SIEMENS

SIMATIC

S7-400
FM 452 electronic cam controller

Operating Instructions

05/2011

A5E01071729-02

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

A5E01071729-02
Ⓟ 07/2011

Table of contents

Preface 1

1.1 Preface

Scope of the manual

This manual contains the description of the FM 452 electronic cam controller as valid at the time it was published. We reserve the right to publish modifications of FM 452 functionality in a separate Product Information.

Content of this manual

This manual describes the hardware and software of the FM 452 electronic cam controller.

It comprises:

- Fundamentals part (Chapters 1 to 8)
- Reference part (Chapters 9 to 13)
- Appendices (A, B, and C)
- Index

Standards

The SIMATIC S7-400 product series fulfills the requirements and criteria of IEC 61131-2.

Recycling and disposal

The FM 452 is low in contaminants and can therefore be recycled. For ecologically compatible recycling and disposal of your old device, contact a certificated 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".

Product overview 2

2.1 FM 452

Description

The FM 452 function module is a single-channel, electronic cam controller for integration in the S7-400 automation system. It supports rotary and linear axes. The module supports proximity switches, and incremental/absolute encoders (SSI) for position feedback. When operating in slave mode, the FM 452 can listen in on the SSI frame of an absolute encoder.

You can program up to 128 position or timing cams that you can assign to 32 cam tracks as required. The first 16 cam tracks are output at the digital outputs of the module. For information about the functions and settings of the cam control, refer to the next chapters.

You can operate several FM 452 stations in parallel. The module also supports combinations with other FM/CP modules. A typical application is the combination of the module with an FM 451 positioning module.

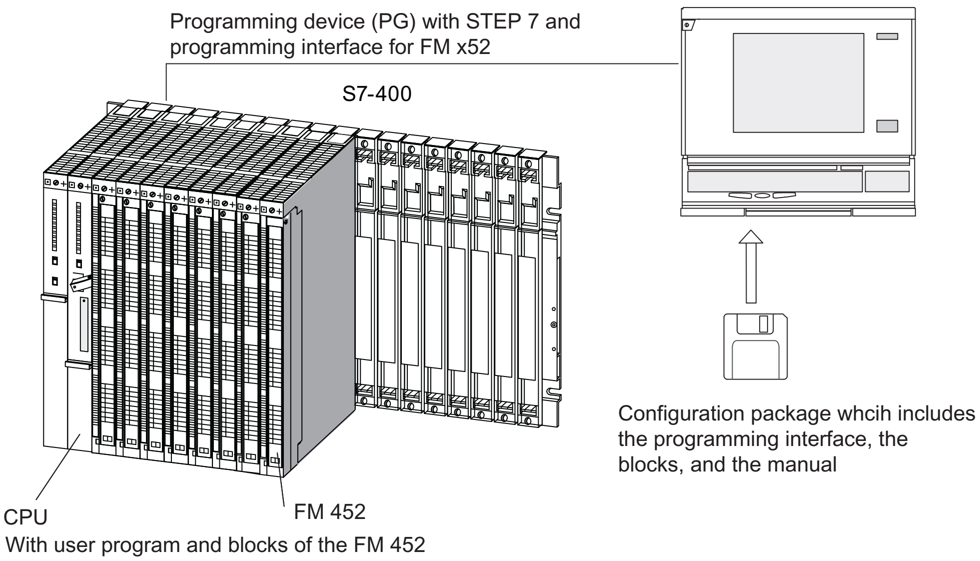


Figure 2-1 Configuration of a SIMATIC S7-400 with FM 452

2.2 Fields of application of FM 452

Example: Applying glue tracks

In the following example, glue tracks are applied to wooden boards. Each cam track controls one glue nozzle via a digital output.

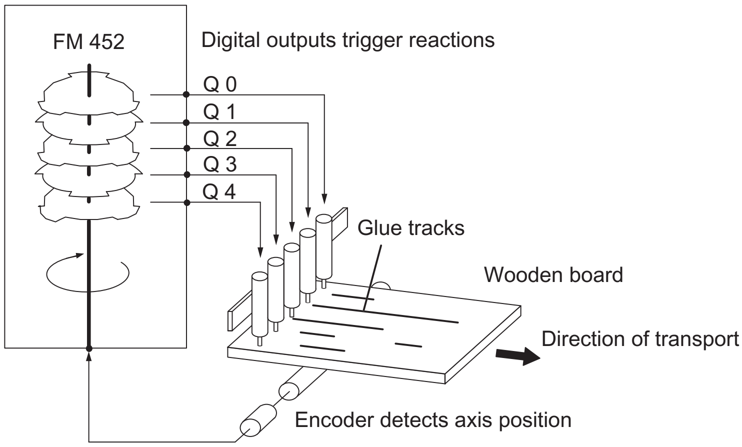


Figure 2-2 Example of an electronic cam control

Example: Press control

The automation of an eccentric press using a cam controller is another typical application.

Press operation is based on a rotary motion, i.e., the rotary axis rotates 360 degrees and then starts the next cycle at zero.

Typical tasks of an electronic cam controller:

- Switching a lubricating system on and off
- Enabling pick-up and release of materials (for example, gripper control)
- Stopping the press at the "upper dead center"

Example: Packaging unit

Preserves are packed on an automatic rotary turntable. The electronic cam controller triggers actions at specific angular positions:

- Inserting and unfolding the cardboard box on the automatic rotary turntable
- Filling the preserves into the cardboard boxes
- Closing the cardboard boxes
- Transfer of the cardboard boxes to a conveyor

2.3 Configuration of an electronic cam control with FM 452

Components of the electronic cam control

The following figure shows the components of an electronic cam control. These are described briefly below.

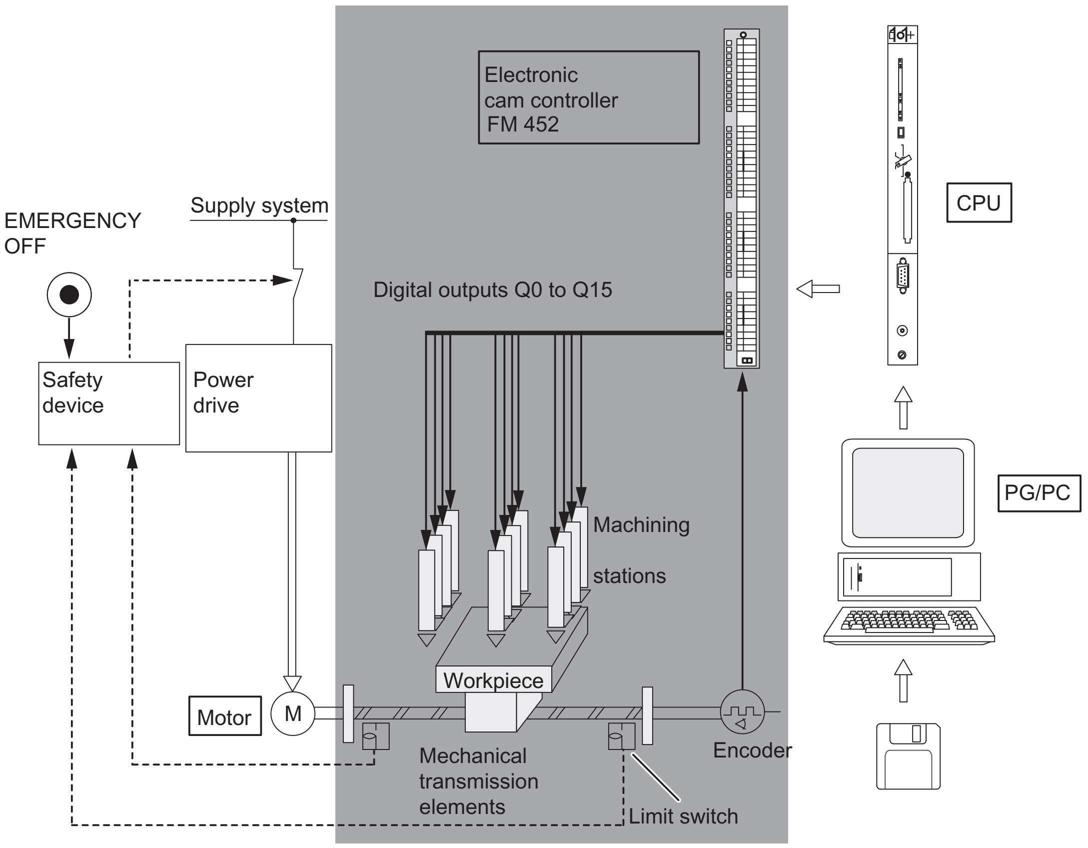


Figure 2-3 Electronic cam control

Power drive and safety system

The motor is controlled by the power drive. The power drive may consist, for example, of a protective circuit that is controlled by an FM 451 positioning module.

