SIEMENS

SIMATIC

S7-300
FM 351 positioning module

Operating Instructions

05/2011
A5E01092700-02

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

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

⚠WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE
indicates that an unintended result or situation can occur if the relevant information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E01092700-02
Ⓟ 06/2011

Preface

Preface

Scope of this manual

This manual contains the description of the FM 351 positioning module that is valid at the time the manual is released. We reserve the right to describe modifications to the functionality of the FM 351 in a separate product information.

Content of this manual

This manual describes the hardware and software of the FM 351 positioning module.

It comprises:

- Basic information: chapters "Product overview" to "Commissioning the FM 351"
- Reference information: chapters "Machine data and increments" to "Examples"
- Appendices: chapters "Technical specifications", "Connection diagrams", and "Data blocks / Error lists"
- an index

Standards

The SIMATIC S7-300 product series fulfills the requirements and criteria of IEC 61131-2.

Recycling and disposal

The FM 351 is low in contaminants and can therefore be recycled. For ecologically compatible recycling and disposal of your old device, contact a certified disposal service for electronic scrap.

Additional support

If you have any further questions about the use of products described in this manual and do not find the right answers here, contact your local Siemens representative (http://www.siemens.com/automation/partner):

A guide to the technical documentation for the various products and systems is available on the Internet:

- SIMATIC Guide manuals (http://www.siemens.com/simatic-tech-doku-portal)

The online catalog and online ordering systems are also available on the Internet:

- A&D Mall (http://www.siemens.com/automation/mall)

Training center

To help you get started with automation technology and systems, we offer a variety of courses. Contact your regional Training Center or the central Training Center in D-90327 Nuremberg, Germany.

- Internet: SITRAIN homepage (http://www.sitrain.com)

Technical Support

You can access technical support for all A&D projects via the following:

- Online support request form: (http://www.siemens.com/automation/support-request)

Service & Support on the Internet

In addition to our documentation, we offer a comprehensive online knowledge base on the Internet at:

Industry Automation and Drive Technologies - Homepage (http://www.siemens.com/automation/service&support)

There you will find the following information, for example:

- The newsletter that provides up-to-date information on your products.
- The documents you need via our Search function in Service & Support.
- A forum for global information exchange by users and specialists.
- Your local partner for Automation and Drives.
- Information about on-site service, repairs, and spare parts. Much more can be found under "Services".

Table of contents

1 Product overview

1.1 FM 351 positioning module

Description of the FM 351

The FM 351 positioning module is used for controlled positioning with rapid traverse/creep speed drives in the S7-300 automation system. The module has 2 independent channels and, thus, controls one rotary axis or one linear axis at a time. The module supports one incremental encoder or one absolute encoder (SSI) for each channel.

You can operate several FM 351 positioning modules simultaneously. It is also possible to combine with other FM / CP modules. A typical application would be combining with an FM 352 electronic cam controller.

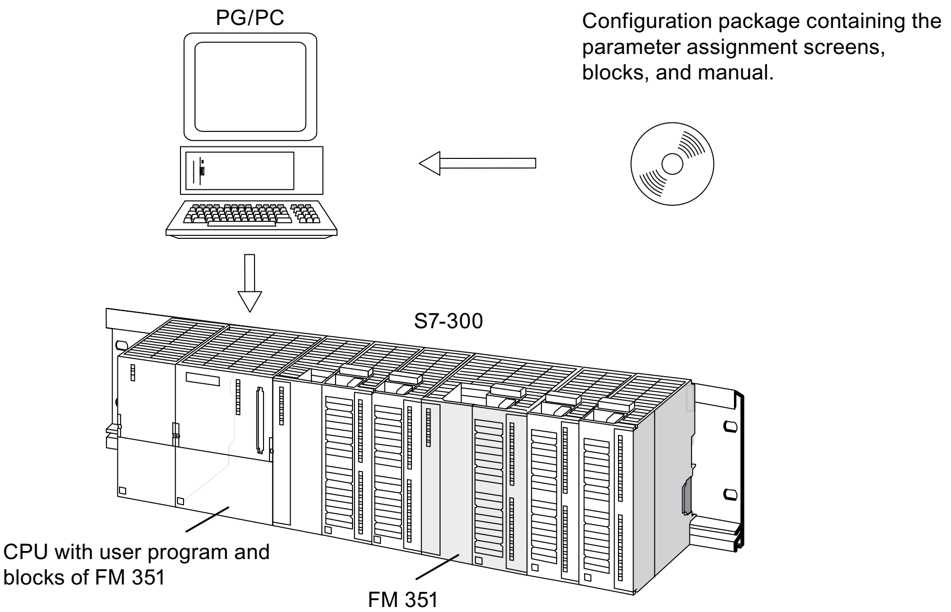


Figure 1-1 Structure of a SIMATIC S7-300 with an FM 351

1.2 Application areas of the positioning module

Overview

- Packaging machines
- Lifting and conveying equipment
- Woodworking machinery

  Example: Controlling delivery processes

  Various wooden parts are processed using a profiling machine. Various work processes and, with that, different cutting heads, are required to machine the wood. The various cutting heads are exchanged by means of a controlled positioning process.
- Paper making machines and printing machines
- Rubber and plastics processing machines

  Example: Simply handling processes

  The injection molded parts in an injection molding machine are removed from the work piece by means of a gripper arm. The gripper arm is controlled by the positioning module.
- Building materials industry
- Machine tools

1.3 Structure of a controlled positioning using an FM 351

Control circuit

The following figure shows the components of a controlled positioning using rapid traverse/creep speed drives.

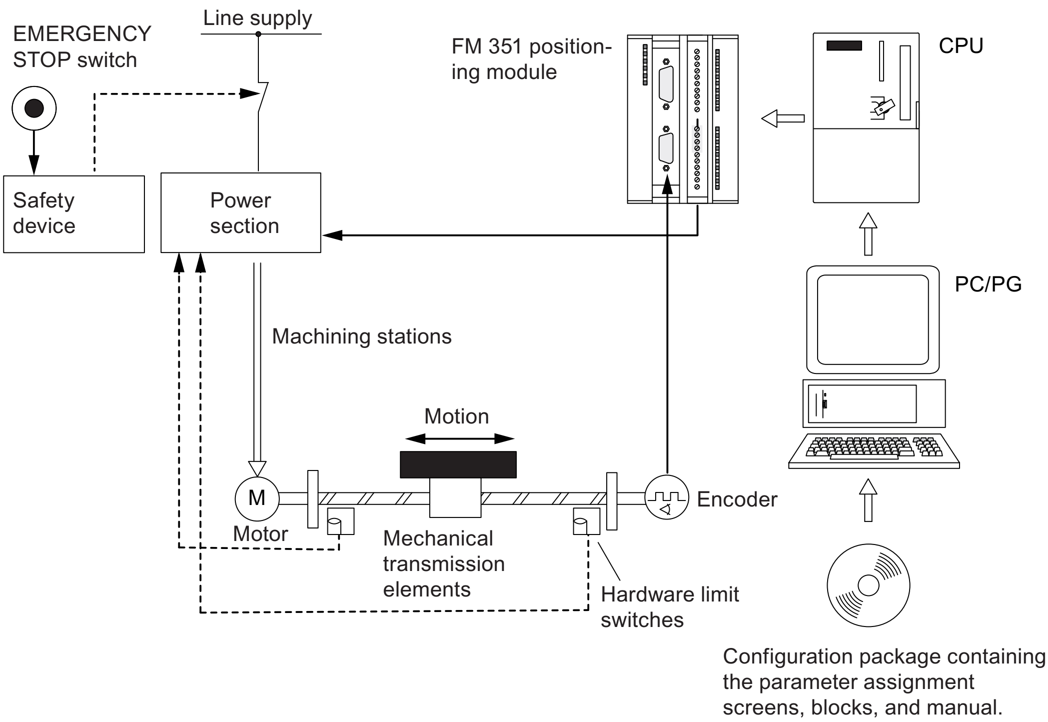


Figure 1-2 Controlled positioning

Power section and safety device

The power section (e.g., a contactor circuit) is controlled by the digital outputs of the FM 351. The FM 351 has 4 control modes (refer to the section entitled "Machine data of the drive (Page 67)").

When the safety device is activated (EMERGENCY STOP switch or hardware limit switch), the power section switches off the motor.

Motor

The motor is controlled by the power section and it drives the axes.

Encoder

The encoder provides distance and direction information. The following encoders can be connected:

- Incremental encoders with 5 V differential signal, symmetrical
- Incremental encoders with 24 V signal, asymmetrical
- SSI absolute encoder

FM 351 positioning module

The FM 351 can position up to 2 axes independently based on the rapid traverse/creep speed process.

The power section is controlled by the 4 digital outputs (refer to the sections entitled "Machine data of the drive (Page 67)" and "Resolution (Page 85)").

The FM 351 positioning module establishes the current actual position value of the axis from the encoder signals, which are proportional to the distance moved (refer to the section entitled "Encoder machine data (Page 77)").

The FM 351 provides the following operating modes and functions:

- "Jog" mode (refer to the section entitled "Configured the "Jog" mode (Page 99)")
- "Reference point approach" mode (refer to the section entitled "Configuring the reference point approach mode (Page 103)")
- "Absolute/relative incremental approach" mode (refer to the section entitled "Configuring the increment drive mode (Page 109)")
- Set actual value (refer to the section entitled "Configuring set actual value / cancel set actual value (Page 117)")
- Set reference point (refer to the section entitled "Configuring set reference point (Page 119)")
- Loop approach (refer to the section entitled "Configuring the loop drive (Page 121)")

CPU

The CPU executes the user program. Data and signals are exchanged between the user program and the module by means of function calls.

Programming device or PC

The programming device or PC is used for

- Assigning parameters:

  You use either the *parameter assignment screens* or the parameter DB to assign parameters for the FM 351 (refer to the section entitled "Content of the parameter DB (Page 184)").
- Programming:

  You program the FM 351 with functions that you can integrate directly in the user program.
- Testing and commissioning:

  You test and commission the FM 351 with the help of the *parameter assignment screens*.

Overview of the positioning module

- 2 axes, types of axis:
  - Linear axis
  - Rotary axis
- 4 digital outputs per axis
- 4 digital inputs per axis
- Typical drives / motors:
  - Standard motor - relay controlled
  - Standard motor on frequency inverter (e.g., Micromaster)
  - Asynchronous motor on power section with vector control
- Position measuring systems:
  - Incremental encoder 5 V, symmetrical
  - Incremental encoder 24 V, asymmetrical
  - SSI absolute encoder
- Monitoring functions:
  - Monitoring the operating range by means of software limit switches
  - Standstill monitoring
  - Encoder monitoring
  - Monitoring for axis movement and target approach

- System environment:
  - Centralized application

    SIMATIC S7-300, from CPU 314 (recommendation: Depends on user memory requirement of the application)

    SIMATIC C7
  - Distributed use with ET 200M
- System integration:
  - Module exchange possible without PG
  - Teleservice possible

Basics of positioning 2

2.1 Controlled positioning

Controlled positioning

Each positioning process is characterized by

- A start position
- The target of the positioning
- Parameters that determine the sequence of the positioning

The target is initially approached at a higher velocity (rapid traverse). At a specified distance from the target, the velocity switches to creep speed. The drive is switched off shortly before the axis reaches the target - also at a specified distance from the target. In doing so, the module monitors the target approach.

The drive is controlled via digital outputs with rapid traverse or creep speed and in the appropriate direction.

2.2 Ranges and switching points of the positioning module

Target

The target is the absolute or relative position on the axis that is approached during a positioning process.

Definition of the switching points and switching ranges

The following areas and positions can be assigned for each controlled positioning:

Range	Description
Operating range	defines the area that you define for your task by means of the software limit switches and the end of the rotary axis.
Changeover difference	defines the distance from the target at which the drive changes from rapid traverse to creep speed.
Changeover point	defines the position at which the drive changes from rapid traverse to creep speed.
Switch-off difference	defines the distance from the target at which the drive is switched off.
Switch-off point	defines the position at which the drive is switched off. The positioning module then adopts the monitoring functions from this point on.
Target range	defines the positioning precision for your application and surrounds the target symmetrically.
Standstill range	defines a symmetrical area around the target that will be monitored by the positioning module.

The following figure shows how the switching points and switching differences can be arranged for a positioning. For simplification purposes, it is assumed that the actual velocity changes linearly over the distance traversed. The emerging ramps can be explained by mechanical inertia or by the parameter assignment possibilities of the power section.

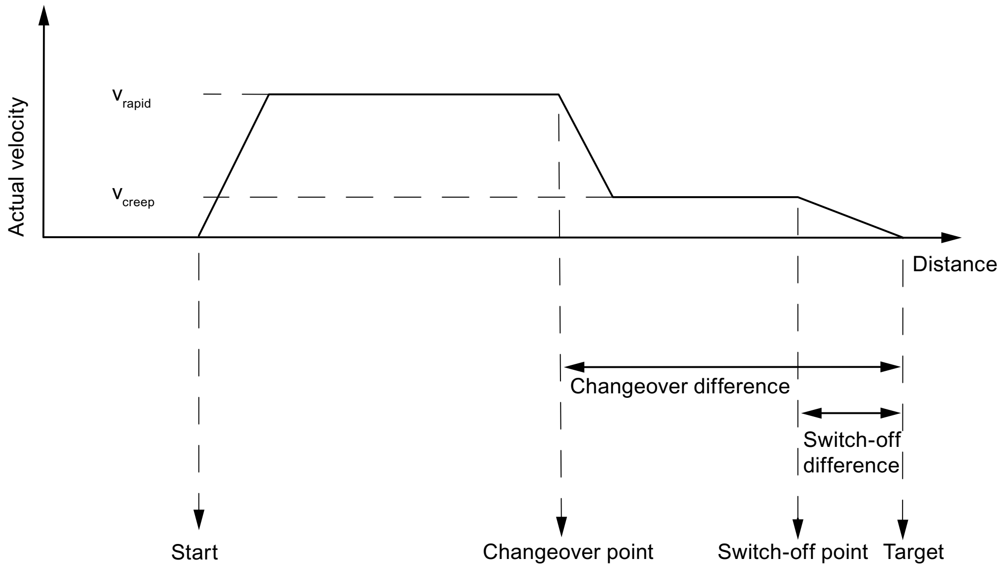
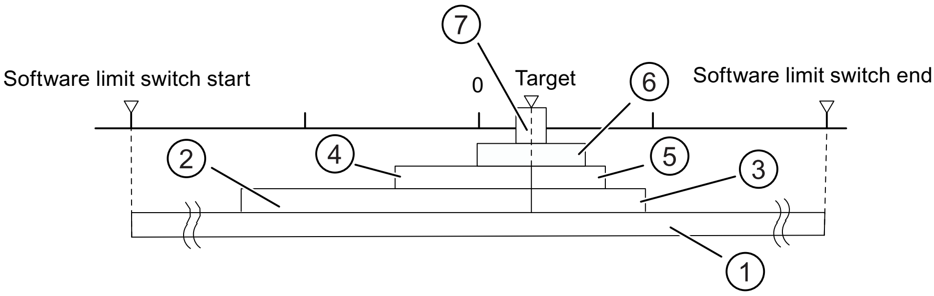


Figure 2-1 Switching points and switching differences

The following figure shows how the switching ranges can be arranged around the target.



① Operating range
② Changeover difference in plus traversing direction
③ Changeover difference in minus traversing direction
④ Switch-off difference in plus traversing direction
⑤ Switch-off difference in minus traversing direction
⑥ Standstill range
⑦ Target range

Figure 2-2 Switching ranges around a target

Installing and removing the FM 351 3

Important safety information

Certain important rules and regulations govern the integrating of an S7-300 with FM 351 in a plant or system. These are described in the Operating Instructions SIMATIC S7-300 CPU 31xC and CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499).

Mounting position of the mounting rails

Horizontal installation of the rail is preferable.

With a vertical installation you must note the restricted ambient temperatures (max. 40 °C).

Establish slots

The FM 351 can be fitted at any slot for signal modules on the mounting rail.

Tools required

You will need a 4.5 mm screwdriver to install or remove the FM 351.

Installing the FM 351 positioning module

1. The FM 351 comes with a bus connector. Insert this on the bus connector of the module to the left of the FM 351. The bus connector is located on the rear, you may have to unfasten the module again.
2. If you want to install further modules to the right of the FM 351, first connect the bus connector of the next module to the right bus connector of the FM 351.

   If the FM 351 is the last module in the line then do not insert a bus connector.
3. Hook the FM 351 onto the mounting rail and swivel it downwards.
4. Screw the FM 351 into place (tightening torque 0.8 ... 1.1 Nm).
5. After installation you can assign a slot number to the FM 351. For this purpose the CPU comes with slot plates.

   The required numbering scheme and the procedure for inserting the slot plates can be found in the Operating Instructions SIMATIC S7-300 CPU 31xC and CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499).
6. Install the shield connection element.

   Order No.: 6ES7 390-5AA00-0AA0

Removing the FM 351 positioning module

1. Switch off the power supply.
2. Switch off the 24 V supply for the FM 351.
3. Switch the CPU to STOP mode.
4. Open the front doors.

   If necessary, remove the labeling strips.
5. Unlock the front connector and remove it.
6. Remove the sub D connector to the encoder.
7. Remove the module fixing screws.
8. Push the module upwards out of the mounting rail and unhook the module.

Wiring the FM 351 4

4.1 Important safety information

Important safety rules

For the safety concept of the plant it is essential that the following switching elements are installed and adapted to the conditions of your plant.

- EMERGENCY STOP switch with which you can switch off the entire plant.
- Hardware limit switch that directly effects the power sections of all drives.
- Motor circuit-breaker

4.2 Description of the encoder interface

Location of the sub D sockets

The figure shows the mounting position and the designation of the sockets on the module. You can connect incremental or absolute encoders (SSI) to the two sub D sockets (refer to the sections entitled "Incremental encoder (Page 131)" and "Absolute encoder (Page 134)").

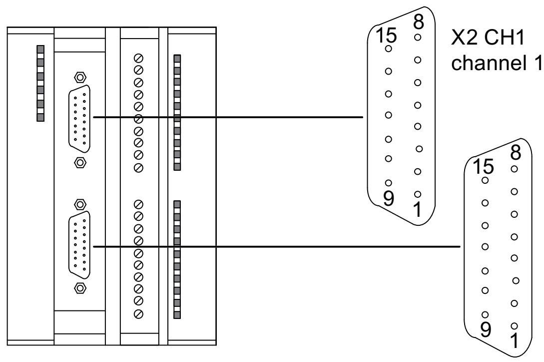


Figure 4-1 Location of the sub D sockets X2 and X3

Assignment of sockets X2 and X3

Pin	Name	Incremental encoders (24 V)	Incremental encoders (5 V)	Absolute encoder
1	A*	Encoder signal A	---	---
2	CLS	---	---	SSI shift clock
3	/CLS	---	---	SSI shift clock inverse
4	B*	Encoder signal B	---	---
5	24 V DC	Encoder supply	Encoder supply	Encoder supply
6	5.2 V DC	---	Encoder supply	Encoder supply
7	M	Ground	Ground	Ground
8	N*	Zero mark signal	---	---
9	Computer unit	Current sourcing/current sinking[1)]	---	---
10	N	---	Zero mark signal	---
11	/N	---	Zero mark signal inverse	---
12	/B	---	Encoder signal B inverse	---
13	B	---	Encoder signal B	---
14	/A / /DAT	---	Encoder signal A inverse	SSI data inverse
15	A / DAT	---	Encoder signal A	SSI data

[1)] Refer to the section entitled "Wiring diagram of the incremental encoder Siemens 6FX 2001-4 (Up = 24 V; HTL) (Page 176)".

4.3 Connecting encoders

Shield connection element

Using the shield connection element you can easily ground all the shielded cables - due to its direct contact to the mounting rail. You will find detailed information in the Operating Instructions SIMATIC S7-300 CPU 31xC and CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499).

Procedure

1. Connecting the connecting cable to the encoder.

   With some encoders it may be necessary to adapt the cable (cable end to the encoder) in accordance with the manufacturer´s instructions.
2. The encoder cables must be shielded.

   Cables A and /A, B and /B, N and /N for the incremental encoder and the cables DAT and /DAT, CLS and /CLS for the absolute encoder must be twisted in pairs.
3. Open the front door and insert the sub D connector onto the FM 351.
4. Secure the connector by tightening the finger screws. Close the front door.
5. Remove the insulation material on the connecting cable and clamp the cable shield into the shield connection element. Use the shield terminal elements for this.

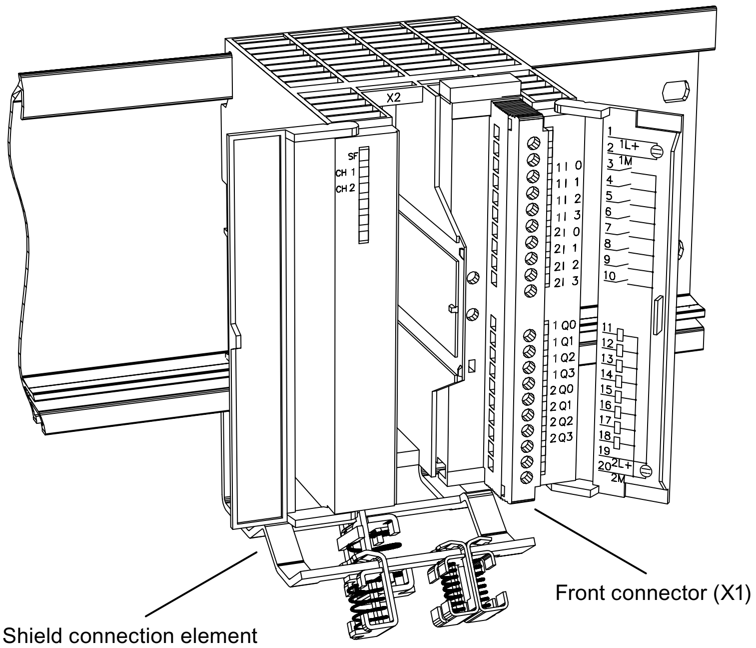


Figure 4-2 Position of the shield connection element

4.4 Description of the front connector

Front connector

You use the 20-pin front connector to connect the supply voltages of the encoder and the digital outputs. The digital outputs and inputs assigned to the channels are also connected.

Front connector (X1) assignment

Terminal	Name	Meaning	Incremental encoder	Absolute encoder
1	1L+	24 V DC auxiliary voltage for the encoder supply		
2	1M	Ground for sensor power supply		
3	1I0	Channel 1: Digital input 0	Reference point switch	Not used
4	1I1	Channel 1: Digital input 1	Reversing switch	Not used
5	1I2	Channel 1: Digital input 2	Enable input	
6	1I3	Channel 1: Digital input 3	Not used	
7	2I0	Channel 2: Digital input 0	Reference point switch	Not used
8	2I1	Channel 2: Digital input 1	Reversing switch	Not used
9	2I2	Channel 2: Digital input 2	Enable input	
10	2I3	Channel 2: Digital input 3	Not used	
11	1Q0	Channel 1: Digital output 0		
12	1Q1	Channel 1: Digital output 1		
13	1Q2	Channel 1: Digital output 2		
14	1Q3	Channel 1: Digital output 3		
15	2Q0	Channel 2: Digital output 0		
16	2Q1	Channel 2: Digital output 1		
17	2Q2	Channel 2: Digital output 2		
18	2Q3	Channel 2: Digital output 3		
19	2L+	24 V DC auxiliary voltage for the load power supply		
20	2M	Mass load power supply		

Auxiliary voltage for the encoder supply (1L+, 1M)

Here you connect a 24 V DC auxiliary voltage for the encoder supply. The reference potential of this supply (1M) is not connected to the ground of the load power supply (2M) in the FM 351.

The 24 V DC auxiliary voltage for the encoder supply is monitored for undervoltage and mass wire break.

The 24 V DC auxiliary voltage for the encoder supply is internally transformed into 5.2 V DC. As a result, on the encoder interface (SUB D socket X2 and X3) 24 V DC and 5.2 V DC are available for the different types of encoder.

Auxiliary voltage for the load power supply (2L+, 2M)

You must connect a 24 V DC auxiliary voltage for the load power supply of the digital outputs to the terminals 2L+ and 2M.

Wiring information for 24 V DC

When wiring, note that the terminals 1L+, 1M and 2L+, 2M must be interconnected so that the module will operate error-free.

If you connect the 1L+, 1M and 2L+, 2M to separate voltage supplies, after a voltage loss on the auxiliary voltage for the load power supply (2L+, 2M) the synchronization of the axes is maintained.

Load power supplies

The DC load power supply must meet the following requirements:

Only a safe, isolated extra-low voltage of ≤ 60 V DC may be used as the load current supply. Safe isolation from mains can be achieved, for example, in accordance with
VDE 0100 Part 410 / HD 384-4-41 / IEC 364-4-41
(as functional extra-low voltage with safe isolation) or
VDE 0805 / EN 60950 / IEC 950
(as safety extra-low voltage SELV) or VDE 0106 Part 101.

8 digital inputs (1I0 to 2I3)

The FM 351 has 4 digital inputs per channel.

You can connect bounce-free switches (24 V p-switching) or contactless sensors (2- or 3-wire proximity switches) to the 8 digital inputs.

The digital inputs are not monitored for short circuit or wire break and are electrically isolated from the encoder supply ground and the CPU ground.

The state of each input can be read off from the respective LED.

8 digital outputs (1Q0 to 2Q3)

The FM 351 has 4 digital outputs per channel.

The power section is controlled by the digital outputs. The function of the digital outputs depends on the control mode. Select the control mode (refer to the section entitled "Machine data of the drive (Page 67)") in the configuration software or in the parameter DB.

The digital outputs are not monitored for short circuit or wire break and are electrically isolated from the encoder supply ground and the CPU ground.

The state of each output can be read off from the respective LED.

Table 4- 1 Functions of the digital outputs, x for channel 1 or 2

Output Q	Control mode			
	1	2	3	4
xQ0	Rapid traverse	Rapid traverse/creep speed	Rapid traverse	Rapid traverse plus
xQ1	Creep speed	Position reached	Creep speed	Creep speed plus
xQ2	Traverse plus	Traverse plus	Traverse plus	Rapid traverse minus
xQ3	Traverse minus	Traverse minus	Traverse minus	Creep speed minus

4.5 Wiring the power section

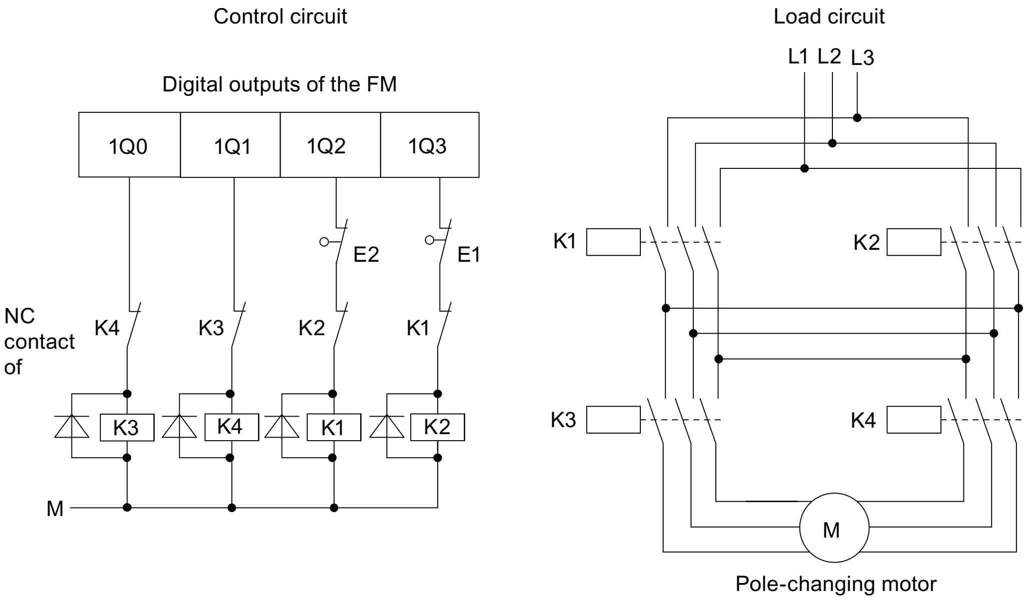
Power section

The power section, e.g. a simple contactor circuit, is connected to the digital outputs of the positioning module and controls the motor.

Contactor circuit

The following figure shows the control and load circuit of a power section.

The functions of the digital outputs correspond to control mode 1(refer to the section entitled "Machine data of the drive (Page 67)").



K1 Direction - plus
K2 Direction - minus
K3 Rapid traverse
K4 Creep speed
E1 Hardware limit switch minus
E2 Hardware limit switch plus

Figure 4-3 Contactor circuit

Operating principle of the contactor circuit

The contactors K1 and K2 control the rotational direction of the motor. Both contactors are mutually locked by means of the opening contacts K2 and K1. The hardware limit switches E1 and E2 are the limit switches minus / plus. The motor is switched off when one of these limit switches is overrun.

The contactors K3 and K4 switch the motor from rapid traverse to creep speed. Both contactors are mutually locked by means of the opening contacts K4 and K3.

CAUTION

Mutually lock the network contactors.

The previous figure shows the mutual locking of the network contactors.

Non-compliance with these instructions could cause a short circuit in the power network.

Note

Direct connection of inductances, e.g, of relays and contactors, is possible without an external protective circuit.

If SIMATIC output circuits can be switched off by means of additionally installed contacts, e.g. relay contacts, additional overvoltage protection devices must be provided for inductances. See the following example of overvoltage protection

Example for overvoltage protection

The following figure illustrates an output power circuit requiring additional overvoltage protectors. DC-activated coils will be wired with diodes or Z diodes.

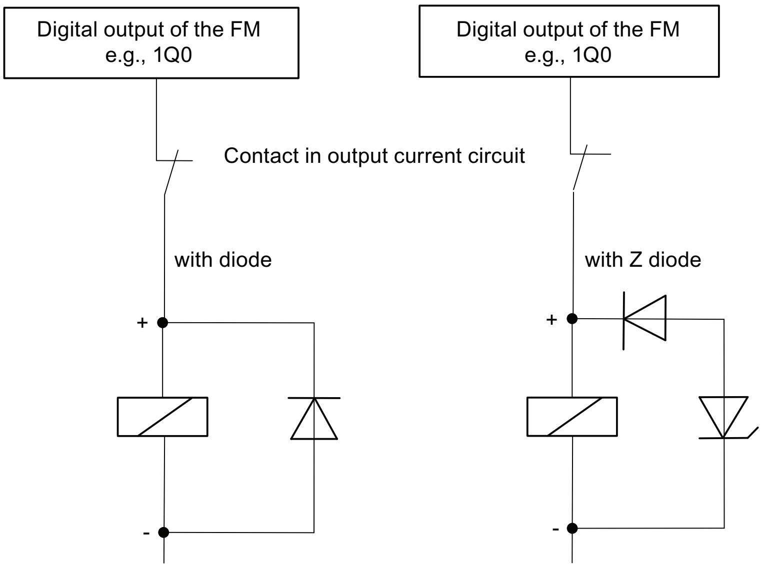


Figure 4-4 Relay contact in the output power circuit

4.6 Wiring Front Connectors

Connecting cables

- As of 100 m cable lengths, the cables for digital inputs and outputs must be shielded.
- The shields of the cables must be arranged on both sides.
- Flexible cable, cross section 0.25 to 1.5 mm^2
- Ferrules are not required. If you do however want to use them, you can use ferrules without insulating collars (DIN 46228, Form A, short design).

Tools required

3.5 mm screwdriver or power screwdriver

Procedure

> **WARNING**
>
> Personal injury and damage to equipment on account of unshielded voltage.
>
> Wiring up the front connector of the FM 351 live could cause injury from an electric shock!
>
> Always switch off the power before you start wiring the FM 351!
>
> If there is no EMERGENCY STOP switch damage could be caused by the connected units.
>
> Install an EMERGENCY STOP switch to be able to switch off the connected drives when using the *configuration software* to operate the FM 351.

1. Strip 6 mm of insulation from the wires. Press on ferrules, if applicable.
2. Open the front panel and place the front connector in the wiring position.
3. Thread the included cable strain relief into the front connector.
4. Connect the strain relief to the connector.
5. If you route the wires out downwards, start the wiring at the bottom. If this is not the case, start at the top. Also screw in terminals that are unassigned.

   The tightening torque is 0.6 to 0.8 Nm.
6. Tighten the strain relief for the cable harness.
7. Push the front connector into the operating position. To do this, press the interlock element.
8. You can complete the labeling field provided and insert it in the front door.

Non-isolation

The ground of the auxiliary voltage for the encoder supply is electrically isolated from the CPU ground. For downward compatibility with FM 351 with order number 6ES7351-1AH01-0AE0, the ground of the auxiliary voltage for the encoder supply (1M) can be connected with the CPU ground (M).

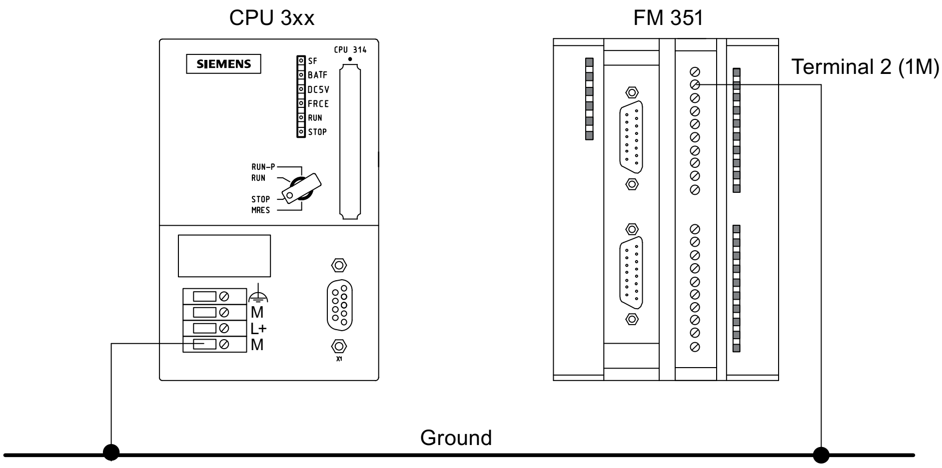


Figure 4-5 Non-isolation

5 Installing the configuration package

Requirement

Before starting to assign parameters for the positioning module, note the requirements in the readme.rtf file, in particular, regarding the required version of STEP 7. The readme.rtf file is available on the included CD.

Content of the configuration package

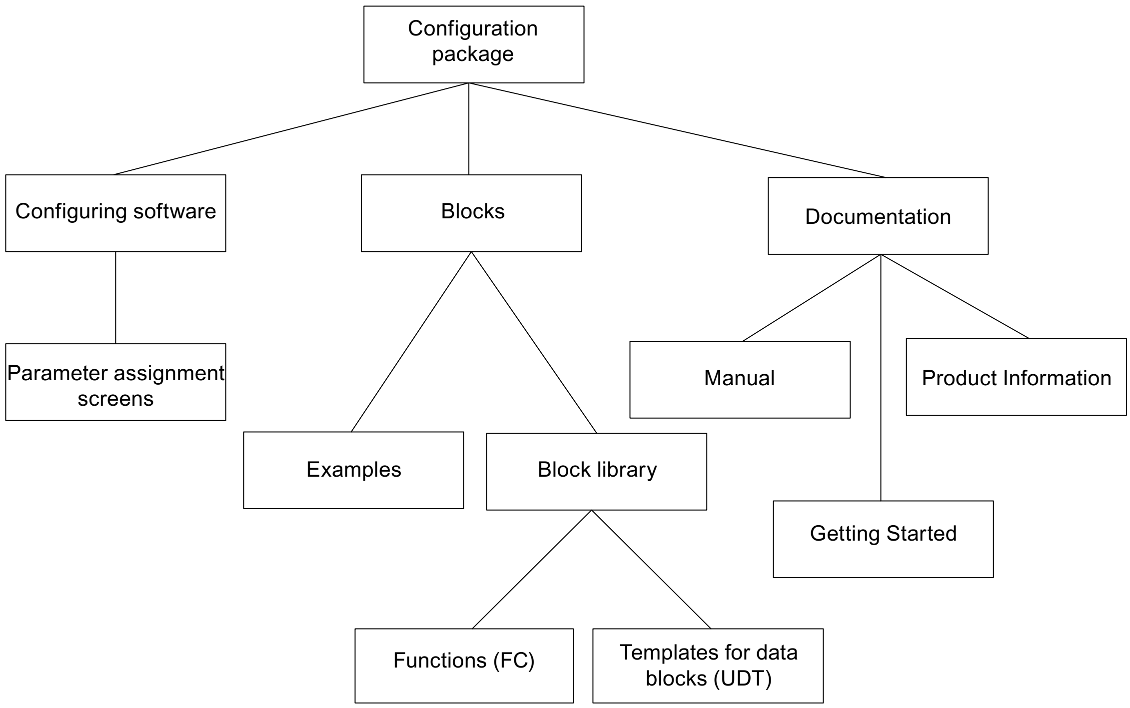


Figure 5-1 Content of the configuration package

Procedure

The entire configuration package can be located on the supplied CD.

1. De-install any already existing configuration packages.
2. Place the CD in the CD drive of your PC or PG.
3. Start Setup.exe and follow the step-by-step setup instructions displayed by the installation program.

Result

The components of the configuration package are installed in the following directories:

- SIEMENS\STEP7\S7LIBS\FMx51LIB: FCs, FBs, and UDTs
- SIEMENS\STEP7\S7FABS: configuration software, Readme, online help
- SIEMENS\STEP7\EXAMPLES: Examples
- SIEMENS\STEP7\S7MANUAL\S7FABS: Getting Started, manuals

Note

If, when installing STEP 7, you chose a directory other than SIEMENS\STEP7, then this directory will be entered.

Programming the FM 351 6

6.1 Overview of the Programming section

If your CPU supports the system blocks SFB 52 and SFB 53 with DPV1 functionality

Then use the blocks from the program folder "FM 351 ABS V2" in the supplied block library to program the FM 351.

In addition to centralized use in the S7-300, these blocks also support distributed use with PROFINET and PROFIBUS DP.

You will find a description in this section.

If your CPU does not support the system blocks SFB 52 and SFB 53 with DPV1 functionality

Then use the blocks from the program folder "FM 351,451 ABS V1" to program the FM 351.

For a description, please refer to the appendix entitled "Programming without SFB 52 and SFB 53 (Page 201)".

6.2 Basics of programming a positioning module

Task

You can assign parameters, control, and commission each channel of the positioning module via a user program. The following sections illustrate how to design a user program to suit your application.

