Program selection is not executed	
		Elimination	Correct program selection and repeat.	
8 (08)	**12 (0C)**	**Program selection unacceptable**		CBS or DB
		Cause	Another program was preselected during a movement	
		Effect	Program selection is not executed	
		Elimination	Use STOP to stop program in progress, or repeat program selection at end of program.	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Traversing program errors		**Error response: "Warning" see Table 11-2**		
8 (08)	**20 (14)**	**Error, program number**		CBS or DB
		Cause	Program numbers in the blocks incorrect	
		Effect	Program is not stored	
		Elimination	Correct program, per cause	
8 (08)	**21 (15)**	**No block in program**		CBS or DB
		Cause	No block in program	
		Effect	Program is not stored	
		Elimination	Correct program, per cause	
8 (08)	**22 (16)**	**Error, block number**		CBS or DB
		Cause	Block number value range incorrect	
		Effect	Program is not stored	
		Elimination	Correct program	
8 (08)	**23 (17)**	**Block number sequence incorrect**		CBS or DB
		Cause	Block number not in ascending order	
		Effect	Program is not stored	
		Elimination	Correct program	
8 (08)	**24 (18)**	**G function 1 unacceptable**		CBS or DB
		Cause	• The number programmed as G function 1 is not allowed. • In block, other data besides M-functions were programmed with dwell time (G04).	
		Effect	Program/block not stored	
		Elimination	Correct program, per cause	
8 (08)	**25 (19)**	**G function 2 unacceptable**		CBS or DB
		Cause	The number programmed as G function 2 is not allowed.	
		Effect	Program/block not stored	
		Elimination	Correct program, per cause	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Traversing program errors		**Error response: “Warning” see Table 11-2**		
8 (08)	**26 (1A)**	**G function 3 unacceptable**		CBS or DB
		Cause	• The number programmed as G function 3 is not allowed. • External block change (G50) was programmed in a block together with continuous operation for setting actual value on the fly (G88/89). • A tool offset (G43, G44) was called up without D number. • In selecting a D number, the direction specification is missing for the tool offset (G43, G44).	
		Effect	Program/block not stored	
		Elimination	Correct program, per cause	
8 (08)	**27 (1B)**	**M function unacceptable**		CBS or DB
		Cause	• The number programmed as M function is not allowed. • At least two of the M functions M0, M2, M18, M30, which cancel each other out, are found in one block.	
		Effect	Program/block not stored	
		Elimination	Correct program, per cause	
8 (08)	**28 (1C)**	**Position/dwell time missing**		CBS or DB
		Cause	• No dwell time specified in block with G04. • Target position missing with external block change (G50). • No new actual value programmed for the function continuous operation with setting actual value on the fly (G88/89).	
		Effect	Program/block not stored	
		Elimination	Correct program, per cause	
8 (08)	**29 (1D)**	**Incorrect D-NO (>20)**		CBS or DB
		Cause	The number for tool offset is greater than 20	
		Effect	Program/block not stored	
		Elimination	Correct program, per cause	
8 (08)	**30 (1E)**	**Error, subroutine**		CBS or DB
		Cause	Subroutine without number of runs	
		Effect	Program is not stored	
		Elimination	Correct program, per cause	
8 (08)	**31 (1F)**	**Velocity missing**		CBS or DB
		Cause	No velocity was programmed	
		Effect	Program/block not stored	
		Elimination	Correct program, per cause	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Table 11-8 General data errors, machine data errors, traversing program errors, continued

Cl.	No.	Error message, error analysis and elimination		Message/ Display
Traversing program errors Error response: "Warning" see Table 11-2				
8 (08)	**32 (20)**	**Error, callup subroutine**		CBS or DB
		Cause	Block syntax for callup subroutine is incorrect	
		Effect	Program is not stored	
		Elimination	Correct program, per cause	
8 (08)	**33 (21)**	**D function unacceptable**		CBS or DB
		Cause	Block syntax for invoking a D function is incorrect	
		Effect	Program is not stored	
		Elimination	Correct program, per cause	
8 (08)	**34 (22)**	**Incorrect program length**		CBS or DB
		Cause	Maximum block number exceeded	
		Effect	Program is not stored	
		Elimination	Correct program, per cause	

Cl. = Detail event class, No. = Detail event number, CBS = checkback signals, DB = data block

Note: Value (xx) = Hexadecimal notation of the error number

Technical Specifications A

Overview

This chapter describes the technical data for the FM 354 positioning module.

- General technical data
- Dimensions and weight
- Load memory
- Encoder inputs
- Setpoint output
- Digital inputs
- Digital outputs

General technical data

General technical data include:

- Electromagnetic compatibility
- Shipping and storage conditions
- Ambient mechanical and climate conditions
- Data on insulation testing, protection class and degree of protection

This information contains standards and test values with which the S7-300 complies, or according to whose criteria the S7-300 was tested.