The power drive shuts off the motor if the safety system responds (EMERGENCY STOP or limit switch).

Motor

The motor drives the axis, controlled by the power drive.

FM 452 electronic cam controller

The electronic cam controller determines the actual position value of the axis based on an encoder signal. It evaluates the encoder signals (for example, by counting the pulses) that are proportional to the distance traveled. Based on the actual position value, it sets or resets the digital outputs ("cams"). The processing stations are controlled by signals at the digital outputs.

Encoder

The encoder returns position and direction data.

CPU

The CPU executes the user program. The user program and the module exchange data and signals by means of function calls.

PG/PC

The electronic cam controller is programmed and assigned its parameters using a PG or PC.

- Programming: You program FM 452 using the programming interface or the parameter DB.
- Programming: You program the FM 452 with functions that you can integrate directly in the user program.
- Testing and commissioning: You test the FM 452 using the programming interface with which you also finally put the system into operation.

Cam control basics 3

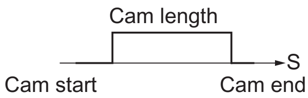
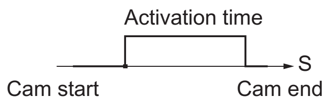
3.1 Properties of the cam types

Cam types

You can assign each cam for operation as a position-based cam or time-based cam.

The following table shows a comparison of the properties of both cam types.

Table 3- 1 Definition and switching of the two cam types

	Position-based cam	Tim-based cam
Representation		
Parameter assignment	The following parameters are assigned: • Cam start • Cam end • Effective direction • Lead time	The following parameters are assigned: • Cam start • Activation time • Effective direction • Lead time
Effective direction	Two effective directions are supported: • positive: The cam is activated at the cam start, if the axis is moving in direction of increasing actual values. • negative: The cam is activated at the cam end, if the axis is moving in direction of decreasing actual values. You can activate both effective directions in parallel.	Two effective directions are supported: • positive: The cam is activated at the cam start, if the axis is moving in direction of increasing actual values. • negative: The cam is activated at the cam start, if the axis is moving in direction of decreasing actual values. You can activate both effective directions in parallel.

	Position-based cam	Tim-based cam
Enabling	The cam is activated: • at the cam start, if the axis is moving in positive direction and positive effective direction is set. • at the cam end, if the axis is moving in negative direction and negative effective direction is set. • when the actual value lies within the cam range.	The cam is activated: • at the cam start when the direction of movement of the axis matches the effective direction. • at the cam start, if the axis is moving in negative direction, and a negative effective direction is set. After it has been activated, the full cam activation time expires even if the direction of movement of the axis changes after the cam is activated. The cam is **not** retriggered if its start position is passed once again within the cam activation time.
Deactivating	The cam is deactivated if: • the assigned distance has been traveled • axis motion in reversed effective direction is detected, and hysteresis is not assigned. • axis motion in reversed effective direction is detected, and the hysteresis is exited • the actual value no longer lies within the cam range, e.g., "Set actual value"/"Set actual value on-the-fly".	The cam is deactivated on expiration of the assigned time, i.e., the activation time is not restarted.
Path length	The path length of the cam is defined by its start and end position. The cam start and end positions belong to the active section of the cam.	The path length of the cam is determined by the axis velocity within the cam activation time.
On period	The on period of the cam is determined by the speed at which the axis travels across the path length of the cam.	The on period of the cam is assigned with the activation time.

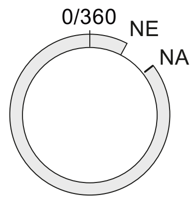
Direction detection

The direction of the axis motion is determined as follows:

- At each pulse of the incremental encoder.
- With each error-free frame of an SSI encoder.

Inverse cam

An inverse cam comes about if the cam start is greater than the cam end. The following table shows the effect of an inverse cam with a linear and rotary axis.

Inverse cam with a linear axis	Inverse cam with a rotary axis
	
The cam start (NA) is greater than the cam end (NE).	The cam start (NA) is more positive than the cam end (NE).
With both types of axes there must always be an interval of at least 4 pulses between the cam start and the cam end.	

3.2 Tracks and track result

3.2.1 Standard tracks

Cam tracks

The 32 tracks can be used to control up to 32 different switching actions. You can evaluate the tracks based on the checkback signals.

Each one of the first 16 tracks (0 to 15) is assigned a digital output (Q0 to Q15) of FM 452 which can be used, for example, for direct control of a contactor relay.

Track result

The system provides up to 128 cams which can be assigned to any track.

Each track can be assigned several cams. The track result is a logic OR operation derived from all cam values of this track (see chapter "Interfaces of the cam controller (Page 23)").

Example of a track result

Define the following cams for track 3 when programming:

Output cam	Cam start	Cam end
1	101 µm	106 µm
2	100 µm	104 µm

This leads to the following track result:

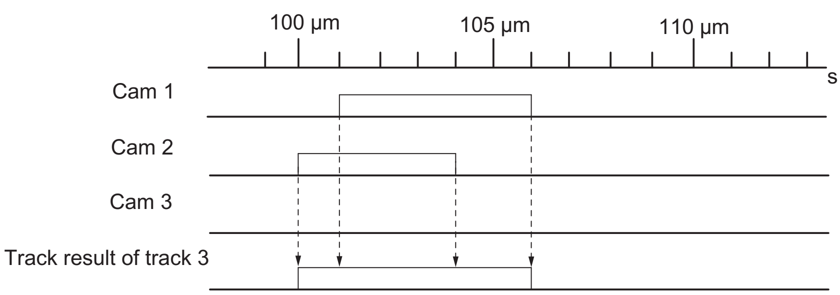


Figure 3-1 Calculating the track result

Track enable

To transfer the results of tracks 0 to 15 as track signals to the digital outputs Q0 to Q15 of FM 452, enable the tracks used.

External enable of tracks 3 to 10

You can set an external enable for tracks 3 to 10 in the machine parameters. The track signals 3 to 10 are then linked by AND logic with digital inputs I3 to I10 before they can switch the respective digital outputs Q3 to Q10 of FM 452.

A digital output (Q3 to Q10) is switched when the following conditions are met:

- The corresponding track is enabled.
- At least one cam on this track is active (track result = 1).
- The corresponding digital input I3 to I10 was set by an external event.

Setting the track signals

The track signals 0 to 15 (according to digital outputs Q0 to Q15) can be set by the cam control system, or by the CPU.

3.2.2 Special tracks

Definition

You can program tracks 0 to 2 for operation as special tracks:

- Track 0 or 1: Counter cam track
- Track 2: Brake cam track

Requirements

The following requirements must be met to allow the use of the special tracks:

- Cams are programmed for the track
- Cam processing is active
- The relevant track is enabled
- The track is selected as a special track

Counter cam track

A counter cam track counts the status transitions of the track results on this track.

Define a counter value, and then start the counter function.

The counter value of the relevant track decrements by the count of 1 at each positive edge of the track result signal.

The track flag bit = 0 as long as the value of the counter cam track is not equal to zero.

When the counter value = 0, the controller sets the track flag bit and, if programmed accordingly, the track signal (see chapter "Interfaces of the cam controller (Page 23)").

It resets the track flag bit, and sets the default value of the counter at the next negative edge of the track result signal (all cams on this track are disabled).