Preparation

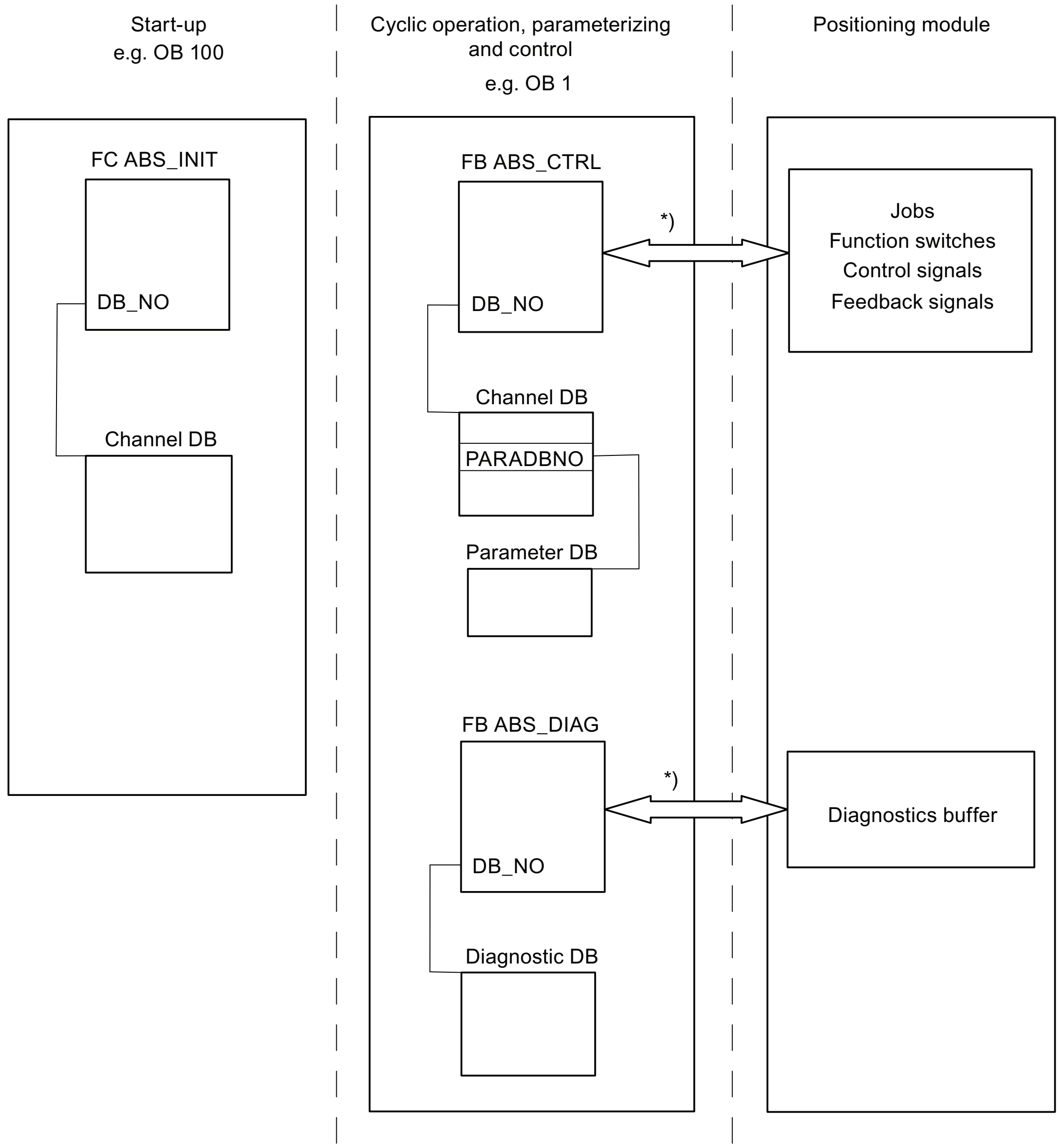
1. Open the block library FMx51LIB in SIMATIC Manager, then copy the required functions (FCs), function blocks (FBs), and block templates (UDTs) to the block container of your project. If the block numbers are already assigned, assign new numbers. The block names are entered unchanged in the symbol table of your S7 program.

Name	Meaning
FC ABS_INIT (FC 0)	You need this FC to initialize the channel DB following a module startup.
FB ABS_CTRL (FB 1)	You need this FB for data exchange and for controlling
FB ABS_DIAG (FB 2)	You need this FB when you process detailed diagnostic information in the program or want to make this information available to an operator control and monitoring system
UDT ABS_CHANTYPE(UDT 1)	You need this UDT to create a channel DB for each channel; this is then used by the FC ABS_INIT and FB ABS_CTRL
UDT ABS_DIAGTYPE (UDT 2)	You need this UDT to create a diagnostic DB for each module; this is then used by the FB ABS_DIAG
UDT ABS_PARATYPE(UDT 3)	You need this UDT to create a parameter DB with parameters; this is then used by the FB ABS_CTRL in order to write or read machine data and increment tables

2. Create the data blocks (DBs) using the UDTs in the block container of your S7 program:
   – a separate channel DB for each channel.
   – If you want to write or read parameters using the user program, you need a separate parameter DB for each channel.
   – If you would like to execute the diagnostics using the user program, you require only a diagnostic DB for each module.
3. Enter the module address in the associated channel DB and, if necessary, also in the corresponding diagnostic DB at the "MOD_ADDR" address (refer to the section entitled Basics of programming a positioning module (Page 202)).
4. Enter the channel number and, if necessary, the number of the parameter DB also in the respective channel DB.

If your programing device or PC is connected to a CPU, you can now download the blocks to the CPU.

The following figure shows you how the positioning module, FCs, DBs and OBs communicate with each other.



*) The module address (channel DB/diagnostic DB) entered in the "MOD_ADDR" parameter is used for accessing the module. We recommend you assign the module address to the channel DB/diagnostic DB in the user program so that the assignment of the module address takes place when you call the user program in OB 100.

Figure 6-1 Data exchange between FCs, FBs, DBs and positioning module

6.3 FC ABS_INIT (FC 0)

Task

The FC ABS_INIT deletes the following data in the channel DB:

- The control signals
- The checkback signals
- The trigger bits, done bits, and error bits of the jobs
- The function switches and their done bits and error bits
- The job management for the FB ABS_CTRL

Call

The function must be executed for each channel after a startup, i.e., after the power supply to the module or CPU is switched on. Call the function, for example, in the start-up OB 100 and the insertion/removal OB 83 or the initialization phase of your user program. This ensures that your user program does not access obsolete data after a CPU or module restart.

Data block used

Channel DB:

The module address must be entered in the channel DB.

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	IN	Number of the channel DB

Return values

The specified function does not supply a return value.

6.4 FB ABS_CTRL (FB 1)

Tasks

With the FB ABS_CTRL you can read the operating data for each channel of the module, assign channel parameters, and control the operation. For these tasks, you use control signals, checkback signals, function switches, and write and read jobs.

Each time it is called, the function block performs the following actions:

- Reading checkback signals:

  The FB ABS_CTRL reads all the checkback signals for a channel and enters them in the channel DB. Because the control signals and jobs are not executed until after this step, the checkback signals reflect the status of the channel before the block was called.

- Job management:

  The FB ABS_CTRL processes the read and write jobs and transmits data between the channel DB, parameter DB, and the module.

- Writing control signals:

  The control signals written to the channel DB are transferred to the module.

Application in the user program

While the FB ABS_CTRL is a multi-instance block, it cannot be used for its own part as a multi-instance in a user block.

Call

The FB ABS_CTRL must be called cyclically for each channel, e.g., in OB 1.

Before you call the FB ABS_CTRL, enter all the data needed to execute the required functions into the channel DB.

Data blocks used

- Channel DB:

  The module address and the channel number must be entered in the channel DB. Incorrect information could result in I/O access errors or in an access to a different module, which in turn gives rise to data corruption.

- Parameter DB:

  If you want to use jobs to write or read the machine data, you need a parameter DB, whose number must be entered in the channel DB.

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	IN	Number of the channel DB
RETVAL	INT	OUT	Return value

Return values

The function provides the following return values:

RETVAL	BR	Description
1	1	At least 1 job is active
0	1	No job is active, no error
-1	0	Error: data error (DATA_ERR) or communication error (JOB_ERR) occurred

Jobs

Data exchange with the module other than the control and checkback signals is handled using jobs.

To deliver a job you set the respective trigger bit in the channel DB and in the case of write jobs also the respective data. You then call FB ABS_CTRL to execute the job.

If you are using the FM 351 centrally, a read job requires exactly one cycle. If you are using the FM 351 as a distributed module, a read job can require several cycles.

On account of the necessary acknowledgements of the module, a write job requires at least 3 calls or OB cycles.

When a job is finished, the block takes back the trigger bit. The next time the block is called, the subsequent job is identified and executed.

For each job, there is not only a trigger bit (extension _EN as in "enable") but also a done bit and an error bit. These have the extension _D as in "done" and _ERR as in "error" in their names. The FB ABS_CTRL updates the done bits and error bits once the processing of a job has ended. After the evaluation or prior to issuing a job, these bits should be set to 0.

If you set the JOBRESET bit, all the done and error bits are reset before the queued jobs are processed. The JOBRESET bit is then reset to 0.

Function switch

The function switches activate and deactivate the states of the channel. A job for writing the function switches will only be executed when there is a change in a switch setting. The setting of the function switch is latched after the job has been executed.

Function switches and jobs can be used at the same time in one FB ABS_CTRL call.

As in the case of the jobs, alongside the function switches there are trigger bits with the ending _ON / _OFF, done bits with the ending _D and error bits with the ending _ERR.

To be able to evaluate the done bits and error bits of the function switches, you should set these bits to 0 before you issue a job to change a function switch.

Order of job processing

You can select several jobs simultaneously. If no jobs are active, the job management of the FB ABS_CTRL searches starting with the MDWR_EN job to determine whether trigger bits are set or changes have been made to function switches. If a job is found, it is processed. Once the job is concluded, the job management searches for the next job to be processed. If the last job ENCVAL_EN has been searched, searching starts over with the MDWR_EN job. This searching process is repeated until all the jobs have been processed.

The jobs are processed in the following technologically appropriate order:

Order	Address in the channel DB	Name	Meaning	Reset from
Write jobs				
1	35.0	MDWR_EN	Write machine data	FB 1
2	35.1 35.2 35.3 36.4	MD_EN DELDIST_EN AVALREM_EN DELDIAG_EN	Enable machine data Delete distance-to-go Cancel set actual value Clear diagnostic buffer	FB 1
3	35.4	TRGL1WR_EN	Write increment table 1	FB 1
4	35.5	TRGL2WR_EN	Write increment table 2	FB 1
5	35.6	REFPT_EN	Set reference point	FB 1
6	 34.0 34.1 34.2	Function switch: PLOOP_ON MLOOP_ON EI_OFF	 Loop approach in plus direction Loop approach in minus direction Do not evaluate enable input	User program
7	35.7	AVAL_EN	Set actual value	FB 1
10	36.2	TRG252_254_EN	Write increment for increment number 254	FB 1
11	36.3	TRG255_EN	Write increment for increment number 255	FB 1

Order	Address in the channel DB	Name	Meaning	Reset from
Read jobs				
12	36.5	MDRD_EN	Read machine data	FB 1
13	36.6	TRGL1RD_EN	Read increment table 1	FB 1
14	36.7	TRGL2RD_EN	Read increment table 2	FB 1
16	37.1	ACTSPD_EN	Read current velocity, distance-to-go, and current increment	FB 1
17	37.2	ENCVAL_EN	Read encoder data	FB 1

This order enables you to initiate a complete positioning operation with a set of jobs and control signals. The jobs start with the writing and activating of machine data, and continue with the setting of the external enable input and, finally, the writing of increments for the incremental approaches.

Control signals

If there is a STOP signal, an operator error or a drive enable is missing, the block resets the control signals START, DIR_M and DIR_P.

You can restart a motion after you have acknowledged the operator error with OT_ERR_A=1. With this acknowledgement you cannot submit any other jobs and control signals.

If there is no operator error pending, the block sets the acknowledgement for the operator error OT_ERR_A to 0.

The block resets the START, DIR_P, and DIR_M start signals when the channel signals the start of the motion, except in "jog" mode.

If the axis parameters are not assigned, the block withholds all the control signals with the exception of the OT_ERR_A operator error acknowledgement.

Jobs and control signals

You can issue several jobs at the same time, also together with the control signals necessary for the positioning. If at least one write job is issued at the same time as the control signals START, DIR_M or DIR_P, the block retains these control signals until the write jobs have been processed.

Jobs during active positioning

If they are issued during a positioning, the write jobs listed in the following table will be retained until the positioning has ended and carried out only after the following call of the block.

Address	Name	Type	Initial value	Comment
34.0	PLOOP_ON	BOOL	FALSE	1 = loop approach in plus direction
34.1	MLOOP_ON	BOOL	FALSE	1 = loop approach in minus direction
34.2	EI_OFF	BOOL	FALSE	1 = do not evaluate enable input
35.1	MD_EN	BOOL	FALSE	1 = enable machine data
35.2	DELDIST_EN	BOOL	FALSE	1 = delete distance-to-go
35.3	AVALREM_EN	BOOL	FALSE	1 = cancel set actual value
35.6	REFPT_EN	BOOL	FALSE	1 = set reference point coordinate
35.7	AVAL_EN	BOOL	FALSE	1 = set actual value
36.4	DELDIAG_EN	BOOL	FALSE	1 = clear diagnostic buffer

Startup

Call FC ABS_INIT at the startup of the module or CPU (refer to the section entitled "FC ABS_INIT (FC 0) (Page 38)"). Among other things, this resets the function switches. The FB ABS_CTRL acknowledges the module startup. During this time, RETVAL and JOBBUSY = 1.

Job status

You can check the status of job execution using the RETVAL return value and the JOBBUSY activity bit in the channel DB. The status of an individual job can be evaluated using the trigger bits, done bits, and error bits of this job.

	RETVAL	JOBBUSY	Trigger bit _EN	Done bit _D	Error bit _ERR
Job active	1	1	1	0	0
Job completed without errors	0	0	0	1	0
Job completed with errors	-1	0	0	1	1
Write job cancelled	-1	0	0	0	1

Response to errors

If faulty data are written during a write job, the channel issues the checkback signal DATA_ERR = 1 to the channel DB. If an error occurs during communication with the module in a write or read job, the cause of error is stored in the JOB_ERR parameter in the channel DB.

- Error with a write job:

  If an error occurs in a job, the trigger bit is canceled and error bit _ERR and done bit _D are set. The trigger bit is also canceled for all write jobs still pending, but only error bit _ERR is set. The pending write jobs are canceled because jobs could pile up in this case.

  The pending read jobs will continue to be processed. JOB_ERR is reset again for each job.

- Error with a read job:

  If an error occurs in a job, the trigger bit is canceled and error bit _ERR and done bit _D are set.

  The read jobs still pending continue to be processed. JOB_ERR is reset again for each job.

For further error information, refer to the JOB_ERR and DATA_ERR parameter descriptions (sections entitled "Diagnosis (Page 137)" and "Data and structure of the diagnostic DB (Page 186)").

Program structure

The following figure shows the basic structure of a user program used to cyclically control a channel of a module following a one-time startup initialization. The RETVAL return value of the FB ABS_CTRL is used in the user program for a general error evaluation.

For every other channel a sequence in accordance with the following figure can be executed parallel and independently.

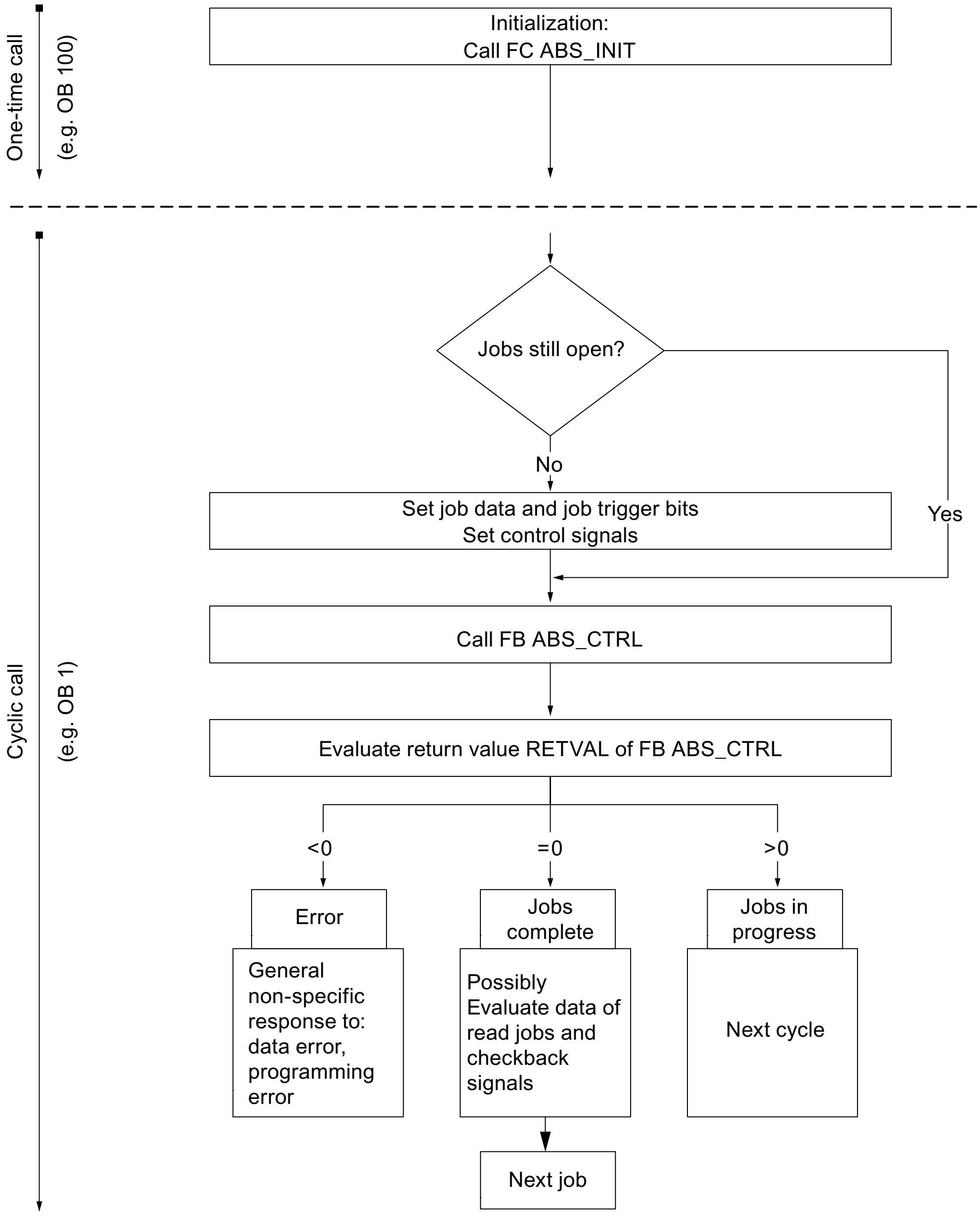


Figure 6-2 General program structure

6.5 FB ABS_DIAG (FB 2)

Tasks

Use FB ABS_DIAG to read the diagnostic buffer of the module and make it available for display on an operator control and monitoring system or for a programmed evaluation.

Application in the user program

While the FB ABS_DIAG is a multi-instance block, it cannot be used for its part as a multi-instance in a user block.

Call

The block must be called cyclically, e.g., in OB 1. An additional call in an alarm OB is not permitted. At least two calls (cycles) are required for a complete execution of the function.

The function reads out the diagnostic buffer when checkback signal DIAG = 1 in the channel DB indicates a new entry in the diagnostic buffer. After the diagnostic buffer is read, the DIAG bit in the channel DB of the module is set to 0.

Data block used

Diagnostic DB:

The module address must be entered in the diagnostic DB. The latest entry in the diagnostic buffer will be entered in the structure DIAG[1] and the oldest entry in the structure DIAG[9].

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	IN	Number of the diagnostic DB
RETVAL	INT	OUT	Return value

Return values

The block returns the following return values in in word 302 in the RETVAL parameter of the diagnostic DB:

RETVAL	BR	Description
1	1	Job active
0	1	No job is active, no error
-1	0	Error

Jobs

You can read the diagnostic buffer independent of a new entry by setting the trigger bit DIAGRD_EN in the diagnostic DB. After the diagnostic buffer is read, the trigger bit is set to 0.

Carry out this job after a CPU start-up and a module start-up. This way you can ensure that the content of the diagnostic DB corresponds with the content of the module´s diagnostic buffer, even if the module has not made a new entry in the diagnostic buffer.

Start

The block does not perform any startup processing.

Response to errors

In the case of a faulty execution, the cause of the error can be found in the diagnostic DB in the JOB_ERR parameter (refer to the sections entitled "Diagnosis (Page 137)" and "Data and structure of the diagnostic DB (Page 186)").

6.6 Data blocks

6.6.1 Templates for data blocks

Block templates UDT

For each data block there is a block template UDT stored in the provided library (FMx51LIB). From these UDTs you can create data blocks with any numbers or names.

6.6.2 Channel DB

Task

The channel DB (refer to the section entitled Content of channel DB (Page 179)) is the data interface between the user program and the positioning module. It contains and accepts all data required for control and operation of a channel.

Structure

The channel DB is divided into different areas:

Channel DB
Module address *)
Channel number
Number of the parameter DB
Control signals
Checkback signals
Function switch
Trigger bits for write jobs
Trigger bits for read jobs
Done bits
Error bits
Job management for functions
Data for jobs

*) You can also enter the address using the configuration software

6.6.3 Diagnostic DB

Task

The diagnostic DB (refer to the section entitled Data and structure of the diagnostic DB (Page 186)) is the data storage for the FB ABS_DIAG and includes the diagnostic buffer of the module that has been processed by this function block.

Structure

Diagnostic DB
Module address
Internal data
Job status
Trigger bit
Processed diagnostic buffer

6.6.4 Parameter DB

Task

If you want to change the machine data and increment tables during operation, you require a parameter DB (refer to the section entitled Contents of parameter DB (Page 184)) in which this data is stored. The parameters can be changed from the user program or from an operator control and monitoring system.

You can export the data displayed in the configuration software into a parameter DB. You can also import a parameter DB into the configuration software and view it there.

Each module channel can have several sets of parameter assignment data, e.g., for different recipes. You can switch among these in your program.

Structure

Parameter DB
Machine data
Increment tables

6.7 Technical specifications of the FCs, FBs, and DBs for the FM 351

Technical specifications

The following table offers you an overview of the technical specifications for the functions and data blocks.

Table 6- 1 Technical specifications of the FM 351 blocks

No.	Block name	Version	Assignment in load memory (bytes)	Assignment in work memory (bytes)	Assignment in local data area (bytes)	MC7 code / data (bytes)	Called system functions
FC 0	FC ABS_INIT	1.0	184	130	2	94	
FB 1	FB ABS_CTRL	1.0	4548	4176	34	4140	SFB 53: WRREC, SFB 52: RDREC
FB 2	FB ABS_DIAG	1.0	1800	1658	42	1622	SFB 52: RDREC
	Channel DB	-	638	184	-	148	
	Parameter DB	-	840	556	-	520	
	Diagnostic DB	-	524	388	-	352	

Module cycle

The checkback signals of a channel are updated by the module every 8 ms.

6.8 Quicker access to module data

Application

In special applications or in an alarm level, a particularly fast access to checkback and control signals could be required. You can reach this data directly via the input and output areas of the module.

To coordinate startup following each module startup, e.g., after inserting a module or after CPU STOP → RUN, you must call the FB ABS_CTRL until the end of the startup is indicated by RETVAL = 0. Then, do not use the FB ABS_CTRL any more.

Note

It is not possible to use the FB ABS_CTRL together with a write access.

Reading checkback signals by means of direct access

The byte addresses must be specified relative to the start address of the outputs of the respective channel. The names of the parameters correspond to the names in the channel DB (refer to the section entitled "Content of the channel DB (Page 179)").

Start address channel 1 = start address of the module

Start address channel 2 = start address of the module + 8

In STL, you access the data using the commands PEB (read 1 byte), PEW (read 2 bytes) and PED (read 4 bytes).

Address	Bit number							
	7	6	5	4	3	2	1	0
Byte 0	PARA	Internal	Internal	DATA_ERR	OT_ERR	DIAG	Internal	Internal
Byte 1	CHGOVER	CUTOFF	ZSPEED	SPEED_OUT	0	WAIT_EI	WORKING	ST_ENBLD
Byte 2	MODE_OUT							
Byte 3	POS_RCD	0	0	0	GO_P	GO_M	0	SYNC
Byte 4	ACT_POS							
Byte 5								
Byte 6								
Byte 7								

Example: Actual position value ACT_POS

The start address of the module is 512.

STL	Description
L PED 516	Read the current actual position value (ACT_POS) from channel 1 //with direct access: //Start address of the channel + 4

Writing control signals using direct access

The byte addresses must be specified relative to the start address of the inputs of the respective channel. The names of the parameters correspond to the names in the channel DB (refer to the section entitled "Content of the channel DB (Page 179)").

Start address channel 1 = start address of the module

Start address channel 2 = start address of the module + 8

In STL, you access the data using the commands PAB (write 1 byte), PAW (write 2 bytes) and PAD (write 4 bytes).

Address	Bit number							
	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	OT_ERR_A	0	0	0
Byte 1	DRV_EN	0	0	0	DIR_P	DIR_M	STOP	START
Byte 2	MODE_IN							
Byte 3	MODE_TYPE							
Byte 4	Reserved							
Byte 5								
Byte 6								
Byte 7								

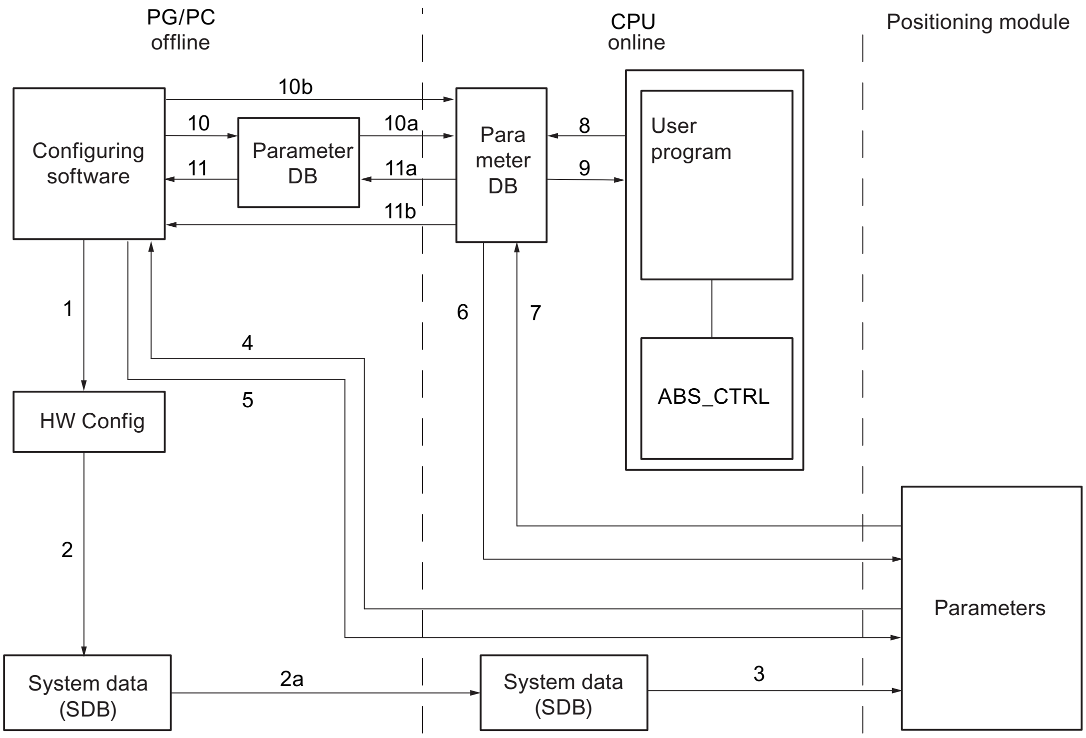
Example: START signals of channel 2

The start address of the module is 512.

STL	Description
L 2#10001000	//Set DRV_EN and DIR_P to 1
T PAB 521	//Write signals with direct access for channel 2: //Start address of the module + 8 + 1

6.9 Parameter transmission paths

Parameters include the following machine data and increments.



1 Saving parameters in the configuration software.

2 Save and compile the HW configuration.

2a Download the HW configuration to the CPU. The CPU automatically carries out step 3.

3 The CPU writes the parameters when system parameter assignments are carried out for the module.

4 With the command "download target system to PG" download the parameters of a channel of the module to the PG.

5 With the command "download target system" download the parameters from the configuration software into a channel of the module.

6 Write parameters using jobs of the user program in a channel of the module.

7 Read parameters using jobs of the user program from a channel of the module.

8 Store parameters from the user program into the online DB.

9 Read in parameters from the online DB into the user program.

10 Export parameters from the configuration software to the offline DB.

10a Download the offline DB to the CPU.

10b Export parameters from the configuration software to the online DB.

11 Import parameters from the offline DB into the configuration software.

11a Load parameters from the online DB to the PG.

11b Import parameters from the online DB into the configuration software.

Figure 6-3 Parameter transmission paths

Some applications for transmission of parameters:

- You process the parameters using the configuration software. The channels of the module should then be automatically assigned upon start-up.

  Carry out steps 1, 2, and 2a.
- You change the parameters during commissioning in debug mode in the configuration software:

  Carry out steps 4 and 5.
- The parameters changed during commissioning should then be automatically loaded upon start-up:

  Carry out steps 1, 2, and 2a.
- You create the parameters using the configuration software. The channels of the module should be assigned upon start-up only by the user program via the data blocks:

  Carry out steps 10, 10a and 6 or 10b and 6.
- You would like to create convenient stored data for recipes:

  Carry out steps 10 and 10a.
- You create the parameters using the configuration software. These should be made available to the user program for temporary changes.

  Carry out steps 1, 2 and 2a for the automatic parameter assignment.

  Carry out the steps 10, 10a, 9, 8 and 6 for access by the user program.
- You change existing parameters solely via the user program:

  Carry out steps 7, 9, 8 and 6.
- You would like to use the configuration software to view the data that has been changed via the user program:

  Carry out steps 11a and 11 or just 11b.
- The parameters changed by the user program should also be automatically loaded upon start-up:

  Carry out steps 11b or 11a, 11 and then 1, 2, 2a.

Commissioning the FM 351 7

Important information

Please observe the points listed in the following warning.

> **WARNING**
>
> Personal injury and equipment damage can occur.
>
> To prevent personal injury and damage to equipment please observe the following points:
>
> - Install an EMERGENCY STOP switch in the immediate vicinity of the computer.
>
>   This is the only way to ensure that the system can be switched off safely in the event of a computer or software failure.
> - Install hardware limit switches that have a direct effect on the power sections of all drives.
> - Always make sure to prevent access to the plant areas containing moving parts.
> - Concurrent controlling and monitoring of the FM 351 from your program and on the "Debug > Commissioning" screen can lead to conflicts with unforeseeable effects. Therefore, always switch the CPU to STOP mode when you are working in the Debug screen, or deactivate your program.

Setting up a project

Set up a project under *STEP 7*.

The following describes the sequence of steps via the SIMATIC Manager (without using the wizards).

Step	Action	✓
1	Install the configuration package (if this has not already happened).	❐
2	In SIMATIC Manager, set up a new project (**File > New**).	❐
3	Insert a station in your project (**Insert > Station**).	❐
4	Select the station, then run the "HW Config" configuration interface by double-clicking "Hardware".	❐
5	Enter your hardware structure in a subrack with: • Power supply (PS) module • CPU • Function module (FM 351)	❐
6	Save this hardware configuration in HW Config (**Station > Save**).	❐

HW installation and wiring

In this first section you install the FM 351 in your S7-300 and wire up the external peripheral elements.

Step	Action	✓
1	**Installing the FM 351** (refer to the section entitled "Installing and removing the FM 351 (Page 19)") Hang the module onto the rail.	 ❐
2	**Wiring the FM 351** (refer to the section entitled "Wiring the FM 351 (Page 21)") • Wiring up the front connector of the FM 351: – Auxiliary voltage for the encoder supply – Auxiliary voltage for the load power supply – Digital inputs – digital outputs • Encoder connection	 ❐ ❐ ❐ ❐ ❐
3	**Checking the safety-related switches** Check the function of the following: • EMERGENCY STOP switches • Hardware limit switches	 ❐ ❐
4	**Front connector** The front connector must be engaged.	 ❐
5	Check the shielding of the individual cables.	❐
6	**Switch on the voltage supply** Switch the CPU to STOP mode (safer mode). Switch on the 24 V supply voltage for the auxiliary voltage.	 ❐ ❐

Prepare for programming

Set up the necessary blocks in your project if you wish to access the module via the user program.

Step	Action	✓
1	In the SIMATIC Manager, select the library FMx51LIB (**File > Open > Libraries**).	❐
2	From the program folder FM 351 ABS V2, copy the FC 0 function and the FB 1 block to the blocks folder.	❐
3	Create a channel DB from the template UDT 1 for each channel and enter the channel number.	❐
4	If you want to use a programmed diagnostic evaluation, copy FB 2 and create a diagnostic DB for each module.	❐
5	If you wish to read or write machine data and increment tables in the user program, you require UDT 3 in order to create a parameter DB for each channel.	❐

Parameter assignment via the configuration software

If you would like to recommission the module, you can assign parameters for this using the parameter assignment screens of the configuration software.

Step	Action	✓
1	Select the line in the subrack with the module FM 351.	❐
2	Now double-click to call the parameter assignment screens for the FM 351.	❐
3	You can change the following settings under **File > Properties**: • General You can change the names and enter a comment. • Addresses If you change the start address, you must also change the end address. **Note the module address that you are shown.** • Basic parameters You can set the type of alarm.	❐
4	Set the respective parameters in the dialog screens **Drive, Axis, Encoder** and **Increments**.	❐
5	You can create your channels under **Edit > Create channel**.	❐
6	Save the parameter assignment with **File > Save**.	❐
7	Close the parameter assignment screens with **File > End**.	❐
8	Save the hardware configuration in HW Config with **Station > Save and compile**.	❐
9	Create an online connection to the CPU and download the hardware configuration to the CPU. In doing so, the parameter assignment data is transferred to the FM 351.	❐

Debug and commissioning

You can use the parameter assignment screens of the configuration software to test your entries and changes made so far.

Step	Action	✓
1	Use the screens **Debug > Commissioning**, **Debug > Error evaluation** and **Debug > Service** to test your data for commissioning.	❐
2	You can change any incorrect machine data in the **Debug > Commissioning** screen. These changes will be valid until the next transition of the CPU from STOP to RUN.	❐
3	You can save the corrected machine data in the CPU by repeating steps 6 to 9 of the previous sequence.	❐

Test steps for modes, jobs and function switches

Use the following tests to check the correct parameter assignment of the FM 351.

<table>
<tr><th>Step</th><th colspan="2">Action</th><th>✓</th></tr>
<tr><td rowspan="3">1</td><td colspan="2">Synchronize the axis</td><td>❒</td></tr>
<tr><td>• Incremental encoder
– Select "Set reference point". To do this enter the required value (refer to the section entitled "Configuring set reference point (Page 119)").
or
– Select the "reference point approach" mode (refer to the section entitled "Configuring the reference point approach mode (Page 103)").</td><td>• Absolute encoder
– In principle, the FM 351 should be synchronized immediately after the parameter assignment.
– Perform an the absolute adjustment (refer to the section entitled "Determining the absolute encoder adjustment (Page 82)").</td><td>❒</td></tr>
<tr><td colspan="2">Check the actual status of the axis. The physical position must match the value output on the display.</td><td>❒</td></tr>
<tr><td rowspan="3">2</td><td colspan="2">Select Jog mode.</td><td>❒</td></tr>
<tr><td colspan="2">• Check the correct interconnection of the outputs (control mode) and the actual value.
– Traverse in creep speed in the plus and minus direction.
– Traverse in rapid traverse in the plus and minus direction.</td><td>❒</td></tr>
<tr><td colspan="2">• Check the encoder resolution (refer to the section entitled "Resolution (Page 85)").
– Traverse the drive a defined distance in a defined direction.
The actual traversing distance must agree with the value indicated in the Debug > Commissioning screen.</td><td>❒</td></tr>
<tr><td>3</td><td colspan="2">Select Incremental approach mode
• absolute with increment number 255
– Check the process with the defined increment dimension
– and use the increment 255 to check the changeover and switch-off differences against the circumstances of your system.</td><td>

❒
❒</td></tr>
<tr><td>4</td><td colspan="2">Test the other function switches and jobs in accordance with your application cases
• e.g., loop approach, set actual value</td><td>❒</td></tr>
</table>

Note

If you are using the FM 351 on PROFIBUS DP, the CPU must be in RUN mode for testing and commissioning. Otherwise you will not be able to control the FM 351.

Note

If, in the commissioning screen, you set the drive enable while the CPU is in STOP mode, and then exit all the parameter assignment screens, the drive enable will be cancelled.

Preparing channel DB

Step	Action	✓
1	Open the channel DB.	❐
2	Check the following entries: • the module address in the MOD_ADDR parameter (refer to the section entitled Basics of programming a positioning module (Page 36)) • the channel number in the CH_NO parameter • if applicable, the number of the parameter DB in the PARADBNO parameter	 ❐ ❐ ❐
3	Save the channel DB (**File > Save**).	❐

Preparing the diagnostic DB

Step	Action	✓
1	Open the diagnostic DB.	❐
2	Make sure that the module address is entered in the MOD_ADDR parameter (refer to the section entitled Basics of programming a positioning module (Page 36)).	❐
3	Save the diagnostic DB (**File > Save**).	❐

Integrating blocks

Step	Action	✓
1	Integrate the required functions and blocks in your user program.	❐

Downloading blocks to the CPU

Step	Action	✓
1	In the SIMATIC Manager, select the blocks and then download them with **Target system > Download to the CPU.**	❐

See also

Important safety information (Page 21)

Machine data and increments 8

8.1 Writing and reading machine data and increment tables

Changing and reading out parameters during operation

This chapter describes how you can use the user program to change and read out parameters during operation.

All parameters are stored in the parameter DB:

- Machine data is located in the parameter DB at the addresses 4.0 to 116.0.
- Increment tables are located in the parameter DB at the addresses 120.0 to 516.0.

The number of the parameter DB must be entered in the respective associated channel DB.

You can enter the parameters with the DB Editor or also easily using the dialog screens "Drive", "Axis", "Encoder" and "Increments" and then write in the parameter DB with the "Export" function.

You can use the function "Import" to import parameters that are in an already available parameter DB into the dialog screens.

Writing, enabling and reading machine data

You can use the machine data to adapt the positioning module to the axis and the encoder.

First parameter assignment

If the channel still contains no machine data, proceed as with a first parameter assignment without the parameter assignment screens in the following way:

1. Enter the new values in the parameter DB and then save these.
2. Download the parameter DB to the CPU.
3. Set the following trigger bit in the channel DB for the job:
   – Write machine data (MDWR_EN).
4. Call the FB ABS_CTRL block in the cyclic user program.

Modifying machine data

Proceed as follows to change machine data using the user program:

1. Enter the new values in the parameter DB.
2. Set the trigger bits in the channel DB for the jobs:
   – Write machine data (MDWR_EN)
   – Enable machine data (MD_EN)
3. Call the FB ABS_CTRL block in the cyclic user program.

If you set the trigger bits for these jobs all at once, the FB ABS_CTRL makes sure the jobs are processed in the correct order.

Otherwise change the machine data always in the following sequence:

- Write machine data
- Enable machine data

Reading machine data

In order to read the current machine data from a channel, proceed as follows:

1. Set the following trigger bit in the channel DB:
   – Read machine data (MDRD_EN)
2. Call the FB ABS_CTRL block in the cyclic user program.

The current machine data is then stored in the parameter DB on the CPU.

Excerpt from the channel DB

Address	Name	Type	Initial value	Comment
35.0	MDWR_EN	BOOL	FALSE	1 = write machine data
35.1	MD_EN	BOOL	FALSE	1 = enable machine data
36.5	MDRD_EN	BOOL	FALSE	1 = read machine data

Writing and reading increment tables

First parameter assignment

If the channel still contains no increment tables, proceed as with a first parameter assignment without the configuration software in the following way:

1. Enter the new values in the parameter DB and then save these.
2. Download the parameter DB to the CPU.
3. Set the trigger bits in the channel DB for the jobs:
   – Write increment table 1 (TRGL1WR_EN) and / or increment table 2 (TRGL2WR_EN)
4. Call the FB ABS_CTRL block in the cyclic user program.