The general technical data are described in the manual “Installing an S7-300.”

UL/CSA certifications

The following certifications are on record for the FM 354:

UL Recognition Mark
Underwriters Laboratories (UL) in compliance with
UL Standard 508, File E 164110

CSA Certification Mark
Canadian Standard Association (CSA) in compliance with
Standard C 22.2 No. 142

FM approval

The FM approval is on record for the FM 354:
FM certification in accordance with Factory Mutual Approval Standard Class Number 3611, Class I, Division 2, Group A, B, C, D.

Warning

Potential for personal injury and property damage.

In areas where there is a risk of explosion, personal injury and property damage may occur if you disconnect plugs while the S7-300 is in operation.

In areas where there is a risk of explosion, always cut off power to the S7-300 before disconnecting plugs.

Warning

WARNING - NEVER DISCONNECT WHILE CIRCUIT IS LIVE
UNLESS LOCATION IS KNOWN TO BE NONHAZARDOUS

CE marking

Our products are in compliance with the EU Guideline 89/336/EEC "Electromagnetic Compatibility" and the harmonized European standards (EN) which it embodies.

The EC Declaration of Conformity in accordance with Article 10 of the EU Guideline referenced above can be found on the Internet at:

http://support.automation.siemens.com/WW/view/de/15257461

Application

SIMATIC products are designed for application in an industrial environment.

Application	Requirement concerning	
	Noise emission	Noise immunity
Industry	EN 50081-2 : 1993	EN 61000-6-2 : 1999

Observe installation guidelines

SIMATIC products meet the requirements, provided you observe the installation guidelines set forth in the manuals during installation and operation.

Power ratings

Technical data: Power ratings

Table A-1 Power ratings

Supply voltage	20.4...28.8 V
Power consumption from 24 V	0.35 A
Power loss	8 W
Startup current	2.2 A
Power consumption from 5 V backplane bus	100 mA

Dimensions and weights

Technical data for dimensions and weights:

Table A-2 Technical data: dimensions and weight

Dimensions W × H × D (mm/in)	80 × 125 × 118
Weight (g/lb)	approx. 550 / 1 lb, 3.3 oz

Memory for parameter data

RAM memory 16 Kbytes FEPROM for retentive storage of parameter data

FM cycle

2 ms

Encoder inputs

Technical data for encoder inputs:

Table A-3 Technical data: encoder inputs

Position detection	• Incremental • Absolute (SSI)
Signal voltages	Inputs: 5 V per RS422
Encoder supply voltage	• 5.2 V/300 mA • 24 V/300 mA
Input frequency and line length for incremental encoder	• Max. 1 MHz wih 10 m conductor length shielded • Max. 500 kHz with 35 m conductor length shielded
Data transm. rates and line length for absolute enc. (SSI)	• Max. 1.25 Mbit/s with 10 m conductor length shielded • Max. 125 kbit/s with 100 m conductor length shielded

Drive port

Table A-4 Technical data: setpoint output

Rated voltage range	–10...10 V
Output current	–3...3 mA

D/A converter resolution see Section 4.2.

Digital inputs

Table A-5 Technical data: digital inputs

Number of inputs	5 (including controller ready)
Supply voltage	24 V DC (allowable range: 20.4...28.8 V)
Electrical isolation	No
Input voltage	• 0 Signal: –3...5 V • 1 Signal: 11...30 V
Input current	• 0 Signal: ≤ 2 mA • 1 Signal: 6...15 mA
Input delay (DI1...4)	• 0 → 1 Signal: typ. 15 µs • 1 → 0 Signal: typ. 150 µs
Connecting a 2-conductor sensor	Possible

Digital outputs

Table A-6 Technical data: digital outputs

Number of outputs	4
Supply voltage	24 V DC (allowable range: 20.4...28.8 V)
Electrical isolation	No
Output voltage	• 0 Signal: Residual current max. 2 mA • 1 Signal: (Power supply –3 V)
Output current on signal “1”	
• at ambient temperature of 40°C	
– Rated value	0.5 A (total current 2 A)
– Permissible value range	5 mA...0.6 A (over power supply range)
– Lamp load	max. 5 W
• at ambient temperature of 60°C	
– Rated value	0.1 A (total current 0.4 A)
– Permissible value range	5 mA...0.12 A (over power supply)
Short circuit protection	Yes
Switching rate	• Resistive load: max. 100 Hz • Inductive load: max. 0.25 Hz

User Data Block (AW-DB)

B

Table B-1 User data block (AW–DB)