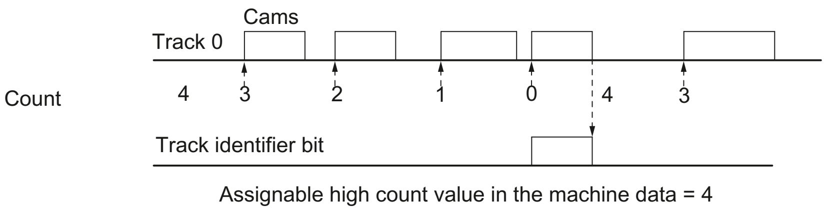


Figure 3-2 Setting a counter cam track

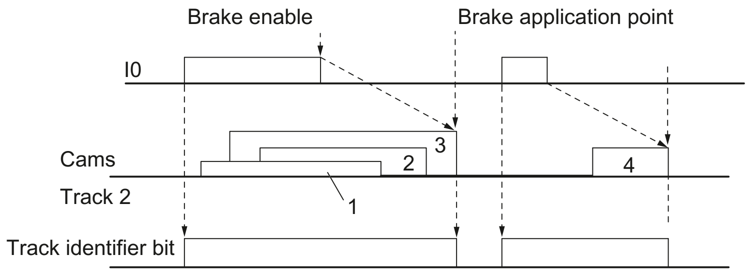
Brake cam track

In order to use track 2 as a brake cam track, interconnect digital input I0.

A positive edge of the I0 signal sets the track flag bit.

The track flag bit is reset again when:

- there is no longer a "1" signal at I0 **and** afterwards
- the controller has detected a negative edge at the track 2 result signal.



1 ...4 denote 4 output cams that act on the brake cam track

Figure 3-3 Setting a brake cam track

In the example, the track flag bit is reset by a negative edge at cam 3 or 4.

3.3 Hysteresis

Definition

Mechanical imbalance at the axis may cause fluctuation of the actual position value. If the actual position value is offset by one edge of a cam, or within an active cam with only one effective direction, this cam's activation would be cycled on and off continuously. A hysteresis prevents this flutter.

A hysteresis setting is dependent on the actual value, and applies globally to all cams. It is enabled when a direction reversal is detected. A hysteresis will always take effect, regardless of whether or not a cam is set at the current axis position.

Rules for the hysteresis range

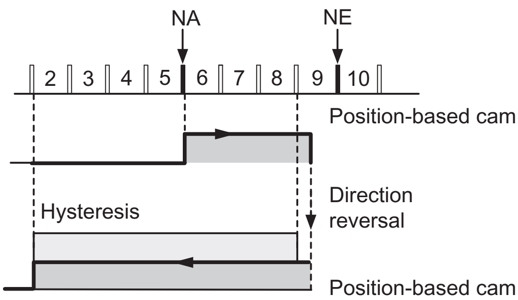
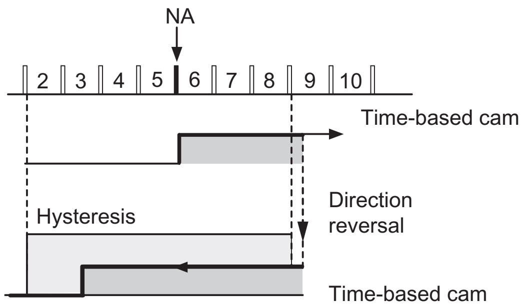
Rules applicable to the hysteresis range:

- The hysteresis will always be set when a directional reversal is detected.
- The indication of the actual value remains constant within the hysteresis.
- The direction is not redefined within the hysteresis.
- A positioning cam is neither set nor reset within the hysteresis.
- A time-based cam is not activated within the hysteresis. An activated time-based cam is deactivated on expiration of the assigned activation time (not only on reaching the hysteresis limit).
- When the value is out of the hysteresis range, the FM 452 sets:
  - the actual position value
  - the current direction of motion of the axis
  - the current states of all cams
- The hysteresis range applies to all cams.

Directional reversal of a cam with hysteresis

The table illustrates the reaction to directional reversal of a cam. A distinction must be made between the reaction of position-based and time-based cams. The effective direction of the cam is **positive**.

Table 3- 2 Reversal of the cam direction

Position-based cam	Time-based cam
	
The hysteresis becomes active after a reversal of direction is detected. The cam is deactivated immediately when the hysteresis range is exited.	The cam always remains active for the duration of the assigned activation time.
Output cam: Hysteresis: NA = Cam start NE = Cam end	

3.4 Dynamic adjustment

Task

The dynamic adjustment is used to compensate delay times of the connected control elements.

Lead time

You can program a delay time and assign it as lead time to each cam. You can assign one lead time to each cam. The lead time applies to the cam start and end position.

Actuation distance

The actuation distance of a cam is calculated continuously based on the current velocity and lead time. The entire cam is shifted in direction of the actual value by this value. The programmed range is the "static range," and the range calculated based on the lead time represents the "dynamic range."

Actuation distance = lead time x actual velocity of the axis

Calculation of the lead distances of all cams is carried out within ¼ of the longest selected lead time in the FM 452.

An extremely high lead time of a cam reduces the dynamic performance of cam processing.

3.5 Interfaces of the cam controller

Overview

The diagram below shows the most important interfaces to illustrate the relationship between data, inputs and outputs.

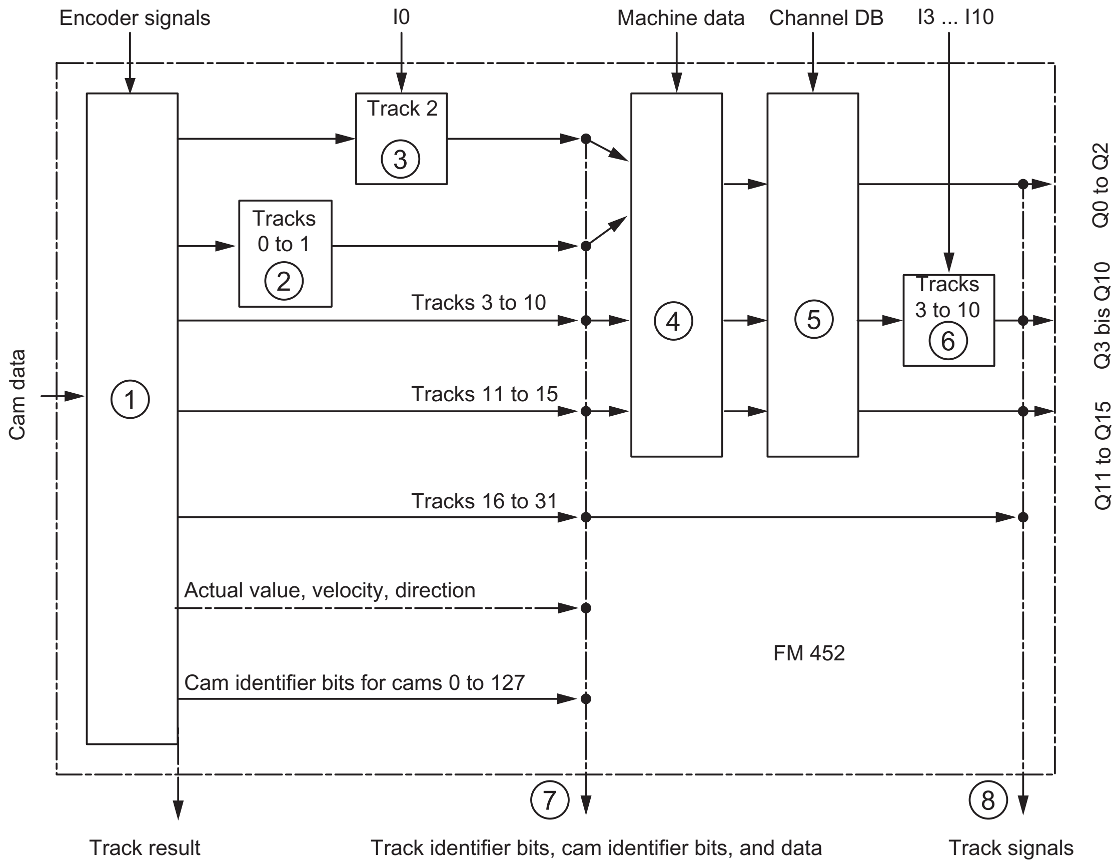


Figure 3-4 Interfaces of FM 452

For information on the diagram, refer to the table below.