Changing increment tables

Proceed as follows to change increment tables using the user program:

1. Enter the new values in the parameter DB.
2. Set the trigger bits in the channel DB for the jobs:
   – Write increment table 1 (TRGL1WR_EN) and / or increment table 2 (TRGL2WR_EN)
3. Call the FB ABS_CTRL block in the cyclic user program.

Reading increment tables

In order to read the increment tables from a channel, proceed as follows:

1. Set the trigger bits in the channel DB for the jobs:
   – Read increment table 1 (TRGL1RD_EN) and / or increment table 2 (TRGL2RD_EN)
2. Call the FB ABS_CTRL block in the cyclic user program.

   The increment tables are then stored in the parameter DB on the CPU.

Excerpt from the channel DB

Address	Name	Type	Initial value	Comment
35.4	TRGL1WR_EN	BOOL	FALSE	1 = write increment table 1 (1 ... 50)
35.5	TRGL2WR_EN	BOOL	FALSE	1 = write increment table 2 (51 ... 100)
36.6	TRGL1RD_EN	BOOL	FALSE	1 = read increment table 1 (1 ... 50)
36.7	TRGL2RD_EN	BOOL	FALSE	1 = read increment table 2 (51 ... 100)

Note

If synchronization-relevant parameters have been changed, when the machine data is enabled the module will carry out the following actions for the respective channel:

- the synchronization is deleted
- the function switches and zero offset will be cancelled
- all the machine data and increment tables so far will become invalid

Synchronization-relevant parameters are:

- Axis type
- End of rotary axis
- Encoder type
- Distance per encoder revolution
- Increments per encoder revolution
- Number of revolutions
- Reference point coordinate
- Absolute encoder adjustment
- Type of reference point approach
- Counting direction

8.2 System of units

Choosing a system of units

In the configuration software of the positioning module, for the input and output of the data you can select from the following systems of units:

- mm (default)
- inch
- degrees

Note

If you change the system of units in the parameter assignment screens under STEP 7, the values are converted into the new system. This may lead to rounding errors.

If you change the system of units programmed via the jobs "Write machine data" and "Enable machine data", the values will not be automatically converted.

System of units in the parameter DB

Address	Name	Type	Initial value	Comment
8.0	UNITS	DINT	L#1	**System of units** 1 = 10^{-3} mm 2 = 10^{-4} inch 3 = 10^{-4} degrees 4 = 10^{-2} degrees 6 = 10^{-3} degrees

Standard system of units

In this manual we will always use the mm system of units to specify the **Limits**. To determine the limits in the other systems of units, carry out the following conversion:

To convert from		you calculate
mm → inch		Limit (inch) = limit (mm) × 0.1 [1)]
mm → degrees	10^{-4} (4 decimal places)	Limit (degrees) = limit (mm) × 0.1
	10^{-3} (3 decimal places)	Limit (degrees) = limit (mm) × 1
	10^{-2} (2 decimal places)	Limit (degrees) = limit (mm) × 10

[1)] The number of decimal places affects the number of pre-decimal places for the maximum value. Four decimal places are used in the "inch" system of units, which means the maximum entry your can make is 100,000.0000 inch. The "millimeter" system of units uses three decimal places, which means the maximum entry you can make is 1,000,000.000 mm.

Connection between increments and system of units

The encoder signals of a connected encoder will be evaluated by the positioning module and converted into the current system of units. The resolution is used for the conversion (refer to the section entitled "Resolution (Page 85)").

If the positioning module

- has counted 10 increments and
- a resolution of 100 μm per increment is assigned in the encoder data,

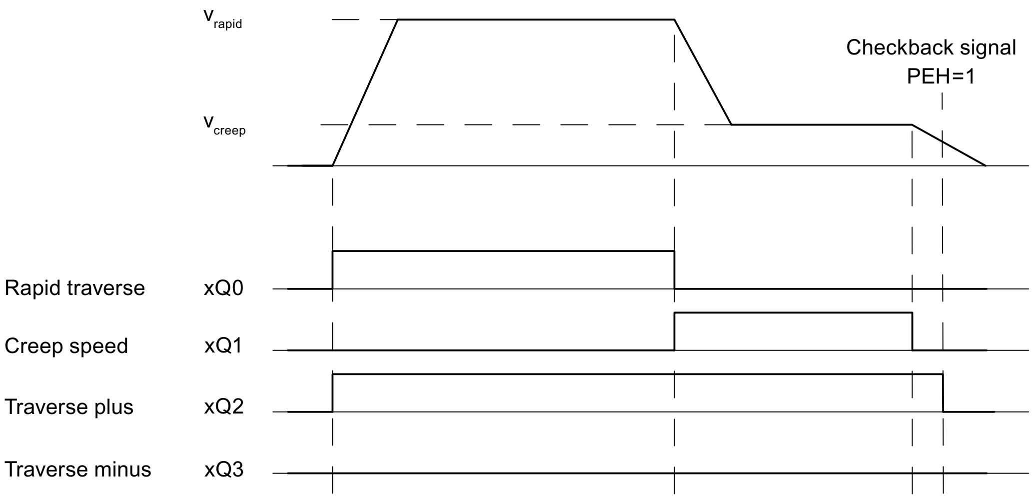
this means that the axis will be moved by a distance of 1 mm.

8.3 Machine data of the drive

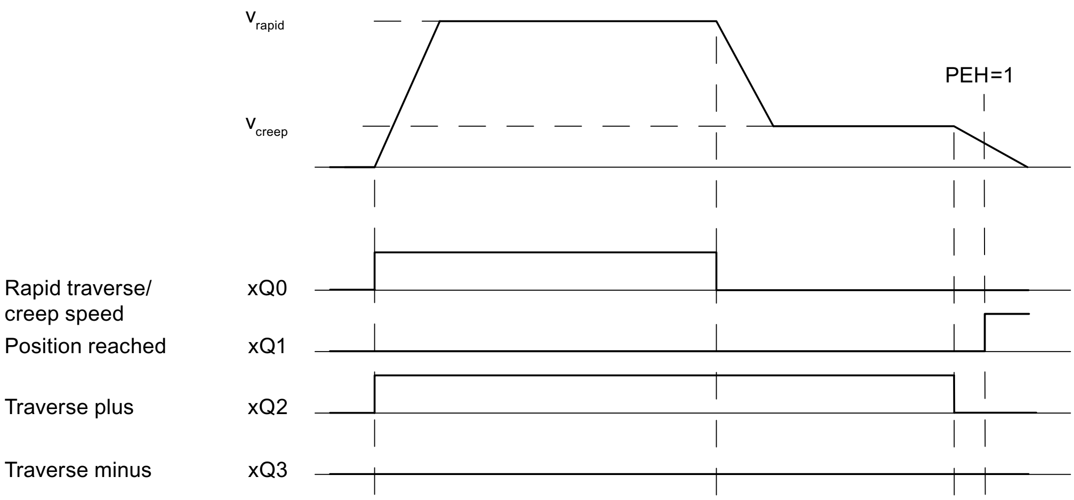
Drive data

Address	Name	Type	Initial value	Comment
92.0	CTRL_TYPE	DINT	L#1	Control mode: The control mode describes how the 4 digital outputs for each channel of a connected motor are operated via the power supply. x stands for channel 1 and 2

Control mode 1



Control mode 2



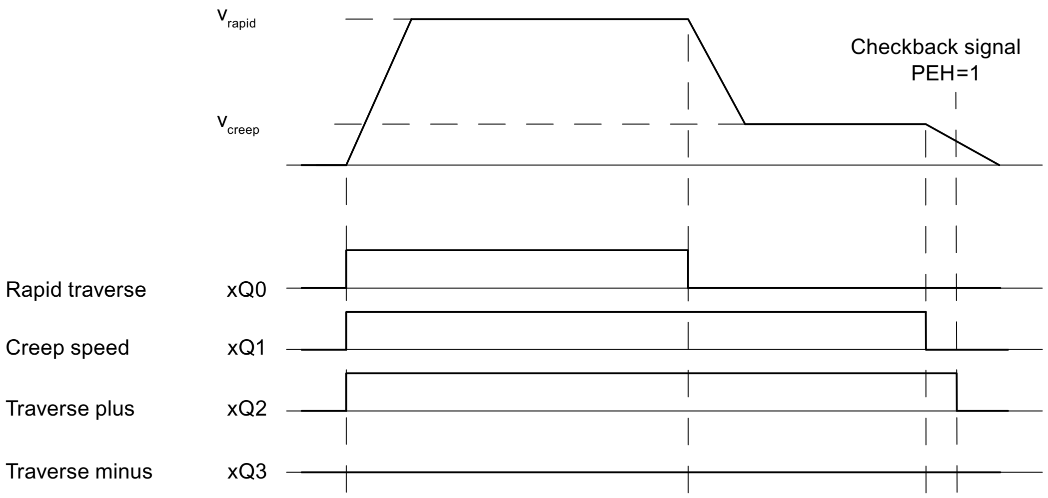
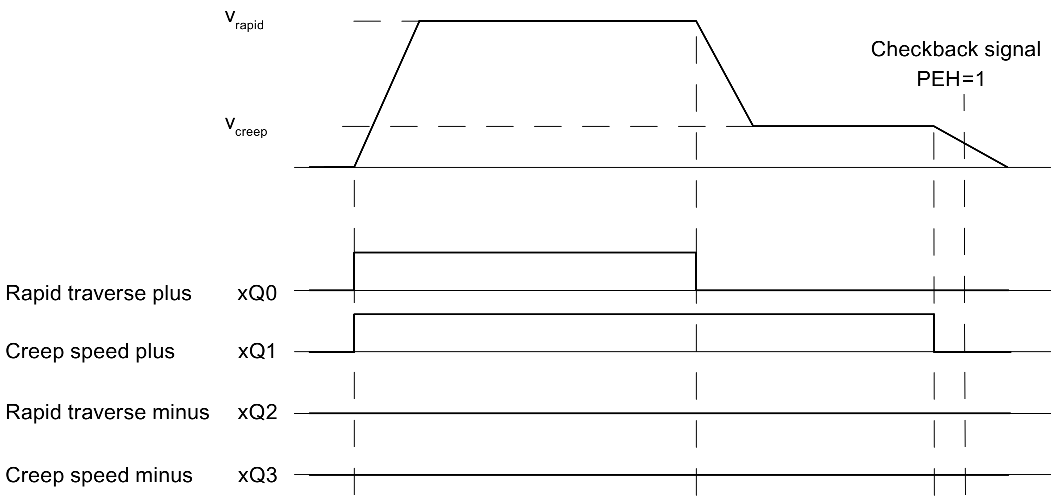
Address	Name	Type	Initial value	Comment
Control mode 3				
				
Control mode 4				
				

Table 8- 1 Table with states of the 4 outputs for each control mode, x stands for channel 1 and 2

Control mode 1	Rapid traverse		Creep speed		PR Position reached stop
	+ direction	- direction	+ direction	- direction	
xQ0	1	1	0	0	-
xQ1	0	0	1	1	-
xQ2	1	0	1	0	-
xQ3	0	1	0	1	-
Control mode 2					
xQ0	1	1	0	0	0
xQ1	0	0	0	0	1
xQ2	1	0	1	0	0
xQ3	0	1	0	1	0
Control mode 3					
xQ0	1	1	0	0	-
xQ1	1	1	1	1	-
xQ2	1	0	1	0	-
xQ3	0	1	0	1	-
Control mode 4					
xQ0	1	0	0	0	-
xQ1	1	0	1	0	-
xQ2	0	1	0	0	-
xQ3	0	1	0	1	-

Address	Name	Type	Initial value	Comment
100.0	CHGDIF_P	DINT	L#5000	Changeover difference plus
104.0	CHGDIF_M	DINT	L#5000	Changeover difference minus
108.0	CUTDIF_P	DINT	L#2000	Switch-off difference plus
112.0	CUTDIF_M	DINT	L#2000	Switch-off difference minus Range: • 1 µm to 1 000 000 000 µm at resolution ≥ 1 µm/pulse • 1 µm to 100 000 000 µm at resolution < 1 µm/pulse

The "changeover difference" defines the changeover point at which the drive changes over from rapid traverse to creep speed.

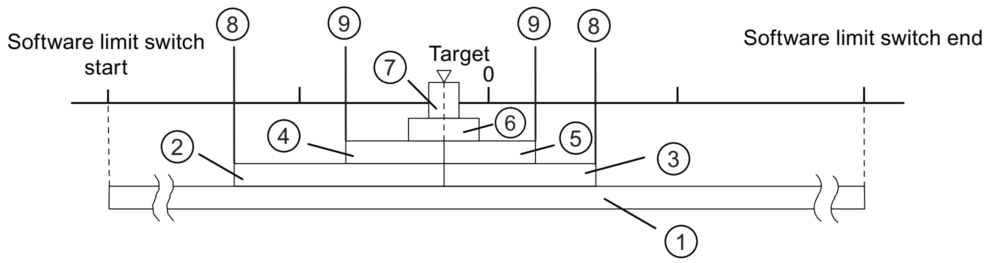
The "switch-off difference" defines the switch-off point at which the drive (creep speed) is switched off. As of this point the FM 351 takes over the monitoring functions.

The values apply to all targets that the FM 351 approaches, with the exception of increment 255.

Rules

- The values for the plus and minus direction can be different.
- The changeover difference must be greater than the switch-off difference.
- The changeover difference must lie within the operating range.
- The changeover difference must be less than the end of rotary axis.
- The switch-off difference must be greater than half the target range.
- Select a sufficient distance between the changeover point and the switch-off point to ensure that the drive can safely switch from rapid traverse to creep speed.
- Select the distance between the switch-off point and the target to ensure that the drive comes to a standstill within the target range.
- The changeover point, the switch-off point, and the start of the target range must be separated from each other by at least 8 ms.

Further information on the arrangement of ranges can be found in the section entitled "Auto-Hotspot".



①	Operating range
②③	Changeover difference plus/minus
④⑤	Switch-off difference plus/minus
⑥	Standstill range
⑦	Target range

Address	Name	Type	Initial value	Comment
⑧	Changeover point			
⑨	Switch-off point			

Address	Name	Type	Initial value	Comment
76.0	TRG_RANGE	DINT	L#1000	**Target range** • 0 = no monitoring Range: • 1 µm to 1 000 000 000 µm at resolution ≥ 1 µm/pulse • 1 µm to 100 000 000 µm at resolution < 1 µm/pulse
The target range lies symmetrically around the target. A value 0 switches off the monitoring of the target range. On the topic of target approach, please refer to the section entitled "End of a positioning (Page 91)".				

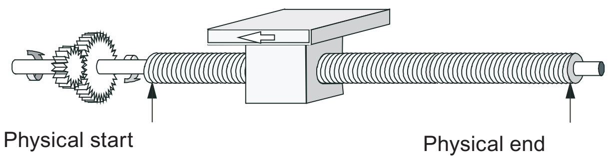
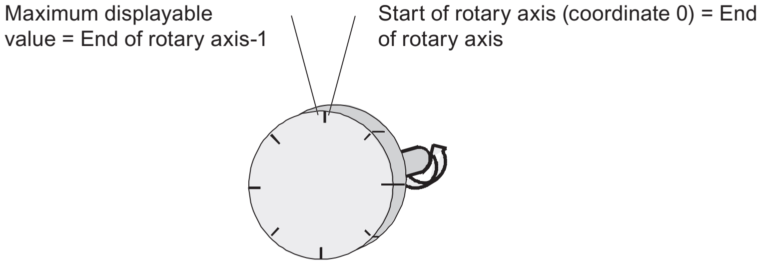
Address	Name	Type	Initial value	Comment
84.0	ZSPEED_R	DINT	L#1000	**Standstill range** • 0 = no monitoring Range: • 1 µm to 1 000 000 000 µm at resolution ≥ 1 µm/pulse • 1 µm to 100 000 000 µm at resolution < 1 µm/pulse
The standstill range lies symmetrically around the target. It is monitored to see if the drive remains standing at an approached target position or if it drifts off. If the standstill range is exited without a valid traversing job, the FM 351 signals an error. The standstill monitoring is switched off with the value 0. Tip: The standstill range should be greater than the target range. Please refer also to the section entitled "End of a positioning (Page 91)", which shows the target approach and the individual monitoring and reports.				

Address	Name	Type	Initial value	Comment
88.0	ZSPEED_L	DINT	L#30000	Standstill velocity • 0 = no monitoring • 1 μm/min to 100 000 μm/min
The standstill velocity serves as a reference velocity for the end of a positioning. Refer also to the section entitled "End of a positioning (Page 91)". The monitoring of the standstill velocity is switched off with the value 0.				

Address	Name	Type	Initial value	Comment
80.0	MON_TIME	DINT	L#2000	Monitoring time • 0 = no monitoring • 1 to 100 000 ms
With the aid of the monitoring time, the module monitors • the movement of the axis up to the switch-off point. The monitoring time starts with the start of a positioning and is retriggered with each actual value change in the traversing direction. • the target approach. The positioning must end within the monitoring time. The monitoring time is retriggered for the last time once the switch-off difference is reached. • the plausibility of the actual values on the switch points. A swinging of the axis on the switch points leads to operating errors. A value 0 switches off the monitoring. **Actual monitoring time** For the monitoring time you can specify all the values from the defined area. • 0: The monitoring is deactivated. • 1 to 100 000 ms: The FM 351 rounds off the specified time to a multiple of 8 ms (module cycle). Therefore, it is best to enter the monitoring time in 8 ms blocks.				

8.4 Axis machine data

Axis data

<table>
<tr><th>Address</th><th>Name</th><th>Type</th><th>Initial value</th><th>Comment</th></tr>
<tr><td>12.0</td><td>AXIS_TYPE</td><td>DINT</td><td>L#0</td><td>Axis type:
Linear axis = 0
Rotary axis = 1</td></tr>
<tr><td colspan="5">The linear axis is an axis that has a restricted physical traversing range.

</td></tr>
<tr><td colspan="5">The rotary axis is an axis whose traversing range is not restricted by mechanical stops.

</td></tr>
</table>

Address	Name	Type	Initial value	Comment
16.0	ENDROTAX	DINT	L#100000	**End of rotary axis:** Range: • 1 µm to 1 000 000 000 µm at resolution ≥ 1 µm/pulse • 1 µm to 100 000 000 µm at resolution < 1 µm/pulse

The value "end of rotary axis" is the theoretical maximum value that the actual value can reach. The theoretically highest value is however never displayed, because it physically identifies the same position as the start of the rotary axis (0).

The maximum value that will be displayed for a rotary axis has the value:

End of rotary axis [µm] - resolution [µm / pulse] * 1 [pulse]

Example: End of rotary axis 1000 mm, resolution 1000 µm / pulse

The display jumps:

- with a positive rotational direction from 999 mm to 0 mm
- with a negative rotational direction from 0 mm to 999 mm.

Rotary axis with absolute encoders

In the case of a rotary axis with an absolute encoder, the rotary axis range (0 to end of rotary axis) must cover the encoder range of the absolute encoder exactly.

$$\text{Rotary axis end}[\mu m] = \text{number of revolutions(encoder)} * \frac{\text{Distance}[\mu m]}{\text{Revolution}}$$

Address	Name	Type	Initial value	Comment
44.0	REFPT	DINT	L#0	**Reference point coordinates:** Range: • -1 000 000 000 µm to 1 000 000 000 µm at resolution ≥1 µm/pulse • -100 000 000 µm to 100 000 000 µm at resolution < 1 µm/pulse

Incremental encoders:

You require the reference point coordinate for the "Reference point approach" mode. If the axis is not synchronized following writing and activating of machine data, the actual value is set to the value of the reference point coordinate.

Absolute encoder (SSI)

You require the reference point coordinates for the mechanical adjustment of the encoder.

For more information, refer to the description of absolute adjustment in the section entitled "Determining the absolute encoder adjustment (Page 82)", which explains the interaction of the absolute adjustment with other data.

The value of the reference point coordinate must lie within the operating range:

- **Linear axis**

  Including the software limit switches
- **Rotary axis**

  Greater than or equal to 0 and smaller than the value "end of rotary axis" (0 ≤ reference point coordinates < "end of rotary axis").

Address	Name	Type	Initial value	Comment
52.0	REFPT_TYPE	DINT	L#0	Type of reference point approach: Areas: 0 = plus, reference point switch in direction + 1 = plus, reference point switch in direction - 2 = minus, reference point switch in direction + 3 = minus, reference point switch in direction -
With Type of reference point approach you determine the conditions for the synchronization of the axes. • The first specification defines the start direction in which the reference point approach is starting. • The second specification defines the location of the zero mark, which leads to synchronization, as far as the reference point switch is concerned. The application of this data is described in the section entitled "Configuring the reference point approach mode (Page 103)".				

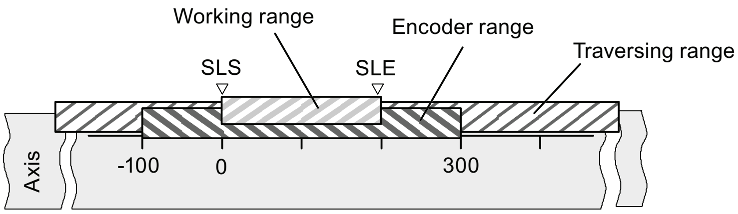
Address	Name	Type	Initial value	Comment
99.0	REFPT_SPD	BOOL	TRUE	Start velocity in the case of a reference point approach 0 = rapid traverse 1 = creep speed
With this data you select the velocity for the start of the reference point approach.				

Address	Name	Type	Initial value	Comment
64.0 68.0	SSW_STRT SSW_END	DINT DINT	L#-100000000 L#100000000	**Software limit switch start** **Software limit switch end** Range: • -1 000 000 000 µm to 1 000 000 000 µm at resolution ≥1 µm/pulse • -100 000 000 µm to 100 000 000 µm at resolution < 1 µm/pulse

This axis data will only be used for a linear axis.

The software limit switches will be monitored if the axis is synchronized. The area that will be restricted by the software limit switches is known as the **Operating range**.

The software limit switch start (SLS) must always be less than the software limit switch end (SLE).



Incremental encoder

After each start-up of the FM 351, the axis is initially not synchronized. Only after a synchronization will the configured software limit switches be monitored.

Absolute encoder (SSI)

The axis is synchronized after the FM 351 has received a complete, error-free message frame for the associated channel. As of this point the software limit switches will be monitored.

The absolute encoder must cover **at least** the operating range including the software limit switches.

Connection: operating range, encoder range, traversing range

- The "operating range" is the area that you establish for your jobs by means of the software limit switches.
- The "encoder range" is the unambiguous range that is covered by the encoder. For a linear axis, the module assigns the "encoder range" symmetrically across across the operating range. That is, the module shifts the encoder range in order to equalize the distances between the software limit switches and the ends of the encoder range (see figure).
- The "traversing range" is the value range that the FM 351 can process. The traversing range depends on the resolution.

8.5 Encoder machine data

Definition

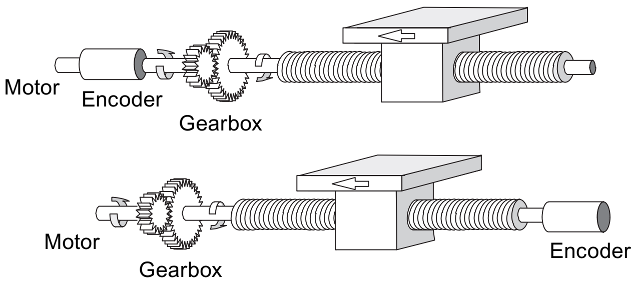
The encoder supplies path information to the module (refer to the section entitled "Encoder (Page 131)"). The module evaluates this and uses the resolution to convert it into an actual value.

The correct definition of the encoder's machine data is essential for ensuring consistency between the calculated and physical actual position of the axis.

Data in the parameter DB

Address	Name	Type	Initial value	Comment
20.0	ENC_TYPE	DINT	L#1	Encoder type and frame length: Range of values: 1 = 5 V incremental 2 = 24 V incremental 3 = SSI 13-bit frame length 4 = SSI 25-bit frame length 5 = SSI 14-bit frame length 6 = SSI 15-bit frame length 7 = SSI 16-bit frame length 8 = SSI 17-bit frame length 9 = SSI 18-bit frame length 10 = SSI 19-bit frame length 11 = SSI 20-bit frame length 12 = SSI 21-bit frame length 13 = SSI 22-bit frame length 14 = SSI 23-bit frame length 15 = SSI 24-bit frame length
With the "frame length" you define the clock frame output by the positioning module.				

Address	Name	Type	Initial value	Comment
24.0	DISP_REV	DINT	L#80000	**Distance per encoder revolution**: Range of values: 1 μm to 1 000 000 000 μm
With the machine data "Distance per encoder revolution" you can inform the positioning module which distance the drive system should cover per encoder revolution. The "Distance per encoder revolution" value depends on the axis configuration and on the way in which the encoder is fitted. You must take into account all the transfer elements like clutches or gearboxes. The section entitled "Resolution (Page 85)" describes the connection between the machine data "Distance per encoder revolution" and "Increments per encoder revolution".				



Address	Name	Type	Initial value	Comment
28.0	b_28	DWORD	L#0	**Parity check for absolute encoders (SSI):** Range of values: 0: No parity check 1: Check for even parity 2: Check for odd parity
You use the machine data element "Parity check for absolute encoders (SSI)" to specify if there is a parity check for the frame when you use an absolute encoder (SSI). This machine data element is only available for the FM 351 with order number 6ES7351-1AH02-0AE0.				

Address	Name	Type	Initial value	Comment
32.0	INC_REV	DINT	L#500	Increments per encoder revolution: Range of values: 1 to 2^{25}

The "Increments per encoder revolution" machine data element specifies the number of increments output by an encoder per revolution. The positioning module uses this value and the "Distance per encoder revolution" machine data element to determine the resolution.

Incremental encoder

Any value within the range of values can be input. The module evaluates the increments in four operations (refer to the section entitled "Incremental encoder (Page 131)").

Absolute encoder

For the limits you have to differentiate between the individual encoder models. Only values to the power of 2 are permitted as input (refer to the section entitled "Absolute encoder (Page 134)").

- Single-turn encoder with (number of revolutions = 1) 13-bit frame length:
  - minimum value = 4
  - maximum value = 8192
- Multiturn encoder (number of revolutions > 1) with 25-bit frame length:
  - minimum value = 4
  - maximum value = 8192
- Single-turn encoder with 25-bit frame length and number of revolutions = 1:
  - minimum value = 4
  - maximum value = 2^{25}

Linear scales will be configured as multiturn encoders as follows:

- Increments per encoder revolution = 8192
- Number of revolutions x 8192 ≥ number of steps in the linear scale

Address	Name	Type	Initial value	Comment
36.0	NO_REV	DINT	L#1	Number of encoder revolutions: Range of values: 1 (single-turn encoder) 2 to 4096 to the power of 2 (Multiturn encoder)

The machine data "Number of encoder revolutions" is used only for absolute encoders. As a result, enter the number of revolutions that are possible with this encoder.

The overall number of steps of the encoder is not machine data. It is calculated as follows:

Overall number of steps = increments per encoder revolution x number of revolutions

Address	Name	Type	Initial value	Comment
40.0	BAUD RATE	DINT	L#0	Baud rate: Range of values: 0 = 188 kHz 1 = 375 kHz 2 = 750 kHz 3 = 1500 kHz
With the machine data "Baud rate" you can determine the data transmission rate from the SSI encoder to the positioning module. This entry is of no significance for incremental encoders. The maximum Baud rate is dependent on the cable length: • 200 m → 188 kHz • 100 m → 375 kHz • 40 m → 750 kHz • 12 m → 1500 kHz				

Address	Name	Type	Initial value	Comment
59.0	CNT_DIR	BOOL	FALSE	Counting direction: 0 = normal 1 = inverted
With the machine data "Counting direction" you can adapt the direction of the position measuring to the direction of movement of the axis. In doing so, take into consideration all rotational directions of the transfer elements (like, for example, clutches and gearboxes). • Normal = incremental count pulses (incremental encoder) or encoder values (absolute encoder) represent increasing actual position values • Inverted = incremental count pulses (incremental encoder) or encoder values (absolute encoder) correspond to decreasing actual position values				

Address	Name	Type	Initial value	Comment
				Monitoring:
63.0	MON_WIRE	BOOL	TRUE	1 = wire break
63.1	MON_FRAME	BOOL	TRUE	1 = frame error (must always be 1)
63.2	MON_PULSE	BOOL	TRUE	1 = missing pulses

Wire break

When monitoring is enabled, the positioning module monitors all cables for a 5 V incremental encoder and an absolute encoder. The monitoring detects:

- Wire break
- Short circuit on individual cables
- Edge distance of the count pulses (also with 24 V incremental encoders)

To monitor with a 24 V incremental encoder you must set a monitoring time of MON_TIME > 0.

With 5 V incremental encoders without zero marks, you must disable the wire break monitoring or interconnect the N and /N signals externally.

Frame error

The module monitors the frame of an absolute encoder (SSI) for:

- Start and stop bit errors

You cannot disable the frame error monitoring with absolute encoders (SSI).

Missing pulses (incremental encoder)

An incremental encoder must always return the same number of increments between two successive zero marks.

The positioning module checks to see if the zero mark of an incremental encoder is occurring at the correct encoder state.

For encoders without zero marks you must switch off the monitoring for missing pulses.

You must also disable the wire-break monitoring or interconnect the N and /N zero mark inputs externally.

8.6 Determining the absolute encoder adjustment

Definition

The absolute encoder adjustment and the reference point coordinate is used to explicitly map the encoder's value range to the coordinate system of the axis.

Determining the correct absolute encoder adjustment

After the first configuration further steps must be taken to ensure that a correct relationship is created between the encoder and coordinate system. The order of events is shown using the parameter assignment screens.

1. Traverse the axis to a defined, reproducible point to which a unique coordinate is assigned.

   This could, for example, be the "software limit switch end".
2. Call the job "Set reference point" with the coordinate of the point defined under 1.

   The positioning module now determines an encoder value for the reference point coordinate entered in the channel DB (REFPT in channel DB), namely the absolute encoder adjustment. You can read out this value in the service screen of the parameter assignment interface.
3. Enter the value read out from the service screen in the "Absolute encoder adjustment" field on the "Axis" tab of the parameter assignment interface.
4. Save your parameter assignment to the corresponding parameter DB using the export function.
5. Close the parameter assignment interface by selecting Save and Exit.
6. Download the data in HW Config to the CPU.
7. Restart the CPU (cold restart) to apply the data.

Note

You perform this adjustment just once during the commissioning. After a configuration, the position module is synchronized during start-up as soon as a complete, faultless message frame of the encoder is received after start-up.

Data in the parameter DB

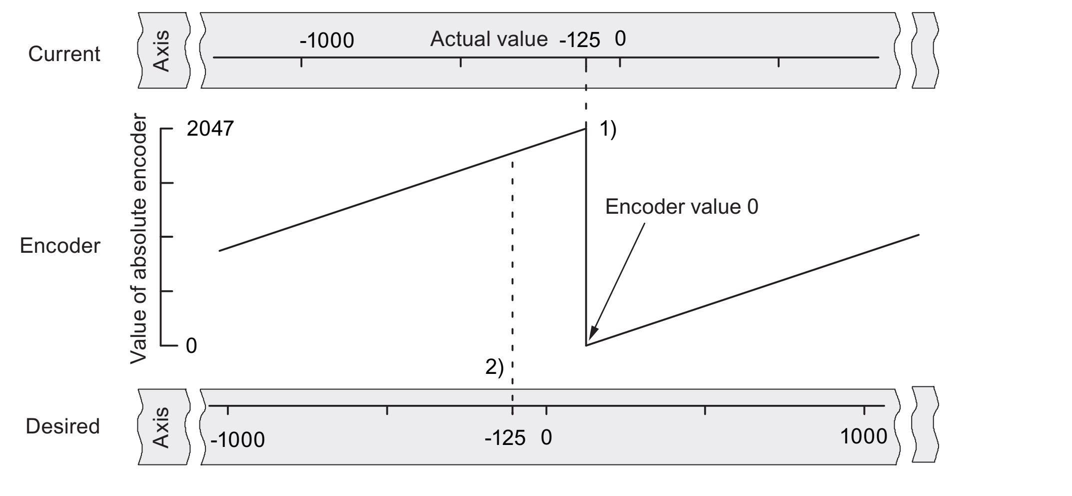
Address	Name	Type	Initial value	Comment
44.0	REFPT	DINT	L#0	**Reference point coordinate** Range: • -1 000 000 000 µm to 1 000 000 000 µm at resolution ≥1 µm/pulse • -100 000 000 µm to 100 000 000 µm at resolution < 1 µm/pulse
48.0	ENC_ADJ	DINT	L#0	**Absolute encoder adjustment**: Range: 0 to (2^{25}-1)

Example of an absolute encoder adjustment

The following assumptions apply to the example:

- Reference point coordinate = -125 mm
- Operating range from SSW_STRT = -1000 mm to SSW_END = 1000 mm
- Absolute encoder adjustment = 0
- Encoder range = 2048 increments with a resolution of 1 mm/pulse
- The absolute encoder used cannot be adjusted mechanically with precision and it also is not possible to set the encoder value specifically.

Example of an absolute encoder adjustment



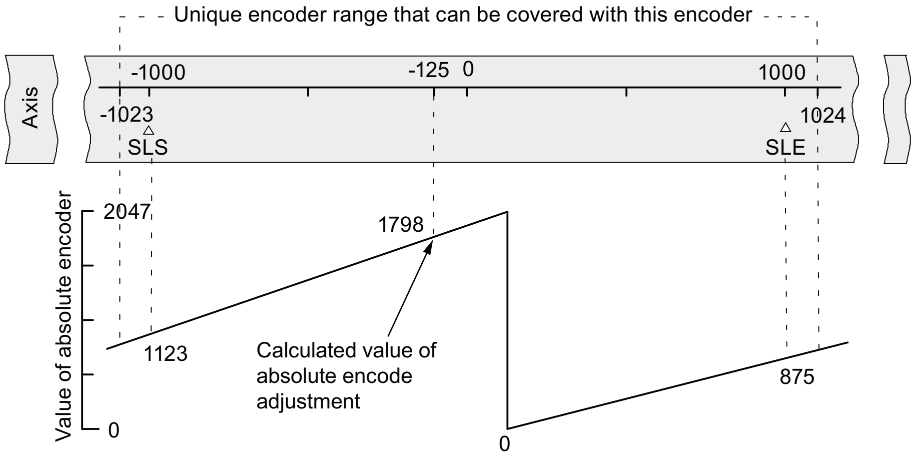
1) Assignment of the coordinate system to the encoder values with the specified absolute encoder adjustment. The encoder value 0 corresponds to the actual value -125.

2) Desired classification of the coordinate system to the encoder. At this position the coordinate should be -125.

Result in accordance with "Set reference point"

After "Set reference point" the relationship looks as follows:

The reference point coordinate on the axis (-125) is assigned to the encoder value (1798) determined from the absolute encoder adjustment.



The encoder supplies 2048 unique values. The operating range is established by the software limit switches. Because of the selected resolution of 1 mm per pulse, the encoder however can cover a greater operating range than the one provided by the software limit switches.

With the set resolution, the operating range is already covered with 2001 values. Therefore, in the example, 47 pulses "remain" and these are symmetrically positioned around the operating range.

Otherwise: Mechanical adjustment of the encoder

A correct relationship between the coordinate system and the encoder can be achieved as follows:

1. Traverse the axis to a reproducible position (e.g., the software limit switch start).
2. Enter this coordinate value in the machine data as the reference point coordinate.
3. Read the encoder value shown at this position from the service screen of the configuration software.
4. Enter this value in the machine data as the absolute encoder adjustment.

After the parameter assignment a correct actual value will always be shown.

Instead of steps 3 and 4, you can also set the encoder to zero via "Reset", if available, and enter the value "0" in the machine data as the absolute encoder adjustment.

8.7 Resolution

Definition

The resolution indicates the traversing distance corresponding to one **pulse**. It is a measurement for the precision of the positioning and also determines the possible maximum traversing range of the positioning module.

The resolution (RES) can be calculated as in the following table:

	Incremental encoder	Absolute encoder
Input values	• Distance per encoder revolution • Increments per encoder revolution: – Pulse evaluation: 4x – 1 increment = 4 pulses	• Distance per encoder revolution • Increments per encoder revolution: – 1 increment = 1 pulse
Calculation	$RES = \dfrac{\dfrac{\text{Distance}}{\text{Encoder revolution}}}{\dfrac{\text{Pulses}}{\text{Encoder revolution}}}$	

Note

All position information is rounded up to the integral multiple of the resolution. This way the entered values differentiate from the used values.

Range of values of the resolution

The chosen system of units determines the range of values of the resolution:

System of units	Information in ...	Range of values of the resolution
mm	10^{-3} mm	0,1•10^{-3} .... 1000*10^{-3} mm/pulse
inch	10^{-4} inch	0,1•10^{-4} .... 1000*10^{-4} inch/pulse
degrees	10^{-4} degrees 10^{-3} degrees 10^{-2} degrees	0,1•10^{-4} .... 1000*10^{-4} degrees/pulse 0,1•10^{-3} .... 1000*10^{-3} degrees/pulse 0,1•10^{-2} .... 1000*10^{-2} degrees/pulse

Example

- An incremental encoder has the following data:
  - Increments per encoder revolution: 5000
  - Distance per encoder revolution: 1000 mm
  - 1 increment = 4 pulses

  This results in the resolution (4 x evaluation):

$$\text{Resolution} = \frac{1000\ \text{mm}}{5000\ \text{increments}} = 0.2000\ \frac{\text{mm}}{\text{Increment}} = 0.2000\ \frac{\text{mm}}{4\ \text{pulses}}$$

$$= 0.0500\ \frac{\text{mm}}{\text{Pulse}}$$

- An SSI encoder has the following data:
  - Increments per encoder revolution: 4096
  - Distance per encoder revolution: 1000 mm
  - 1 increment = 1 pulse

This results in the resolution:

$$\text{Resolution} = \frac{1000\ \text{mm}}{4096\ \text{increments}} = 0.2441\ \frac{\text{mm}}{\text{Increment}} = 0.2441\ \frac{\text{mm}}{\text{Pulse}}$$

Relationship between the traversing range and resolution

The traversing range is restricted by the numerical representation in the position module. This numerical representation varies depending on the resolution. You should thus make sure that your specifications are always within the permitted limits.

The maximum traversing range is illustrated in the following table:

Resolution (RES) lies in the range	Maximum traversing range
0,1 $^{\mu m}/_{pulse}$ ≤ RES < 1 $^{\mu m}/_{pulse}$	-10^8 µm to 10^8 µm (-100 m to +100 m)
1 $^{\mu m}/_{pulse}$ ≤ AUFL ≤ 1000 $^{\mu m}/_{pulse}$	$-1\ 0^9$ µm to 10^9 µm (-1000 m to +1000 m)

8.8 Increments

8.8.1 Increments

Definition

Increments are target specifications that can be controlled by the positioning module with the **relative/absolute incremental approach** mode.

Prerequisites for increments

The target that is to be approached must be at a distance in front of the respective software limit switch of at least one-half the target range.