Absolute address	Variable	Data type	Comment
General addresses			
0	MOD_ADR	INT	Module address
2...13.5			Reserved
13.6	MODE_BUSY	BOOL	Execution started
13.7	POS_REACHED	BOOL	Position
Control signals			
14.0			Reserved
14.1	TEST_EN	BOOL	Switch to P bus Start-up
14.2			Reserved
14.3	OT_ERR_A	BOOL	Acknowledge operator/traversing error
14.4...14.7			Reserved
15.0	START	BOOL	Start
15.1	STOP	BOOL	Stop
15.2	DIR_M	BOOL	Negative direction
15.3	DIR_P	BOOL	Positive direction
15.4	ACK_MF	BOOL	Acknowledge M function
15.5	READ_EN	BOOL	Read-in enable
15.6	SKIP_BLK	BOOL	Block skip
15.7	DRV_EN	BOOL	Drive enable
16	MODE_IN	BYTE	Operating mode
17	MODE_ TYPE	BYTE	Operating mode parameters
18	OVERRIDE	BYTE	Override
19...21			Reserved
Checkback signals			
22.0			Reserved
22.1	TST_STAT	BOOL	Switch to P bus completed
22.2			Reserved
22.3	OT_ERR	BOOL	Operator/traversing error
22.4	DATA_ERR	BOOL	Data error

Table B-1 User data block (AW−DB), continued

Absolute address	Variable	Data type	Comment
22.5...22.6			Reserved
22.7	PARA	BOOL	Channel parameterizell
23.0	ST_ENBLD	BOOL	Start enable
23.1	WORKING	BOOL	Machining in progress
23.2	WAIT_EI	BOOL	Wait for external enable
23.3...23.4			Reserved
23.5	DT_RUN	BOOL	Dwell in progress
23.6	PR_BACK	BOOL	Reverse prog. scan
23.7			Reserved
24	MODE_OUT	BYTE	Active operating mode
25.0	SYNC	BOOL	Channel synchronized
25.1	MSR_DONE	BOOL	End of measurement
25.2	GO_M	BOOL	Negative travel
25.3	GO_P	BOOL	Positive travel
25.4	ST_SERVO	BOOL	Servo enable status
25.5	FVAL_DONE	BOOL	On-the-fly setting of actual value completed
25.6			Reserved
25.7	POS_RCD	BOOL	Position reached. Stop.
26	NUM_MF	BYTE	M function number
27.0...27.3			Reserved
27.4	STR_MF	BOOL	M function strobe signal
27.5			Reserved
28...31	ACT_POS	DINT	Actual position
32...33			Reserved
Initiation signals for single settings			
34.0	SERVO_EN	BOOL	Controller enable
34.1	GAUG_FLY	BOOL	On-the-fly measuring
34.2...34.5			Reserved
34.6	PARK_AX	BOOL	Parking axis
34.7	SIM_ON	BOOL	Simulation on
35.0...35.1			Reserved
35.2	MSR_EN	BOOL	Length measurement
35.3	REF_TRIG	BOOL	Retrigger reference point
35.4	DI_OFF	BOOL	Enable input disabled
35.5			Reserved
35.6	SSW_DIS	BOOL	Software limit positions disabled

Table B-1 User data block (AW−DB), continued

Absolute address	Variable	Data type	Comment
35.7			Reserved
Initiation signals for single commands			
36			Reserved
37.0	MD_EN	BOOL	Activate MD
37.1	DELDIST_EN	BOOL	Delete residual distance
37.2	SEARCH_F	BOOL	Automatic block advance
37.3	SEARCH_B	BOOL	Automatic block return
37.4			Reserved
37.5	RESET_AX	BOOL	Restart
37.6	AVALREM_EN	BOOL	Rescind setting of actual value
37.7			Reserved
Initiation signals for Write requests			
38.0	VLEV_EN	BOOL	Speed levels 1, 2
38.1	CLEV_EN	BOOL	Frequency levels 1, 2
38.2	TRG254_EN	BOOL	Setpoint for incremental dimension
38.3	MDI_EN	BOOL	MDI block
38.4	MDIFLY_EN	BOOL	On-the-fly MDI block
38.5			Reserved
38.6	REFPT_EN	BOOL	Set reference point
38.7	AVAL_EN	BOOL	Set actual value
39.0	FVAL_EN	BOOL	On-the-fly setting of actual value
39.1	ZOFF_EN	BOOL	Zero offset
39.2			Reserved
39.3	PARCH_EN	BOOL	Modify parameters/data
39.4	DIGO_EN	BOOL	Digital outputs
39.5	PROGS_EN	BOOL	Program selection
39.6	REQAPP_EN	BOOL	Request application data
39.7	TEACHIN_EN	BOOL	Teach-in
40...41			Reserved
Initiation signals for Read requests			
42.0	OPDAT_EN	BOOL	Basic operating data
42.1	ACT_BL_EN	BOOL	Active NC block
42.2	NXT_BL_EN	BOOL	Next NC block
42.3	BLEXT_EN	BOOL	Actual value block change
42.4	SERVDAT_EN	BOOL	Service data
42.5	OC_ERR_EN	BOOL	Operating error no.