No.	Description
①	The cam processing functions of FM 452 calculate the cam identifier bits based on the switching conditions and actual values. The module also determines the track results according to the assignment of the cams to the tracks.
②	If you programmed track 0 or 1 as a counter cam track, the track result of the cam control system (point 1) is logically linked to the counter result in order to produce the track identifier bit. The track identifier bit is otherwise equivalent to the track result.
③	If you programmed track 2 as a brake cam track, the track result of the cam control system (point 1) is logically linked to input I0 to produce the track identifier bit. The track identifier bit is otherwise equivalent to the track result.
④	Using machine parameters, you can control whether the previously determined track identifier bits of tracks 0 to 15 of the cam controller are passed on, or whether they are set directly by the track enable signal (TRACK_EN).
⑤	You enable the track signals of tracks 0 to 15 by setting TRACK_EN, and the count function by setting CNTC0_EN / CNTC1_EN.
⑥	The track signals of tracks 3 to 10 can be ANDed with digital inputs I3 to I10 if you have enabled this option in the machine parameters (EN_IN_I3 to EN_IN_I10).
⑦	All track and cam identifier bits can be read at this location (i.e., before these are logically linked with machine and channel data) using the ACTPOS_EN or CAMOUT_EN job. At tracks 3 to 31, the track identifier bit is equivalent to the track result (point 1).
⑧	After having been logically linked with the machine/channel data, the track signals of tracks 0 to 15 are available in the return signals. The track signals of tracks 16 to 31 and the track identifier bits of point 7 are identical. The track signals of tracks 0 to 15 are also available at the digital outputs Q0 to Q15.

Installing and removing the FM 452

4

Important safety rules

Certain important rules and regulations govern the integrating of an S7-400 with FM 452 in a plant or system. These are described in the Installation Manual SIMATIC Automation System S7-400: Hardware and Installation (http://support.automation.siemens.com/WW/view/en/1117849).

Selecting slots

The electronic cam controller FM 452 can be installed in a central or expansion rack just like a signal module.

Mechanical configuration

Refer to the Installation Manual SIMATIC Automation System S7-400, Hardware and Installation (http://support.automation.siemens.com/WW/view/en/1117849) for options on the mechanical setup and how to proceed during configuration.

Installation and removal tools

You require a 4.5 mm screwdriver to install or remove the FM 452.

Installing the FM 452 electronic cam controller

1. Hook in the FM 452 at the top and swing it down.
2. Secure the FM 452 with screws (torque approximately 0.8 to 1.1 N/m.)
3. Label the FM 452 with its slot number. Use the number wheel included with the rack.

   The required numbering scheme and the procedure for defining the slot numbers are described in the Installation Manual SIMATIC Automation System S7-400: Hardware and Installation (http://support.automation.siemens.com/WW/view/en/1117849).

Removing the FM 452 electronic cam controller

1. Switch off the power control unit.
2. Release the front connector and remove it.
3. Loosen the mounting screws on the module.
4. Swing the module upwards and remove it.

Wiring FM 452 5

5.1 Before you start wiring

Important safety rule

It is essential for the safety of the system to install the elements listed below and to adapt these to your system.

- EMERGENCY OFF switch to shut off the entire system.
- EMERGENCY OFF limit switches directly affecting the power units of all drives.
- Motor circuit-breakers.

5.2 Terminal assignment of the front connector

Front connector

Connect the encoder, the digital inputs and outputs, and the auxiliary power supplies using the 48-pin front connector.

Terminal assignment of the front connector

Pin	Name	Proximity switch	Incremental encoder	Absolute encoder
1		---		
2		---		
3	1L+	Auxiliary supply 24 V DC		
4	A/DAT	---	Encoder signal A (5 V)	SSI data
5	/A / /DAT	---	Encoder signal A inverse (5 V)	SSI data inverse
6	B/CLI [1]	---	Encoder signal B (5 V)	SSI shift clock input [1]
7	/B / /CLI [1]	---	Encoder signal B inverse (5 V)	SSI shift clock input inverse [1]
8	N	---	Zero mark signal (5 V)	---
9	/N	---	Zero mark signal inverse (5 V)	---
10	CLS [2]	---	---	SSI shift clock output
11	/CLS [2]	---	---	SSI shift clock output inverse
12	A*	Encoder signal A (24 V)		---
13	B*	---	Encoder signal B (24 V)	---
14	N*	---	Zero mark signal (24 V)	---
15	Q0	Digital output 0		
16	Q1	Digital output 1		
17	Q2	Digital output 2		
18	Q3	Digital output 3		
19	Q4	Digital output 4		
20	Q5	Digital output 5		
21	Q6	Digital output 6		
22	Q7	Digital output 7		
23	5.2 V DC	---	Encoder supply (5.2 V)	
24	24 V DC	Encoder supply (24 V)		
25	M [3]	Encoder ground		
26	2L+	Auxiliary supply 24 V DC		
27	Computer Unit	---	Current sourcing/current sinking[4]	---
28	Q8	Digital output 8		
29	Q9	Digital output 9		
30	Q10	Digital output 10		

Pin	Name	Proximity switch	Incremental encoder	Absolute encoder
31	Q11	Digital output 11		
32	Q12	Digital output 12		
33	Q13	Digital output 13		
34	Q14	Digital output 14		
35	Q15	Digital output 15		
36	3L+	Auxiliary supply 24 V DC		
37	I0	Brake cam track enable		
38	I1	Length measurement/ edge detection/ set actual value on-the-fly		
39	I2	Reference point switch		
40	I3	Enable track signal 3		
41	I4	Enable track signal 4		
42	I5	Enable track signal 5		
43	I6	Enable track signal 6		
44	I7	Enable track signal 7		
45	I8	Enable track signal 8		
46	I9	Enable track signal 9		
47	I10	Enable track signal 10		
48	M [3]	Ground for auxiliary voltages		

[1] In listen-in mode

[2] In master mode

[3] Ground connections wired on module

[4] See chapter "Wiring Diagram of the Incremental Encoder Siemens 6FX 2001-4 (Up = 24 V; HTL) (Page 174)".

Auxiliary voltage for sensors and DA (1L+, 2L+, 3L+)

The 24 V DC auxiliary voltage of the encoders and digital outputs is monitored:

- for wire-break of the 24 V feed line
- for power failure

The 24 V DC auxiliary supply is converted internally to 5.2 V DC. This means that 24 V DC (terminal 24) and 5.2 V DC (terminal 23) are available at the front connector for the different types of encoders.

The general technical data and requirements of the DC load power supplies are described in the Manual SIMATIC S7-400 Automation System Module Data (http://support.automation.siemens.com/WW/view/en/1117740).

11 digital inputs (I0 to I10)

You can connect bounce-free switches (24 V current sourcing) or non-contact sensors (2 or 3-wire proximity switches) via 11 digital inputs.

The digital inputs are not monitored for short circuits or wire break and have a non-isolated connection to the module chassis.

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

16 digital outputs (Q0 to Q15)

The state (on/off) of tracks 0 to 15 is output at 16 digital outputs. The digital outputs have a non-isolated connection to the module chassis.

Loads supported:

- Operating voltage 24 V
- Current load 0.5 A / short-circuit proof

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

5.3 Wiring front connectors

Connecting cables

- The cables for digital I/O must be shielded if they exceed a certain lengths:
  - Digital inputs: As of cable length > 32 m
  - Digital outputs: As of cable length > 100 m
- The encoder cables must be shielded.
- The shields of the encoder cables must be terminated at the shielding / ground conductor busbar, and on the IO connectors.
- Always use twisted-pair cables for the A, /A, B, /B and N, /N signals of the incremental encoder.
- Use flexible connecting cables with a conductor cross-section of 0.25 to 1.5 mm^2
- Ferrules are not required. However, should you prefer to use them, you can crimp and wire two wires with a conductor cross-section of 0.25 to 0.75 mm^2 using a single ferrule without insulation collar (DIN 46228 (design A, short version).

Note

If you connect measuring probes or proximity switches, you must use shielded cables to achieve optimum noise immunity.

Wiring information for 24 V DC

Wire the 24 V DC auxiliary voltage for the encoders and digital outputs to pins 36, 26, and 3.

When wiring, note that the terminals 1L+ to 3L+ must be interconnected so that the module will operate error-free. You can use up to three voltage supplies.