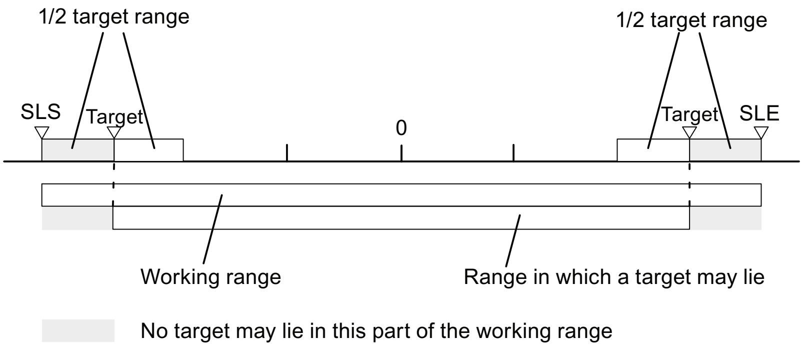


Figure 8-1 Limits for increment specifications

8.8.2 Increment number 1 to 100

Increment number 1 to 100

You have the option of entering up to 100 increments in a table. These are valid both for the **relative incremental approach** as well as for the **absolute incremental approach**.

Note that the positioning module does not permit negative values for the **relative incremental approach.** The values will be interpreted by the positioning module - depending on the direction of movement - as either a positive or a negative difference.

Note

The entry is made in the unit in accordance with the set system of units. In doing so, please observe the post-decimal places.

Numerical example:

- **Incremental**: 800 mm
- **System of units**: 10^{-3} mm
- **Entry in the parameter DB**: 800000

Tip: in the increment table define separate areas for relative and absolute increments.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
35.4	TRGL1WR_EN	BOOL	FALSE	1 = write increment table 1 (increments 1 ... 50)
35.5	TRGL2WR_EN	BOOL	FALSE	1 = write increment table 2 (increments 51 ... 100)
36.6	TRGL1RD_EN	BOOL	FALSE	1 = read increment table 1 (increments 1 ... 50)
36.7	TRGL2RD_EN	BOOL	FALSE	1 = read increment table 2 (increments 51 ... 100)

Data used in the parameter DB

Address	Name	Type	Initial value	Comment	
120.0	TRGL1.TRG[1]	DINT	L#0	Increment number 1	Increment table 1
. . .	. . .	. .	. . .	. . .	
316.0	TRGL1.TRG[50]	DINT	L#0	Increment number 50	
320.0	TRGL2.TRG[51]	DINT	L#0	Increment number 51	Increment table 2
. . .	. . .	. . .	. . .	. . .	
516.0	TRGL2.TRG[100]	DINT	L#0	Increment number 100	

8.8.3 Increment number 254

Increment number 254

Independently of the increment table you can use the increment number 254 as an additional distance specification. For the changeover and switch-off differences, the entries from the parameter DB are valid for this increment.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
36.2	TRG252_254_EN	BOOL	FALSE	1 = write increment for increment number 254
96.0	TRG252_254	DINT	L#0	Increment for increment number 254

Data used in the parameter DB

Address	Name	Type	Initial value	Comment
100.0	CHGDIF_P	DINT	L#5000	Changeover difference plus
104.0	CHGDIF_M	DINT	L#5000	Changeover difference minus
108.0	CUTDIF_P	DINT	L#2000	Switch-off difference plus
112.0	CUTDIF_M	DINT	L#2000	Switch-off difference minus

8.8.4 Increment number 255

Increment number 255

The increment number 255 gives you an additional guideline for the path.

The switch-off differences and the changeover differences are presented together with the increment. Unlike the situation with the other increments, the increment 255 uses the values established in the channel DB for the switch-off and changeover differences. The entries from the machine data have no validity for this increment.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
36.3	TRG255_EN	BOOL	FALSE	1 = write increment for increment number 255
100.0	TRG255	DINT	L#0	Increment for increment number 255
104.0	CHGDIF255	DINT	L#0	Changeover differences for increment number 255
108.0	CUTDIF255	DINT	L#0	Switch-off differences for increment number 255

9 Operating modes and jobs

9.1 End of a positioning

Definition

The end of a positioning is indicated by the checkback signal WORKING = 0. It can be reached in three different ways:

- Target approach
- Shutdown
- Cancel

Monitoring

During the end of a positioning, the following monitoring systems are active:

- Monitoring time

  The monitoring time is retriggered at the switch-off point for the last time and then loses its validity with the end of the positioning.

  Within this time, the end of the positioning must be reached, otherwise the outputs will be switched off and the operating error "Error during target approach" (error number 5) will be issued.

- Monitoring of target range

  The FM 351 sets a symmetrical range around each target, thereby defining the positioning precision of your application. The axis must come to a standstill within this range during a target approach. A value 0 switches off the tolerance during the target approach.

- Monitoring of standstill velocity

  The standstill velocity is used to determine that the drive comes to a standstill within the target range. After reaching the switch-off point it is checked to see if it has fallen short of the area.

  The velocity must fall below the standstill velocity within the target range, otherwise the FM 351 issues the operating error "Target range overrun" (error number 10).

  Falling short of the standstill velocity is monitored only once per target approach.

  Note that the velocity can briefly fall below the standstill velocity for the velocity determination of the module if the axis moves at a very slow positioning velocity (less than 2 pulses per 8 ms).

- Monitoring of standstill range

  Having ended a positioning it is monitored to see if the drive remains standing at an approached target position or if it drifts off.

  The standstill range is monitored

  – after the FM 351 has reported the checkback signal "PR"

  – if the monitoring time is exceeded

  – if the velocity falls below the standstill velocity.

  If the standstill range is exited without a valid traversing job, the FM 351 reports the "Standstill range exited" error (error number 6).

Target approach

The target approach in the "Absolute/relative incremental approach" modes starts once the switch-off point has been reached. As of this point the drive is switched off, and the FM 351 adopts the monitoring functions.

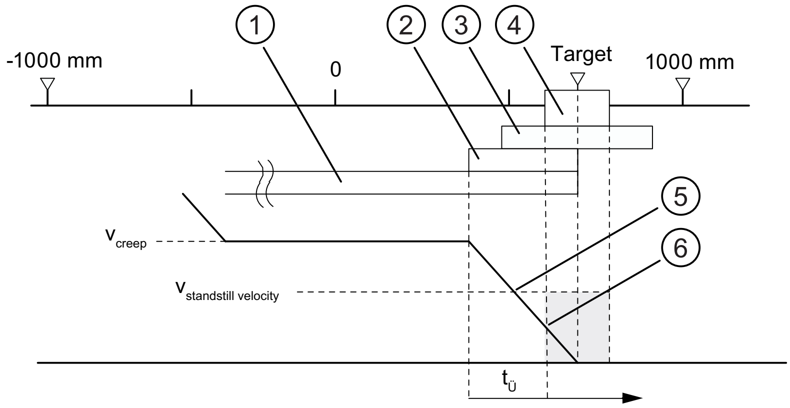
Depending on which monitoring you have assigned, there are different cases for creating the checkback signal "PR (POS_RCD)". The positioning is canceled if checkback signal "PR (POS_RCD)" is not generated.

Scenario 1: You have assigned:

- Target range (TRG_RANGE) > 0
- Standstill velocity (ZSPEED_L) > 0
- Monitoring time (MON_TIME) > 0

  PR is generated when the velocity falls below the standstill velocity and the target range has been reached. It does not matter which condition is fulfilled first.

  PR is not generated if the actual value has not reached the target range within the monitoring time or the target range has been overrun without the velocity falling below the standstill velocity.



①	Changeover difference plus
②	Switch-off difference plus
③	Standstill range
④	Target range
⑤	Standstill velocity reached
⑥	Target range with V_{still} reached: PR is set
$t_Ü$	Monitoring time

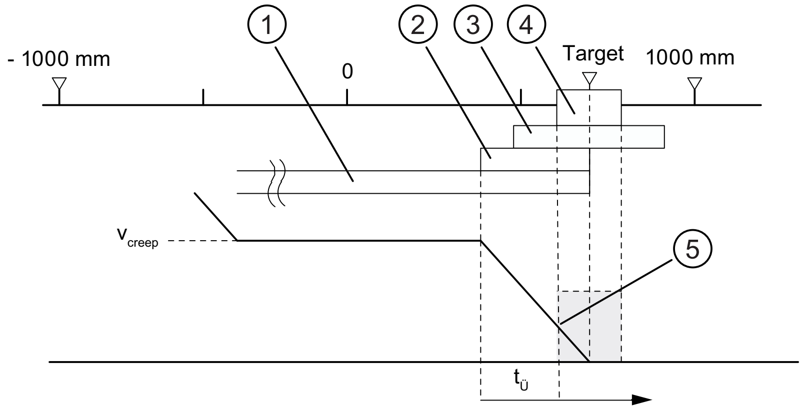
Figure 9-1 Target approach of an incremental approach

Scenario 2: You have assigned:

- Target range (TRG_RANGE) > 0
- Standstill velocity (ZSPEED_L) = 0
- Monitoring time (MON_TIME) > 0

  PR is generated when the target range has been reached.

  PR is not generated if the actual value does not reach the target range within the monitoring time.



①	Changeover difference plus
②	Switch-off difference plus
③	Standstill range
④	Target range
⑤	Target range with V_{still} reached: PR is set
$t_Ü$	Monitoring time

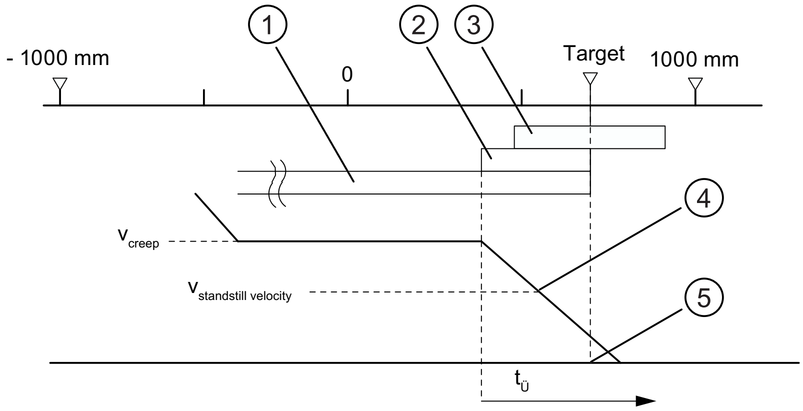
Figure 9-2 Target approach of an incremental approach

Scenario 3: You have assigned:

- Target range (TRG_RANGE) = 0
- Standstill velocity (ZSPEED_L) > 0
- Monitoring time (MON_TIME) > 0

  PR is generated if the velocity falls below the standstill velocity and the target is then reached.

  PR is not generated if the actual value has not reached the target within the monitoring time or the target range has been overrun without the velocity falling below the standstill velocity.



①	Changeover difference plus
②	Switch-off difference plus
③	Standstill range
④	Standstill velocity reached
⑤	Target reached: PR is set
$t_Ü$	Monitoring time

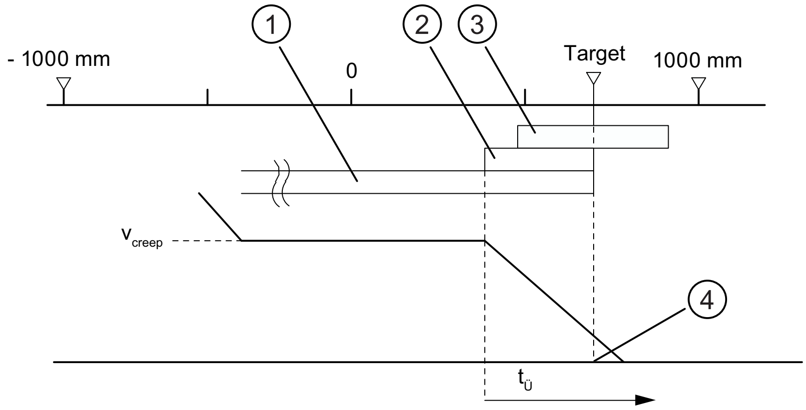
Figure 9-3 Target approach of an incremental approach

Scenario 4: You have assigned:

- Target range (TRG_RANGE) = 0
- Standstill velocity (ZSPEED_L) = 0
- Monitoring time (MON_TIME) > 0

  PR is generated when the target has been reached.

  PR is not generated if the actual value does not reach the target within the monitoring time.



①	Changeover difference plus
②	Switch-off difference plus
③	Standstill range
④	Target reached: PR is set
$t_Ü$	Monitoring time

Figure 9-4 Target approach of an incremental approach

Scenario 5: You have assigned:

- Target range (TRG_RANGE) ≥ 0
- Standstill velocity (ZSPEED_L) ≥ 0
- Monitoring time (MON_TIME) = 0

  If, in this case, the positioning comes to a standstill before the target range, the end of the positioning is not detected. PR is not generated and the checkback signal WORKING remains set. You can cancel the positioning only by deleting the drive enable (DRV_EN = 0).

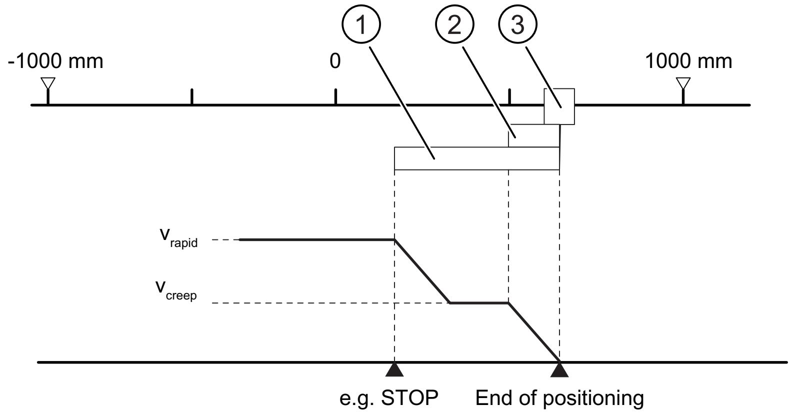
Shutdown without specified target

Shutdown means: the positioning operation is terminated in a controlled manner, complying with the differences of rapid traverse via creep speed.

The positioning is shut down, if

- the FM 351 receives a STOP signal (STOP = 1)
- The "Jog" and "Reference point approach" modes are terminated
- there is an operator error

The "PR (POS_RCD)" checkback signal is not set. The sequences are analogous to target approach.



① Changeover difference plus
② Switch-off difference plus
③ Target range

Figure 9-5 Shutdown of a positioning

Cancel

Cancel means: the positioning is immediately ended without using the changeover difference and switch-off difference from rapid traverse or creep speed to standstill. In addition, all the relevant outputs of the respective control mode are immediately switched off and also:

- Increment = actual value
- Distance-to-go = zero

The positioning is canceled if

- The drive enable signal is deleted (DRV_EN=0)
- The CPU goes to STOP
- Diagnostic error or any operating error except the "Target overrun" operating error (error number 9) occurred

The "PR (POS_RCD)" checkback signal is not set in "Incremental approach" mode.

If the standstill velocity is assigned, the standstill monitoring will become active as soon as the velocity falls below the standstill velocity. If the standstill velocity is not assigned, the standstill monitoring will be active once the outputs have been switched off.

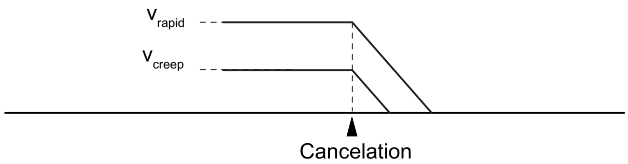


Figure 9-6 Cancelation of a positioning

Data used in the parameter DB

Address	Name	Type	Initial value	Comment
76.0	TRG_RANGE	DINT	L#1000	Target range
80.0	MON_TIME	DINT	L#2000	Monitoring time
84.0	ZSPEED_R	DINT	L#1000	Standstill range
88.0	ZSPEED_L	DINT	L#30000	Standstill velocity

Checkback signal in the channel DB

Address	Name	Type	Initial value	Comment
23.1	WORKING	BOOL	FALSE	1 = Positioning running (processing running)
25.7	POS_RCD	BOOL	FALSE	1 = Position reached

9.2 Configured the "Jog" mode

Definition

In "Jog" mode you can move the drive in one direction at the press of a button. You must install a button for both directions (plus and minus) respectively. You can use "Jog" mode for both synchronized and non-synchronized axes.

For a non-synchronized axis, jogging involves a positioning in the specified direction.

For a synchronized axis (linear axis), jogging involves a positioning to the software limit switches.

Requirement

The axis parameters must be assigned.

Sequence of "Jog" mode

1. Set the control signal for "Jog" mode (MODE_IN=1).
2. Set the control signal for the drive enable (DRV_EN=1).
3. Set the function switch for "Do not evaluate enable input" (EI_OFF=1) or wire the enable input for the corresponding channel.
4. Enter the start velocity.
   – Rapid traverse (MODE_TYPE=1)
   – Creep speed (MODE_TYPE=0)

5. Set the control signal for the plus or minus traversing direction (DIR_P=1 or DIR_M=1).
6. Call the FB ABS_CTRL.

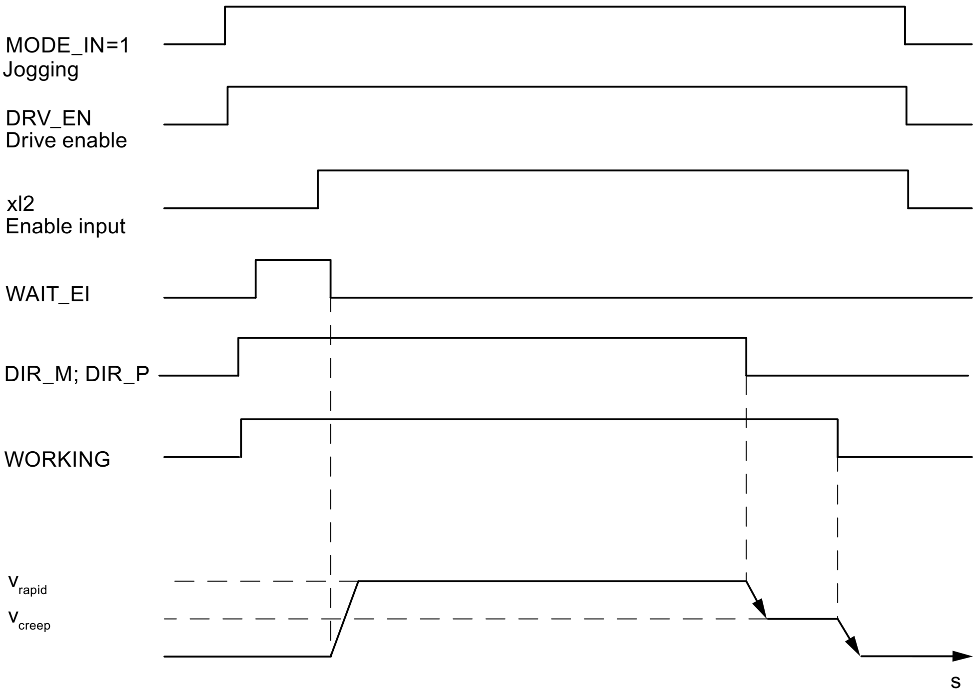


Figure 9-7 Example for "Jog" mode

Data used in the channel DB

Address	Name	Type	Initial value	Comment
15.2	DIR_M	BOOL	FALSE	1 = Minus direction
15.3	DIR_P	BOOL	FALSE	1 = Plus direction
15.7	DRV_EN	BOOL	FALSE	1 = Switch on drive enable
16.0	MODE_IN	BYTE	B#16#0	1 = jog
17.0	MODE_TYPE	BYTE	B#16#0	1 = rapid traverse 0 = creep speed
23.0	ST_ENBLD	BOOL	FALSE	1 = Start enabled
23.1	WORKING	BOOL	FALSE	1 = Positioning running (processing running)
23.2	WAIT_EI	BOOL	FALSE	1 = axis waiting for external enable
34.2	EI_OFF	BOOL	FALSE	1 = Do not evaluate enable input

Shutdown of jogging

"Jog" mode is shut down, if

- You release the button used for jogging (DIR_M or DIR_P=0)
- The FM 351 receives a STOP signal (STOP = 1)
- The actual value reaches the limit of the operating range for a synchronized linear axis
  The axis can only continue moving in the opposite direction.

After traversing is shut down, it can continue in any direction.

Canceling jogging

"Jog" mode is canceled, if

- the drive enable signal is deleted (DRV_EN=0)
- a traversing range limit for a linear axis has been overrun

Monitoring

In jog mode, the following monitoring functions are not active at the end of the positioning:

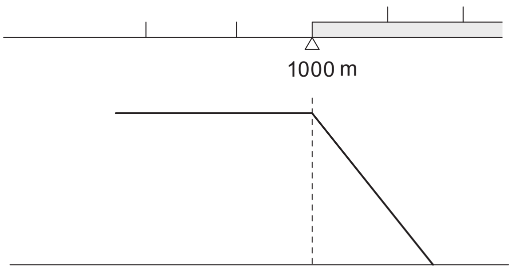
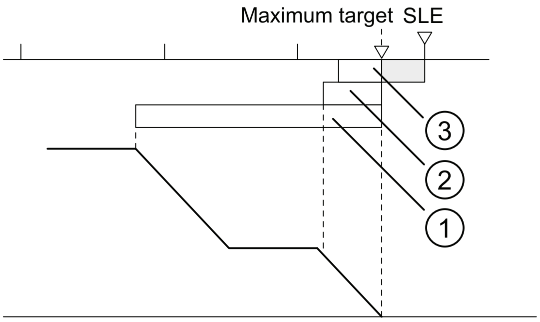
- Monitoring of standstill range
- Monitoring of target range
- Monitoring of standstill velocity

This operation does not result in proper positioning after which the "Position reached" signal is set.

Operating range limit for a linear axis

The limits for "Jog" mode differ between a synchronized and an unsynchronized axis.

Table 9- 1 Jogging with a synchronized and an unsynchronized axis

Axis is not synchronized	Axis is synchronized
If, when jogging, the traversing range limit is overrun, • the actual value indicated is no longer valid. • the positioning is aborted.  ▭ The actual value display is no longer valid	Jogging is a positioning on a target that is located the whole target range away from the software limit switches. The operating range limits are determined from: • **SLE 1/2 target range** for the end of the linear axis in plus direction • **SLE + 1/2 target range** for the end of the linear axis in minus direction If you do not open the button beforehand, the FM 351 shuts down at a target point that is positioned half the target range away from the respective software limit switch. The FM 351 sets all the ranges necessary for the correct shutdown around this target point  ▭ = Part of operating range in which a target position cannot be located ① = Changeover difference plus ② = Switch-off difference plus ③ = 1/2 target range

9.3 Configuring the reference point approach mode

Definition

In "Reference point approach" mode you can synchronize the axes on account of a repeating external event.

Requirements

- An incremental encoder with zero mark.
- The axis parameters must be assigned.

Connections	Channel 1	Channel 2
Reference point switch	Digital input 1I0	Digital input 2I0
	The reference point switch should be provided such that the drive can safely brake from rapid traverse to creep speed in the range of the switch.	
Reversing switch	Digital input 1I1	Digital input 2I1
	When assigning parameters, ensure that the start of the reference point approach is assigned in the direction of the reversing switch. Only in this way can you guarantee that the reference point switch will always be found.	
Enable input	Digital input 1I2	Digital input 2I2

Sequence of "Reference point approach" mode

The buttons for the positive and negative traversing direction must be installed for each channel.

1. Enter the value of the reference point coordinate in the parameter DB (REFPT).
2. Enter the type of "Reference point approach" in the parameter DB.

   The following options are available:

Start in direction ...	That leads to synchronization ...	
plus	First zero mark after exiting the reference point switch in **plus** direction	REFPT_TYPE=0
plus	First zero mark after exiting the reference point switch in **minus** direction	REFPT_TYPE=1
minus	First zero mark after exiting the reference point switch in **plus** direction	REFPT_TYPE=2
minus	First zero mark after exiting the reference point switch in **minus** direction	REFPT_TYPE=3

3. Enter the start velocity.
   – Rapid traverse (REFPT_SPD=0)
   – Creep speed (REFPT_SPD=1)
4. Write and enable the machine data.
5. Set the control signal for "Reference point approach" mode (MODE_IN=3).
6. Set the control signal for the drive enable (DRV_EN=1).
7. Set the function switch for "Do not evaluate enable input" (EI_OFF=1) or wire the enable input for the corresponding channel.
8. Set the control signal for the plus or minus traversing direction (DIR_P=1, DIR_M=1 or START=1)
9. Call the FB ABS_CTRL.

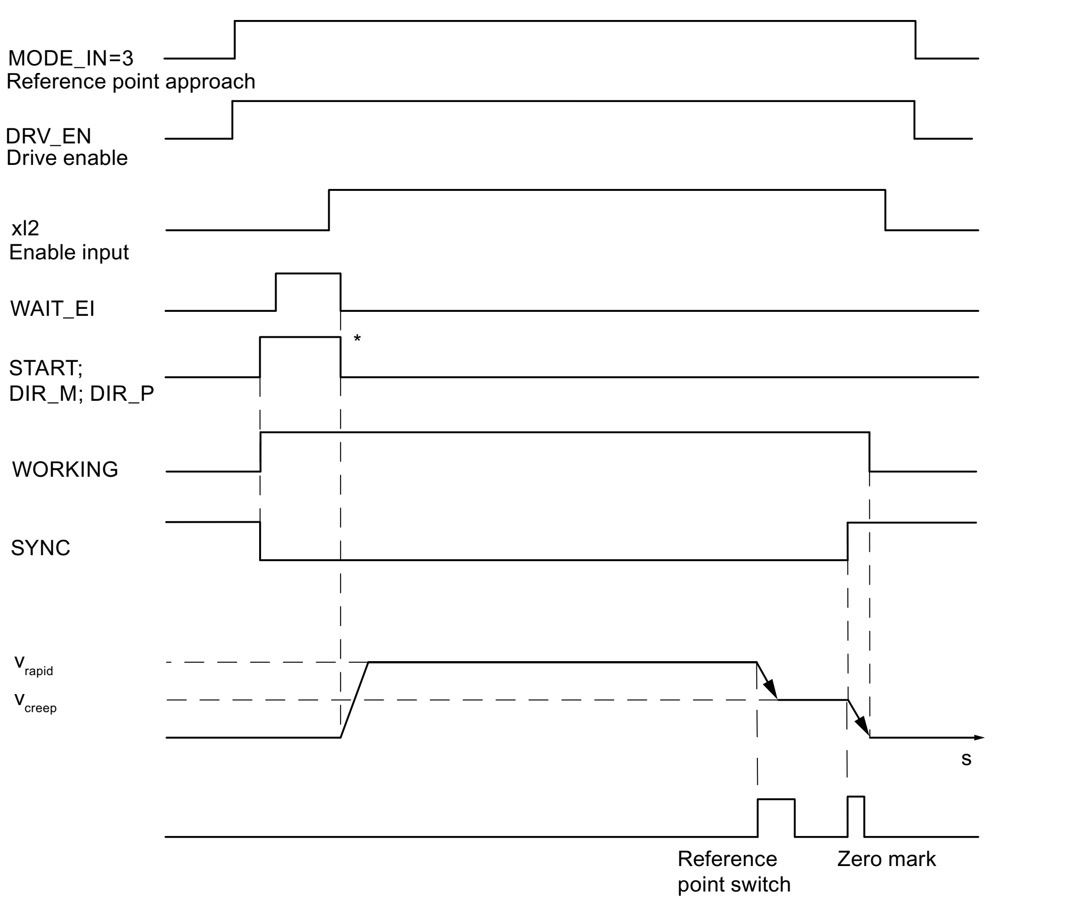
Table 9- 2 Start command for a reference point approach

Start command	Task	Comment
DIR_P	The drive starts in the direction of positive values. It therefore moves in the direction of the traversing range end.	If a negative direction is entered in the machine data, the FM 351 issues an operator error. No reference point approach is carried out.
DIR_M	The drive starts in the direction of negative values. It therefore moves in the direction of the traversing range start.	If a positive direction is entered in the machine data, the FM 351 issues an operator error. No reference point approach is carried out.
START	The drive starts in the direction that was entered in the machine data.	

Note

The following applies to the rotary axis: Reproducibility of the reference point is only ensured if the ratio of the **End of rotary axis** value and the **Distance per encoder revolution** value is an integer.

Example for "Reference point approach" mode



* The start signals are reset by the FB ABS_CTRL.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
15.0	START	BOOL	FALSE	1 = start positioning
15.2	DIR_M	BOOL	FALSE	1 = Minus direction
15.3	DIR_P	BOOL	FALSE	1 = Plus direction
15.7	DRV_EN	BOOL	FALSE	1 = Switch on drive enable
16.0	MODE_IN	BYTE	B#16#0	3 = reference point approach
23.0	ST_ENBLD	BOOL	FALSE	1 = Start enabled
23.1	WORKING	BOOL	FALSE	1 = Positioning running (processing running)
23.2	WAIT_EI	BOOL	FALSE	1 = axis waiting for external enable
25.0	SYNC	BOOL	FALSE	1 = axis is synchronized
34.2	EI_OFF	BOOL	FALSE	1 = Do not evaluate enable input

Data used in the parameter DB

Address	Name	Type	Initial value	Comment
44.0	REFPT	DINT	L#0	Reference point coordinate
52.0	REFPT_TYPE	DINT	L#0	Type of reference point approach
99.0	REFPT_SPD	BOOL	TRUE	Start velocity for the reference point approach 0 = rapid traverse 1 = creep speed

Effects of operating mode

- Synchronization is canceled when traversing starts.
- The actual position is set to the value of the reference point coordinate when the "SYNC" checkback signal is set.
- The operating range is specified on the axis.
- The individual points within the operating range maintain their original value, but have new physical positions.

Canceling the reference point approach

"Reference point approach" mode is canceled, if

- the drive enable signal is deleted (DRV_EN=0)
- a traversing range limit for a linear axis has been overrun

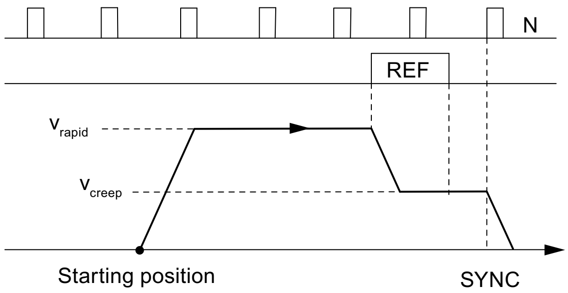
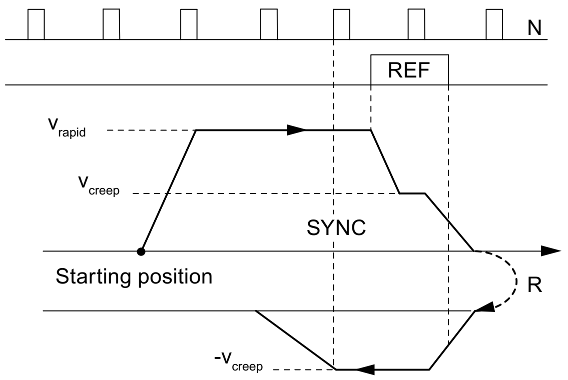
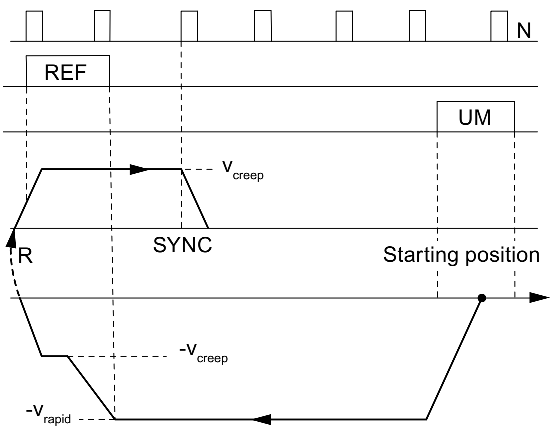
Reference point approach in dependence on the start position

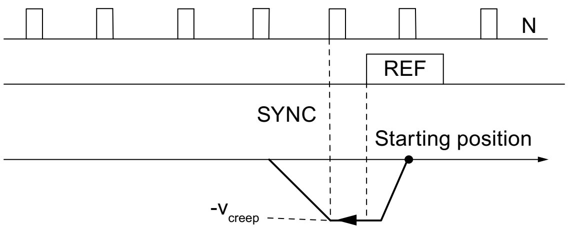
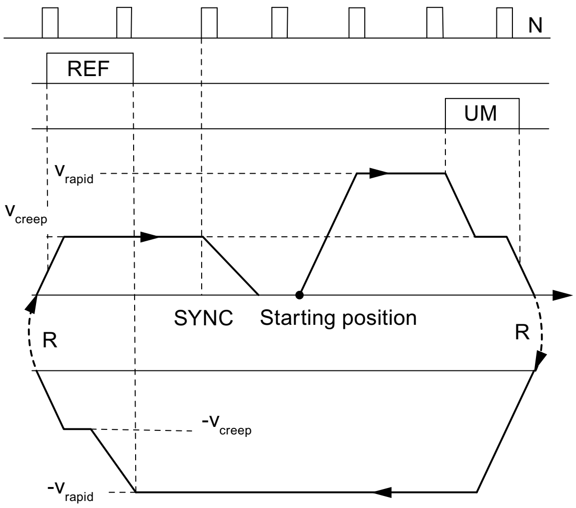
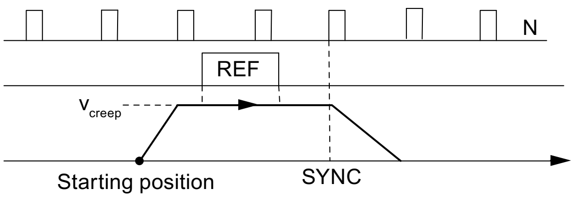
For a reference point approach you must distinguish between various scenarios that depend on the following:

- Location of the drive at the time a reference point approach is started
- Assigned start direction
- Assigned location of the zero mark relative to the reference point switch

The cases for REFPT_TYPE 0 and 1 are explained in the following table. The figures apply analogously to REFPT_TYPE 2 and 3.

Table 9- 3 Possibilities of a reference point approach

Conditions for reference point approach	Sequence of reference point approach
Example of reference point approach (REFPT_TYPE=0): • Start direction is plus. • The location of the zero mark of the reference point switch is assigned in the plus direction.	
Example of reference point approach (REFPT_TYPE=1): • Start direction is plus. • The location of the zero mark of the reference point switch is assigned in the minus direction.	
R = Direction reversal REF = Reference point switch UM = Reversing switch N = Zero mark of encoder SYNC = Synchronization was reached	
Example of reference point approach (REFPT_TYPE=0): • The start direction must be assigned in the plus direction. • The location of the zero mark of the reference point switch is assigned in the plus direction. • The reversing switch is more positive than the reference point switch.	

Conditions for reference point approach	Sequence of reference point approach
Example of reference point approach (REFPT_TYPE=1): • Start direction is plus. • The location of the zero mark of the reference point switch is assigned in the minus direction. • The start position of the reference point approach is on the reference point switch.	
Example of reference point approach (REFPT_TYPE=0): • Start direction is plus. • The location of the zero mark of the reference point switch is assigned in the plus direction. • The reversing switch is more positive than the reference point switch.	
Example of reference point approach (REFPT_TYPE=0): • Start direction is plus. • The location of the zero mark of the reference point switch is assigned in the plus direction. • Start velocity = creep speed	
R = Direction reversal REF = Reference point switch UM = Reversing switch N = Zero mark of encoder SYNC = Synchronization was reached	

9.4 Configuring the increment drive mode

Definition

In "incremental approach" mode the FM 351 can move the drive

- to **absolute** targets,
- **relatively** by an incremental distance in a specified direction.

The target or the relative distances are specified as increments of the FM 351. You have the option of entering up to 100 increments in a table. These are valid both for **relative incremental approach** mode as well as for **absolute incremental approach** mode. Independently of the increment table, you can specify the distance using the increments 254 and 255 (refer to the section entitled "Increments (Page 87)").

Requirements

- The axis parameters must be assigned.
- The axis must be synchronized.
- The increments must be available on the module.

Interpretation of the increments

The FM 351 interprets the specifications differently, depending on the "incremental approach" you select.

- Absolute incremental approach:
  The increments are interpreted as an absolute target position.
- Relative incremental approach:
  The increments are interpreted as a relative distance from the start position.

Note

Only positive increments are permissible for "Relative incremental approach" mode.

The signs of the increments result from the DIR_P and DIR_M direction specifications.