Table B-1 User data block (AW−DB), continued

Absolute address	Variable	Data type	Comment
42.6...43.2			Reserved
43.3	PARRD_EN	BOOL	Parameter/data
43.4	DIGIO_EN	BOOL	Dig. inputs/outputs
43.5	OPDAT1_EN	BOOL	Additional operating data
43.6	APPDAT_EN	BOOL	Application data
43.7	MSRRD_EN	BOOL	Read measured values
Ready signals (for Initiation signals, single settings)			
44.0	SERVO_D	BOOL	Controller enable
44.1	GAUG_FLY_D	BOOL	On-the-fly measuring
44.2...44.4			Reserved
44.5	TRAV_MON_D	BOOL	Rotation monitoring
44.6	PARK_AX_D	BOOL	Parking axis
44.7	SIM_ON_D	BOOL	Simulation on
45.0...45.1			Reserved
45.2	MSR_D	BOOL	Length measurement
45.3	REF_TRIG_D	BOOL	Retrigger reference point
45.4	DI_OFF_D	BOOL	Enable input disabled
45.5			Reserved
45.6	SSW_DIS_D	BOOL	Software limit positions disabled
45.7			Reserved
Ready signals (for Initiation signals, single commands)			
46			Reserved
47.0	MD_D	BOOL	Activate MD
47.1	DELDIST_D	BOOL	Delete residual distance
47.2	SEARCH_F_D	BOOL	Automatic block advance
47.3	SEARCH_B_D	BOOL	Automatic block return
47.4			Reserved
47.5	RESET_AX_D	BOOL	Restart
47.6	AVALREM_D	BOOL	Rescind setting of actual value
47.7			Reserved
Ready signals (for Initiation signals for Write requests)			
48.0	VLEV_D	BOOL	Speed levels 1, 2
48.1	CLEV_D	BOOL	Frequency levels 1, 2
48.2	TRG254_D	BOOL	Setpoint for incremental dimension
48.3	MDI_D	BOOL	MDI block
48.4	MDIFLY_D	BOOL	On-the-fly MDI block
48.5			Reserved
48.6	REFPT_D	BOOL	Set reference point
48.7	AVAL_D	BOOL	Set actual value
49.0	FVAL_D	BOOL	On-the-fly setting of actual value
49.1	ZOFF_D	BOOL	Zero offset

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
49.2			Reserved
49.3	PARCH_D	BOOL	Modify parameters/data
49.4	DIGO_D	BOOL	Digital outputs
49.5	PROGS_D	BOOL	Program selection
49.6	REQAPP_D	BOOL	Request application data
49.7	TEACHIN_D	BOOL	Teach-in
50...51			Reserved
Ready signals (for Initiation signals for Read requests)			
52.0	OPDAT_D	BOOL	Basic operating data
52.1	ACT_BL_D	BOOL	Active NC block
52.2	NXT_BL_D	BOOL	Next NC block
52.3	BLEXT_D	BOOL	Actual value block change
52.4	SERVDAT_D	BOOL	Service data
52.5	OC_ERR_D	BOOL	Operating error read
52.6	OT_ERR_D	BOOL	Operator/traversing error read
52.7	DA_ERR_D	BOOL	Data error read
53.0...53.2			Reserved
53.3	PARRD_D	BOOL	Parameter/data
53.4	DIGIO_D	BOOL	Dig. inputs/outputs
53.5	OPDAT1_D	BOOL	Additional operating data
53.6	APPDAT_D	BOOL	Application data
53.7	MSRRD_D	BOOL	Read measured values
Error signals (for Initiation signals, single settings)			
54.0	SERVO_ERR	BOOL	Controller enable
54.1	GAUG_FLY_ERR	BOOL	On-the-fly measuring
54.2...54.4			Reserved
54.5	TRAV_MON_ERR	BOOL	Rotation monitoring
54.6	PARK_AX_ERR	BOOL	Parking axis
54.7	SIM_ON_ERR	BOOL	Simulation on
55.0...55.1			Reserved
55.2	MSR_ERR	BOOL	Length measurement
55.3	REF_TRIG_ERR	BOOL	Retrigger reference point
55.4	DI_OFF_ERR	BOOL	Enable input disabled
55.5			Reserved
55.6	SSW_DIS_ERR	BOOL	Software limit positions disabled
55.7			Reserved
Error signals (for Initiation signals, single commands)			
56			Reserved
57.0	MD_ERR	BOOL	Activate MD
57.1	DELDIST_ERR	BOOL	Delete residual distance
57.2	SEARCH_F_ERR	BOOL	Automatic block advance