Make sure that all voltage supplies have a common ground potential.

A diagnostics event is displayed if the auxiliary voltage is missing.

⚠ CAUTION

The module can be damaged.

If you reverse the polarity of the encoder supply, the module will be destroyed and must be replaced!

Verify the correct polarity of the 24 V DC supply "auxiliary voltage 1L+, 2 L+, 3L+, and ground M).

Non-isolation

The ground of the auxiliary voltages is interconnect with CPU ground potential; i.e., you have to wire pin 48 (M) to CPU ground using a low-impedance connection.

For an external encoder supply, you must also connect the ground of the external encoder supply to the CPU ground using a low-resistance connection.

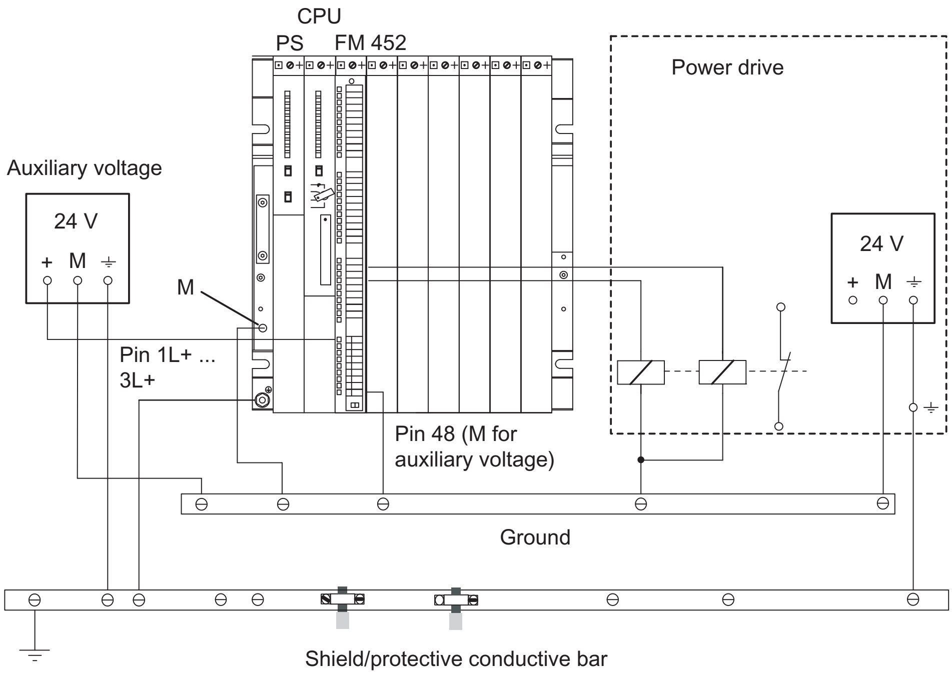


Figure 5-1 Diagram of the potential interconnection

Tools required

3.5-mm screwdriver or power screwdriver

Wiring

CAUTION
Personal injury and damage to equipment on account of unshielded voltage. If you wire the FM 452 front connector while the system is in live state, you will risk injury from electric shock! Always switch off power before you wire the FM 452! If no EMERGENCY OFF switch is installed, damage may be caused by connected aggregates. Install an EMERGENCY OFF switch to be able to shut down the connected drives while operating the FM 452 via the *parameter assignment interface*.

Proceed as follows when wiring the front connector:

1. Remove the cover from the front connector.
2. Strip the wires (length 6 mm).
3. Are you using conductor end sleeves?

   If yes: Squeeze the conductor end sleeves to the conductors.
4. Feed the enclosed strain relief clamp into the front connector.
5. Start wiring at the bottom. Tighten any unused screw terminals on a front connector equipped with such (tightening torque 0.6 to 0.8 N/m).
6. Tighten the strain relief clamp for the cable chain.
7. Close the front connector.
8. Label the terminals using the enclosed labeling strip.

Further information

A detailed description of the wiring of a front connector can be found in the Installation Manual SIMATIC Automation System S7-400: Hardware and Installation (http://support.automation.siemens.com/WW/view/en/1117849).

6 Installing software

Introduction

You assign the parameters for the FM 452 using the *parameter assignment interface.* This interface is provided both for FM 452 and FM 352. A description of the *parameter assignment interface* is available in the *Online Help*.

Requirements

Before starting to assign parameters for the FM 452 Electronic Cam Controller, note the requirements in the readme.rtf file, in particular, the required version of STEP 7. The readme.rtf file is available on the included CD.

Installation

The software package is available on the included CD. How to install the software:

1. Place the CD into the drive of your PG/PC.
2. In the dialog, select the CD drive, and in the **Setup** directory, select the **Setup.exe** file and start installation.
3. Follow the setup instructions.

   Result: Setup installs the software in the folders listed below:

   – **SIEMENS\STEP7\S7LIBS\FMx52LIB**: FCs and UDTs

   – **SIEMENS\STEP7\S7FCAM**: Parameter assignment interface, Readme, Online Help

   – **SIEMENS\STEP7\EXAMPLES\zEn19_01 and zEn19_02**: Examples for FM 452 and FM 352

   – **SIEMENS\STEP7\MANUAL**: Manual

   Note

   If you installed *STEP 7* in a folder other than SIEMENS\STEP7, this path will be entered.

Configuring and assigning parameters

For more information, refer to the section entitled "Commissioning the FM 452 (Page 61)".

7 Programming FM 452

7.1 Basics of Programming an FM 452

Task

You can assign parameters, control, and commission the FM 452 module from a user program. To exchange data between the user program and module, you use the functions (FCs) and data blocks (DBs) described below.

Preparatory steps

- Open the FMx52LIB block library in SIMATIC Manager. Copy the required functions (FCs) and block templates (UDTs) to the block folder 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.
  - CAM_INIT (FC 0):
    Required to initialize the channel DB after a module startup
  - CAM_CTRL (FC 1):
    Required for data exchange with the module

    Note

    You can also use the FB CAM_CTRL provided for the FM 352.

  - CAM_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

    ---

    Note

    You can also use the FB CAM_DIAG provided for the FM 352.

  - CAM_MSRM (FC 3):
    Required to immediately read the results of a length measurement or edge detection after a hardware interrupt
  - CAM_CHANTYPE (UDT1):
    Required to generate a channel DB, which is used by the FCs CAM_INIT, CAM_CTRL, and CAM_MSRM
  - CAM_DIAGTYPE (UDT2):
    Required to generate a diagnostic DB, which is used by FC CAM_DIAG.
  - CAM_P016TYPE (UDT3):
    Required to generate a parameter DB with machine data and data for 16 cams, which is used by FC CAM_CTRL to write or read machine or cam data.

- CAM_P032TYPE (UDT4):
  Same as CAM_P016TYPE, but for 32 cams.
- CAM_P064TYPE (UDT5):
  Same as CAM_P016TYPE, but for 64 cams.
- CAM_P128TYPE (UDT6):
  Same as CAM_P016TYPE, but for 128 cams.

* Create data blocks using the UDTs in the block folder of your S7 program. You require a separate set of data blocks for each module used.
* Enter the module address in the channel DB and, if necessary, also in the 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.

* If your PG/PC is connected to a CPU, you can now download the FCs and DBs to the CPU.

7.2 FC CAM_INIT (FC 0)

Tasks

FC CAM_INIT initializes the following data in the channel DB:

- The control signals
- The return signals
- The trigger, done and error bits of the jobs
- The function switches and their done and error bits
- Job management, and the internal buffers for FC CAM_CTRL and FC CAM_MSRM

Call-up

The function must executed after a startup (power on) of the module or CPU. You should therefore install it, for example, in the warm restart OB (OB100) and the removal/insertion OB (OB83), or call it in the initialization phase of your user program. This ensures that your user program does not access obsolete data after a CPU or module restart.

Call parameters

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

Return values

This function does not return a return value.

7.3 FC CAM_CTRL (FC 1)

Tasks

You can call FC CAM_CTRL to read operating data from the module, to initialize the module and control it at runtime. For these tasks, you use the control signals, checkback signals, and write and read jobs.