Sequence of "incremental approach" mode with increment number 1 - 100

<table>
<tr><th rowspan="2">Step</th><th>Absolute incremental approach</th><th>Relative incremental approach</th></tr>
<tr><th colspan="2">Increment number 1 - 100</th></tr>
<tr><td>1</td><td>Set the control signal for "Absolute incremental approach" mode (MODE_IN=5).</td><td>Set the control signal for "Relative incremental approach" mode (MODE_IN=4).</td></tr>
<tr><td>2</td><td colspan="2">Enter the increments in the tables (TRGL1; TRGL2).</td></tr>
<tr><td>3</td><td colspan="2">Write the increment tables (TRGL1/2WR_EN=1).</td></tr>
<tr><td>4</td><td colspan="2">Set the control signal for the drive enable (DRV_EN=1).</td></tr>
<tr><td>5</td><td colspan="2">Set the function switch for "Do not evaluate enable input" (EI_OFF=1) or wire the enable input for the corresponding channel.</td></tr>
<tr><td>6</td><td colspan="2">Enter the increment number (MODE_TYPE=1...100).</td></tr>
<tr><td>7</td><td>Set the control signal:
• Linear axis:
– START; the direction is determined explicitly by the target and the current actual value.
• Rotary axis:
– START; the target is approached on the shortest path.
– DIR_P; start in plus direction
– DIR_M; start in minus direction</td><td>Set the control signal:
• Linear axis:
– DIR_P; start in plus direction
– DIR_M; start in minus direction
• Rotary axis:
– DIR_P; start in plus direction
– DIR_M; start in minus direction</td></tr>
<tr><td>8</td><td colspan="2">Call the FB ABS_CTRL.</td></tr>
<tr><td colspan="3">Steps 2 and 3 are only required if no increments are given or the existing increments are to be changed.</td></tr>
</table>

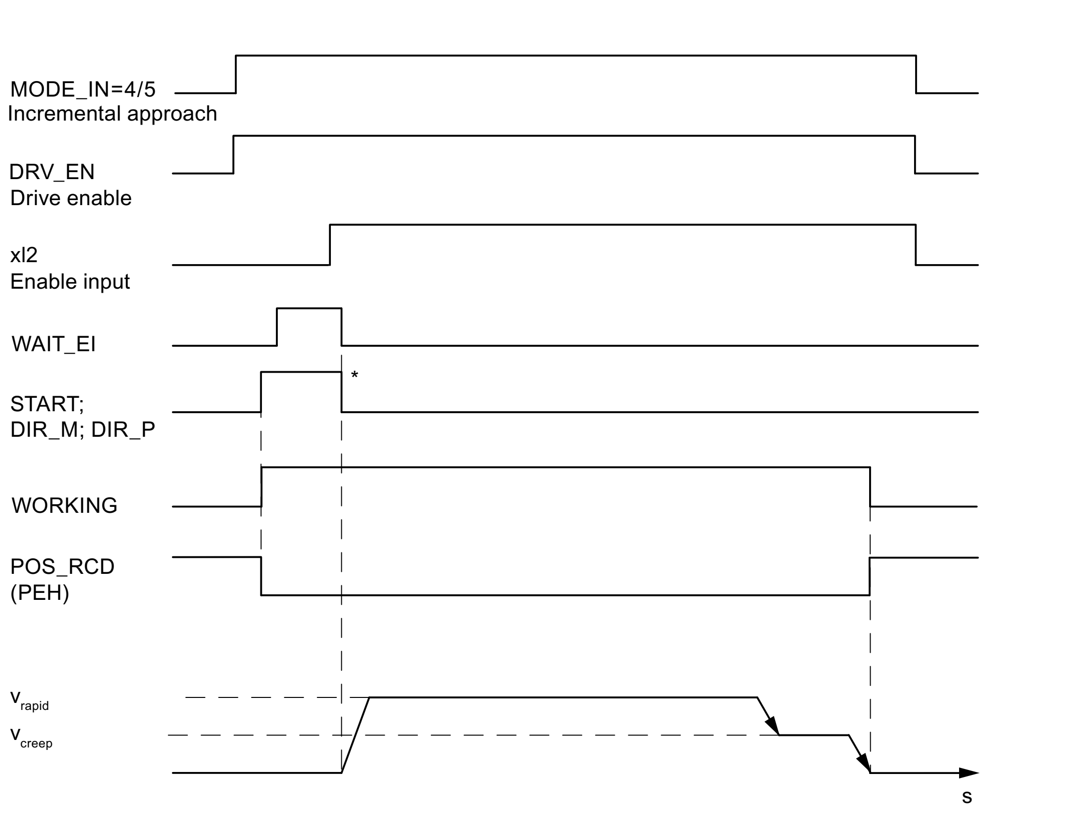
Sequence of "incremental approach" mode with increment number 254

<table>
<tr><th rowspan="2">Step</th><th>Absolute incremental approach</th><th>Relative incremental approach</th></tr>
<tr><th colspan="2">Increment number 254</th></tr>
<tr><td>1</td><td>Set the control signal for "Absolute incremental approach" mode (MODE_IN=5).</td><td>Set the control signal for "Relative incremental approach" mode (MODE_IN=4).</td></tr>
<tr><td>2</td><td colspan="2">Set the control signal for the drive enable (DRV_EN=1).</td></tr>
<tr><td>3</td><td colspan="2">Set the function switch for "Do not evaluate enable input" (EI_OFF=1) or wire the enable input for the corresponding channel.</td></tr>
<tr><td>4</td><td colspan="2">Enter the increment number (MODE_TYPE=254).</td></tr>
<tr><td>5</td><td colspan="2">Enter the increment for the increment number 254 (TRG252_254).</td></tr>
<tr><td>6</td><td colspan="2">Set the trigger bit for writing the increment (TRG252_254_EN=1).</td></tr>
<tr><td>7</td><td>Set the control signal:
• Linear axis:
– START; the direction is determined explicitly by the target and the current actual value.
• Rotary axis:
– START; the target is approached on the shortest path.
– DIR_P; start in plus direction
– DIR_M; start in minus direction</td><td>Set the control signal:
• Linear axis:
– DIR_P; start in plus direction
– DIR_M; start in minus direction
• Rotary axis:
– DIR_P; start in plus direction
– DIR_M; start in minus direction</td></tr>
<tr><td>8</td><td colspan="2">Call the FB ABS_CTRL.</td></tr>
</table>

Sequence of "incremental approach" mode with increment number 255

Step	Absolute incremental approach	Relative incremental approach
	Increment number 255	
1	Set the control signal for "Absolute incremental approach" mode (MODE_IN=5).	Set the control signal for "Relative incremental approach" mode (MODE_IN=4).
2	Set the control signal for the drive enable (DRV_EN=1).	
3	Set the function switch for "Do not evaluate enable input" (EI_OFF=1) or wire the enable input for the corresponding channel.	
4	Enter the increment number (MODE_TYPE=255).	
5	Enter the increment for the increment number 255 (TRG255).	
6	Enter the value for the changeover difference of the increment number 255 (CHGDIF255).	
7	Enter the value for the switch-off difference of the increment number 255 (CUTDIF255).	
8	Set the trigger bit for writing the increment, switch-off difference and changeover difference (TRG255_EN=1).	
9	Set the control signal: • Linear axis: – START; the direction is determined explicitly by the target and the current actual value. • Rotary axis: – START; the target is approached on the shortest path. – DIR_P; start in plus direction – DIR_M; start in minus direction	Set the control signal: • Linear axis: – DIR_P; start in plus direction – DIR_M; start in minus direction • Rotary axis: – DIR_P; start in plus direction – DIR_M; start in minus direction
10	Call the FB ABS_CTRL.	

Example for "incremental approach" mode



* The start signals are reset by the FB ABS_CTRL.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
15.0	START	BOOL	FALSE	1 = start positioning
15.2	DIR_M	BOOL	FALSE	1 = Minus direction
15.3	DIR_P	BOOL	FALSE	1 = Plus direction
15.7	DRV_EN	BOOL	FALSE	1 = Switch on drive enable
16.0	MODE_IN	BYTE	B#16#0	4 = relative incremental approach 5 = absolute incremental approach
17.0	MODE_TYPE	BYTE	B#16#0	Increment number 1 - 100, 254 or 255
23.0	ST_ENBLD	BOOL	FALSE	1 = Start enabled
23.1	WORKING	BOOL	FALSE	1 = Positioning running (processing running)
23.2	WAIT_EI	BOOL	FALSE	1 = axis waiting for external enable
25.7	POS_RCD	BOOL	FALSE	1 = Position reached
34.2	EI_OFF	BOOL	FALSE	1 = Do not evaluate enable input
36.2	TRG252_254_EN	BOOL	FALSE	1 = write increment for increment number 254
36.3	TRG255_EN	BOOL	FALSE	1 = write increment for increment number 255
35.4	TRGL1WR_EN	BOOL	FALSE	1 = write increment table 1 (increment number 1 ... 50)
35.5	TRGL2WR_EN	BOOL	FALSE	1 = write increment table 2 (increment number 51 ... 100)
96.0	TRG252_254	DINT	L#0	Increment for increment number 254
100.0	TRG255	DINT	L#0	Increment for increment number 255
104.0	CHGDIF_255	DINT	L#0	Changeover difference for increment number 255
108.0	CUTDIF_255	DINT	L#0	Switch-off difference for increment number 255

Data used in the parameter DB

Address	Name	Type	Initial value	Comment	
100.0	CHGDIF_P	DINT	L#5000	Changeover difference plus	
104.0	CHGDIF_M	DINT	L#5000	Changeover difference minus	
108.0	CUTDIF_P	DINT	L#2000	Switch-off difference plus	
112.0	CUTDIF_M	DINT	L#2000	Switch-off difference minus	
120.0	TRGL1.TRG[1]	DINT	L#0	Increment number 1	Increment table 1
. . .	. . .	. . .	. . .	. . .	
316.0	TRGL1.TRG[50]	DINT	L#0	Increment number 50	
320.0	TRGL2.TRG[51]	DINT	L#0	Increment number 51	Increment table 2
. . .	. . .	. . .	. . .	. . .	
516.0	TRGL2.TRG[100]	DINT	L#0	Increment number 100	

Distance-to-go

The distance-to-go is the specified difference between the target (increment) and the actual value.

In the case of a rotary axis, the displayed distance-to-go cannot be used.

Shutdown of incremental approach

"Incremental approach" mode is shut down when the FM 351 receives a STOP signal (STOP = 1).

After traversing is shut down, the distance-to-go remains.

The remaining distance-to-go with a "relative incremental approach" can be traveled to the end if all of the following conditions are met:

- The operating mode is unchanged
- The increment number is unchanged
- The direction is unchanged
- The remaining distance-to-go is greater than the assigned switch-off difference

You travel the distance-to-go by starting the "relative incremental approach" without any changes.

Canceling the incremental approach

"Incremental approach" mode is canceled when the "drive enable" signal is deleted (DRV_EN=0).

Delete distance-to-go

With the job "Delete distance-to-go" you can delete any awaiting distance-to-go.

You also delete the awaiting distance-to-go by calling a different operating mode or by starting the operating mode in the other direction.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
35.2	DELDIST_EN	BOOL	FALSE	1 = delete distance-to-go

9.5 Configuring set actual value / cancel set actual value

Definition

The job "Set actual value" assigns a new coordinate to the current encoder state. The operating range is projected on a different area of the axis.

Ascertain the shift of the operating range with ($ACTUAL_{new}$ -$ACTUAL_{current}$).

- $ACTUAL_{new}$ is the default value
- $ACTUAL_{current}$ is the actual value at the time of the execution

Requirements

- The axis parameters must be assigned.
- The axis must be synchronized.

Job sequence

1. Enter the coordinate AVAL for the actual value ($ACTUAL_{new}$).
   – Linear axis:

     You must select the specified actual value such that the software limit switches are still within the permissible traversing range after the job is called.

     The absolute value of the shift resulting from ($ACTUAL_{new}$ - $ACTUAL_{current}$) must be less than or equal to the absolute value of the permitted traversing range (max. 100 m or 1000 m).

   – Rotary axis:

     Rule for the defined actual value:

     0 ≤ actual value < end of rotary axis
2. Set the respective trigger bit (AVAL_EN=1).

If issued during a positioning, the "set actual value" job is held back until the end of the positioning, and it is carried out only after the subsequence call of the block.

Data used in the channel DB

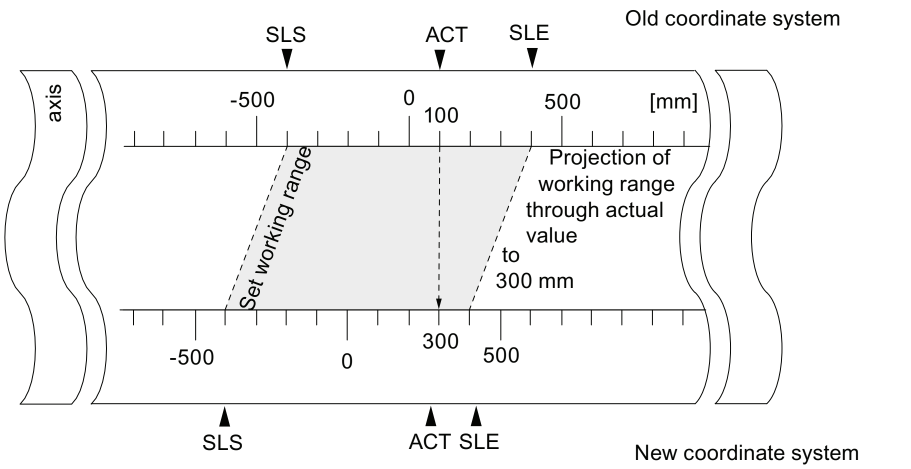
Address	Name	Type	Initial value	Comment
35.7	AVAL_EN	BOOL	FALSE	1 = set actual value
84.0	AVAL	DINT	L#0	Coordinate for "Set actual value"

Effects of the job

In the example "set actual value " to 300 mm, you can see how this job projects the operating range onto a certain position of the axis. This has the following results:

- The actual position is set to the value of the actual value coordinate.
- The operating range is shifted on the axis.
- The individual points (e.g., software limit switch end) within the operating range maintain their original value, but have new physical positions.

Table 9- 4 Shifting of the operating range on the axis by means of "set actual value"

<table>
<tr><th>Set actual value</th><th>SLS</th><th>ACTUAL</th><th>SLE</th></tr>
<tr><td rowspan="2">
</td><td>-400</td><td>100</td><td>400</td></tr>
<tr><td>-400</td><td>300</td><td>400</td></tr>
</table>

Withdrawing the job (cancel set actual value)

With the job "cancel set actual value" you reset all the operating range shifts that were created by means of the "set actual value".

The total of all operating range shifts must not exceed the traversing range, so that this job can be correctly executed.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
35.3	AVALREM_EN	BOOL	FALSE	1 = cancel set actual value

9.6 Configuring set reference point

Definition

You use the "set reference point" job to synchronize the axes. The job shifts the operating range. All the shifts that were created with set actual value are retained.

The setting projects the operating range on the axis. Therefore, by entering different values, the operating range can be moved to any position within the physical area of the axis.

Prerequisites

- The positioning must have ended.
- The axis parameters must be assigned.

Job sequence

1. Enter the value for the reference point coordinate (REFPT).
   – Linear axis:

   The reference point coordinate must not lie outside of the software limit switches. This also applies to the reference point coordinate in a shifted coordinate system.

   – Rotary axis:

   Rule for the reference point coordinate:

   0 ≤ reference point coordinate < end of rotary axis
2. Set the respective trigger bit (REFPT_EN).

Data used in the channel DB

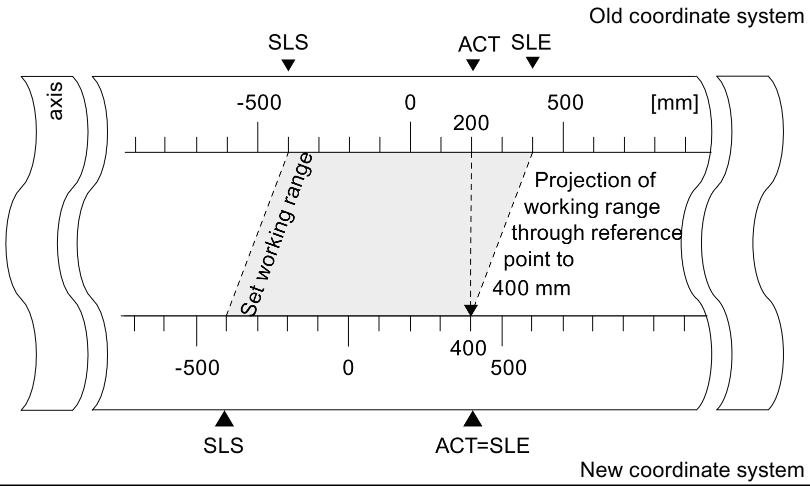
Address	Name	Type	Initial value	Comment
25.0	SYNC	BOOL	FALSE	1 = axis is synchronized
35.6	REFPT_EN	BOOL	FALSE	1 = set reference point
92.0	REFPT	DINT	L#0	Reference point coordinate

Effects of the job

In the example "set reference point" to 400 mm you can see how this job projects the operating range onto a specific, physical position of the axis. This has the following results:

- The actual position is set on the value of the reference point coordinate.
- The operating range is shifted on the axis.
- The individual points (e.g. software limit switch end) maintain their original value, but are located at new positions.
- The SYNC bit in the checkback signals is set.

Table 9- 5 Shift of the operating range on the axis by means of "set reference point"

<table>
<tr><th>Set reference point</th><th>SLS</th><th>ACTUAL</th><th>SLE</th></tr>
<tr><td rowspan="2">
</td><td>-400</td><td>200</td><td>400</td></tr>
<tr><td>-400</td><td>400</td><td>400</td></tr>
</table>

Special features of the absolute encoder

This job is required for an absolute adjustment (refer to section entitled "Determining the absolute encoder adjustment (Page 82)").

9.7 Configuring the loop drive

Definition

With "loop approach" you specify the direction in which a target is approached with a frictional connection. You can use loop approach if a frictional connection between the motor and axis can only be ensured in one direction.

A target that is approached in the direction opposite the specified direction is initially overrun. Then the FM 351 carries out an about-turn and approaches the target in the specified direction.

Requirements

- The axis parameters must be assigned.
- The axis must be synchronized.
- In the case of a loop approach in the opposite direction to that of the traversing direction of the target the maximum target position is:
  - traversing direction plus

    target < SLE - $^1/_2$ target range - switch-off difference plus - changeover difference minus
  - traversing direction minus

    target > SLS + $^1/_2$ target range + switch-off difference minus + changeover difference plus
- A loop approach will not be carried out if the target is being approached in the direction of the loop approach. In this case, an incremental approach without an about-turn in direction will be carried out.
- The sequence of the "incremental approach" mode must be known (refer to the section entitled "Configuring the increment drive mode (Page 109)").

Sequence of events for the loop approach

1. Set the control signal for the "absolute / relative incremental approach" mode (MODE_IN=4/5).
2. Set the control signal for the drive enable (DRV_EN=1).
3. Set the function switches for "Do not evaluate enable input" (EI_OFF=1) or wire the corresponding channel for the input enable.
4. Enter the increment number (MODE_TYPE=1...100, 254, 255).
5. Set the function switches (PLOOP_ON / MLOOP_ON=1).
6. Start the incremental approach.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
15.0	START	BOOL	FALSE	1 = start positioning
15.2	DIR_M	BOOL	FALSE	1 = minus direction
15.3	DIR_P	BOOL	FALSE	1 = plus direction
15.7	DRV_EN	BOOL	FALSE	1 = switch on drive enable
16.0	MODE_IN	BYTE	B#16#0	4/5 = relative / absolute incremental approach
17.0	MODE_TYPE	BYTE	B#16#0	Increment number 1-100, 254 or 255
34.0	PLOOP_ON	BOOL	FALSE	1 = loop approach in plus direction
34.1	MLOOP_ON	BOOL	FALSE	1 = loop approach in minus direction
34.2	EI_OFF	BOOL	FALSE	1 = do not evaluate enable input

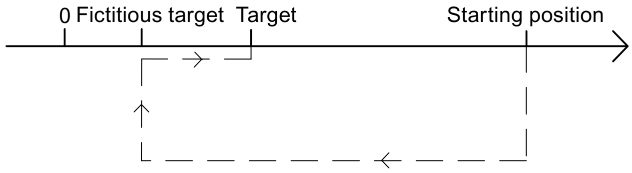
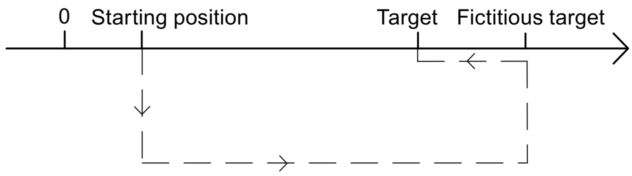
Fictitious target

If you start positioning to a target that is located in the direction opposite that of the assigned loop approach, the FM 351 determines a fictitious target for this target. The FM 351 performs a direction reversal at the fictitious target and then approaches the target in the proper direction.

This fictitious target **must** be located by at least half the target range before the respective software limit switch.

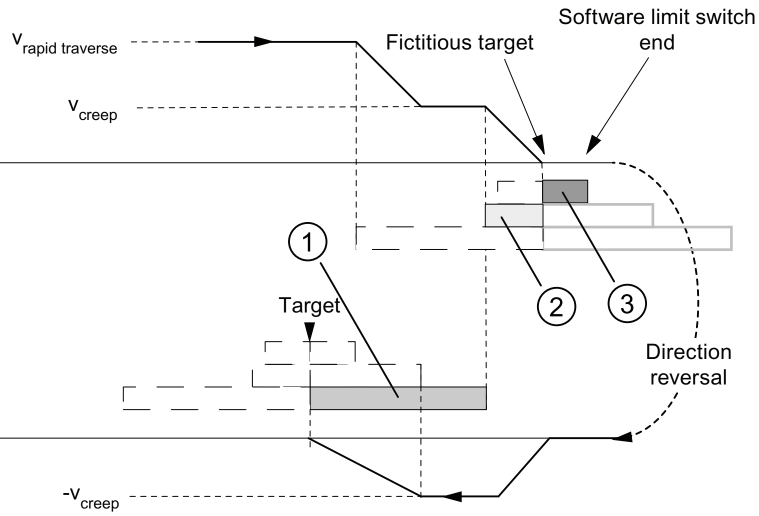
The distance of the fictitious target from the assigned target is determined depending on the direction:

Table 9- 6 Calculating the location of the fictitious target with a loop approach

Guidelines	Location of fictitious target
Parameter assignment: Loop + (frictional connection plus) and traverse in minus direction.	The fictitious target (**$target_f$**) has the value: **$target_f$ = target - switch-off difference minus - changeover difference plus** 
Parameter assignment: Loop - (frictional connection minus) and traverse in plus direction.	The fictitious target (**$target_f$**) has the value: **$target_f$ = target + switch-off difference plus + changeover difference minus** 

Example

Using a positioning with loop approach minus to a maximum target we can illustrate the location of the fictitious target.



① Changeover difference negative
② Switch-off difference positive
③ 1/2 target range

Figure 9-8 Loop approach minus to a maximum target

9.8 Enable input

Definition

The enable input is an external input with which a positioning can be enabled by an external event.

Evaluating the enable input (EI_OFF=0)

The respective enable input (xI2) must be wired for the channel.

With this you have the possibility of preparing the start of a positioning. Start the positioning, independently of the program flow of your user program, setting a "1" signal on the enable input.

The journey begins when you set a "1" signal on the enable input and is shut down when you set a "0" signal on the enable input.

Do not evaluate enable input (EI_OFF=1)

If you switch off the evaluation of the enable input, an operating mode starts immediately after the start signal is detected. Then it is not possible to prepare an operating mode and start it at a defined later time.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
34.2	EI_OFF	BOOL	FALSE	1 = do not evaluate enable input

9.9 Read position data

Definition

With the "read position data" job you can read the increment, distance-to-go and speed at the current time.

Job sequence

1. Set the trigger bit in the channel DB (ACTSPD_EN=1).
2. The data is then stored in the channel DB.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
37.1	ACTSPD_EN	BOOL	FALSE	1 = read position data
112.0	ACTSPD	DINT	L#0	Current speed
116.0	DIST_TO_GO	DINT	L#0	Distance-to-go
120.0	ACT_TRG	DINT	L#0	Current increment

9.10 Read encoder data

Definition

With the "read encoder data" job you can read the current data of the encoder as well as the value for the absolute encoder adjustment.

Prerequisites

You can read the value for the absolute adjustment after you have carried out the "set reference point" job (refer to the section entitled "Determining the absolute encoder adjustment (Page 82)").

Job sequence

1. Set the trigger bit in the channel DB (ENCVAL_EN=1).
2. The data is then stored in the channel DB.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
37.2	ENCVAL_EN	BOOL	FALSE	1 = read encoder values
124.0	ENCVAL	DINT	L#0	Actual encoder value (internal representation)
128.0	ZEROVAL	DINT	L#0	Last zero mark value (internal representation)
132.0	ENC_ADJ	DINT	L#0	Absolute encoder adjustment

9.11 Checkback signals for the positioning

Definition

The "checkback signals for the positioning" inform you of the current status of the positioning.

Sequence

The data are stored in the channel DB at each call of FB ABS_CTRL.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
23.0	ST_ENBLD	BOOL	FALSE	1 = Start enabled
23.1	WORKING	BOOL	FALSE	1 = Positioning running (processing running)
23.2	WAIT_EI	BOOL	FALSE	1 = axis waiting for external enable
23.4	SPEED_OUT	BOOL	FALSE	0 = creep speed 1 = rapid traverse
23.5	ZSPEED	BOOL	FALSE	1 = axis is located in the standstill range
23.6	CUTOFF	BOOL	FALSE	1 = axis is located in the switch-off range
23.7	CHGOVER	BOOL	FALSE	1 = axis is located in the changeover range
24.0	MODE_OUT	BYTE	B#16#0	Active operating mode
25.2	GO_M	BOOL	FALSE	1 = axis moves in minus direction
25.3	GO_P	BOOL	FALSE	1 = axis moves in plus direction
25.7	POS_RCD	BOOL	FALSE	1 = Position reached
26.0	ACT_POS	DINT	L#0	Current actual value (current position of axis)

9.12 Checkback signals for the diagnostics

Definition

The "checkback signals for the diagnostics" informs you of occurring diagnostic events.

Sequence

1. If the module enters a new event in the diagnostic buffer, it sets the DIAG bit in all channels in the checkback interface. Every time an error occurs, all of the error classes listed in the appendix "Data blocks / Error lists" create an entry in the diagnostic buffer. The DIAG bit is also set when the diagnostic buffer is deleted.
2. If it is not possible to call an operating mode or control an active operating mode, or if this is carried out incorrectly, the module sets an operator error OT_ERR in the checkback interface. The cause of error is then entered in the diagnostic buffer. While the operator error exists, you can neither start a new operating mode nor continue with the halted one. You acknowledge an existing operator error with OT_ERR_A=1.
3. The module sets the DATA_ERR bit in the checkback interface when it detects faulty data in a write job. The cause of error is then entered in the diagnostic buffer.
4. The checkback signals are stored in the channel DB.
5. When the diagnostic buffer is read, the module sets the DIAG bit back to 0 in all channels.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
22.2	DIAG	BOOL	FALSE	1 = delete diagnostic buffer
22.3	OT_ERR	BOOL	FALSE	1 = operator error
22.4	DATA_ERR	BOOL	FALSE	1 = data error

Encoder 10

10.1 Incremental encoder

Connectable incremental encoders

Only incremental encoders with two pulses electrically offset at 90° with or without zero marks are supported:

- Encoders with asymmetrical output signals with 24 V level
  - Limit frequency = 50 kHz
  - cable up to 100 m long
- Encoders with symmetrical output signals with 5 V differential interface in accordance with RS 422
  - Limit frequency = 400 kHz
  - with 5 V voltage supply: cable up to 32 m long
  - with 24 V voltage supply: cable up to 100 m long

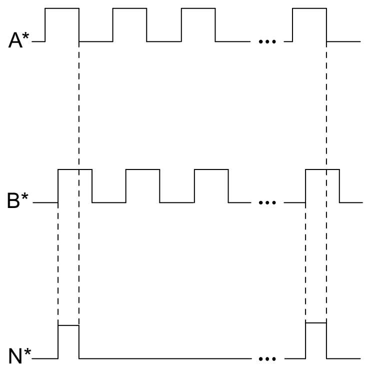
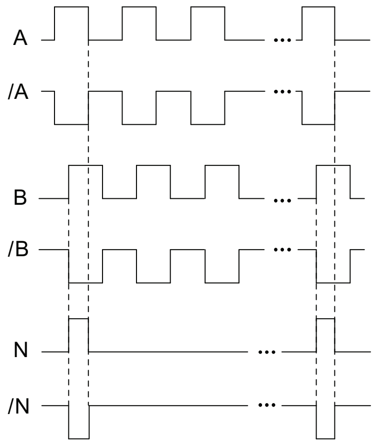
Note

If the 5 V encoder does not output a zero mark signal and wire-break monitoring is enabled, you must interconnect the zero mark signal inputs N and /N externally so that the inputs will exhibit different signal levels (for example, N to 5V, /N to ground).

Signal forms

The following figure illustrates the signal forms of encoders with asymmetrical and symmetrical output signals.

Table 10- 1 Incremental encoder signal forms

asymmetrical	symmetrical
	

Signal evaluation

increments

An increment identifies a signal period of the two signals (A and B) of an encoder. This value is specified in the technical specifications of an encoder and / or on its ID plate.

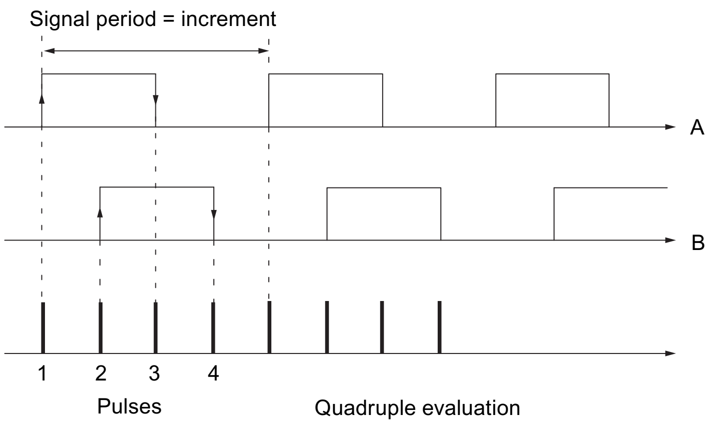


Figure 10-1 Increments and pulses

Pulses

The positioning module evaluates all 4 edges of signals A and B in every increment (quadruple evaluation).

Pulses
1 increment (encoder specification) = 4 pulses (FM evaluation)

Response times

The positioning module has the following response times for connected incremental encoders:

Response times
Response time = switching cycle of the connected switching elements

Note

You can compensate for the minimum response time by assigning the changeover difference and switch-off difference accordingly.

Indecision

The indecision influences the precision of the positioning. In the case of incremental encoders the indecision is negligible.

10.2 Absolute encoder

Single-turn and multiturn encoders

Absolute encoders are divided into the categories:

- Single-turn encoders

  The total range of single-turn encoders is scaled to one revolution.
- Multiturn encoders

  The total range of multiturn encoders is scaled to several revolutions.

Connectable absolute encoders

Only absolute encoders with serial interfaces are supported. The transfer of the path information takes place synchronously on the basis of the SSI protocol (**s**ynchronous **s**erial **i**nterface). The FM 351 supports only GRAY code. The data formats 25-bit (fir tree) and 13-bit (half fir tree) result from the arrangement of the data bits in the message frames.

Encoder type	Message frame length
Single-turn encoders	13 bits
Single-turn encoders	25 bits
Multiturn encoders	25 bits

Data Transmission

The baud rate for data transmission depends on the cable length (refer to the appendix section entitled "Technical specifications (Page 167)").

Pulse evaluation of absolute encoders

Pulse evaluation of absolute encoders
1 increment (encoder specification) = 1 pulse (FM evaluation)

Response times

The FM 351 has the following response times for absolute encoders:

Response times
Minimum response time = message frame time + switching cycle of the connected switching elements
Maximum response time = 2 x message frame time + monoflop time + switching cycle of the connected switching elements
With programmable absolute encoders: Maximum response time = message frame time + monoflop time + switching cycle of the connected switching elements +1/max. step sequence frequency

Monoflop time

The monoflop time is 64 µs.

Encoders with values greater than the limits specified here are not permitted.

Message frame times

The message frame times depend on the baud rate:

Baud rate	Message frame time with 13 bit	Message frame time with 25 bit
0.188 MHz	75 µs	139 µs
0.375 MHz	38 µs	70 µs
0.750 MHz	19 µs	35 µs
1.500 MHz	10 µs	18 µs

Example of response times

This example shows you how to calculate the minimum and maximum response time. The example does not use a programmable encoder.

- Hardware switching cycle: approx. 150 µs
- Message frame time: 18 µs at 1.5 MHz baud rate (25 Bit message frames)
- Monoflop time: 64 µs

Minimum response time = 18 µs + 150 µs = 168 µs

Maximum response time = 2 x 18 µs + 64 µs + 150 µs = 250 µs

Note

You can compensate for the minimum response time by assigning the changeover and switch-off differences accordingly.

Indecision

The indecision is the difference between the maximum and the minimum response time. With an absolute encoder this amounts to

Indecision
Indecision = message frame time + monoflop time
With programmable absolute encoders: Indecision = message frame time + monoflop time + 1/max. step sequence frequency

Diagnosis 11

11.1 Possibilities of error display and error evaluation

Information regarding errors

You receive information regarding errors in the following ways:

- Observe the error LEDs on the module.

  You can look up the meaning of the error LEDs in the section entitled "Meaning of the error LEDs (Page 139)".
- Connect your PG with the CPU and open the error evaluation screen of the configuration software. The current (error) status of the module is shown with the error class, error number, and in plain text. If necessary, update the display by pressing the "update" button. Causes and remedies for the displayed error messages can be found in the error list, in the appendix under "Error classes (Page 189)".
- Provide your user program with a detailed error evaluation (refer to the section entitled "Error display with OP (Page 140)") or a response to a diagnostic alarm (refer to the sections entitled "Error display with OP (Page 140)" and "Diagnostics interrupts (Page 147)").
- For a display in an OP: Read out the diagnostic buffer of the module cyclically in your user program. Evaluate the diagnostic DB in the OP. The meaning of error class and error number can be found in the error list, in the appendix under "Error classes (Page 189)".

Delete diagnostic buffer

In order for you to chronologically organize error messages better, the positioning module offers the possibility of completely deleting the diagnostic buffer. But this is then only possible when the positioning has already ended and the channel has been assigned.

11.2 Types of error

11.2.1 Synchronous errors

Description

These errors occur synchronously to a job or to the start of a positioning. Synchronous errors are operator errors (error class 2), data errors (error class 4), machine data errors (error class 5), increment table errors (error class 6) (refer to the appendix for a description of error classes).

11.2.2 Asynchronous errors

Description

These errors occur during operation on account of external events. They trigger a diagnostic interrupt. Asynchronous errors are operating errors (error class 1) and diagnostic errors (error class 128) (refer to the appendix for a description of error classes).

11.3 Meaning of the error LEDs

Error LEDs

The status and error display shows different error states.

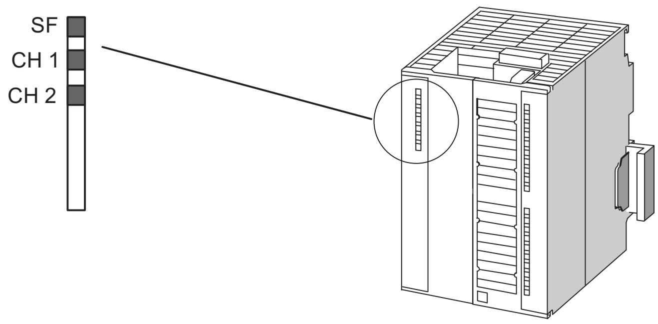


Figure 11-1 FM 351 status and error displays

Display	Meaning	Explanations
SF (red) LED - ON	Group errors	This LED indicates an error state on the FM 351. Diagnostic interrupt (internal or external (channel) errors) To remedy the error, refer to the error list, in the appendix under "Error classes (Page 189)".
CH 1 (red) CH 2 (red)	Channel error 1 Channel error 2	These LEDs show a channel error on channel 1 or on channel 2. • Encoder wire break • Absolute encoder error • Missing pulses of Incremental encoder • Operating error • Parameter assignment error in a parameter assignment from the SDB.

11.4 Error display with OP

Program structure

The following figure shows the "General program structure" of a user program presented in the section entitled "FB ABS_CTRL (FB 1) (Page 39)", expanded to include the readout of the diagnostic buffer for display on an OP. The FB ABS_DIAG stores the diagnostic buffer in a diagnostic DB that can be displayed by the OP.

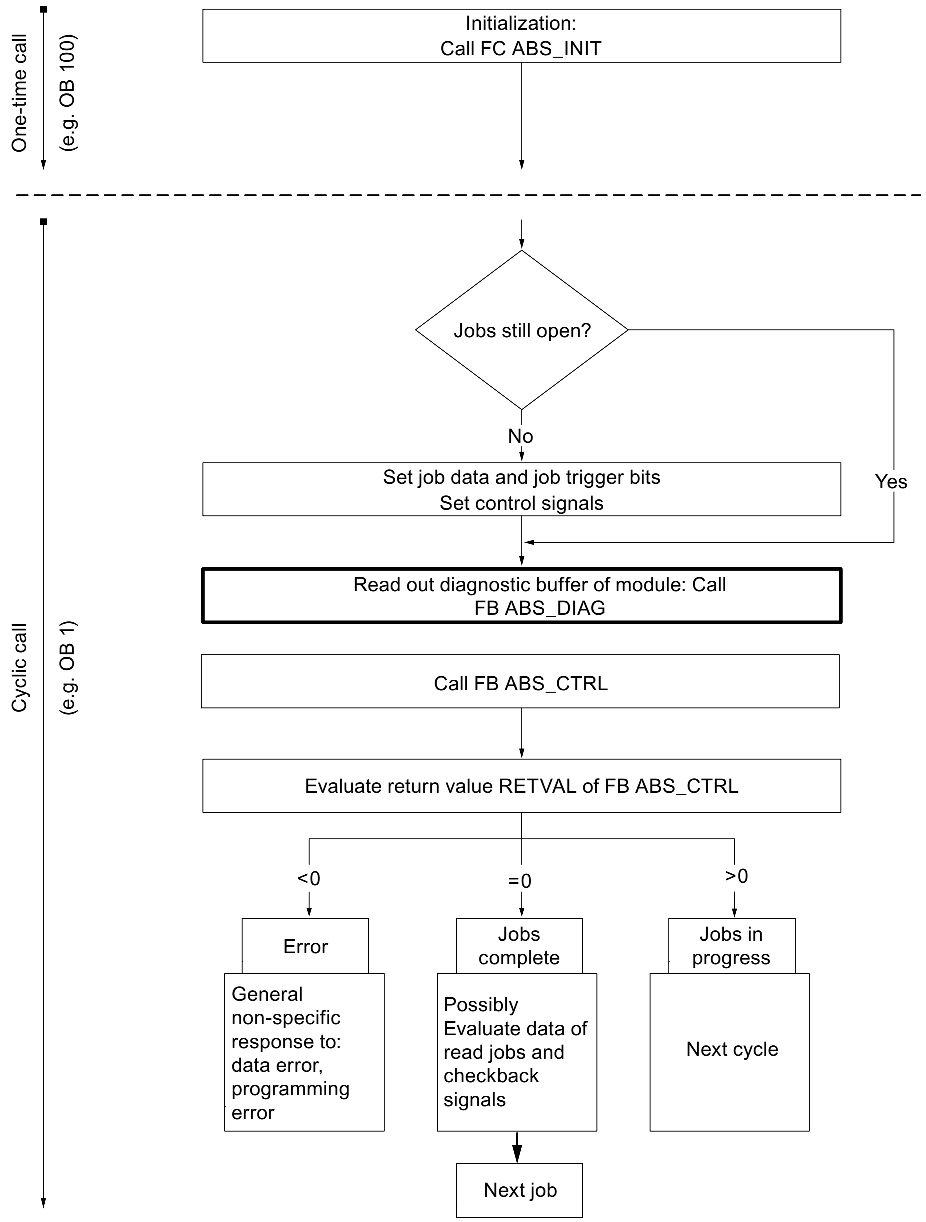


Figure 11-2 Program structure with diagnostic display for the OP

11.5 Error evaluation in the user program

Error response in the user program

In the user program you can respond selectively to errors. The following methods can be used:

- The return value RETVAL of the integrated standard FBs:

  this value is ascertained each time the block is called.

  RETVAL = -1 is a group display for a synchronous error in a job or when communicating with the module.
- An error bit _ERR belongs to each job. This is a group display for an error in the job or one of its predecessors in a job chain:

  the error bit is set for a write job and its subsequent jobs when a data error is reported by the module or a communication error occurs.

  With reading jobs the error bit is set for the job in question when a communication error occurs.

  The FB ABS_CTRL resets the error bits once a job has been processed. These should however be canceled by the user program in the case of an error assessment.
- The checkback signal DATA_ERR as a group display for an error that the module has detected during a write job. The signal will be established again with the next write job.
- The checkback signal OT_ERR (operator error) as a group display for an error that the module has detected when starting a traverse. The error must be acknowledged with OT_ERR_A=1 after the cause has been remedied.
- The checkback signal DIAG is set when the contents of the diagnostic buffer have changed. This signal can come along later than the signals DATA_ERR and OT_ERR.
- The communication error JOB_ERR includes the error code in the case of a communication problem between the FB and the module (refer to list of JOB_ERR messages in the appendix "List of JOB_ERR messages (Page 188)"). The value is determined again after a job has been processed and is stored in the channel DB for the FB ABS_CTRL and in the diagnostic DB for the FB ABS_DIAG.
- The FB ABS_DIAG for reading out the diagnostic buffer of the module. Here you can discover the error causes for synchronous and asynchronous events.
- Diagnostic interrupts for the fast response to events in the diagnostic interrupt OB (OB 82).

In the following figure you will find a possible program structure with which you can respond to the checkback signals "data error" (DATA_ERR), "operator error" (OT_ERR) and the error bits of the job (_ERR).