Table B-1 User data block (AW−DB), continued

Absolute address	Variable	Data type	Comment
57.3	SEARCH_B_ERR	BOOL	Automatic block return
57.4			Reserved
57.5	RESET_AX_ERR	BOOL	Restart
57.6	AVALREM_ERR	BOOL	Rescind setting of actual value
57.7			Reserved
Error signals (for Initiation signals for Write requests)			
58.0	VLEV_ERR	BOOL	Speed levels 1, 2
58.1	CLEV_ERR	BOOL	Frequency levels 1, 2
58.2	TRG254_ERR	BOOL	Setpoint for incremental dimension
58.3	MDI_ERR	BOOL	MDI block
58.4	MDIFLY_ERR	BOOL	On-the-fly MDI block
58.5			Reserved
58.6	REFPT_ERR	BOOL	Set reference point
58.7	AVAL_ERR	BOOL	Set actual value
59.0	FVAL_ERR	BOOL	On-the-fly setting of actual value
59.1	ZOFF_ERR	BOOL	Zero offset
59.2			Reserved
59.3	PARCH_ERR	BOOL	Modify parameters/data
59.4	DIGO_ERR	BOOL	Digital outputs
59.5	PROGS_ERR	BOOL	Program selection
59.6	REQAPP_ERR	BOOL	Request application data
59.7	TEACHIN_ERR	BOOL	Teach-in
60...61			Reserved
Error signals (for Initiation signals for Read requests)			
62.0	OPDAT_ERR	BOOL	Basic operating data
62.1	ACT_BL_ERR	BOOL	Active NC block
62.2	NXT_BL_ERR	BOOL	Next NC block
62.3	BLEXT_ERR	BOOL	Actual value block change
62.4	SERVDAT_ERR	BOOL	Service data
62.5	OC_ERR_ERR	BOOL	Operating error read
62.6	OT_ERR_ERR	BOOL	Operator/traversing error read
62.7	DA_ERR_ERR	BOOL	Data error read
63.0...63.2			Reserved
63.3	PARRD_ERR	BOOL	Parameter/data
63.4	DIGIO_ERR	BOOL	Dig. inputs/outputs
63.5	OPDAT1_ERR	BOOL	Additional operating data
63.6	APPDAT_ERR	BOOL	Application data
63.7	MSRRD_ERR	BOOL	Read measured values
64...65			Reserved

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
Error messages/status signals from POS_CTRL			
66	JOB_ERR	INT	Error code SFC 58/59 (POS_CTRL) SFB 52/53 (POS_CTRL)
68.0	JOBBUSY_WR	BOOL	Write job active
68.1	IMPO_WR	BOOL	Write request not possible
68.2	JOBBUSY_RD	BOOL	Read job active
68.3	IMPO_RD	BOOL	Read request not possible
68.4...69.0			Reserved
69.1	JOBRESET	BOOL	Reset status/error
69.2...69.7			Reserved
Diagnostic interrupt data (POS_DIAG)			
70.0	MDL_DEFECT	BOOL	Module/group errors (coming and going)
70.1	INT_FAULT	BOOL	Internal error/hardware error (group error DBB72, 73)
70.2	EXT_FAULT	BOOL	External error
70.3	PNT_INFO	BOOL	External channel error (group error bytes 78)
70.4...70.5			Reserved
70.6	NO_CONFIG	BOOL	Module not initialized
70.7			Reserved
71	MDL_TYPE	BYTE	Module type class for FM 453 = 08H / Channel information available
72.0			Reserved
72.1	COMM_FAULT	BOOL	Communication error (K bus)
72.2			Reserved
72.3	WTCH_DOG_FLT	BOOL	Response from watchdog timer
72.4	INT_PS_FLT	BOOL	Internal supply voltage to the module failed (NMI)
72.5...73.1			Reserved
73.2	EPROM_FLT	BOOL	FEPROM error
73.3	RAM_FLT	BOOL	RAM error
73.4...73.5			Reserved
73.6	HW_INTR_FLT	BOOL	Process interrupt lost
73.7			Reserved
74	POS_ID	BYTE	FM pos. ID (74H)
75	LEN_INFO	BYTE	Length of the diagnostic information (16)
76	CHEN_NO	BYTE	Number of channels (1)
77.0	CH_ERR_VE1	BOOL	Channel error vector
77.1...78.6			Reserved
78.7	OC_ERR_EN1	BOOL	Operating error for channel