On each call, the function performs the following actions:

- Read checkback signals:

  FC CAM_CTRL reads all checkback signals from the module and enters them in the channel DB. The control signals and jobs are not executed until this task is completed, and thus the checkback signals reflect the module status prior to the block call.
- Write control signals:

  The control signals written to the channel DB are transferred to the module. Enabling of cam processing, however, is delayed as long as the trigger for a "Set reference point" job or "Write cam data" job is set. Activation (or reactivation) of cam processing is delayed by this time.
- Execute job:

  The next job is executed based on the trigger bits for jobs entered in the channel DB.

Call

This function must be called cyclically.

Before you call the function, enter all the data in the channel DB that are required to execute the required functions.

Data used

- Channel DB:

  The module address must be entered in the channel DB.
- Parameter DB:

  If you want to write or read machine or cam data using jobs, you require a parameter DB whose number must be entered in the channel DB. The size of the parameter DB must be adequate for the number of cams.

Parameters

Parameter	Declaration	Data type	Description
DB_NO	INPUT	INT	Number of the channel DB
RETVAL	OUTPUT	INT	Return value

Jobs

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

To start a job, set the corresponding trigger bit in the channel DB and provide the relevant data for write jobs. Next, call FC CAM_CTRL to execute the job.

A read job is executed immediately. Due to the required confirmation required from the module, a write job requires at least three calls (or OB cycles).

You can transfer several jobs simultaneously and alongside with control signals. With the exception of the job for writing the function switches, the jobs are executed in the order of the trigger bits specified in the channel DB. Once a job has been completed, the trigger bit is canceled. The next time the block is called, the subsequent job is identified and executed.

In addition to a trigger bit, a done bit and an error bit are provided for all jobs. Instead of the ending _EN (for "enable"), the name of those jobs have the ending _D (for "done") or _ERR (for "error"). Done and error bits of the job should be set to 0 after they were evaluated, or before this job is started.

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 switches

The function switches activate and deactivate module states. A job for writing the function switches is only executed if changes were made to a switch setting. It is always executed between the "Set reference point" (REFPT_EN) and "Set actual value" (AVAL_EN) jobs. The setting of the function switch is latched after the job has been executed.

Length measurements and edge detection must not be activated concurrently. FC CAM_CTRL makes sure that only one of the function switches can be in active state. However, the length measurement function is activated if you do enable both function switches at the same time ($0 \rightarrow 1$).

Function switches and jobs can be used in the same call of FC CAM_CTRL.

Similar to the jobs, done bits with the ending _D and error bits with the ending _ERR are provided for the function switches.

To be able to evaluate the done and error bits, you should set these bits to 0 when changing a function switch.

Startup

Call FC CAM_INIT at the startup of the module or CPU. Among other things, this also resets resets the function switches.

FC CAM_CTRL acknowledges the module startup. During this time, RET_VAL and JOBBUSY = 1.

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	I	Number of the channel DB
RET_VAL	INT	O	Return value

Return values

The function provides the following return values:

RET_VAL	BR	Description
1	1	At least 1 job active
0	1	No job active, no error
-1	0	Error: Data error (DAT_ERR) or Communication error (JOB_ERR)

Job status

You can read the status of the job execution using the RET_VAL return value and the JOBBUSY activity bit in the channel DB. You can determine the status of a single job by evaluating its trigger, done, and error bits.

	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 aborted	-1	0	0	0	1

Reaction to errors

The module returns the message DATA_ERR = 1 if the data written by a job is faulty. If an error occurs during communication with the module for a write or read job, the cause of the error is entered in the JOB_ERR parameter in the channel DB.

- Error in a write job:

  If an error occurs in a job, the trigger bit is canceled and the error bit (_ERR) and done bit (_D) are set. The trigger bit is also canceled and the error bit (_ERR) is set for all write jobs still pending.

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

- Error in a read job:

  If an error occurs in a job, the trigger bit is canceled and the 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 parameters (see chapter "Diagnosis (Page 147)" and "Data and Structure of the Diagnostic DB (Page 186)").

7.4 FC CAM_DIAG (FC 2)

Tasks

Use FC CAM_DIAG to read the data of the diagnostic buffer of the module and make these available for visualization on an operator control and monitoring system or for a programmed evaluation.

Call

This function must be called cyclically. It is not allowed to initiate a second call in an interrupt OB. At least two calls (cycles) are required to complete execution of this function..

The function reads the diagnostics buffer if checkback signal DIAG = 1 reports a new entry in the buffer. The module sets DIAG to 0 after the diagnostics buffer was read.

Data used

The module address must be entered in the diagnostic DB. The newest entry in the diagnostics buffer is written in the DIAG[1] structure and the oldest entry is written in the DIAG[4] structure.

Jobs

You can read the diagnostics buffer regardless of any new entry by setting the DIAGRD_EN trigger bit. After the diagnostics buffer was read, the trigger bit is set to 0.

Parameters

Parameter	Declaration	Data type	Description
DB_NO	INPUT	INT	Number of the diagnostic DB
RETVAL	OUTPUT	INT	Return value

Startup

The function does not perform a startup routine.

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	I	Number of the diagnostic DB
RET_VAL	INT	O	Return value

Return values

The function provides the following return values:

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

Reaction to errors

The cause of a job error can be found in the JOB_ERR parameter of the diagnostic DB (see chapter "Diagnosis (Page 147)" and "Data and Structure of the Diagnostic DB (Page 186)").

7.5 FC CAM_MSRM (FC 3)

Tasks

Use FC CAM_MSRM only if you want to evaluate length measurement or edge detection data directly in the process interrupt OB.

Call

The function is called in a process interrupt OB (e.g. OB 40).

Data used

The module address must be entered in the channel DB.

Startup

The function does not perform a startup routine.

Call parameters

Name	Data type	P-type	Meaning
DB_NO	INT	I	Number of the channel DB
RET_VAL	INT	O	Return value

Return values

The function returns the following values:

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

Measurement results and status information

The measurement results and status information are available in the channel DB:

Table 7- 1 Alarm measurement data the channel DB

Address	Name	Type	Start value	Comment
112.0	BEG_VAL	DINT	L#0	Start value
116.0	END_VAL	DINT	L#0	End value
120.0	LEN_VAL	DINT	L#0	Length
56.0	JOB_ERR_M	INT	0	Communication error
58.0	JOBBUSY_M	BOOL	FALSE	Job active

Reaction to errors

The cause of a job error can be read from the JOB_ERR_M parameter of the channel DB (see chapter "Diagnosis (Page 147)" and "Data and Structure of the Diagnostic DB (Page 186)").

7.6 Data blocks

7.6.1 Templates for data blocks

Templates for data blocks

The included library (FMx52LIB) contains a block template (UDT) for each data block. Based on this UDT, you can create data blocks with user-specific numbers and names.

Optimizing the UDT

To save memory, you can delete unused data areas at the end of the UDT CAM_CHANTYPE. Save the modified UDT under a different name.

You can then generate a channel DB based on this UDT you optimized for your application.

Functions which access deleted data areas can no longer be used.

The included UDT for the machine and cam data are already tuned to the possible numbers of cams. They can be optimized in steps of 16 cams.

7.6.2 Channel DB

Task

The channel DB forms the data interface between the user program and the FM 452 electronic cam controller. It contains and accepts all data required for controlling and operating the module.

Configuration

The channel DB is divided into various areas:

Areas of the channel DB
Address* / version switch
Control signals
Return 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 enter the address in the programming interface

7.6.3 Diagnostics DB

Task

The diagnostics DB provides the data storage for FC CAM_DIAG, and contains the module's diagnostics buffer prepared by this function.

Configuration

Structure of the diagnostics DB
Module address
Internal data
Job status
Trigger bit
Prepared diagnostics buffer

7.6.4 Parameter DB

Task

All machine and cam data are saved to the parameter DB. These parameters can be modified by the user program, or by an operating and monitoring system. The modified data can be imported to the programming interface and visualized there. You can export data visualized on the programming interface to a parameter DB.

A module may contain several parameter sets (for example, for various recipes) that you can select program-controlled.