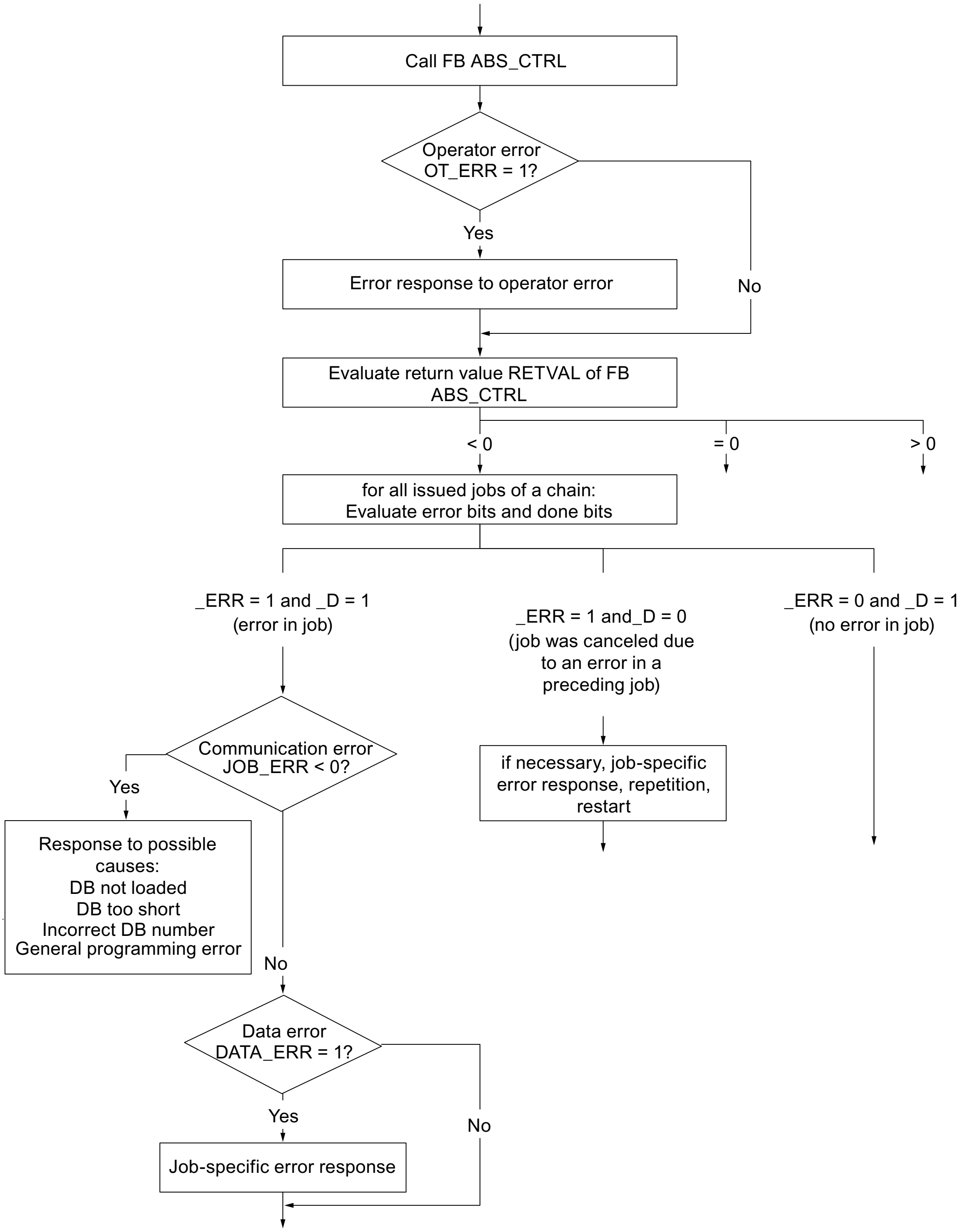


Figure 11-3 User program with evaluation of the error bits in the jobs

The following image offers a possible program structure using which you can evaluate all the errors via the entries in the diagnostic DB. In this way you can respond to the program if one or more errors are reentered in the diagnostic buffer of the module. Some possible program responses are listed in the following detailed figures.

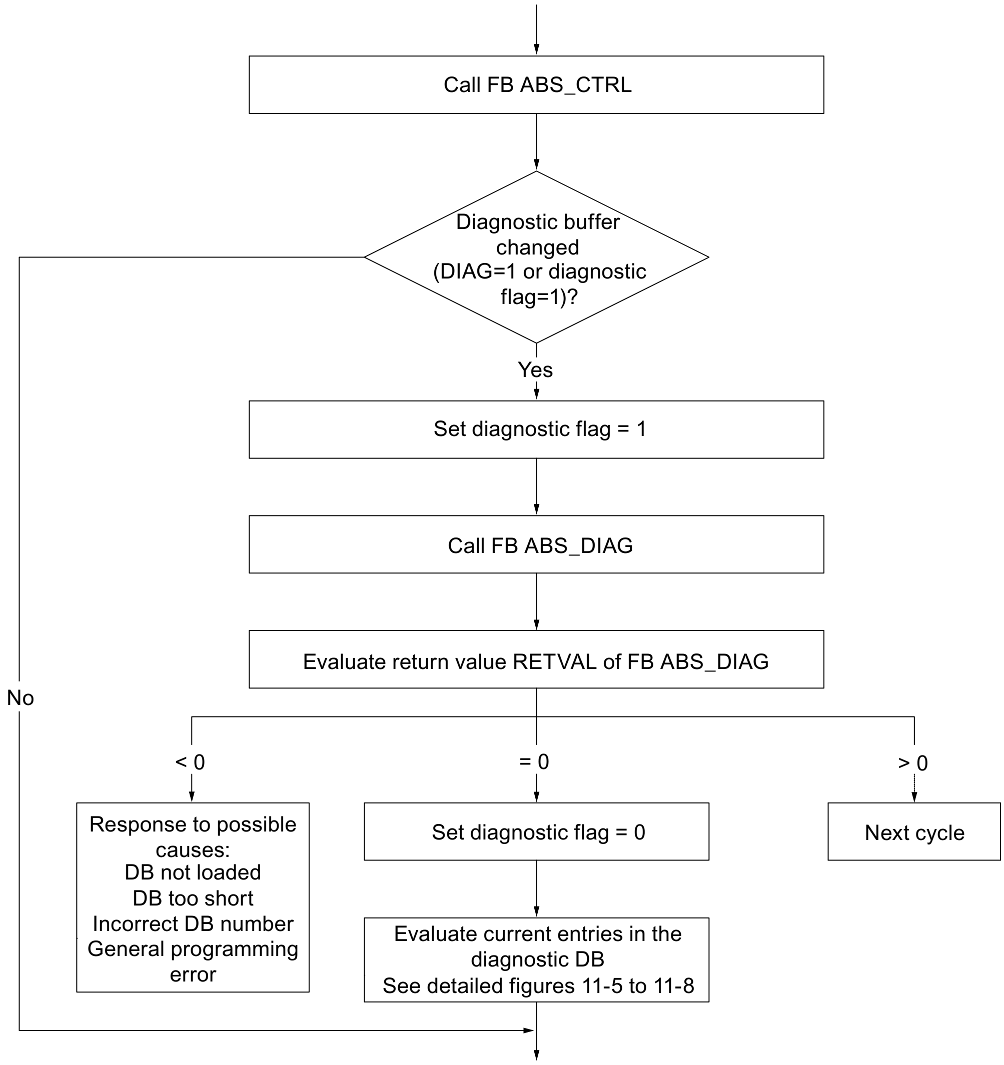


Figure 11-4 User program with complete error evaluation via the diagnostic DB

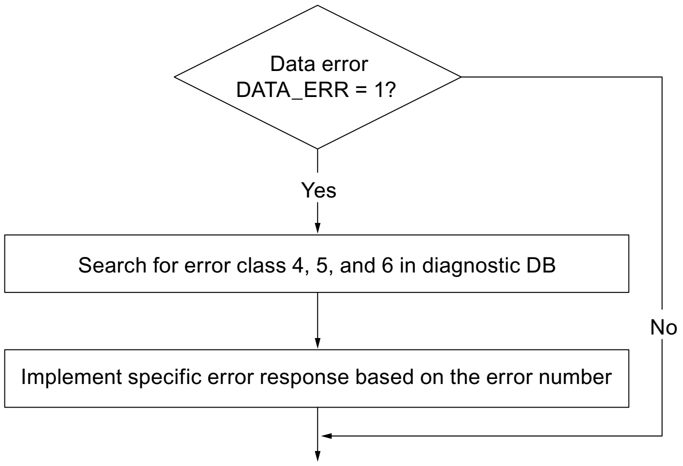


Figure 11-5 Possible evaluation of a data error

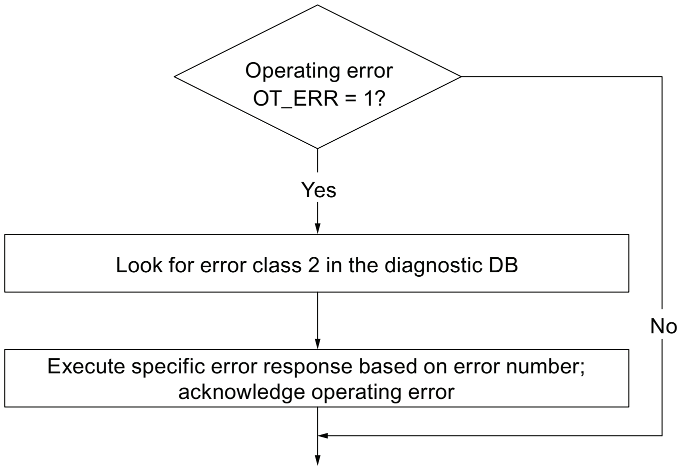


Figure 11-6 Possible evaluation of an operator error

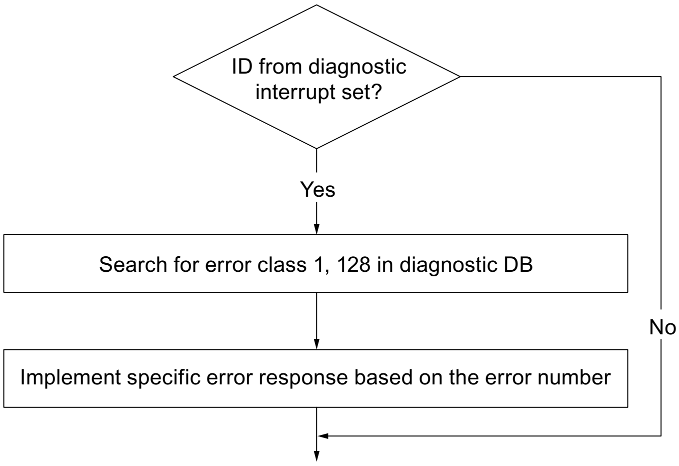


Figure 11-7 Possible evaluation of a diagnostic interrupt

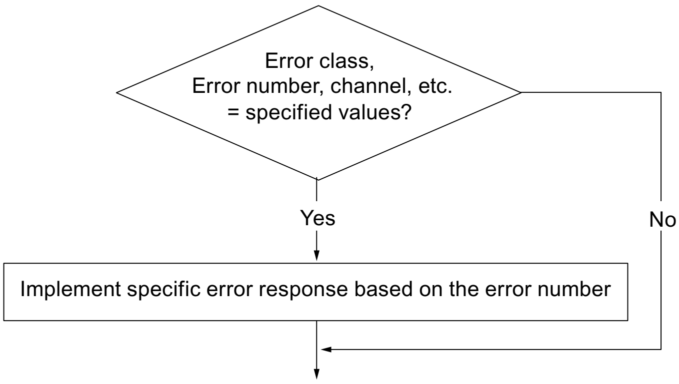


Figure 11-8 Possible evaluation of a specifically stipulated error

11.6 Diagnostics buffer of the module

Diagnostic events

The module´s diagnostic buffer includes maximum 9 diagnostic events and is organized as a ring buffer.

A diagnostic event is written in the buffer if an "incoming" (error) message is detected. This can be a message, a synchronous error (data error, operator error) or even an asynchronous error (operational error and diagnostics error). From a cause of error several entries can occur as follow-up errors. Outgoing messages do not create any entries in the diagnostic buffer.

For each diagnostic event the following is given:

- Status (always incoming)
- Internal error
- External error
- Error class
- Error number
- Channel number
- Increment number (in the case of increment table errors)

When a diagnostic event is written in the diagnostic buffer, the checkback signal DIAG=1 is set in all assigned channels.

With the FB ABS_DIAG, the diagnostic buffer can be transferred as a whole into a data block (diagnostic DB) or displayed via the error evaluation screen of the configuration software. If the diagnostic buffer is read, the module sets the checkback signal DIAG=0.

Note

If the diagnostic buffer is read simultaneously from the FB ABS_DIAG and the error evaluation screen, it is possible that a new diagnostic event is not detected by the program.

11.7 Diagnostics interrupts

Interrupt processing

The FM 351 can trigger diagnostic interrupts. You process these interrupts in an interrupt OB. If an interrupt is triggered without the associated OB having been loaded, the CPU will switch to STOP mode (see SIMATIC Programming With STEP 7 Manual (http://support.automation.siemens.com/WW/view/en/45531107)).

You enable the processing of the diagnostic interrupts as follows:

1. Select the module in HW Config
2. Enable the diagnostic interrupt under **Edit > Object properties > Basic parameters.**
3. Save and compile the hardware configuration.
4. Download the hardware configuration to the CPU.

Overview of diagnostic interrupts

The following events and errors trigger a diagnostic interrupt:

- Operating error
- Incorrect machine data (in parameter assignment via SDB)
- Diagnostic errors

These errors are explained in detail in the appendix section entitled "Error classes (Page 189)".

Response of the FM 351 in the case of an error with diagnostic interrupt

- The positioning is canceled.
- The synchronization is cleared when any of the following diagnostic interrupts occur:
  - Front connector missing, external auxiliary voltage for the encoder supply is missing
  - Zero mark error detected, cable fault (5 V encoder signals)
  - Traversing range exited (indicated by an operating error)
  - Set actual value cannot be executed (indicated by an operating error).
- The control signals START, DIR_P and DIR_M will no longer be processed apart for one exception

  Exception:

  When an operating error occurs, jogging in the direction of the operating range is still possible.
- Function switches and jobs will no longer be processed.

FM 351 detects an error "incoming"

A diagnostic interrupt is "incoming", if at least one error is given. If not all errors are remedied, the remaining existing errors will be reported again as "incoming".

Sequence:

1. The FM 351 detects one or more errors and triggers a diagnostic interrupt. The "SF" LED and, depending on error, the "CH1" / "CH2" LEDs illuminate. The error event is entered in the diagnostic buffer.
2. The operating system of the CPU calls OB 82.
3. You can evaluate the start info of OB 82.
4. Via the parameter OB82_MOD_ADDR you can read which module has triggered the interrupt.
5. For further information call the FB ABS_DIAG.

FM 351 detects the transition to the error-free state ("outgoing")

A diagnostic interrupt is then only "outgoing" if the last error on the module has been remedied.

Sequence:

1. The FM 351 detects that all errors have been cleared and triggers a diagnostic interrupt. The "SF" LED no longer lights up. The diagnostic buffer is not changed.
2. The operating system of the CPU calls OB 82.
3. Via the parameter OB82_MOD_ADDR you can read which module has triggered the interrupt.
4. Evaluate the OB82_MDL_DEFECT bit.

   If this bit is "0", there are no errors on the module. You can end your evaluation here.

Diagnostic interrupts in dependence on the CPU status

- When the CPU is in STOP, the diagnostic interrupts are blocked by the FM 351.
- If, with the CPU in STOP, not all the existing errors have been remedied, the FM 351 again reports the unremedied errors after the transition to RUN state as "incoming".
- If all the existing errors have been remedied with the CPU in STOP, then an error-free status of the FM 351 after transition to RUN is **not** reported with a diagnostic interrupt.

Evaluation of a diagnostic interrupt in the user program

The following entries in the local data of the diagnostic interrupt OB (OB 82) will be set by the FM 351. The errors are also entered in the diagnostic buffer (error class 128, for meaning and possible remedies, see appendix "Error classes (Page 189)"):

Address	Name	Type	Comment
0.0	OB82_EV_CLASS	BYTE	Event class and identifiers: B#16#38: Outgoing event B#16#39: Incoming event
1.0	OB82_FLT_ID	BYTE	Error code (B#16#42)
2.0	OB82_PRIORITY	BYTE	Priority class: B#16#1A in RUN mode B#16#1C in STARTUP mode
3.0	OB82_OB_NUMBR	BYTE	OB number (82)
4.0	OB82_RESERVED_1	BYTE	Reserved
5.0	OB82_IO_FLAG	BYTE	Input module: B#16#54
6.0	OB82_MDL_ADDR	INT	Logical base address of the module in which the error has occurred
8.0	OB82_MDL_DEFECT	BOOL	Module fault
8.1	OB82_INT_FAULT	BOOL	Internal error
8.2	OB82_EXT_FAULT	BOOL	External error
8.3	OB82_PNT_INFO	BOOL	Channel error
8.4	OB82_EXT_VOLTAGE	BOOL	Missing external auxiliary supply
...	Not used		
10.3	OB82_WTCH_DOG_FLT	BOOL	Time monitoring has responded
...	Not used		
12.0	OB82_DATE_TIME	DATE_AND_TIME	Date and time when the OB was called

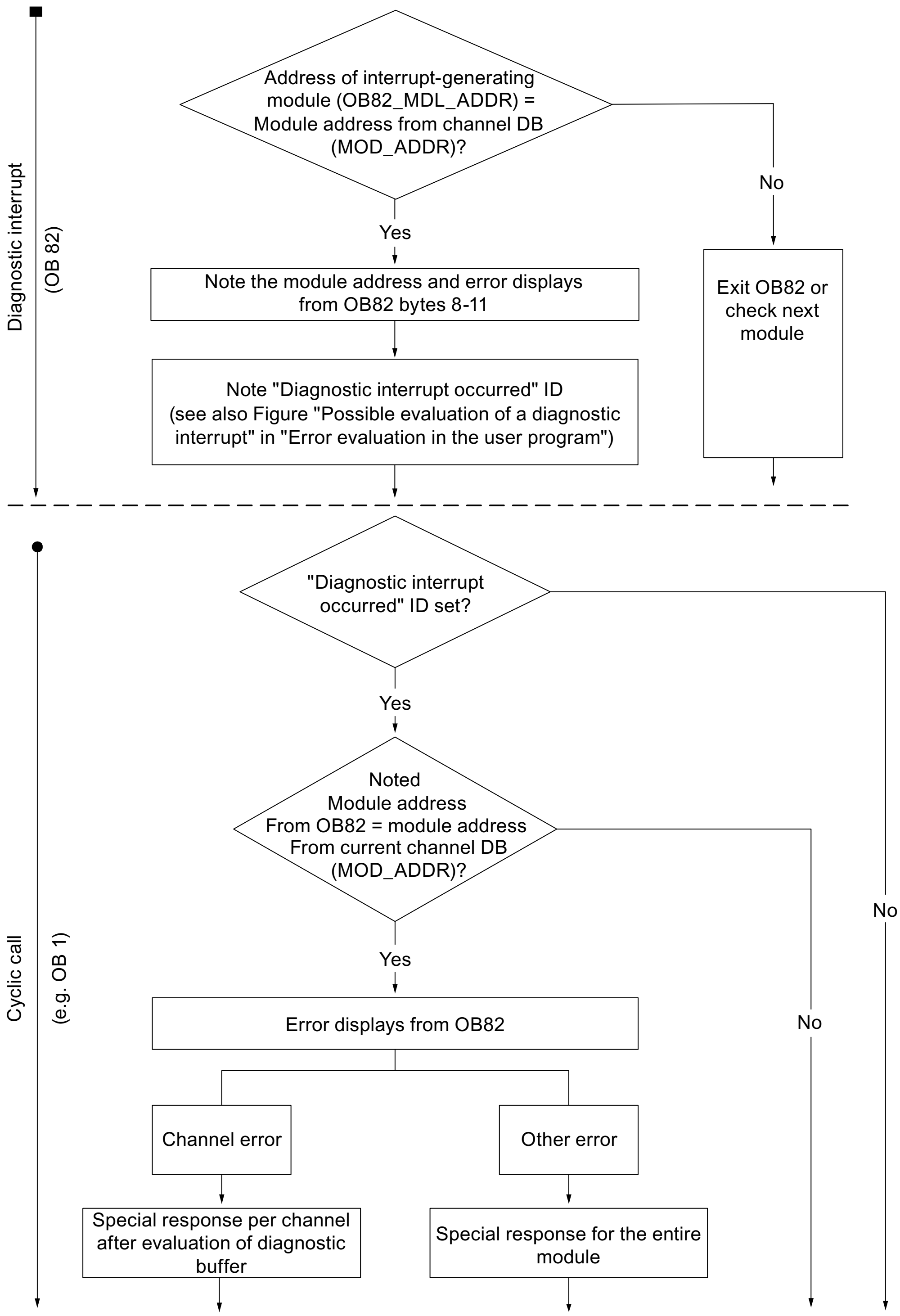


Figure 11-9 Possible evaluation of a diagnostic interrupt

12 Examples

12.1 Introduction

Example project

The FM 351 configuration package you installed contains example projects showing you several typical applications based on a number of selected functions.

The English example project is located in the following folder:

...\STEP7\EXAMPLES\zEn18_01

It contains several S7 programs of various complexities and objectives.

12.2 Requirements

Conditions for executing the examples

The following requirements must be met:

- A completely wired S7 station, consisting of a power supply module, a CPU, and an FM 351 module, version V3 or higher. The behavior of older versions may deviate from that described here.
- STEP 7 as well as the configuration package for the FM 351 are correctly installed on your PC or programming device. The description of handling is based on STEP 7 V5.0. Deviations may occur in the case of other versions.
- The programming device is connected to the CPU.

You can use the examples to operate an FM 351 or an FM 451.

12.3 Prepare examples

Procedure

In order to work through the examples online, you must prepare as follows:

1. Use SIMATIC Manager to open the example project **zEn18_01_FMx51___Prog** in the **\STEP7\EXAMPLES** folder (use the detailed representation to display the symbolic name) and copy it under a suitable name in your project directory (**File > Save as**).
2. Insert in your project a station that complies with your hardware set-up.
3. Complete the hardware configuration with HW Config and save the configuration.
4. Select an example program and copy the program to the offline CPU.
5. Assign the FM 351 parameters using "Getting Started". Use **File > Export** to export the parameters that are adapted to your system to the parameter DBs of all the examples apart from the the "Getting Started" example.
6. Enter the module address in the associated channel DB and, if necessary, also in the corresponding diagnostic DB in the "MOD_ADDR" parameter (refer to the section entitled Basics of programming a positioning module (Page 36)).
7. Download the hardware configuration to your CPU.
8. Download the blocks to your CPU.
9. If you would like to try the next example, go to step 4.

12.4 Example codes

Examples in STL

The examples are written in STL. You can view them directly via the LAD/FBD/STL editor.

Select the view with "symbolic display", "icon selection" and "comment". If you have sufficient space on the screen, you can also view the "icon information".

12.5 Testing an example

Test sequence

Once you have made all the entries necessary for the respective example, download the entire block container to the CPU.

The example programs include variable tables (VATs) you can use to view and change data blocks online, i.e., in CPU RUN mode. Select the views "icon" and "icon comment" from the variable table. Open a variable table, connect it with the configured CPU and cyclically monitor the variables. This way the displayed variables are constantly updated. By transmitting the control values you can change the values in the online data blocks.

All examples assume that you have entered and saved the machine data with the parameter assignment screens. This allows you to work through the examples sequentially.

If "continuous reading" by the FM 351 (e.g., of position values) is programmed, the CPU S7-300 can encounter restrictions when updating the parameter assignment screens.

12.6 Continuing to use an example

Procedure

You can continue using the code of examples directly as a user program.

The code of examples is not optimized and also is not designed for all eventualities. Error evaluations have not been extensively programmed in the example programs so as not to make the programs too comprehensive.

The "AllFunctions" example program can be used as a copy template in which, by means of modifying, you can compile the functions that can then be used as a basic template for your user program.

The examples are prepared for channel 1 ("severalchannels" for channel 1 and 2). Use the LAD/FBD/STL editor to change the channel number.

12.7 Example program 1 "FirstSteps"

Objective

With this example you can commission your positioning module whose parameters you have assigned with the aid of "Getting Started".

This example expands the program in the "Linking to the User Program" chapter of the "Getting Started" by an error evaluation.

Requirements

You have assigned the positioning module parameters as described in "Getting Started".

In the channel DB, the address of your module is entered correctly in the MOD_ADDR parameter, and the channel number is correctly entered in the CH_NO parameter.

Startup

In the startup OB (OB 100) you call the FC ABS_INIT, which resets all the control and checkback signals as well as the job management in the channel DB.

Cyclic operation

Open the variable table (VAT_CTRL_1), create the connection to the configured CPU and then monitor the variables. Transfer the prepared control values. Activate the "CHAN_1".DRV_EN: the drive is now enabled ("CHAN_1".ST_ENBLD=1). If the drive is not enabled, please check your enable inputs.

CAUTION

Start the drive by means of the two next steps.

You can stop the drive again by using one of the following measures:

- Reset the control value for the direction to 0 and activate
- Reset the control value for the drive enable to 0 and activate
- Bring the CPU to STOP

Set DIR_P=1 in order to move in the plus direction with selected "Jog" mode. If you set DIR_P=0, the drive is shut down properly.

Error evaluation

Create a data error by setting in the VAT_CTRL_1 the reference point coordinate "CHAN_1".REFPF outside of the operating range or the end of rotary axis. Then activate the job "set reference point" with "CHAN_1".REFPT_EN=1. The CPU goes to STOP. In an example, this is the simplest method of indicating an error. You can of course program a different error evaluation.

Open the hardware configuration and double-click the FM 351 or FM 451. This opens the parameter assignment software. Display the cause of error via the **Debug > Error evaluation** screen.

The status values in the VAT_CTRL_1 still show the status before the STOP of the CPU. Update the status values in order to view the done bits and error bits of the jobs.

Proceed as follows to remedy the error:

1. Enter a permissible value in the control value.
2. Switch the CPU to STOP.
3. Switch the CPU to RUN mode.
4. Enable the control values. If you have already enabled the control values before the CPU is restarted, the initialization will reset them again in the OB 100 and they will therefore be ineffective.

12.8 Example program 2 "Commissioning"

Aim

In this example you commission the positioning module without using the parameter assignment screens. You control and monitor by means of variable tables (VAT).

Requirements

You have assigned the positioning module parameters as described in "Getting Started".

In the channel DB, the address of your module is entered correctly in the MOD_ADDR parameter, and the channel number is correctly entered in the CH_NO parameter.

In the diagnostic DB, the address of your module is entered correctly in the MOD_ADDR parameter.

The supplied channel DB already includes the DB number 30 of the parameter DB for the machine data in the PARADBNO parameter.

The machine data of your system are stored in the PARADB_1 data block.

Start-up

In the start-up OB (OB 100) call the FC ABS_INIT in order to initialize the channel DB. Then set the trigger bits for all jobs that you require after start-up of the module.

Cyclic operation

Open the variable table (VAT_CTRL_1), create the connection to the configured CPU and then monitor the variables.

Transfer the prepared control values. "Jog" mode is set and the necessary enables are set. The drive turns by means of DIR_P=1. The actual value must change. In order to stop the drive, set STOP to "1" and transfer the control values.

Activate and transfer the control value "CHAN_1".REFPT_EN (set reference point). The checkback signal "CHAN_1".SYNC =1 means: the channel is synchronized.

In VAT_DIAG you can see the most important entries of the diagnostic buffer of the module. The meaning of the error classes and error numbers can be found in the manual in the appendix section entitled "Error classes (Page 189)".

Error evaluation

Try to create more errors:

- Specify a reference point coordinate that is greater than the operating range or the end of rotary axis.
- Switch off the external auxiliary voltage.
- Delete the PARADB_1 on the online CPU and try to write the machine data. In the example, the error evaluation is programmed in such a way that the CPU goes to STOP. When you update VAT_CTRL_1 again, the error code for this error is displayed in the "CHAN_1".JOB_ERR. The meaning of the error codes can be found in the manual in the appendix section entitled "List of JOB_ERR messages (Page 188)".

12.9 Example program 3 "AllFunctions"

Objective

This example uses all the functions of the FM 351 / 451:

- Operating modes
- Function switch
- Write jobs
- Read jobs

You can use the example programs as copy templates. By means of modification, compile the functions that will form the basic template for your user program. The data that you must adapt to your application are identified with ***. Some functions are only available with the FM 451.

Responses to external events and the error evaluation are system-specific and are therefore not included in this example.

Requirements

You have assigned the positioning module parameters as described in "Getting Started".

In the channel DB, the address of your module is entered correctly in the MOD_ADDR parameter, and the channel number is correctly entered in the CH_NO parameter.

The supplied channel DB already includes the DB number 30 of the parameter DB for the machine data in the PARADBNO parameter.

The machine data of your system are stored in the PARADB_1 data block.

Startup

In the startup OB (OB 100) call the FC ABS_INIT in order to initialize the channel DB. Then set the trigger bits for all jobs that you require after startup of the module.

Operation

The CPU is in STOP. Open the variable table USER_VAT and enter the necessary job number in the control values for your user program. The job numbers are explained in the code of the example.

In doing so, the correct combination of user data "USER_DB".CTRL_SIG, "USER_DB".FUNC_SW, "USER_DB".WR_JOBS, "USER_DB".RD_JOBS and "USER_DB".RETVAL_CTRL is necessary.

For more information, refer to the section entitled "Operating modes and jobs (Page 91)".

Create the connection to the configured CPU and transfer and activate the control values.

Start the CPU (STOP > RUN). Monitor the checkback signals and actual values.

You can repeat the processing of the step chain by means of a new STOP RUN transition of the CPU. Obviously, this action is not suitable for continuous operation. In the example we use this action so that the module is always reinitialized.

12.10 Example program 4 "OneChannel"

Objective

In this example you control a drive by means of the user program. The user program commissions the module after a CPU restart. Next, it executes a step sequence that is triggered by certain events.

Using the variable tables, you define events, monitor the reactions of the module, and evaluate the diagnostic buffer.

In this slightly more complex example, you can get to know the following block possibilities:

- Issuing several jobs simultaneously
- Mixing write and read jobs
- Reading with a continuous job, without waiting for the end of the job
- Evaluation of the checkback signals of the block
- Evaluation of the checkback signals for an individual job
- Resetting of done bits and error bits for individual jobs or all jobs
- Central call of ABS_CTRL at the end of the user program

Requirements

You have assigned the positioning module parameters as described in "Getting Started".

In the channel DB, the address of your module is entered correctly in the MOD_ADDR parameter, and the channel number is correctly entered in the CH_NO parameter.

The supplied channel DB already includes the DB number 30 of the parameter DB for the machine data in the PARADBNO parameter.

The machine data of your system are stored in the PARADB_1 data block.

Startup

In the startup OB (OB 100) you set the startup identifier (step 0) for the user program in the associated instance DB (USER_DB).

Operation

The CPU is in STOP. Open the variable table USER_VAT, adapt the increment ("USER_DB".TRG_INC_1, "USER_DB".TRG_INC_2), the changeover difference ("USER_DB".CHGDIF) and the switch-off difference ("USER_DB".CUTDIF) to your system and transfer the control values.

Start the CPU (STOP > RUN). Observe the step number of the step chain ("USER_DB".STEPNO), the checkback signals and the actual values. After the initialization a "relative incremental approach" is carried out. The drive travels in a negative direction to its first position ("USER_DB".TRG_INC_1).

Then the program waits in step 6 for an external trigger ("USER_DB".START_INC_2), in order to commence the next incremental approach in direction plus. When the position is reached, the step chain is at its end value (-2). The incremental approach with increment number 255 allows the transfer of the changeover difference and switch-off difference. With that you can test the behavior of your target approach.

You can repeat the processing of the step chain by means of a new start (STOP > RUN) of the CPU. Obviously, this action is not suitable for continuous operation. In the example we use this action so that the module is always reinitialized.

Error evaluation

In the case of an error in the processing, the step chain is brought to a halt. Step number -1 will be entered.

Try to create errors that will be stored by the central error evaluation as group errors in the "USER_DB".ERR bit.

- In the USER_VAT, enable the prepared control value for the increment number 1 ("USER_DB".TRG_INC_1), which is greater than the software limit switch.

  The step chain is brought to a stop and -1 is shown as the step number. Check the errors by means of the error evaluation screen.
- In the USER_VAT enable further control values one after the other for the increment number 1 ("USER_DB".TRG_INC_1), increment number 255 ("USER_DB".TRG_INC_2), and the change-over difference ("USER_DB".CHGDIF) and switch-off difference ("USER_DB".CUTDIF). For the error inspection proceed as you would with increment 1.

User program FB 1 (USER_PROG)

The user program uses the data in the module-specific data blocks (USER_DB) in the form <block name>.<symbolic identifier>. This way, the user program can operate a channel precisely.

In this program mode you can access the data in the data block by means of the symbolic identifiers. The indirect addressing for several channels can be found in the example program 6 "SeveralChannels".

The user program executes a step sequence as follows:

Step 0: The positioning module is initialized. The jobs with the associated data to be executed at a restart of the module are set.

Step 1: The program waits for the jobs set in Step 0 to be executed.

Step 2: The assigned value of the increment "USER_DB".TRG_INC_1 is entered in the increment table. Then the increment table is written in the module. The control signals for the first incremental approach will be issued simultaneously. The FB ABS_CTRL ensures the correct sequence of execution from step 2.

Step 3: The program waits for the set write job to be executed.

Step 4: The program waits for the checkback signal "PR" and the updated position values form the first incremental approach.

Step 5: The assigned values for the second incremental approach, changeover difference and switch-off difference are entered in the channel DB. Then, with "USER_DB".START_INC_2 the second incremental approach is started with increment number 255.

Step 6: The program waits for the set jobs to be executed.

Step 7: The program waits for the checkback signal "PR" and the updated position values form the second incremental approach.

12.11 Example program 5 "DiagnosticsAndInterrupts"

Objective

This example includes a user program with the same formulation as in the example program 4 "OneChannel". In addition, we show you how you evaluate a diagnostic interrupt for specific modules and then make this into a general module error in the user program.

Requirements

You have assigned the positioning module parameters as described in "Getting Started".

In the channel DB, the address of your module is entered correctly in the MOD_ADDR parameter, and the channel number is correctly entered in the CH_NO parameter.

In the diagnostic DB, the address of your module is entered correctly in the MOD_ADDR parameter.

The supplied channel DB already includes in the PARADBNO parameter the DB number (30) of the parameter DB for the machine data.

The machine data of your system are stored in the PARADB_1 data block.

In the HW Config, enable the diagnostic interrupt for this module via **Edit > Object properties > Basic parameters > Select interrupt > Diagnostics**. Compile the hardware configuration, and then download it to the CPU.

Startup

In the startup OB (OB 100) you set the startup identifier (step 0) for the user program in the instance DB.

Operation

As in example program 4 "OneChannel".

Error evaluation

In the case of an error in the processing, the step chain is brought to a halt. Step number -1 will be entered. In the USER_VAT you will find the latest entry of the diagnostic buffer. You can determine the cause of error via the error class and error number (refer to the appendix section entitled "Error classes (Page 189)").

Try to create errors that will be stored by the central error evaluation as group errors in the "USER_DB".ERR bit.

- In the USER_VAT, enable the prepared control value for the increment number 1 ("USER_DB".TRG_INC_1), which is greater than the software limit switch.

  The step chain is brought to a stop and -1 is shown as the step number. Check the errors via the error evaluation screen or the diagnostic data in the USER_VAT.
- In the USER_VAT enable further control values one after the other for the increment number 1 ("USER_DB".TRG_INC_1), increment number 255 ("USER_DB".TRG_INC_2), and the change-over difference ("USER_DB".CHGDIF) and switch-off difference ("USER_DB".CUTDIF). For the error inspection proceed as you would with increment 1.
- Create the diagnostic interrupts by disconnecting the auxiliary voltage of the module or removing the front connector. The diagnostics error "USER_DB".ERR_MOD and group error "USER_DB".ERR will be 1 and the step number will be -1.

User program (FB PROG)

The formulation is as in example program 4 "OneChannel".

In this example, no special measures are taken for placing after the error remedy.

Diagnostic interrupt (OB 82)

In the diagnostic interrupt, depending on the address of the interrupt-triggering module (OB82_MDL_ADDR) the error identifier in the associate instance DB (USER_DB) of the user program is entered. A response takes place in the cyclic user program.

12.12 Example program 6 "SeveralChannels"

Objective

This example contains the same user program as example program 4 "OneChannel", but it operates 2 channels of the module. The user program uses a separate instance of ABS_CTRL and ABS_DIAG for each channel, a multiple instance is not possible. The user program expects a channel number as input parameter. The DB numbers for channel and diagnostic DBs associated with this channel are stored as constants in the program and can by adjusted by you.

Requirements

You have assigned the channel 1 parameters as described in "Getting Started". Use **Edit > Copy channel** to copy channel 1 to channel 2. If necessary, adapt the parameters of channel 2. Save the hardware configuration and download it to the CPU.

In the channel DB, the address of your module is entered correctly in the MOD_ADDR parameter, and the channel number is correctly entered in the CH_NO parameter.

In the diagnostic DB, the address of your module is entered correctly in the MOD_ADDR parameter.

The supplied channel DBs already include the DB number 30 or 31 of the parameter DB for the machine data in the PARADBNO parameter.

In the data blocks PARADB_1 and PARADB_2 the machine data is stored for in each case one channel of your system.

In HW Config, enable the diagnostic interrupt for this module with **Edit > Object Properties > Basic Parameters > Interrupt Selection > Diagnostics**. Compile the hardware configuration, and then download it to the CPU.

A variable table is set up for each channel.

Startup

In the startup OB (OB 100) you set the startup identifier (step 0) for the user program in both instance DBs (USER_DB_1, USER_DB_2).

Operation

The CPU is in STOP. Open the USER_VAT_1 and USER_VAT_2 and transfer their control values.

Start the CPU (STOP > RUN). You can see how the actual positions of both channels change.

Error evaluation

As in example program 5 "DiagnosticsAndInterrupts", however separately for each channel.

User program (FB PROG)

Objective and order of events of the user program are as in example program 5 "DiagnosticsAndInterrupts" and in example program 4 "OneChannel".

The user program is designed for operation with several channels, as it indirectly accesses the module-specific data blocks (channel DBs, diagnostic DB and parameters DBs). The channel number specified during call up is used in the user program to select the instance DBs. With this type of programming, you cannot use symbolic names for the data in the data blocks because of the "Open global data block" instruction used in the user program.

Diagnostic interrupt (OB 82)

In the diagnostic interrupt, the error identifier is entered in the associated instance DB of the user program according to the address of the interrupt-triggering channel (OB82_MDL_ADDR).

Technical specifications A

A.1 General Technical Specifications

The following technical data are available in the Operating Instructions SIMATIC S7-300, CPU 31xC and CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499).

- Electromagnetic compatibility
- Transportation and storage conditions
- Mechanical and climatic environmental conditions
- Specifications for insulation tests, protection class, and degree of protection
- Certifications and standards

Observe the installation guidelines

SIMATIC products fulfill the requirements provided during installation and operation, the manual´s installation guidelines are followed.