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
79..85			Reserved
86	OC_ERR_NO	BYTE	Error number (DS 164) – Detail event class
87	OC_REE_CL	BYTE	Error number (DS 164) – Detail event number
88...89			Reserved
90	OT_ERR_NO	BYTE	Error number (DS 162) – Detail event class
91	OT_ERR_CL	BYTE	Error number (DS 162) – Detail event number
92...93			Reserved
94	DA_ERR_NO	BYTE	Error number (DS163) – Detail event class
95	DA_ERR_CL	BYTE	Error number (DS163) – Detail event number
96	DIAG_ERR	INT	Error code for POS_DIAG (return code SFC 51)
98	MSRM_ERR	INT	Error code for POS_MSRM (return code SFC 59/SFB 52))
100		ARRAY [100 ..139] BYTE	Internal, Reserved
Data for the requests			
140	ZOFF	DINT	Zero offset
144	AVAL	DINT	Set actual value
148	FVAL	DINT	On-the-fly setting of actual value
152	REFPT	DINT	Set reference point
156	TRG254	DWORD	Setpoint for incremental dimension
160	VLEVEL_1	DWORD	Speed level 1
164	VLEVEL_2	DWORD	Speed level 2
168	CLEVEL_1	DWORD	Frequency level 1
172	CLEVEL_2	DWORD	Frequency level 2
MDI block			
176	MDIB	STRUCT	MDI block
+ 0...+1			Reserved
+ 2.0	G_1_EN	BOOL	G function group 1
+ 2.1	G_2_EN	BOOL	G function group 2
+ 2.2...+ 2.3			Reserved
+ 2.4	X_T_EN	BOOL	Position/ dwell
+ 2.5...+ 2.7			Reserved
+ 3.0	V_EN	BOOL	Speed
+ 3.1	M_1_EN	BOOL	M function group 1
+ 3.2	M_2_EN	BOOL	M function group 2
+ 3.3	M_3_EN	BOOL	M function group 3
+ 3.4...+ 3.7			Reserved

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
+ 4	G_1_VAL	BYTE	G function no. of group 1
+ 5	G_2_VAL	BYTE	G function no. of group 2
+ 6...+ 7			Reserved
+ 8	X_T_VAL	DINT	Value for position/dwell
+ 12	V_VAL	DINT	Value for speed
+ 16	M_1_VAL	BYTE	M function no. of group 1
+ 17	M_2_VAL	BYTE	M function no. of group 2
+ 18	M_3_VAL	BYTE	M function no. of group 3
+ 19			Reserved
		END_STRUCT	
Modify parameter/data			
196	PAR_CHAN	STRUCT	Modify parameter/data
+ 0	TYP	BYTE	DB type
+ 1	NUMB	BYTE	Number
+ 2	COUN	BYTE	Quantity
+ 3	JOB	BYTE	Request
+ 4	DATA	ARRAY [200 ..219] BYTE	Data array, structure/data type of Write data as per bytes 1 to 4 of this structure
		END_STRUCT	
Digital inputs/outputs (for read and write)			
220.0	D_IN0	BOOL	Digital input 0
220.1	D_IN1	BOOL	Digital input 1
220.2	D_IN2	BOOL	Digital input 2
220.3	D_IN3	BOOL	Digital input 3
220.4...7			Reserved
221.0	D_OUT0	BOOL	Digital output 0
221.1	D_OUT1	BOOL	Digital output 1
221.2	D_OUT2	BOOL	Digital output 2
221.3	D_OUT3	BOOL	Digital output 3
221.4...7			Reserved
On-the-fly MDI block			
222	MDI_F	STRUCT	On-the-fly MDI block
+ 0...+ 1			Reserved
+ 2.0	G_1_EN	BOOL	G function group 1
+ 2.1	G_2_EN	BOOL	G function group 2

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
+ 2.2...+ 2.3			Reserved
+ 2.4	X_T_EN	BOOL	Position/dwell
+ 2.5...+ 2.7			Reserved
+ 3.0	V_EN	BOOL	Speed
+ 3.1	M_1_EN	BOOL	M function group 1
+ 3.2	M_2_EN	BOOL	M function group 2
+ 3.3	M_3_EN	BOOL	M function group 3
+ 3.4...+ 3.7			Reserved
+ 4	G_1_VAL	BYTE	G function no. of group 1
+ 5	G_2_VAL	BYTE	G function no. of group 2
+ 6...+ 7			Reserved
+ 8	X_T_VAL	DINT	Value for position/dwell
+ 12	V_VAL	DINT	Value for speed
+ 16	M_1_VAL	BYTE	M function no. of group 1
+ 17	M_2_VAL	BYTE	M function no. of group 2
+ 18	M_3_VAL	BYTE	M function no. of group 3
+ 19			Reserved
		END_STRUCT	
Program selection			
242	PROG_NO	BYTE	Program number
243	BLCK_NO	BYTE	Block number
244	PROG_DIR	BYTE	Direction of processing
245			Reserved
Request for application data			
246	CODE_AP1	BYTE	Application data 1
247	CODE_AP2	BYTE	Application data 2
248	CODE_AP3	BYTE	Application data 3
249	CODE_AP4	BYTE	Application data 4
Teach-in			
250	TEA_PROG_NO	BYTE	Program number
251	TEA_BLCK_NO	BYTE	Block number
252	FELD2_INTERN	ARRAY [252...309] BYTE	Internal, Reserved
Basic operating data			
310	ACT_VAL	DINT	Actual position