Configuration

Structure of the parameter DB
CAM_P016TYPE (UDT3) Machine data Cam data of cams 0 to 15
CAM_P032TYPE (UDT4) Machine data Cam data of cams 0 to 31
CAM_P064TYPE (UDT5) Machine data Cam data of cams 0 to 63
CAM_P0128TYPE (UDT6) Machine data Cam data of cams 0 to 127

7.7 Interrupts

7.7.1 Interrupt processing

Procedure

The FM 452 can trigger hardware and diagnostic interrupts. Process those interrupts in an interrupt OB. If an interrupt is generated and the corresponding OB is not loaded, the CPU changes to STOP (refer to the manual *Programming with STEP* 7).

You can enable interrupt processing at the following levels:

Enabling general interrupts for the entire module:

- Select the module in HW Config.
- Using the menu command Edit > Object Properties > Basic Parameters, enable diagnostic and/or hardware interrupts.
- Select the OB number for the hardware interrupt using Edit > Object Properties > Addresses.
- Save and compile the hardware configuration.
- Download the hardware configuration to the CPU.

Enabling events for hardware interrupts in the machine data.

Setting parameters for hardware interrupts in the cam data for cams 0 to 7.

7.8 Evaluation of a hardware interrupt

Hardware interrupt data

If FM 452 generates a hardware interrupt, the following information is available at variable OB40_POINT_ADDR (or at the corresponding variable of a different hardware interrupt OB):

Content of double word OB40_POINT_ADDR

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0	0
1	0	0	Start of measure ment	0	0	Cams	End of measure ment	0
2	Cam 7 on	Cam 7 off	Cam 6 on	Cam 6 off	Cam 5 on	Cam 5 off	Cam 4 on	Cam 4 off
3	Cam 3 on	Cam 3 off	Cam 2 on	Cam 2 off	Cam 1 on	Cam 1 off	Cam 0 on	Cam 0 off

Byte 1 identifies the cause of the interrupt:

- Cam: evaluate byte 2 and byte 3 according to the table.
- Measurement start/measurement end: use the CAM_MSRM function to read the current measured values from the module.

Lost hardware interrupts

If the processing of a hardware interrupt is not yet completed in the hardware interrupt OB, the module registers all subsequent hardware interrupt events. If an event occurs again before the hardware interrupt could be triggered, the module triggers the "hardware interrupt lost" diagnostic interrupt.

7.9 Evaluating a diagnostics interrupt

Diagnostics interrupt data

Following a diagnostic interrupt, the diagnostic information is available in the local data of the OB82 and can be used for fast analysis. Call FC CAM_DIAG to find out the exact cause of error by reading the diagnostics buffer.

The local data of the diagnostics interrupt OB that are supported are listed below.

Tag	Data type	Description
OB82_MDL_DEFECT	BOOL	Module error
OB82_INT_FAULT	BOOL	Internal error
OB82_EXT_FAULT	BOOL	External error
OB82_PNT_INFO	BOOL	Channel error
OB82_EXT_VOLTAGE	BOOL	Missing external auxiliary supply
OB82_FLD_CONNCTR	BOOL	Front connector missing
OB82_WTCH_DOG_FLT	BOOL	Watchdog timeout
OB82_INT_PS_FLT	BOOL	Internal power supply of the module failed
OB82_HW_INTR_FLT	BOOL	Hardware interrupt lost

7.10 Technical data

Overview

The following table provides an overview of the technical specifications of the functions.

No.	Block name	Version	Allocation in load memory (bytes)	Allocation in work memory (bytes)	Allocation in the local data area (bytes)	MC7 code / data (bytes)	System functions called
FC 0	FC CAM_INIT	1.0	192	138	2	102	
FC 1	FC CAM_CTRL	1.0	5232	4754	32	4718	SFC 58: WR_REC, SFC 59: RD_REC
FC 2	FC CAM_DIAG	1.0	1782	1638	42	1602	SFC 59: RD_REC
FC 3	FC CAM_MSRM	1.0	296	226	16	190	SFC 59: RD_REC
	Channel DB	-	986	804	-	372	
	Parameter DB 16 Parameter DB 32 Parameter DB 64 Parameter DB 128	- - - -	616 808 1192 1960	336 528 912 1680	- - - -	300 492 876 1644	
	Diagnostic DB	-	460	338	-	302	

Module cycle

The module updates the checkback data - except in the pulse measuring system - at intervals of 4 ms.

In the pulse measuring system, the data for the actual position value and track signals is available after 0.5 ms.

7.11 High-speed access to module data

Application

Special applications or alarm levels require particularly fast access to checkback and control signals. You can access this data directly via the module I/O.

To coordinate each module startup (e.g., after inserting a module, CPU STOP → RUN), you must call FC CAM_CTRL until completion of the startup is indicated by RET_VAL = 0.

Note

When accessing to FM 452 data directly, you may only use the non-internal data and method described in this section. Otherwise, the user program will encounter difficulties accessing the module.

Reading checkback signals by means of direct access

The byte addresses are specified relative to the output address of the module. The bit names correspond to the names in the channel DB.

In STL you access the data using the PIB (read 1 byte) and PID (read 4 bytes) commands.

Address	Bit number							
	7	6	5	4	3	2	1	0
Byte 0	PARA	Internal	Internal	DATA_ERR	Internal	DIAG	Internal	Internal
Byte 1	0	0	0	CAM_ACT	0	0	0	0
Byte 2	Internal							
Byte 3	0	0	FVAL_DONE	HYS	GO_P	GO_M	MSR_DONE	SYNC
Byte 4	ACT_POS							
Byte 5								
Byte 6								
Byte 7								
Byte 8	TRACK_OUT							
Byte 9								
Byte 10								
Byte 11								

Writing control signals via direct access

The byte addresses are specified relative to the input address of the module. The bit names correspond to the names in the channel DB.

In STL you access the data using the PQB (write 1 byte) and PQW (write 2 bytes) commands.

Address	Bit number							
	7	6	5	4	3	2	1	0
Byte 0	Internal							
Byte 1	0	CNTC1_EN	CNTC0_EN	CAM_EN	DIR_P	DIR_M	0	0
Byte 2	TRACK_EN							
Byte 3								

Example: Actual position value (ACT_POS)

The start address of the module is 512

STL	
L PID 516	Reading the actual position value (ACT_POS) by means of direct access: Module start address + 4

7.12 Parameter transfer routes

Transfer routes

The term parameter refers to the following machine and cam data.