A.2 Technical Specifications of the FM 351

Technical specifications

Technical specifications	
Dimensions and weight	
Dimensions W x H x D (mm)	80 × 125 × 120
Weight	Approx. 535 g
Current, voltage and output	
Current consumption from backplane bus	Max. 200 mA
Power loss	Typ. 7.9 W
Auxiliary voltage for the encoder supply	Auxiliary voltage: 24 V DC (X1, terminal 1) (permissible range: 20.4 to 28.8 V)
Encoder supply	• Horizontal installation of S7-300, 20 °C: – 5.2 V/500 mA (for both channels together) – 24 V/800 mA (for both channels together) • Horizontal installation of S7-300, 60 °C: – 5.2 V/500 mA (for both channels together) – 24 V/600 mA (for both channels together) • Vertical set-up S7-300, 40 °C: – 5.2 V/500 mA (for both channels together) – 24 V/600 mA (for both channels together) • Current consumption from 1L+ (without load): max. 100 mA (X1, terminal 1) • Encoder supply 24 V, uncontrolled – L+ -2 V (X2/X3, terminal 5) – Short-circuit protection: yes, thermal • Encoder supply 5.2 V (X2/X3, terminal 6) short-circuit protection: yes, electronic • Permissible potential difference between input (ground) and central ground connection of the CPU: – 60 V AC – 75 V DC
Auxiliary voltage for the load power supply	Auxiliary voltage: 24 V DC (X1, terminal 19) (permissible range: 20.4 to 28.8 V)

Technical specifications	
Supply of the digital inputs and outputs	• Current consumption from 2L+ (without load): max. 50 mA (X1, terminal 19) • Permissible potential difference between input of ground connection 1M (X1, terminal 2) and the central grounding point (shield): 60 V AC; 75 V DC – Insulation tested with 500 V DC • Permissible potential difference between input of ground connection 2M (X1, terminal 20) and the central grounding point (shield): 60 V AC; 75 V DC – Insulation tested with 500 V DC
Load voltage incorrect polarity protection	Yes
Encoder inputs	
Position measuring	• Incremental • Absolute
Signal voltages	• Symmetrical inputs: 5 V in accordance with RS 422 • Asymmetrical inputs: 24 V/ typically. 4 mA
Input frequency and cable length for symmetrical incremental encoder with 5 V supply	Max. 400 kHz with 32 m cable length, shielded
Input frequency and cable length for symmetrical incremental encoder with 24 V supply	Max. 400 kHz with 100 m cable length, shielded
Input frequency and cable length for asymmetrical incremental encoder with 24 V supply	• Max. 50 kHz with 25 m cable length, shielded • Max. 25 kHz with 100 m cable length, shielded
Data baud rate and cable length for absolute encoders	• Max. 188 kHz with 200 m cable length, shielded • Max. 375 kHz with 100 m cable length, shielded • Max. 750 kHz with 40 m cable length, shielded • Max. 1.5 MHz with 12 m cable length, shielded
Listen-in mode for absolute encoders	No
Input signals	• Incremental: 2 pulse trains, 90° shift, 1 zero pulse • Absolute: Absolute value

Technical specifications	
Digital inputs	
Number of digital inputs	8
Number of simultaneously controllable digital inputs	8
Galvanic isolation	Yes, optocoupler
Status display	Yes, green LED for each digital input
Input voltage	• 0-signal: -3 ... 5 V • 1-signal: 11 ... 30 V
Input current	• 0-signal: ≤ 2 mA (quiescent current) • 1-signal: 6 mA
Input delay (1I0, 1I1, 1I2 and 2I0, 2I1, 2I2)	• 0- → 1-signal: Typ. 3 ms • 1-→ 0-signal: Typ. 3 ms
Input delay (1I3 and 2I3)	• 0- → 1-signal: Typ. 300 μs • 1- → 0-signal: Typ. 300 μs
Connection of a 2-wire BERO	Supported
Cable length, unshielded (1I0, 1I1, 1I2 and 2I0, 2I1, 2I2)	100 m
Cable length, shielded (1I0, 1I1, 1I2 and 2I0, 2I1, 2I2)	Max. 600 m
Cable length shielded (1I3 and 2I3)	Max. 100 m
Insulation testing	VDE 0160
Digital outputs	
Number of outputs	8
Galvanic isolation	Yes, optocoupler
Status display	Yes, green LED for each digital output
Output voltage	• 0-signal: 0.5 mA • 1-signal: 0.5 A (permissible range: 5 ... 600 mA) • Lamp load: 5 W
Output delay when output voltage = 0.5 A	• 0- → 1-signal: max. 300 μs • 1- → 0-signal: max. 300 μs
Signal level with 1-signal	L+ -0.8 V
Control of a digital input	Yes
Control of a count input	No, on account of 50 μs missing pulse
Short-circuit protection	Yes, thermally clocking switching threshold 1 A
Limiting of inductive switch-off voltage	Typ. L+: -48 V
Switching frequency	• Resistive load: Max. 100 Hz • Inductive load: Max. 0.5 Hz
Total current of the digital outputs with the horizontal installation of S7-300	Demand factor 75%: at 20 °C and 60 °C: 3 A
Total current of the digital outputs with the vertical installation of S7-300	Demand factor 75%: at 40 °C: 3 A

Technical specifications	
Unshielded cable length	Max. 100 m
Shielded cable length	Max. 600 m
Insulation testing	VDE 0160

Note

If the 24 V supply voltages are connected via a mechanical contact, the FM 351 issues a pulse to the outputs. Within the permissible output current range the pulse can amount to 50 μs. You must observe this when you use the FM 351 in connection with fast counters.

Connection diagrams B

B.1 Overview

Overview

The following table describes encoders that you can connect to the positioning module. The connection diagrams for these encoders are described in this appendix:

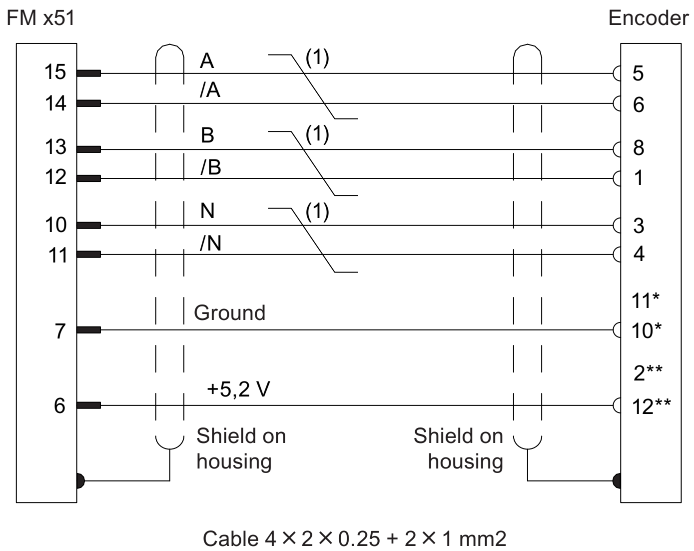
In the chapter	... you will find the pin assignment for	Connecting cable	Comment
Connection diagram for incremental encoder Siemens 6FX2001-2 (Up=5 V; RS 422)	Incremental encoder Siemens 6FX2001-2	4 x 2 x 0.25 + 2 x 1 mm^2	Incremental encoders: U_p=5 V, RS 422
Connection diagram for incremental encoder Siemens 6FX2001-2 (Up=24 V; RS 422)	Incremental encoder Siemens 6FX2001-2	4 x 2 x 0.5 mm^2	Incremental encoders: U_p=24 V, RS 422
Connection diagram for incremental encoder Siemens 6FX2001-4 (Up=24 V; HTL)	Incremental encoder Siemens 6FX2001-4	4 x 2 x 0.5 mm^2	Incremental encoders: U_p=24 V, HTL
Connection diagram for absolute encoder Siemens 6FX2001-5 (Up=24 V; SSI)	Absolute encoder Siemens 6FX2001-5	4 x 2 x 0.5 mm^2	Absolute encoder: U_p=24 V, SSI

B.2 Connection diagram for incremental encoder Siemens 6FX 2001-2 (Up=5V; RS 422)

Connection diagram

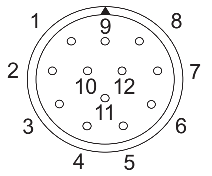
The following figure shows the connection diagram for the incremental encoder Siemens 6FX 2001-2 (U_p=5 V: RS 422):

Connection diagram

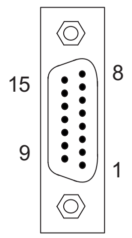


(1) Twisted-pair cables

* Pins 10 and 11 are internally bridged.
** Pins 2 and 12 are internally bridged.



12-pin circular connector Socket
Siemens 6FX2003-0SU12
Connection side (solder side)



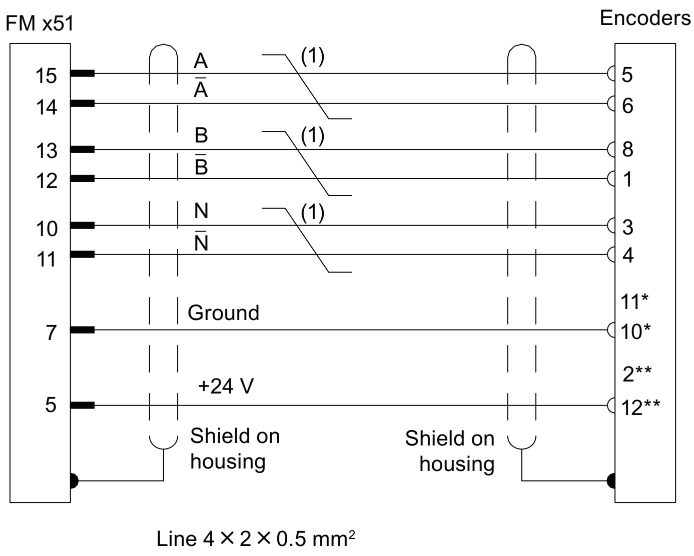
15-pin sub D connector, connection side (solder side) metallized housing with screw interlock 6FC9341-1HC

B.3 Connection diagram for incremental encoder Siemens 6FX 2001-2 (Up=24V; RS 422)

Connection diagram

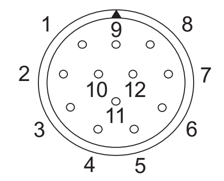
The following figure shows the connection diagram for an incremental encoder Siemens 6FX2001-2 (U_p=24 V; RS 422):

Connection diagram

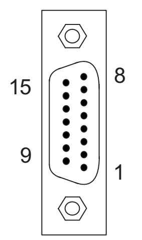


(1) Twisted-pair cables

* Pins 10 and 11 are internally bridged.
** Pins 2 and 12 are internally bridged.



Circular connector 12-pin socket
Siemens 6FX2003-0SU12
Connection side (solder side)



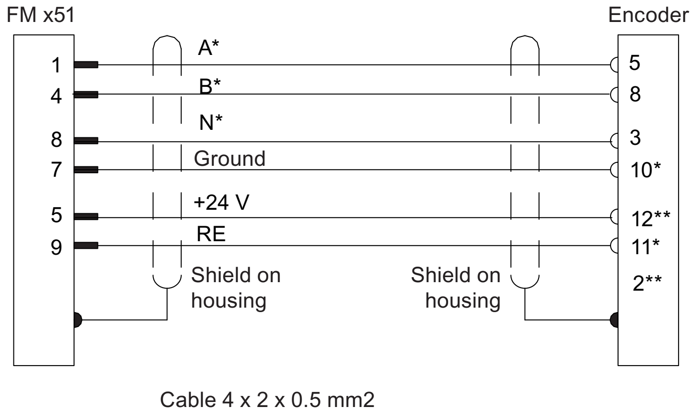
15-pin sub D connector, connection side (solder side) metallized housing with screw interlock 6FC9341-1HC

B.4 Wiring diagram of the incremental encoder Siemens 6FX 2001-4 (Up = 24 V; HTL)

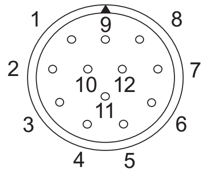
Connection diagram

The following figure shows the connection diagram for an incremental encoder Siemens 6FX2001-4 (Up=24 V; HTL):

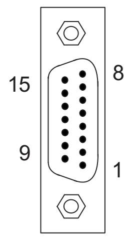
Connection diagram



* Pins 10 and 11 are internally bridged.
** Pins 2 and 12 are internally bridged.



Circular connector 12-pin socket
Siemens 6FX2003-0SU12
Connection side (solder side)



15-pin sub D connector, connection side (solder side) metallized housing with screw interlock 6FC9341-1HC

Note

To connect a non-SIEMENS incremental encoder in a push-pull configuration (current sourcing/sinking), observe the following:

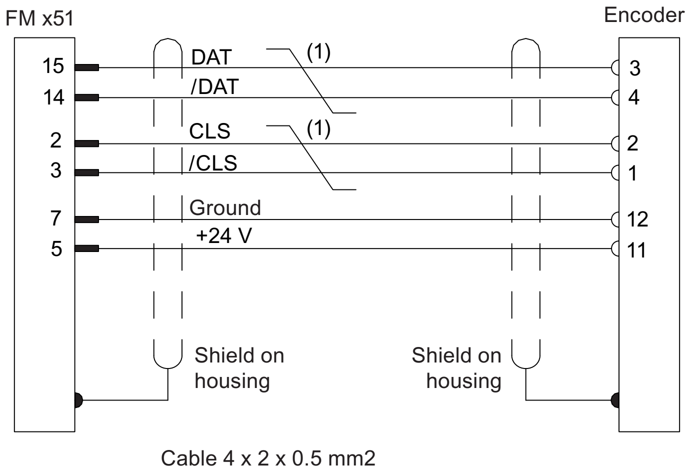
- Current sourcing: Connect RE (9) to ground (7).
- Current sinking: Connect RE (9) to +24 V (5).

B.5 Connection diagram for absolute encoder Siemens 6FX 2001-5 (Up=24V; SSI)

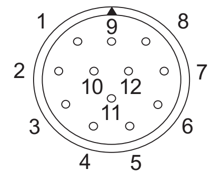
Connection diagram

The following figure shows the connection diagram for an absolute encoder Siemens 6FX2001-5 (Up=24 V; SSI):

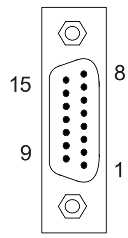
Connection diagram



(1) Twisted-pair cables



Circular connector 12-pin socket
Siemens 6FX2003-0SU12
Connection side (solder side)



15-pin sub D connector, connection side (solder side) metallized housing with screw interlock 6FC9341-1HC

Data blocks and error lists C

C.1 Content of the channel DB

Data of channel DB

Note

You must not change any data that is not listed in this table.

Table C- 1 Content of the channel DB

Address	Name	Type	Initial value	Comment
Addresses				
0.0	MOD_ADDR	INT	0	Module address
2.0	CH_NO	INT	1	Channel number
10.0	PARADBNO	INT	-1	Number of the parameter DB
Control signals				
14.3	OT_ERR_A	BOOL	FALSE	1 = acknowledge operator error
15.0	START	BOOL	FALSE	1 = start positioning
15.1	STOP	BOOL	FALSE	1 = stop active traversing
15.2	DIR_M	BOOL	FALSE	1 = minus direction
15.3	DIR_P	BOOL	FALSE	1 = plus direction
15.6	SPEED252	BOOL	FALSE	Not used
15.7	DRV_EN	BOOL	FALSE	1 = switch on drive enable
16.0	MODE_IN	BYTE	B#16#0	Required operating mode 0 = no operating mode 1 = Jog 3 = reference point approach 4 = relative incremental approach 5 = absolute incremental approach
17.0	MODE_TYPE	BYTE	B#16#0	• Start velocity for jog mode 0 = creep speed 1 = rapid traverse • Increment number for incremental approach mode

Address	Name	Type	Initial value	Comment
Checkback signals				
22.2	DIAG	BOOL	FALSE	1 = Diagnostic buffer changed
22.3	OT_ERR	BOOL	FALSE	1 = operator error occurred
22.4	DATA_ERR	BOOL	FALSE	1 = data error
22.7	PARA	BOOL	FALSE	1 = axis parameters assigned
23.0	ST_ENBLD	BOOL	FALSE	1 = start enabled
23.1	WORKING	BOOL	FALSE	1 = Positioning running (processing running)
23.2	WAIT_EI	BOOL	FALSE	1 = axis waiting for external enable
23.4	SPEED_OUT	BOOL	FALSE	0 = creep speed 1 = rapid traverse
23.5	ZSPEED	BOOL	FALSE	1 = axis is located in the standstill range
23.6	CUTOFF	BOOL	FALSE	1 = axis is located in the switch-off range
23.7	CHGOVER	BOOL	FALSE	1 = axis is located in the changeover range
24.0	MODE_OUT	BYTE	B#16#0	Active operating mode
25.0	SYNC	BOOL	FALSE	1 = axis is synchronized
25.1	MSR_DONE	BOOL	FALSE	Not used
25.2	GO_M	BOOL	FALSE	1 = axis moves in minus direction
25.3	GO_P	BOOL	FALSE	1 = axis moves in plus direction
25.5	FVAL_DONE	BOOL	FALSE	Not used
25.7	POS_RCD	BOOL	FALSE	1 = Position reached
26.0	ACT_POS	DINT	L#0	Current actual value (current position of axis)
Function switch				
34.0	PLOOP_ON	BOOL	FALSE	1 = loop approach in plus direction
34.1	MLOOP_ON	BOOL	FALSE	1 = loop approach in minus direction
34.2	EI_OFF	BOOL	FALSE	1 = do not evaluate enable input
34.3	EDGE_ON	BOOL	FALSE	Not used
34.4	MSR_ON	BOOL	FALSE	Not used
Trigger bits for write jobs				
35.0	MDWR_EN	BOOL	FALSE	1 = write machine data
35.1	MD_EN	BOOL	FALSE	1 = enable machine data
35.2	DELDIST_EN	BOOL	FALSE	1 = delete distance-to-go
35.3	AVALREM_EN	BOOL	FALSE	1 = cancel set actual value
35.4	TRGL1WR_EN	BOOL	FALSE	1 = write increment table 1 (increment number 1 ... 50)
35.5	TRGL2WR_EN	BOOL	FALSE	1 = write increment table 2 (increment number 51 ... 100)
35.6	REFPT_EN	BOOL	FALSE	1 = set reference point
35.7	AVAL_EN	BOOL	FALSE	1 = set actual value

Address	Name	Type	Initial value	Comment
36.0	FVAL_EN	BOOL	FALSE	Not used
36.1	ZOFF_EN	BOOL	FALSE	Not used
36.2	TRG252_254_EN	BOOL	FALSE	1 = write increment for increment number 254
36.3	TRG255_EN	BOOL	FALSE	1 = write increment for increment number 255
36.4	DELDIAG_EN	BOOL	FALSE	1 = clear diagnostic buffer
Trigger bits for read jobs				
36.5	MDRD_EN	BOOL	FALSE	1 = Read machine data
36.6	TRGL1RD_EN	BOOL	FALSE	1 = read increment table 1 (increment number 1 ... 50)
36.7	TRGL2RD_EN	BOOL	FALSE	1 = read increment table 2 (increment number 51 ... 100)
37.0	MSRRD_EN	BOOL	FALSE	Not used
37.1	ACTSPD_EN	BOOL	FALSE	1 = Read actual speed, distance-to-go, and current increment
37.2	ENCVAL_EN	BOOL	FALSE	1 = read encoder values
Done bits for function switches				
38.0	PLOOP_D	BOOL	FALSE	1 = "loop approach in plus direction" job completed
38.1	MLOOP_D	BOOL	FALSE	1 = "loop approach in minus direction" job completed
38.2	EI_D	BOOL	FALSE	1 = "Do not evaluate enable input" job completed
38.3	EDGE_D	BOOL	FALSE	Not used
38.4	MSR_D	BOOL	FALSE	Not used
Done bits for write jobs				
39.0	MDWR_D	BOOL	FALSE	1 = "Write machine data" job completed
39.1	MD_D	BOOL	FALSE	1 = "Enable machine data" job completed
39.2	DELDIST_D	BOOL	FALSE	1 = "Delete distance-to-go" job completed
39.3	AVALREM_D	BOOL	FALSE	1 = "Cancel set actual value" job completed
39.4	TRGL1WR_D	BOOL	FALSE	1 = "Write increment table 1" job completed
39.5	TRGL2WR_D	BOOL	FALSE	1 = "Write increment table 2" job completed
39.6	REFPT_D	BOOL	FALSE	1 = "Set reference point" job completed
39.7	AVAL_D	BOOL	FALSE	1 = "Set actual value" job completed
40.0	FVAL_D	BOOL	FALSE	Not used
40.1	ZOFF_D	BOOL	FALSE	Not used
40.2	TRG252_254_D	BOOL	FALSE	1 = "Write increment for increment number 254" job completed
40.3	TRG255_D	BOOL	FALSE	1 = "Write increment for increment number 255" job completed
40.4	DELDIAG_D	BOOL	FALSE	1 = "Clear diagnostic buffer" job completed

Address	Name	Type	Initial value	Comment
Done bits for read jobs				
40.5	MDRD_D	BOOL	FALSE	1 = "Read machine data" job completed
40.6	TRGL1RD_D	BOOL	FALSE	1 = "Read increment table 1" job completed
40.7	TRGL2RD_D	BOOL	FALSE	1 = "Read increment table 2" job completed
41.0	MSRRD_D	BOOL	FALSE	Not used
41.1	ACTSPD_D	BOOL	FALSE	1 = "Read actual speed, distance-to-go, and current increment" job completed
41.2	ENCVAL_D	BOOL	FALSE	1 = "Read encoder values" job completed
Error bits for function switches				
42.0	PLOOP_ERR	BOOL	FALSE	1 = Error in "Loop approach in plus direction" job
42.1	MLOOP_ERR	BOOL	FALSE	1 = Error in "Loop approach in minus direction" job
42.2	EI_ERR	BOOL	FALSE	1 = Error in "Do not evaluate enable input" job
42.3	EDGE_ERR	BOOL	FALSE	Not used
42.4	MSR_ERR	BOOL	FALSE	Not used
Error bits for write jobs				
43.0	MDWR_ERR	BOOL	FALSE	1 = Error in "Write machine data" job
43.1	MD_ERR	BOOL	FALSE	1 = Error in "Activate machine data" job
43.2	DELDIST_ERR	BOOL	FALSE	1 = Error in "Delete distance-to-go" job
43.3	AVALREM_ERR	BOOL	FALSE	1 = Error in "Cancel set actual value" job
43.4	TRGL1WR_ERR	BOOL	FALSE	1 = Error in "Write increment table 1" job
43.5	TRGL2WR_ERR	BOOL	FALSE	1 = Error in "Write increment table 2" job
43.6	REFPT_ERR	BOOL	FALSE	1 = Error in "Set reference point" job
43.7	AVAL_ERR	BOOL	FALSE	1 = Error in "Set actual value" job
44.0	FVAL_ERR	BOOL	FALSE	Not used
44.1	ZOFF_ERR	BOOL	FALSE	Not used
44.2	TRG252_254_ERR	BOOL	FALSE	1 = Error in "Write increment for increment table 254" job
44.3	TRG255_ERR	BOOL	FALSE	1 = Error in "Write increment for increment table 255" job
44.4	DELDIAG_ERR	BOOL	FALSE	1 = Error in "Clear diagnostic buffer"
Error bits for read jobs				
44.5	MDRD_ERR	BOOL	FALSE	1 = Error in "read machine data" job
44.6	TRGL1RD_ERR	BOOL	FALSE	1 = Error in "Read increment table 1" job
44.7	TRGL2RD_ERR	BOOL	FALSE	1 = Error in "Read increment table 2" job
45.0	MSRRD_ERR	BOOL	FALSE	Not used
45.1	ACTSPD_ERR	BOOL	FALSE	1 = Error in "Read actual speed, distance-to-go, and current increment" job
45.2	ENCVAL_ERR	BOOL	FALSE	1 = Error in "Read current encoder values" job
Job management for the FB ABS_CTRL				
48.0	JOB_ERR	INT	0	Error number of the communication error
50.0	JOBBUSY	BOOL	FALSE	1 = at least one job is running
50.1	JOBRESET	BOOL	FALSE	1 = reset all error bits and done bits

Address	Name	Type	Initial value	Comment
Data element for "Zero offset" job (FM 451)				
80.0	ZOFF	DINT	L#0	Not used
Data element for "Set actual value" job				
84.0	AVAL	DINT	L#0	Coordinate for "Set actual value"
Data element for "Set actual value on-the-fly" (FM 451) job				
88.0	FVAL	DINT	L#0	Not used
Data element for "Set reference point" job				
92.0	REFPT	DINT	L#0	Coordinate for "Set reference point"
Data element for "Write increment for increment table 254" job				
96.0	TRG252_254	DINT	L#0	Increment for increment number 254
Data for "Write increment for increment number 255" job				
100.0	TRG255	DINT	L#0	Increment for increment number 255
104.0	CHGDIF255	DINT	L#0	Changeover difference for increment number 255
108.0	CUTDIF255	DINT	L#0	Switch-off difference for increment number 255
Data for "Read position data" job				
112.0	ACTSPD	DINT	L#0	Current velocity
116.0	DIST_TO_GO	DINT	L#0	Distance-to-go
120.0	ACT_TRG	DINT	L#0	Current increment
Data for "Read encoder data" job				
124.0	ENCVAL	DINT	L#0	Actual encoder value (internal representation)
128.0	ZEROVAL	DINT	L#0	Last zero mark value (internal representation)
132.0	ENC_ADJ	DINT	L#0	Absolute encoder adjustment
Data for "Length measurement/edge detection" job (FM 451)				
136.0	BEG_VAL	DINT	L#0	Not used
140.0	END_VAL	DINT	L#0	Not used
144.0	LEN_VAL	DINT	L#0	Not used

C.2 Content of the parameter DB

Parameter DB data

Note

You must not change any data that is not listed in this table.

Table C- 2 Content of the parameter DB

Address	Name	Type	Initial value	Comment
Machine data				
4.0	EDGEDIST	DINT	L#0	Not used
8.0	UNITS	DINT	L#1	System of units
12.0	AXIS_TYPE	DINT	L#0	0 = linear axis 1 = rotary axis
16.0	ENDROTAX	DINT	L#100000	End of rotary axis
20.0	ENC_TYPE	DINT	L#1	Encoder type, message length
24.0	DISP_REV	DINT	L#80000	Length per encoder revolution
28.0	b_28	DWORD	L#0	Parity check for SSI absolute encoder (this machine data element is only available for the FM 351 with order number 6ES7351-1AH02-0AE0)
32.0	INC_REV	DINT	L#500	Increments per encoder revolution
36.0	NO_REV	DINT	L#1	Number of encoder revolutions
40.0	BAUD RATE	DINT	L#0	Baud rate
44.0	REFPT	DINT	L#0	Reference point coordinate
48.0	ENC_ADJ	DINT	L#0	Absolute encoder adjustment
52.0	REFPT_TYPE	DINT	L#0	Type of reference point approach
59.0	CNT_DIR	BOOL	FALSE	Counting direction: 0 = normal 1 = inverted
63.0	MON_WIRE	BOOL	TRUE	1 = wire break monitoring
63.1	MON_FRAME	BOOL	TRUE	1 = message error monitoring
63.2	MON_PULSE	BOOL	TRUE	1 = glitch monitoring
64.0	SSW_STRT	DINT	L#-100000000	Start of software limit switch
68.0	SSW_END	DINT	L#100000000	End of software limit switch
76.0	TRG_RANGE	DINT	L#1000	Target range
80.0	MON_TIME	DINT	L#2000	Monitoring time [ms]
84.0	ZSPEED_R	DINT	L#1000	Standstill range
88.0	ZSPEED_L	DINT	L#30000	Upper limit of the standstill speed
92.0	CTRL_TYPE	DINT	L#1	Steer mode (1 - 4)

Address	Name	Type	Initial value	Comment
Machine data				
99.0	REFPT_SPD	BOOL	TRUE	Start speed in the case of a reference point approach: 0 = rapid traverse 1 = creep traverse
99.1	EI_TYPE	BOOL	FALSE	Not used
100.0	CHGDIF_P	DINT	L#5000	Plus changeover difference
104.0	CHGDIF_M	DINT	L#5000	Minus changeover difference
108.0	CUTDIF_P	DINT	L#2000	Plus switch-off difference
112.0	CUTDIF_M	DINT	L#2000	Minus switch-off difference
Increment table 1				
120.0	TRGL1.TRG[1]	DINT	L#0	Increment number 1
. . .	. . .	. . .	. . .	Increment table 1
316.0	TRGL1.TRG[50]	DINT	L#0	Increment number 50
Increment table 2				
320.0	TRGL2.TRG[51]	DINT	L#0	Increment number 51
. .	. .	. .	. .	Increment table 2
516.0	TRGL2.TRG[100]	DINT	L#0	Increment number 100

C.3 Data and structure of the diagnostic DB

Data and structure of the diagnostic DB

Note

You must not change any data that is not listed in this table.

Table C- 3 Structure of the diagnostic DB

Address	Name	Type	Initial value	Comment
0.0	MOD_ADDR	INT	0	Module address
256.0	JOB_ERR	INT	0	Error number of the communication error
258.0	JOBBUSY	BOOL	FALSE	1 = job active
258.1	DIAGRD_EN	BOOL	FALSE	1 = definitely read diagnostics buffer
260.0	DIAG_CNT	INT	0	Number of valid entries in the list
262.0	DIAG[1]	STRUCT		Diagnostics data - latest entry
272.0	DIAG[2]	STRUCT		Diagnostics data - second entry
282.0	DIAG[3]	STRUCT		Diagnostics data - third entry
292.0	DIAG[4]	STRUCT		Diagnostics data - fourth entry
302.0	DIAG[5]	STRUCT		Diagnostics data - fifth entry
312.0	DIAG[6]	STRUCT		Diagnostics data - sixth entry
322.0	DIAG[7]	STRUCT		Diagnostics data - seventh entry
332.0	DIAG[8]	STRUCT		Diagnostics data - eighth entry
342.0	DIAG[9]	STRUCT		Diagnostics data - ninth entry

The structure of the diagnostics entry DIAG[n] is developed as follows:

Table C- 4 Structure of the diagnostics entry

Address	Name	Type	Initial value	Comment
+0.0	STATE	BOOL	FALSE	0 = incoming event 1 = outgoing event
+0.1	INTF	BOOL	FALSE	1 = internal error
+0.2	EXTF	BOOL	FALSE	1 = external error
+2.0	FCL	INT	0	Error class: 1: Operating error 2: Operating errors 4: Data error 5: Machine data error 6: Increment table error 15: Messages 128: Diagnostics errors
+4.0	FNO	INT	0	Error number
+6.0	CH_NO	INT	0	Channel number
+8.0	TRG_NO	INT	0	Increment number

C.4 List of JOB_ERR messages

JOB_ERR messages

JOB_ERR (Hex)	JOB_ERR (Dez)	JOB_ERR (Int)	Meaning
80A0	32928	-32608	Negative acknowledgement when reading from the module. The module has been removed during the read operation - or the module is faulty.
80A1	32929	-32607	Negative acknowledge when writing to the module. The module has been removed during the write operation - or the module is faulty.
80A2	32930	-32606	DP protocol error with layer 2
80A3	32931	-32605	DP protocol error with user interface / user
80A4	32932	-32604	Communication on C bus interrupted
80B0	32944	-32592	Data record / job unknown.
80B1	32945	-32591	Incorrect length specification. Parameter FM_TYPE in the channel DB is not correctly set for the module being used.
80B2	32946	-32590	The configured slot is not assigned.
80B3	32947	-32589	Actual module type dissimilar to the set module type.
80C0	32960	-32576	The module does not yet have ready the data to be read.
80C1	32961	-32575	The data of a same-named write job is not yet ready on the module.
80C2	32962	-32574	The module is currently processing the possible maximum of jobs.
80C3	32963	-32573	Required equipment (memory etc.) is currently in use.
80C4	32964	-32572	Communication error
80C5	32965	-32571	Distributed I/O unavailable.
80C6	32966	-32570	Priority class cancellation (restart or background)
8522	34082	-31454	Channel DB or parameter DB too short. The data cannot be read from the DB. (write job)
8532	34098	-31438	The DB number of the parameter is too large. (write job)
853A	34106	-31430	Parameter DB does not exist. (write job)
8544	34116	-31420	Error with the n-th (n > 1) read access on a DB after an error has occurred. (write job)
8723	34595	-30941	Channel DB or parameter DB too short. The data cannot be written in the DB. (read job)
8730	34608	-30928	Parameter DB in the CPU is write-protected. The data cannot be written in the DB (read job).
8732	34610	-30926	The DB number of the parameter is too large. (read job)
873A	34618	-30918	Parameter DB does not exist. (read job)
8745	34629	-30907	Error with the n-th (n > 1) write access on a DB after an error has occurred. (read job)
The errors 80A2..80A4 as well as 80Cx are temporary, i.e. they can be remedied after a waiting time without you having to do anything.			

C.5 Error classes

Error classes

Class 1: Operating error

Operating errors are detected asynchronously to an operator input/control. Operating errors cause the positioning to be canceled, except in the case of error number 9. Error number 9 causes the positioning to be shut down.

No.	Meaning		Diagnostic interrupt
1	**Software limit switch start overrun**		Yes
	Cause	The actual value is outside the operating range.	
2	**Software limit switch end overrun**		Yes
	Cause	The actual value is outside the operating range.	
3	**Traversing range start overrun**		Yes
	Cause	Traversing range limit overrun (the coordinates of the traversing range limits belong with the traversing range).	
4	**Traversing range end overrun**		Yes
	Cause	Traversing range limit overrun (the coordinates of the traversing range limits belong with the traversing range).	
5	**Error on target approach**		Yes
	Cause	Target range has not been reached within the monitoring time.	
6	**Standstill range exited**		Yes
	Cause	The actual value is outside the standstill range.	
7	**Positive feedback**		Yes
	Cause	Actual value change > 1/2 standstill range in the incorrect direction.	
8	**Missing or too slight actual value change**		Yes
	Cause	No actual value change or an actual value change against the set direction within the monitoring time.	
9	**Target overrun (FM 451)**		Yes
	Cause	The target was overrun during "Set actual value on-the-fly".	
10	**Target range overrun**		Yes
	Cause	Target range has been overrun after target approach.	
11	**Changeover point switched incorrectly**		Yes
	Cause	Axis is oscillating in the changeover point.	
12	**Switch-off point switched incorrectly**		Yes
	Cause	Axis is oscillating in the switch-off point/reversal point.	
13	**Start of target range switched incorrectly**		Yes
	Cause	Axis is oscillating in the target range.	

No.	Meaning		Diagnostic interrupt
14	Change greater than half the rotary axis range		Yes
	Cause	The velocity/frequency is too high or there are faulty actual value jumps	
15	Change greater than the rotary axis range		Yes
	Cause	The velocity/frequency is too high or there are faulty actual value jumps	
16	Increment for increment number 252 not transferred (FM 451)		Yes
	Cause	The increment was not transferred.	
17	Increment for increment number 252 cannot be approached (FM 451)		Yes
	Cause	The distance between the current actual position and the specified increment is less than the changeover difference or switch-off difference.	
18	Incorrect increment for increment number 252 (FM 451)		Yes
	Cause	The increment is outside the operating range.	

Class 2: Operator errors

Operator errors are detected when control signals in the user data area are changed. The operator errors cause the positioning to shut down.

No.	Meaning		Diagnostic interrupt
1	Impermissible operating mode		No
	Cause	The selected operating mode is not permitted.	
3	Inadmissible interface job		No
	Cause	The selected signal is not permitted with this operating mode.	
4	Incorrect operating mode parameter		No
	Cause	• In "jog" mode the velocity specification is not equal to the rapid traverse or creep speed. • In "incremental approach" mode the increment is not equal to 1 to 100 or not equal to 254 and 255.	
5	Start enable not available		No
	Cause	Start enable not available upon start.	
7	Target / target range is outside the operating range		No
	Cause	Specified or calculated target is located outside of the software limit switches.	
8	Axis parameters not assigned		No
	Cause	Incorrect machine data or no machine data has been assigned for the axis.	
9	Axis not synchronized		No
	Cause	"Incremental approach" mode is possible only with an already synchronized axis.	

No.	Meaning		Diagnostic interrupt
10	**Target/incremental distance cannot be positioned**		No
	Cause	The distance between the current actual position and the specified target is less than the switch-off difference.	
17	**Reference point approach not possible**		No
	Cause	An SSI encoder is connected.	
18	**Relative or absolute incremental approach is not possible**		No
	Cause	The increment is invalid.	
19	**The switch-off difference is not greater than 1/2 the target range with increment number 255**		No
	Cause	The switch-off difference for increment 255 is less than half the target range.	
20	**Inadmissible travel in the specified direction**		No
	Cause	The distance to the software limit switch is insufficient.	

Class 4: Data error

Data errors are detected synchronously to an operator input/control. Data errors do not result in an error response.

No.	Meaning		Diagnostic interrupt
6	**Specified increment too great**		No
	Cause	The value is outside ±100 m or ±1000 m. The increment/target must not be greater than the traversing range. For a rotary axis, the coordinate must be ≥0 and less than the end of rotary axis.	
10	**Faulty zero offset (FM 451)**		No
	Cause	The zero offset is greater than ±100 m or ±1000 m. The software limit switches are outside of the traversing range (-100 m to +100 m or -1000 m to +1000 m) after setting the zero offset. Rotary axis: The absolute value of the zero offset is greater than the end of rotary axis.	
11	**Incorrect actual value**		No
	Cause	Linear axis: The coordinate is outside the current (possibly shifted) software limit switches. Rotary axis: The coordinate is < 0 or greater than the end of rotary axis.	
12	**Incorrect reference point**		No
	Cause	Linear axis: The coordinate is outside the current (possibly shifted) software limit switches. Rotary axis: The coordinate is < 0 or greater than the end of rotary axis.	

No.	Meaning		Diagnostic interrupt
20	Not permissible to activate machine data		No
	Cause	There are no new, error-free machine data on the module.	
27	Unauthorized bit-coded setting		No
	Cause	Bits that are not used and not described here are not equal to 0.	
29	Inadmissible bit coding		No
	Cause	Bits that are not used and not described here are not equal to 0.	
34	Not possible to cancel set actual value		No
	Cause	With an SSI encoder and a linear axis, the actual position value would be outside the operating range after making the setting.	
36	Incorrect changeover difference with the increment number 255		No
	Cause	The value is outside the permissible range of numbers of ±100 m or ±1000 m. For a rotary axis, the coordinate must be ≥0 and less than the end of rotary axis.	
37	Incorrect switch-off difference with the increment number 255		No
	Cause	The value is outside the permissible range of numbers of ±100 m or ±1000 m. The switch-off difference must be less than the changeover difference.	
107	Axis parameters not assigned		No
	Cause	The axis has no machine data or the machine data are not enabled.	
108	Axis not synchronized		No
	Cause	One of the "set actual value" and "set actual value on-the-fly" jobs has been triggered even though the axis is not synchronized.	

Class 5: Machine data error

The diagnostic interrupt is only triggered if an error is detected in the system data block (SDB). Machine data errors do not result in an error response.

No.	Meaning		Diagnostic interrupt
5	**Error in hardware interrupt setting**		Yes
	Cause	You have attempted to select a hardware interrupt that the module does not support.	
6	**Incorrect minimum edge distance (FM 451)**		Yes
	Cause	You have entered a value < 0 or > 10^9 µm as the minimum edge distance.	
7	**Incorrect system of units**		Yes
	Cause	The value for the system of units is outside the permissible range of 1 to 4 and 6.	
8	**Incorrect axis type**		Yes
	Cause	You have entered neither 0 nor 1 as an axis type.	
9	**Incorrect end of rotary axis**		Yes
	Cause	The value for the end of rotary axis is outside the permissible range of 1 to 10^9 µm or 1 to 10^8 µm (depending on resolution).	
10	**Incorrect encoder type**		Yes
	Cause	The value for the encoder type is outside the permissible range of 1 to 4.	
11	**Incorrect distance per encoder revolution**		Yes
	Cause	The value for distance per encoder revolution is outside the permissible range of 1 to 10^9 µm (irrespective of the resolution).	
13	**Incorrect increments per encoder revolution** (refer to section entitled "Encoder machine data (Page 77)")		Yes
14	**Incorrect number of revolutions** (refer to section entitled "Encoder machine data (Page 77)")		Yes
15	**Incorrect baud rate**		Yes
	Cause	You have specified a value for the baud rate that lies outside the permissible range of 0 to 3.	
16	**Incorrect reference point coordinate**		Yes
	Cause	The coordinate is outside the range of -100 m to +100 m or -1000 m to +1000 m (depending on the resolution). Linear axis: The coordinate is outside the operating range. Rotary axis: The coordinate is greater than the end of rotary axis or < 0.	
17	**Incorrect absolute encoder adjustment**		Yes
	Cause	SSI navigator: The value of the absolute encoder adjustment is not in the encoder range (increments per encoder revolution x number of revolutions - 1).	