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
314	SPEED	DWORD	Actual speed
318	REM_DIST	DINT	Residual distance
322	SET_POS	DINT	Setpoint position
326	SUM_OFST	DINT	Sum of active coordinate offset, tool offset, zero offset
330	TRAV_SPE	DWORD	Rotational speed
334...338			Reserved
Active NC block			
342	ACT_BL	STRUCT	Active NC block
+ 0	PROG_NO	BYTE	Program number
+ 1	BLCK_NO	BYTE	Block number
+ 2.0	G_1_EN	BOOL	G function group 1
+ 2.1	G_2_EN	BOOL	G function group 2
+ 2.2	G_3_EN	BOOL	G function group 3
+ 2.3			Reserved
+ 2.4	X_T_EN	BOOL	Position/dwell
+ 2.5	SR_L_EN	BOOL	No. of UP calls
+ 2.6	SR_N_EN	BOOL	UP call
+ 2.7	SKIP_EN	BOOL	Block skip
+ 3.0	V_EN	BOOL	Speed
+ 3.1	M_1_EN	BOOL	M function group 1
+ 3.2	M_2_EN	BOOL	M function group 2
+ 3.3	M_3_EN	BOOL	M function group 3
+ 3.4	TO_EN	BOOL	Tool offset
+ 3.5...+ 3.7			Reserved
+ 4	G_1_VAL	BYTE	G function no. of group 1
+ 5	G_2_VAL	BYTE	G function no. of group 2
+ 6	G_3_VAL	BYTE	G function no. of group 3
+ 7			Reserved
+ 8	X_T_VAL	DINT	Value for position/dwell
+ 12	V_VAL	DINT	Value for speed
+ 16	M_1_VAL	BYTE	M function no. of group 1
+ 17	M_2_VAL	BYTE	M function no. of group 2
+ 18	M_3_VAL	BYTE	M function no. of group 3
+ 19	TO_VAL	BYTE	Tool offset no.
		END_STRUCT	

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
Next NC block			
362	NXT_BL	STRUCT	Next NC block
+ 0	PROG_NO	BYTE	Program number
+ 1	BLCK_NO	BYTE	Block number
+ 2.0	G_1_EN	BOOL	G function group 1
+ 2.1	G_2_EN	BOOL	G function group 2
+ 2.2	G_3_EN	BOOL	G function group 3
+ 2.3			Reserved
+ 2.4	X_T_EN	BOOL	Position/dwell
+ 2.5	SR_L_EN	BOOL	No. of UP calls
+ 2.6	SR_N_EN	BOOL	UP call
+ 2.7	SKIP_EN	BOOL	Block skip
+ 3.0	V_EN	BOOL	Speed
+ 3.1	M_1_EN	BOOL	M function group 1
+ 3.2	M_2_EN	BOOL	M function group 2
+ 3.3	M_3_EN	BOOL	M function group 3
+ 3.4	TO_EN	BOOL	Tool offset
+ 3.5...+ 3.7			Reserved
+ 4	G_1_VAL	BYTE	G function no. of group 1
+ 5	G_2_VAL	BYTE	G function no. of group 2
+ 6	G_3_VAL	BYTE	G function no. of group 3
+ 7			Reserved
+ 8	X_T_VAL	DINT	Value for position/dwel
+ 12	V_VAL	DINT	Value for speed
+ 16	M_1_VAL	BYTE	M function no. of group 1
+ 17	M_2_VAL	BYTE	M function no. of group 2
+ 18	M_3_VAL	BYTE	M function no. of group 3
+ 19	TO_VAL	BYTE	Tool offset no.
		END_STRUCT	
Application data			
382	APP1	DINT	Application data 1
386	APP2	DINT	Application data 2
390	APP3	DINT	Application data 3
394	APP4	DINT	Application data 4
Actual value block change			
398	BLCK_EXT	DINT	Actual value block change

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
Service data			
402	OUT_VAL	DINT	frequency output value
406	ENC_VAL	DINT	pulse output counter
410...414			Reserved
418	FOLL_ERR	DINT	Difference between setpoint and actual position
422			Reserved
426	OSC_ERR	DINT	Setpoint overshoot value/switch adjustment
430			Reserved
Additional operating data			
434	OVERRIDE1	BYTE	Override
435	PROG_NO1	BYTE	NC traversing program no.
436	BLCK_NO1	BYTE	NC block no.
437	LOOP_NO1	BYTE	UP call counter
438	G90_91	BYTE	Active G90/91
439	G60_64	BYTE	Active G60/64
440	G43_44	BYTE	Active G43/44
441	TO_NO	BYTE	Active D number
442.0			Reserved
442.1	LIM_SP	BOOL	Speed limit
442.2... 442.7			Reserved
443.0	LIM_FR	BOOL	Start/stop frequency limited
443.1	LIM_FV	BOOL	Frequency value for acceleration limited
443.2			Reserved
443.3	LIM_FS	BOOL	Acceleration/deceleration value limited
443.4... 445			Reserved
Parameter/data			
446	PAR_RD	STRUCT	Parameter/data
+ 0	TYP	BYTE	DB type
+ 1	NUMB	BYTE	Number
+ 2	COUN	BYTE	Quantity
+ 3	JOB	BYTE	Request
+ 4	DATA1	ARRAY [450 ..469] BYTE	Array, structure/data type according to data, to be read as per bytes 1 to 4 of this structure
		END_STRUCT	