No.	Meaning		Diagnostic interrupt
18	Incorrect type of reference point approach		Yes
	Cause	You have specified a value outside of the permissible value quantities of 0, 1, 2 and 3.	
19	Incorrect counting direction		Yes
	Cause	You have specified a value outside of the permissible value quantities of 0 and 1.	
20	It is not possible to monitor the hardware		Yes
	Cause	You have set the monitoring of message errors in the parameter DB to "FALSE". The monitoring of missing pulses is not possible with the used encoder. Disable the MON_PULSE parameter.	
21	Incorrect software limit switch start		Yes
	Cause	Linear axis: The software limit switch start is outside the traversing range (-100 m to +100 m or -1000 m to +1000 m, depending on resolution). Linear axis: The software limit switch start (including any existing zero offset) is less than -100 m or -1000 m (depending on the resolution).	
22	Incorrect software limit switch end		Yes
	Cause	Linear axis: The software limit switch end is outside the traversing range (-100 m to +100 m or -1000 m to +1000 m, depending on the resolution) or is less than the software limit switch start. Linear axis: The software limit switch end (including any existing zero offset) is greater than +100 m or +1000 m (depending on the resolution).	
23	Incorrect maximum velocity		Yes
	Cause	The data that are not listed in the parameter DB must be 0.	
24	Incorrect target range		Yes
	Cause	Linear axis: Range between 0 to 100 m or 1000 m, depending on the resolution. Rotary axis: Range greater than the end of rotary axis.	
25	Incorrect monitoring time		Yes
	Cause	The value for the monitoring time is outside the permissible range from 0 to 100 000 ms.	
26	Incorrect standstill range		Yes
	Cause	Linear axis: Range between 0 to 100 m or 1000 m, depending on the resolution. Rotary axis: Range greater than the end of rotary axis.	
127	Incorrect standstill velocity		Yes
	Cause	The value for the standstill velocity is outside the permissible range from 0 to 100 000 μm/min.	
128	Incorrect control mode		Yes
	Cause	You have specified a value for the control mode that lies outside the permissible range of 1 to 4.	

No.	Meaning		Diagnostic interrupt
129	**Incorrect start velocity for the reference point approach**		Yes
	Cause	You have entered neither 0 nor 1 as a start velocity.	
130	**Incorrect changeover difference in the + direction**		Yes
	Cause	Linear axis: Range between 0 to 100 m or 1000 m (depending on the resolution). Rotary axis: Range greater than the end of rotary axis and less than 1/2 the target range.	
131	**Incorrect changeover difference in the - direction**		Yes
	Cause	Linear axis: Range between 0 to 100 m or 1000 m (depending on the resolution). Rotary axis: Range greater than the end of rotary axis and less than 1/2 the target range.	
132	**Incorrect switch-off difference in the + direction**		Yes
	Cause	The switch-off difference is greater than the plus changeover difference, less than 1/2 target range, or is outside the permissible range between 0 to 100 m or 0 to 1000 m (depending on resolution).	
133	**Incorrect switch-off difference in the - direction**		Yes
	Cause	The switch-off difference is greater than the minus changeover difference, less than 1/2 target range, or is outside the permissible range between 0 to 100 m or 0 to 1000 m (depending on resolution).	
200	**Incorrect resolution**		Yes
	Cause	You have specified a resolution < 0.1 μm/pulse or > 1000 μm/pulse. You have specified a distance per encoder revolution and a quantity of pulses per encoder revolution that results in a resolution < 0.1 or > 1000.	
201	**Encoder does not fit the operating range / rotary axis range**		Yes
	Cause	SSI encoders and rotary axes: The encoder does not cover the exact rotary axis range. Linear axis: The encoder does not cover at least the operating range (including software limit switches).	

Class 6: Increment table error

The increment table errors do not result in an error response.

No.	Meaning		Diagnostic interrupt
6	Specified increment in the increment table too great		No
	Cause	The value is outside ±100 m or ±1000 m. The increment/target must not be greater than the traversing range. For a rotary axis, the coordinate must be ≥0 and less than the end of rotary axis.	

Class 15: Messages

Messages do not result in an error response.

No.	Meaning		Diagnostic interrupt
1	Start of parameter assignment		No
	Cause	The module has detected a parameter assignment via a system data block.	
2	End of parameter assignment		No
	Cause	The module has processed the parameter assignment via a system data block error-free.	
11	Distance to changeover point too short		No
	Cause	The hardware response times cannot be adhered to because the distance between the switching points is too short.	
12	Distance to reversal point too short		No
	Cause	The hardware response times cannot be adhered to because the distance between the switching points is too short.	
14	Distance to switch-off point too short		No
	Cause	The hardware response times cannot be adhered to because the distance between the switching points is too short.	
15	Distance to target range start too short		No
	Cause	The hardware response times cannot be adhered to because the distance between the switching points is too short.	

Class 128: Diagnostic errors

No.	Meaning		Diagnostic interrupt
4	**Missing external auxiliary supply**		Yes
	Cause	• External auxiliary voltage 24 V is not connected or has failed. • The fuse on the module is defective. • Undervoltage • Ground wire break • Short circuit (e.g., at the connected encoder)	
	Effect	• The positioning is cancelled on all channels. • Outputs are switched off • Synchronization is cancelled for incremental encoders when the auxiliary voltage for the encoder supply is missing. • The FM 351 parameters are not assigned. • Start enable is deleted	
	Remedy	Ensure a correct 24 V connection (if 24 V connection is correct, then the module is defective).	
5	**Front connector is missing (FM 451)**		Yes
	Cause	Front connector of the positioning module is not connected.	
	Effect	• No external 24 V auxiliary supply • Module not ready	
	Remedy	Plug the front connector into the positioning module	
51	**Time monitoring triggered (watch dog)**		Yes
	Cause	• Strong interference on the FM 351 • Errors in the FM 351	
	Effect	• Module is reset • All outputs are switched off • If, after resetting the module, no module fault is detected, the module is ready for operation again • The module signals the expired WATCHDOG with "incoming" and "outgoing"	
	Remedy	• Eliminate the interference • Please contact the responsible sales department, whereby information regarding the precise circumstances that caused the error are of considerable importance. • Replace the FM 351	

No.	Meaning		Diagnostic interrupt
144	**Encoder wire break**		Yes
	Cause	• The encoder cable has sheared off or is not plugged in • Encoders without cross signals • Incorrect pin assignment • Cable length too long • Encoder signals short-circuited • Encoder signal edge fault • Maximum input frequency of the encoder input has been exceeded • Encoder supply failure	
	Effect	• The positioning is canceled • Outputs are switched off • Delete the synchronization with incremental encoders • Start enable is deleted	
	Remedy	• Check encoder cable • Comply with encoder specification • Monitoring can be temporarily disabled using the parameter assignment screen, but at the responsibility of the operator. • Comply with the technical specifications of the module	
145	**Absolute encoder error**		Yes
	Cause	The message frame traffic between the FM 351 and the absolute encoder (SSI) is faulty or has been interrupted: • The encoder cable has sheared off or is not plugged in • Incorrect encoder type • The encoder is incorrectly set (programmable encoder) • The message length has been incorrectly specified • The encoder is supplying incorrect values (encoder is faulty) • Interference on measuring system cable • Selected baud rate is too high • Monoflop time of the encoder is greater than 64 μs	
	Effect	• The positioning is canceled • Outputs are switched off • Start enable is deleted	
	Remedy	• Check encoder cable • Check encoder • Check the message traffic between the encoder and the FM 351	

No.	Meaning		Diagnostic interrupt
146	**Missing pulses of incremental encoder**		Yes
	Cause	• Encoder monitoring has detected missing pulses. • Number of increments per encoder revolution incorrectly entered. • Encoder faulty: is not supplying the specified number of pulses • incorrect or no zero mark • Crosstalk on the encoder cable	
	Effect	• The positioning is canceled. • Outputs are switched off • Start enable is deleted	
	Remedy	• Enter the correct number of increments per encoder revolution (parameter assignment screen) • Check the encoder and encoder cable • Comply with the shielding and grounding directives • Monitoring can be temporarily disabled using the parameter assignment screen, but at the responsibility of the operator.	

Programming without SFB 52 and SFB 53 D

D.1 Overview of the Programming without SFB 52 and SFB 53 section

If your CPU does not support the system blocks SFB 52 and SFB 53 with DPV1 functionality

Then use the blocks from the program folder "FM 351,451 ABS V1" to program the FM 351.

You will find a description in this section.

D.2 Basics of programming a positioning module

Task

You can assign parameters, control, and commission each channel of the positioning module via a user program. The following chapter illustrates how to design a user program to suit your application.

Preparation

1. In SIMATIC Manager, open the block library FMx51LIB and copy the required functions (FC) and block templates (UDT) to the block container of your project. If the block numbers are already being used, assign new numbers. The block names are entered unchanged in the symbol table of your S7 program.

Name	Meaning
FC ABS_INIT (FC 0)	Required to initialize the channel DB after a module start-up
FC ABS_CTRL (FC 1)	Required for data exchange and for controlling
FC ABS_DIAG (FC 2)	Required if you are processing detailed diagnostic information in the program or want to make this available for a operator control and monitoring system
UDT ABS_CHANTYPE(UDT 1)	Required in order to create a channel DB for each channel; this is then used by the FC ABS_INIT and FC ABS_CTRL
UDT ABS_DIAGTYPE (UDT 2)	Required in order to create a diagnostic DB for each module; this is then used by the FC ABS_DIAG
UDT ABS_PARATYPE(UDT 3)	Required in order to create a parameter DB with parameters; this is then used by the FC ABS_CTRL in order to write or read machine data and increment tables

2. Create the data blocks (DBs) using the UDTs in the block container of your S7 program:
   - a separate channel DB for each channel.
   - If you want to write or read parameters using the user program, you need a separate parameter DB for each channel.
   - If you would like to execute the diagnostics using the user program, you require only a diagnostic DB for each module.

3. Enter the module address in the associated channel DB and, if necessary, also in the corresponding diagnostic DB in the "MOD_ADDR" parameter.

   Proceed as follows to enter the module address:

   – Recommended procedure:

     Assign the module address to the channel DB/diagnostic DB in the user program so that the assignment of the module address takes place when you call the user program in OB 100.

   – Alternative procedure:

     You can have the module address entered automatically if you select the module in HW Config, open the "Properties" dialog with the menu command **Edit > Object Properties**, and use the "Mod_Adr" button there to select a channel DB and diagnostic DB, if necessary. But in this case the values entered in the channel DB/diagnostic DB (including the module address) are reset to their initial values in the event of a consistency check (menu command **Edit > Check block consistency** opens the "Check block consistency" dialog) followed by a compilation (menu command **Program > Compile All** in the "Check block consistency" dialog box).

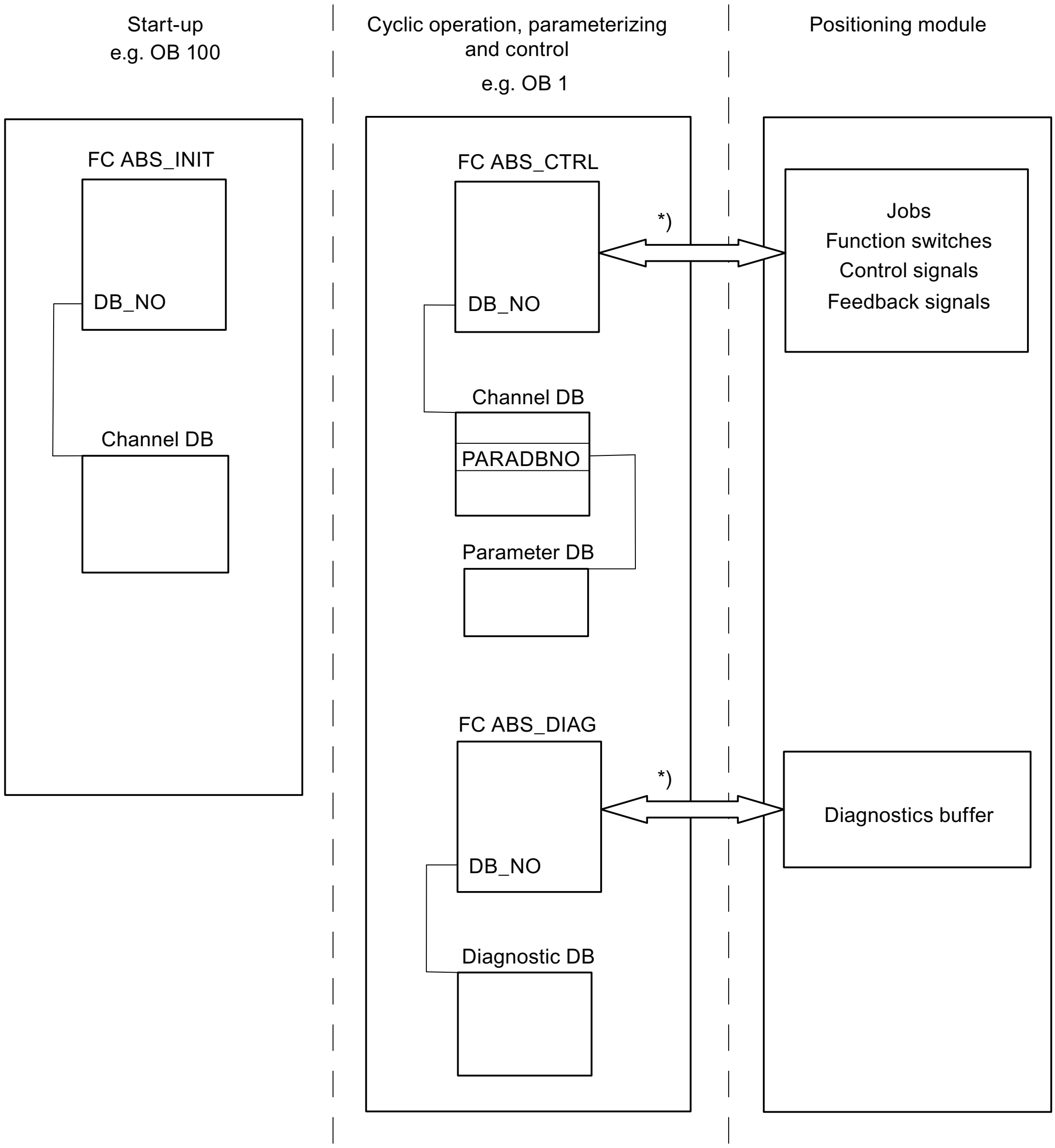
     The values are not changed if there is only a consistency check without compilation.

     The menu command **Edit > Compile All** is only required within the consistency check if the project was last edited with STEP 7 V5.0 Service Pack 2 or later.

4. Enter the channel number and, if necessary, the number of the parameter DB also in the respective channel DB.

5. If your programming device or PC is connected to a CPU, you can now download the FCs and DBs to the CPU.

The following figure shows you how the positioning module, FCs, DBs and OBs communicate with each other.



*) The module address (channel DB/diagnostic DB) entered in the "MOD_ADDR" parameter is used for accessing the module. We recommend you assign the module address to the channel DB/diagnostic DB in the user program so that the assignment of the module address takes place when you call the user program in OB 100.

Figure D-1 Data exchange between FCs, DBs and positioning module

D.3 FC ABS_INIT (FC 0)

Task

The FC ABS_INIT deletes the following data in the channel DB:

- The control signals
- The checkback signals
- The trigger bits, done bits, and error bits of the jobs
- The function switches and their done bits and error bits
- The job management for the FC ABS_CTRL

Call

The function must be run through for each channel after start-up (supply voltage on) of the module or the CPU. Call the function, for example, in the start-up OB 100 and the insertion/removal OB 83 or the initialization phase of your user program. This ensures that your user program does not access obsolete data after a CPU or module restart.

Data block used

Channel DB:

The module address must be entered in the channel DB.

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	IN	Number of the channel DB

Return values

The specified function does not supply a return value.

D.4 FC ABS_CTRL (FC 1)

Tasks

With the FC ABS_CTRL you can read the operating data for each channel of the module. You can also configure the channels and control them during operation. To do this, use the control signals, checkback signals, function switches as well as the write and read jobs.

With each call the function carries out the following activities:

- Reading checkback signals:

  The FC ABS_CTRL reads all the checkback signals for a channel and enters them in the channel DB. Because the control signals and jobs are not executed until after this step, the checkback signals reflect the status of the channel before the block was called.

- Job management:

  The FC ABS_CTRL processes the read and write jobs and transfers the data between the channel DB, parameter DB and the module.

- Writing control signals:

  The control signals that are entered in the channel DB are transferred to the module.

Call

The FC ABS_CTRL must be called cyclically for each channel, e.g., in OB 1.

Before you call the FC ABS_CTRL, enter all the data required to execute the intended functions in the channel DB.

Data blocks used

- Channel DB:

  The module address and the channel number must be entered in the channel DB. Incorrect information could result in I/O access errors or in an access to a different module, which in turn gives rise to data corruption.

- Parameter DB:

  If you want to use jobs to write or read the machine data, you need a parameter DB, whose number must be entered in the channel DB.

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	IN	Number of the channel DB
RET_VAL	INT	OUT	Return value

Return values

The function provides the following return values:

RET_VAL	BR	Description
1	1	At least 1 job is active
0	1	No job is active, no error
-1	0	Error: Data error (DATA_ERR) or communication error (JOB_ERR) has occurred

Jobs

Data exchange with the module other than the control and checkback signals is handled using jobs.

To deliver a job you set the respective trigger bit in the channel DB and in the case of write jobs also the respective data. Then call the FC ABS_CTRL in order to execute the job.

If you are using the FM 351 centrally, a read job requires exactly one cycle. If you are using the FM 351 as a distributed module, a read job can require several cycles.

On account of the necessary acknowledgements of the module, a write job requires at least 3 calls or OB cycles.

When a job is finished, the block takes back the trigger bit. The next time the block is called, the following job will be ascertained and executed.

For each job, there is not only a trigger bit (extension _EN as in "enable") but also a done bit and an error bit. These have the extension _D as in "done" and _ERR as in "error" in their names. The FC ABS_CTRL updates the done bits and error bits once the processing of a job has ended. After the evaluation or prior to issuing a job, these bits should be set to 0.

If you set the JOBRESET bit, all the done bits and error bits will be reset prior to processing the awaiting jobs. The JOBRESET bit is then set back to 0.

Function switch

The function switches switch the states of the channel on and off. A job for writing the function switches will only be executed when there is a change in a switch setting. The setting of the function switch is retained after the job has been executed.

Function switches and jobs can be used simultaneously with a call of the FC ABS_CTRL.

As in the case of the jobs, alongside the function switches there are trigger bits with the ending _ON / -OFF, done bits with the ending _D and error bits with the ending _ERR.

To be able to evaluate the done bits and error bits of the function switches, you should set these bits to 0 before you issue a job to change a function switch.

Order of job processing

You can select several jobs simultaneously. If no jobs are active, the job management of the FC ABS_CTRL searches as of job MDWR_EN to see whether trigger bits are set or changes to the function switches have been made. If a job is found, it is processed. Once the job is concluded, the job management searches for the next job to be processed. If the last job ENCVAL_EN has been searched, searching starts over with the MDWR_EN job. This searching process is repeated until all the jobs have been processed.

The jobs are processed in the following technologically appropriate order:

Order	Address in the channel DB	Name	Meaning	Reset from
Write jobs				
1	35.0	MDWR_EN	Write machine data	FC 1
2	35.1 35.2 35.3 36.4	MD_EN DELDIST_EN AVALREM_EN DELDIAG_EN	Enable machine data Delete distance-to-go Cancel set actual value Delete diagnostic buffer	FC 1
3	35.4	TRGL1WR_EN	Write increment table 1	FC 1
4	35.5	TRGL2WR_EN	Write increment table 2	FC 1
5	35.6	REFPT_EN	Set reference point	FC 1
6	 34.0 34.1 34.2	Function switch: PLOOP_ON MLOOP_ON EI_OFF	 Loop approach in plus direction Loop approach in minus direction Do not evaluate enable input	User program
7	35.7	AVAL_EN	Set actual value	FC 1
10	36.2	TRG252_254_EN	Write increment for increment number 254	FC 1
11	36.3	TRG255_EN	Write increment for increment number 255	FC 1
Read jobs				
12	36.5	MDRD_EN	Read machine data	FC 1
13	36.6	TRGL1RD_EN	Read increment table 1	FC 1
14	36.7	TRGL2RD_EN	Read increment table 2	FC 1
16	37.1	ACTSPD_EN	Read current velocity, distance-to-go and current increment	FC 1
17	37.2	ENCVAL_EN	Read encoder data	FC 1

This order enables you to completely trigger a positioning with a set of jobs and control signals. The jobs go from writing and activating the machine data through the setting of the external enable input up to writing the increments for the incremental approaches.

Control signals

If there is a STOP signal, an operator error or a drive enable is missing, the block resets the control signals START, DIR_M and DIR_P.

You can restart a motion after you have acknowledged the operator error with OT_ERR_A=1. With this acknowledgement you cannot submit any other jobs and control signals.

If there is no operator error pending, the block sets the acknowledgement for the operator error OT_ERR_A to 0.

The block resets the START, DIR_P, and DIR_M start signals when the channel signals the start of the motion, except in "jog" mode.

If the axis parameters are not assigned, the block withholds all the control signals with the exception of the OT_ERR_A operator error acknowledgement.

Jobs and control signals

You can issue several jobs at the same time, also together with the control signals necessary for the positioning. If at least one write job is issued at the same time as the control signals START, DIR_M or DIR_P, the block retains these control signals until the write jobs have been processed.

Jobs during an ongoing positioning

If they are issued during a positioning, the write jobs listed in the following table will be retained until the positioning has ended and carried out only after the following call of the block.

Address	Name	Type	Initial value	Comment
34.0	PLOOP_ON	BOOL	FALSE	1 = loop approach in plus direction
34.1	MLOOP_ON	BOOL	FALSE	1 = loop approach in minus direction
34.2	EI_OFF	BOOL	FALSE	1 = do not evaluate enable input
35.1	MD_EN	BOOL	FALSE	1 = enable machine data
35.2	DELDIST_EN	BOOL	FALSE	1 = delete distance-to-go
35.3	AVALREM_EN	BOOL	FALSE	1 = cancel set actual value
35.6	REFPT_EN	BOOL	FALSE	1 = set reference point coordinate
35.7	AVAL_EN	BOOL	FALSE	1 = set actual value
36.4	DELDIAG_EN	BOOL	FALSE	1 = delete diagnostic buffer

Start-up

On start-up of the module or the CPU, call the FC ABS_INIT. Among other things, this resets the function switches. The FC ABS_CTRL acknowledges the module start-up. During this time the RET_VAL and JOBBUSY = 1.

Job status

The status of the job processing can be read off from the return value RET_VAL and from the active bit JOBBUSY in the channel DB. The status of an individual job can be evaluated using the trigger bits, done bits, and error bits of this job.

	RET_VAL	JOBBUSY	Trigger bit _EN	Done bit _D	Error bit _ERR
Job active	1	1	1	0	0
Job completed without errors	0	0	0	1	0
Job completed with errors	-1	0	0	1	1
Write job cancelled	-1	0	0	0	1

Response to errors

If faulty data are written during a write job, the channel issues the checkback signal DATA_ERR = 1 to the channel DB. If an error occurs during communication with the module in a write or read job, the cause of error is stored in the JOB_ERR parameter in the channel DB.

- Error with a write job:

  If an error occurs in a job, the trigger bit is canceled and error bit _ERR and done bit _D are set. The trigger bit is also canceled for all write jobs still pending, but only error bit _ERR is set. The awaiting write jobs will be withdrawn because jobs could stack up on top of each other.

  The awaiting read jobs will continue to be processed. In doing so, the JOB_ERR is then reset for each job.

- Error with a read job:

  If an error occurs in a job, the trigger bit is canceled and error bit _ERR and done bit _D are set.

  The awaiting read jobs will continue to be processed. In doing so, the JOB_ERR is then reset for each job.

For further error information, refer to JOB_ERR and DATA_ERR parameter descriptions in the section entitled "Diagnosis (Page 137)".

Program structure

The following figure shows the basic structure of a user program used to cyclically control a channel of a module following a one-time startup initialization. The return value RET_VAL of the FC ABS_CTRL is used in the user program for a general error evaluation.

For every other channel a sequence in accordance with the following figure can be executed parallel and independently.

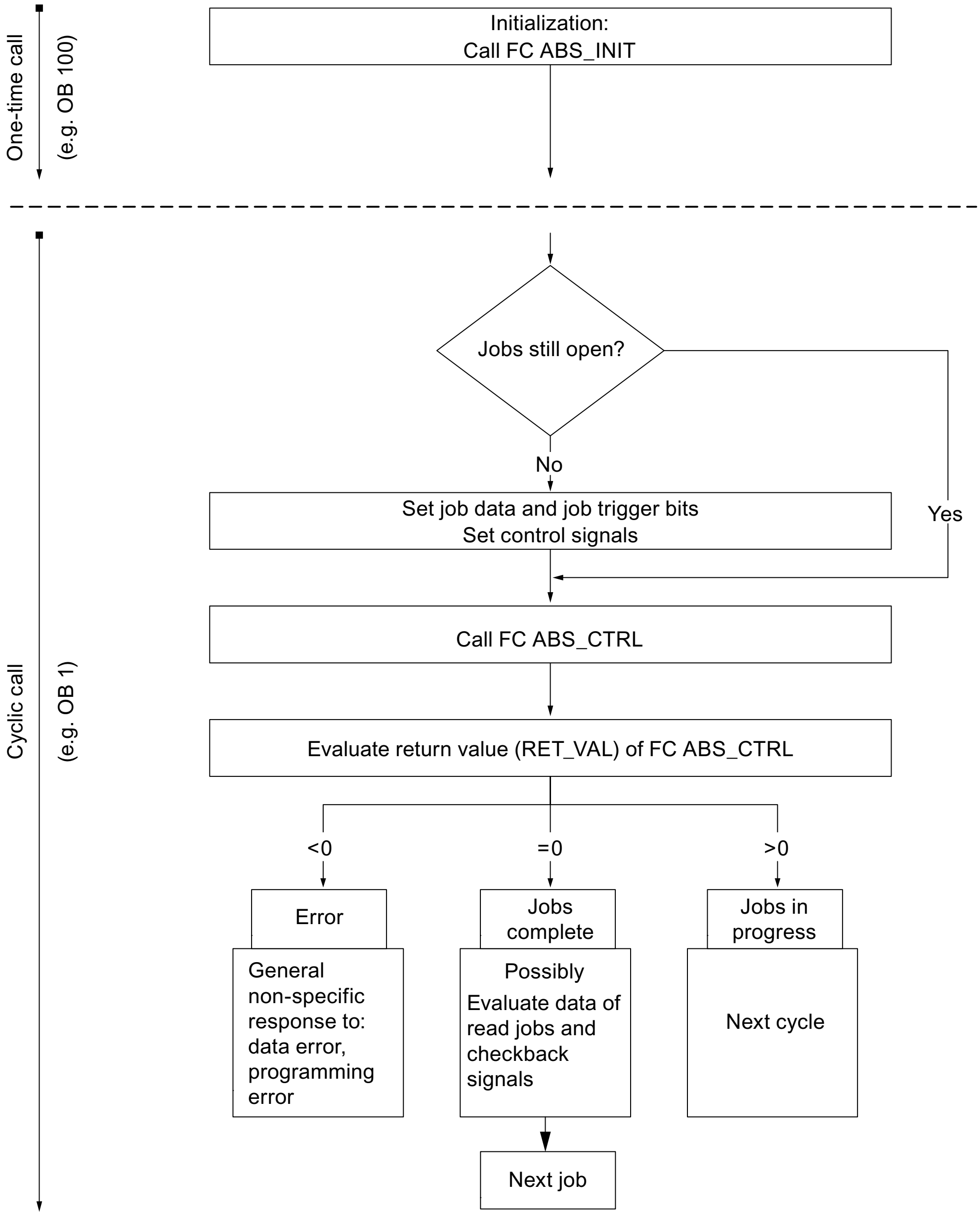


Figure D-2 General program structure

D.5 FC ABS_DIAG (FC 2)

Tasks

With the FC ABS_DIAG you read out the diagnostic buffer of the module and make it available for a display in the operator control and monitoring system or for a programmed evaluation.

Call

The function must be called cyclically, e.g., in OB 1. An additional call in an interrupt OB is not permitted. At least two calls (cycles) are required for a complete execution of the function.

The function reads out the diagnostic buffer when a new entry in the diagnostic buffer is indicated by means of the checkback signal DIAG = 1 in the channel DB. After the diagnostic buffer is read, the DIAG bit in the channel DB of the module is set to 0.

Data block used

Diagnostic DB:

The module address must be entered in the diagnostic DB. The latest entry in the diagnostic buffer will be entered in the structure DIAG[1] and the oldest entry in the structure DIAG[9].

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	IN	Number of the diagnostic DB
RET_VAL	INT	OUT	Return value

Return values

The function provides the following return values:

RET_VAL	BR	Description
1	1	Job active
0	1	No job is active, no error
-1	0	Error

Jobs

You can read the diagnostic buffer independent of a new entry by setting the trigger bit DIAGRD_EN in the diagnostic DB. After the diagnostic buffer is read, the trigger bit is set to 0.

Carry out this job after a CPU start-up and a module start-up. This way you can ensure that the content of the diagnostic DB corresponds with the content of the module´s diagnostic buffer, even if the module has not made a new entry in the diagnostic buffer.

Start-up

The function does not perform a start-up processing.

Response to errors

In the case of a faulty execution, the cause of the error can be found in the diagnostic DB in the JOB_ERR parameter; refer to the section entitled "Diagnosis (Page 137)".

D.6 Data blocks

D.6.1 Templates for data blocks

Block templates UDT

For each data block there is a block template UDT stored in the provided library (FMx51LIB). From these UDTs you can create data blocks with any numbers or names.

D.6.2 Channel DB

Task

The channel DB is the data interface between the user program and the FM 351. It includes and accepts all the data that is required for controlling and operating a channel.

Structure

The channel DB is divided into different areas:

Channel DB
Module address *)
Channel number
Number of the parameter DB
Control signals
Checkback signals
Function switch
Trigger bits for write jobs
Activation bits for read jobs
Done bits
Error bits
Job management for functions
Data for jobs

*) You can also enter the address using the configuration software

D.6.3 Diagnostic DB

Task

The diagnostic DB (refer to the section entitled Data and structure of the diagnostic DB) is the data storage for the FB ABS_DIAG and includes the diagnostic buffer of the module that has been processed by this function block.

Structure

Diagnostic DB
Module address
Internal data
Job status
Trigger bit
Processed diagnostic buffer

D.6.4 Parameter DB

Task

If you want to change the machine data and increment tables during operation, you require a parameter DB in which this data is stored. The parameters can be changed from the user program or from an operator control and monitoring system.

You can export the data displayed in the configuration software into a parameter DB. You can also import a parameter DB into the configuration software and view it there.

Each module channel can have several sets of parameter assignment data, e.g., for different recipes. You can switch among these in your program.

Structure

Parameter DB
Machine data
Increment tables

D.7 Technical specifications of the FCs and DBs for the FM 351

Technical specifications

The following table offers you an overview of the technical specifications for the functions and data blocks.

Table D- 1 Technical specifications of the functions and data blocks for the FM 351

No.	Block name	Version	Assignment in load memory (bytes)	Assignment in work memory (bytes)	Assignment in local data area (bytes)	MC7 code / data (bytes)	Called system functions
FC 0	FC ABS_INIT	1.0	184	130	2	94	
FC 1	FC ABS_CTRL	1.0	4548	4176	34	4140	SFC 58: WR_REC, SFC 59: RD_REC
FC 2	FC ABS_DIAG	1.0	1800	1658	42	1622	SFC 59: RD_REC
	Channel DB	-	638	184	-	148	
	Parameter DB	-	840	556	-	520	
	Diagnostic DB	-	524	388	-	352	

Module cycle

The checkback signals of a channel are updated by the module every 8 ms.

D.8 Quicker access to module data

Application

In special applications or in an alarm level, a particularly fast access to checkback and control signals could be required. You can reach this data directly via the input and output areas of the module.

For start-up coordination, after each start-up of the module (e.g. after connecting the module, after CPU STOP → RUN) you must call the FC ABS_CTRL until the end of the start-up is indicated by RET_VAL = 0. After this you must not use the FC ABS_CTRL any more.

Note

It is not possible to use the FC ABS_CTRL together with a write access.

Reading checkback signals by means of direct access

The byte addresses must be specified relative to the start address of the outputs of the respective channel. The names of the parameters correspond with the names in the channel DB.

Start address channel 1 = start address of the module

Start address channel 2 = start address of the module + 8

In STL, you access the data using the commands PEB (read 1 byte), PEW (read 2 bytes) and PED (read 4 bytes).

Address	Bit number							
	7	6	5	4	3	2	1	0
Byte 0	PARA	Internal	Internal	DATA_ERR	OT_ERR	DIAG	Internal	Internal
Byte 1	CHGOVER	CUTOFF	ZSPEED	SPEED_OUT	0	WAIT_EI	WORKING	ST_ENBLD
Byte 2	MODE_OUT							
Byte 3	POS_RCD	0	0	0	GO_P	GO_M	0	SYNC
Byte 4	ACT_POS							
Byte 5								
Byte 6								
Byte 7								

Example: Actual position value ACT_POS

The start address of the module is 512

STL	
L PED 516	Read the current actual position value (ACT_POS) from channel 1 with direct access: Start address of the channel + 4

Write control signals using direct access

The byte addresses must be specified relative to the start address of the inputs of the respective channel. The names of the parameters correspond with the names in the channel DB.

Start address channel 1 = start address of the module

Start address channel 2 = start address of the module + 8

In STL, you access the data using the commands PAB (write 1 byte), PAW (write 2 bytes) and PAD (write 4 bytes).

Address	Bit number							
	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	OT_ERR_A	0	0	0
Byte 1	DRV_EN	0	0	0	DIR_P	DIR_M	STOP	START
Byte 2	MODE_IN							
Byte 3	MODE_TYPE							
Byte 4	Reserved							
Byte 5								
Byte 6								
Byte 7								

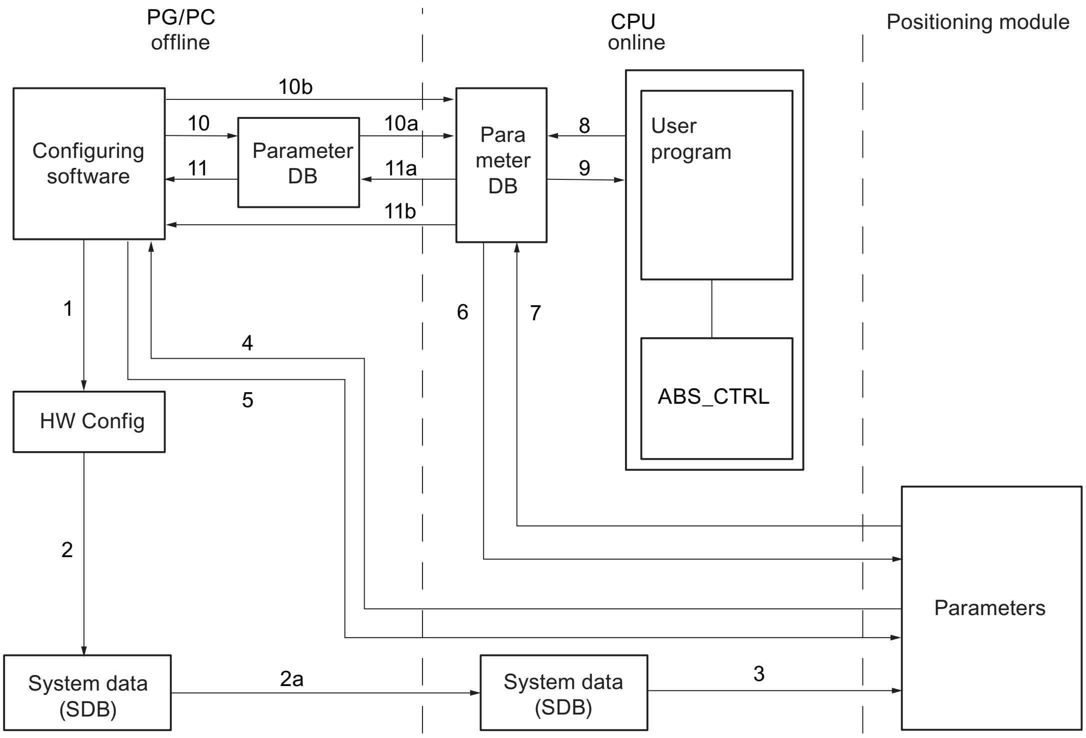
Example: START signals of channel 2

The start address of the module is 512

STL	
L 2#10001000	Set DRV_EN and DIR_P to 1
T PAB 521	Write signals with direct access for channel 2: Start address of the module +8 + 1

D.9 Parameter transmission paths

Parameters include the following machine data and increments.



1 Saving parameters in the configuration software.
2 Save and compile the HW configuration.
2a Download the HW configuration to the CPU. The CPU automatically carries out step 3.
3 The CPU writes the parameters when system parameter assignments are carried out for the module.
4 With the command "download target system to PG" download the parameters of a channel of the module to the PG.
5 With the command "download target system" download the parameters from the configuration software into a channel of the module.
6 Write parameters using jobs of the user program in a channel of the module.
7 Read parameters using jobs of the user program from a channel of the module.
8 Store parameters from the user program into the online DB.
9 Read in parameters from the online DB into the user program.
10 Export parameters from the configuration software to the offline DB.
10a Download the offline DB to the CPU.
10b Export parameters from the configuration software to the online DB.
11 Import parameters from the offline DB into the configuration software.
11a Load parameters from the online DB to the PG.
11b Import parameters from the online DB into the configuration software.

Figure D-3 Parameter transmission paths

Some applications for transmission of parameters:

- You process the parameters using the configuration software. The channels of the module should then be automatically assigned upon start-up.

  Carry out steps 1, 2, and 2a.

- You change the parameters during commissioning in debug mode in the configuration software:

  Carry out steps 4 and 5.

- The parameters changed during commissioning should then be automatically loaded upon start-up:

  Carry out steps 1, 2, and 2a.

- You create the parameters using the configuration software. The channels of the module should be assigned upon start-up only by the user program via the data blocks:

  Carry out steps 10, 10a and 6 or 10b and 6.

- You would like to create convenient stored data for recipes:

  Carry out steps 10 and 10a.

- You create the parameters using the configuration software. These should be made available to the user program for temporary changes.

  Carry out steps 1, 2 and 2a for the automatic parameter assignment.

  Carry out the steps 10, 10a, 9, 8 and 6 for access by the user program.

- You change existing parameters solely via the user program:

  Carry out steps 7, 9, 8 and 6.

- You would like to use the configuration software to view the data that has been changed via the user program:

  Carry out steps 11a and 11 or just 11b.

- The parameters changed by the user program should also be automatically loaded upon start-up:

  Carry out steps 11b or 11a, 11 and then 1, 2, 2a.

Index

H

I

J

L

M

N

O

P

R

T

U

W

X