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
470		ARRAY [470...485] BYTE	Internal, Reserved
Measured values			
486	BEGIN_VAL	DINT	Initial value or on-the-fly measured value
490	END_VAL	DINT	Final value
494	LENGTH_VAL	DWORD	Measured length value
Operator control and monitoring			
498	USR	STRUCT	Operator control and monitoring
+ 0.0	BITC_0	BOOL	Write MD
+ 0.1	BITC_1	BOOL	Read MD
+ 0.2	BITC_2	BOOL	MDI block transferred
+ 0.3	BITC_3	BOOL	Prog. sel. transferred
+ 0.4	BITC_4	BOOL	Teach-in transferred
+ 0.5	BITC_5	BOOL	Incremental dimension transferred
+ 0.6	BITC_6	BOOL	Speed levels transferred
+ 0.7	BITC_7	BOOL	Frequency levels transferred
+ 1.0	BITC_8	BOOL	MDI block transferred on-the-fly
+ 1.1	BITC_9	BOOL	Set actual value transferred
+ 1.2	BITC_10	BOOL	Zero offset transferred
+ 1.3...+ 1.4			Reserved
+ 1.5	BITC_13	BOOL	Diagnostic interrupt
+ 1.6	BITC_14	BOOL	Data error
+ 1.7	BITC_15	BOOL	Operator/traversing error
+ 2	MD_NO	WORD	MD number
+ 4	MD_VALUE	DINT	MD value
+ 8	INC_NO	BYTE	Incremental dimension number
+ 9			Reserved
+ 10	PICT_NO	WORD	Display number
+ 12	KEY_CODE	WORD	Keyboard code
+ 14...+15			Reserved
+ 16.0	BITA_0	BOOL	Open-loop control mode
+ 16.1	BITA_1	BOOL	Approach to reference point
+ 16.2	BITA_2	BOOL	Incremental mode (relative)
+ 16.3	BITA_3	BOOL	MDI
+ 16.4	BITA_4	BOOL	Auto/single block mode
+ 16.5	BITA_5	BOOL	Auto mode

Table B-1 User data block (AW–DB), continued

Absolute address	Variable	Data type	Comment
+ 16.6	BITA_6	BOOL	Jog mode
+ 16.7... +17.5			Reserved
+ 17.6	BITA_14	BOOL	Acknowledge error
+ 17.7	BITA_15	BOOL	Acknowledge diagnostic interrupt
		END_STRUCT	

■

List of Abbreviations

C

AS	Automation system
BA	Mode
BA “A/AE”	“Automatic/Automatic single block” mode
BA “REF”	“Reference point approach” mode
BA “SM”	“Incremental approach” mode
BA “STE”	“Open-loop control” mode
BA “T”	“Jogging” mode
BP	Mode parameter
BR	Binary result
CPU	Central Processing Unit of the SIMATIC S7
DB	Data block
DBB	Data block byte
DBX	Data block bit
DB-MD	Data block for machine data
DB-NC	Data block for traversing programs
DB-SM	Data block for increments
DB-SS	Data block for status messages
DB-WK	Data block for tool offset data
DEKL	Detail event class
DENR	Detail event number
DP	Distributed I/O
EMC	Electromagnetic compatibility
EN	Enable (input parameter in LAD representation)
ENO	Enable Output (output parameter in LAD representation)
EPROM	Erasable programmable read-only memory
ESD	Electrostatic sensitive device
EXE	External pulse shaper
FB	Function block

FC	Function
FEPROM	Flash EPROM: Read/Write memory
FM	Function module
HEX	Hexadecimal
HMI	Device for operating and monitoring of a process
I	Input parameter
IM	Interface module (SIMATIC S7)
I/Q	Throughput parameter (initialization parameter)
LAD	Ladder program
LED	Light Emitting Diode
MDI	Manual Data Input
MLFB	Machine-readable order designation
MPI	Multi Point Interface
MSR	Measurement system raster
OB	Organization block
OP	Operator panel
PG	Programming device
PLC	Programmable controller
PS	Power Supply (SIMATIC S7)
PWM	Pulse width modulation
Q	Output parameter
RFG	Controller enable
RPS	Reference point switch
S7-300	PLC of medium performance range
SDB	System data block
SFC	System Function Call (integrated fucntions)
SM	Signal module (SIMATIC S7, e.g. input/output module)
SSI	Synchronous Serial Interface
STEP 7	Programming device software for SIMATIC S7
STL	Statement list
SZL	System status list
TF	Technology function
UP	User program

Index