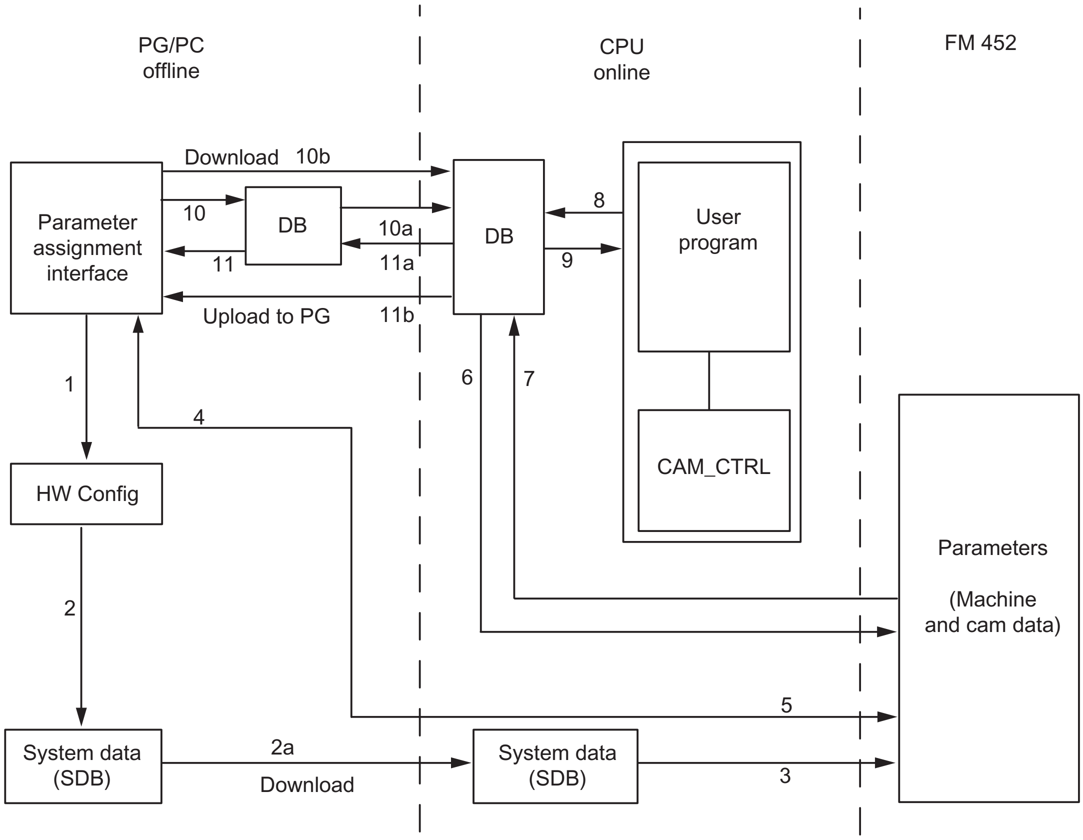


Figure 7-1 Parameter transmission paths

1	Save the parameters in the parameter assignment interface.
2	Saving and compiling the HW configuration, and download to the CPU.
3	The CPU writes the parameters to the module during system parameter assignment.
4	Upload the module parameters to the PG by selecting the "PLC > Upload to PG" command.
5	Download parameters from FM Config to the module with the "Download" command.
6	Write parameters to the module using jobs in the user program.
7	Read parameters from the module using jobs in the user program.
8	Save parameters from the user program to the online DB.
9	Write parameters from the online DB to the user program.
10	Export parameters from the parameter assignment interface to the DB (offline or online DB); an offline DB must then be downloaded to the CPU.
11	Import parameters from an online or offline DB into the programming interface.

Some use cases for the transfer of parameters:

Use case	Steps
You edit the parameters with the parameter assignment interface. The parameters should then be assigned automatically to the module during startup.	Perform steps 1, 2, and 3.
You edit parameters on the parameter assignment interface during commissioning in test mode.	Perform steps 4 and 5
The parameters edited during commissioning should be downloaded automatically during startup.	Perform steps 1, 2, and 3.
You create the parameters using the parameter assignment interface. When it starts up, the module should be assigned parameters only by the user program using data blocks.	Perform steps 10 and 6
You modify existing parameters (exclusively) using the user program.	Perform steps 7, 9, 8, and 6
You create the parameters using the parameter assignment interface. These should be available to the user program for temporary modifications.	Execute steps 1, 2, 3 for automatic parameter assignment. Execute steps 10, 7 for access with the user program.
You need a comfortable means of creating data records for recipes.	Perform step 10
You want to view the data modified by the user program on the parameter assignment interface.	Perform step 11
Parameters modified by the user program should also be loaded automatically during startup.	Perform steps 6, 11, 1, 2, and 3

Commissioning the FM 452 8

General information

Please observe the points listed in the following warnings.

> **⚠ WARNING**
>
> To avoid injury and material damage, observe the following items:
>
> - Install an EMERGENCY STOP switch in the range 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 an EMERGENCY STOP limit switch which directly affects the power units of all drives.
> - Ensure that no individuals can access system areas which contain moving parts.
> - If controlling and monitoring FM 452 both in your program and in the **Test > Commissioning** dialog box, you risk the generation of conflicts with indefinite effects. Therefore, always set the CPU to STOP when using the test dialog box, or disable the user program.

HW installation and wiring

In this first section, you install an FM 452 in your S7-400 and wire the front connector.

Step	Action	✓
1	**Installing FM 452** (see chapter "Installing and removing the FM 452 (Page 25)") Insert the module into one of the available slots.	☐
2	**Wiring FM 452** (see chapter "Wiring FM 452 (Page 27)") Wire the front connector for FM 452: • Digital inputs • Digital outputs • Encoder connection • Power supply to FM 452	 ☐ ☐ ☐ ☐
3	**Checking the safety-relevant limit switches** Check the function of the following: • Limit switches • Emergency stop equipment	 ☐ ☐
4	**Front connectors** The front connector must be engaged.	☐

Step	Action	✓
5	Check the shielding of the cables.	☐
6	**Switching on the power supply** Switch the CPU to STOP (safe state). Switch on the 24 V supply for FM 452.	 ☐ ☐

Setting up a project

Now, set up a project in *STEP 7*.

The section below describes the corresponding steps in SIMATIC Manager (without assistance from the wizard).

Step	Action	✓
1	Install the parameter assignment interface (if not already installed).	☐
2	Select **File > New** to create a new project in SIMATIC Manager.	☐
3	Add a station to your project (**Insert > Station**).	☐
4	Select the station, then double-click "Hardware" to open the "HW Config" configuration interface.	☐
5	Enter your hardware configuration in a rack, including: • Power supply module (PS) • CPU • Function module (FM)	 ☐ ☐ ☐
6	Save this hardware configuration to HW Config (**Station > Save**).	☐

Assigning parameters using the parameter assignment interface

Use the parameter assignment interface to assign the module parameters when commissioning the module. Observe the following order:

Step	Action	✓
1	Select the tier in the rack that contain the FM 452 module.	☐
2	Next, double-click the entry to open the parameter assignment interface for FM 452.	☐
3	You can change the following settings by selecting **File > Properties**: • General You can change the name and enter a comment. • Addresses You can edit the start address, and assign the address area to a process image partition. **Note the module address displayed.** • Basic parameters You can set the interrupt class and the reaction to CPU STOP.	☐

Step	Action	✓
4	In the block diagram, you can select the Axis, Encoders, Cams, Tracks and **and** Interrupt Enable dialog boxes, and set the relevant parameters.	☐
5	Save the parameter assignment with **File > Save**.	☐
6	Close the parameter assignment interface by selecting **File > Exit**.	☐
7	Save the hardware configuration in HW Config with **Station > Save and compile**.	☐
8	Go online to the CPU and download the hardware configuration to the CPU. This data is transferred to FM 452 at every STOP to RUN transition.	☐
9	Select **Test > Commissioning**.	☐

Test and commissioning

You can now test your entries and changes.

Step	Action	✓
1	To test your commissioning data, select the **Test > Commission**, **Test > Service** and **Test > Error Evaluation** dialog boxes.	☐
2	You can change incorrect machine data in the **Test > Commissioning** interface. Those changes remain valid until the next STOP-RUN transition of the CPU.	☐
3	You can save the revised machine data to the CPU by repeating steps 7 to 9 of the sequence.	☐

Steps for testing axis synchronization and switching characteristics

Use the following tests to validate the FM 452 parameter assignment.

<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". Enter the corresponding value (see chapter "Execute "Set reference point" (Page 112)").
or
– Set the "Retrigger reference point" function switch (see chapter "Execute "Retrigger reference point" (Page 122)").</td><td>• Absolute encoder
– FM 452 is always synchronized immediately after parameter assignment.
– Adjust the absolute encoder (see chapter "Determining the correct absolute encoder adjustment (Page 82)"). You might have to determine the precise value using the "set reference point" function.</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>
</table>

Step	Action	✓
2	Check the switching characteristics of the assigned output cams and tracks. • Enable testing. • Execute the "set reference point" command. • Activate cam processing. • Enable the track signals. • Rotate the encoder accordingly **or** • Set the **simulation** function switch.	☐
3	Test the other settings according to your applications • Set reference point • Set actual value	☐

Getting prepared for programming

You still need to create the blocks required in your project.

Step	Action	✓
1	Select the FMX52LIB library in SIMATIC Manager (**File > Open > Libraries**).	☐
2	Copy the FC0 and FC1 functions and the channel DB template UDT1 from the library to the blocks container.	☐
3	Create a channel DB for each module (based on the UDT1 template).	☐
4	If you want to use a programmed diagnostics evaluation, copy FC2 and UDT2 and create a diagnostic DB for each module.	☐
5	Copy FC3 if you require length measurements or edge detection with hardware interrupt.	☐
6	To write and read machine data in the user program, you require UDT3 for 16 output cams, UDT4 for 32 output cams, UDT5 for 64 output cams, or UDT6 for 128 output cams.	☐

Preparing the channel DB

Step	Action	✓
1	Open the channel DB.	☐
2	Make sure that the module address is entered in the MOD_ADDR parameter (refer to the section entitled Basics of Programming an FM 452 (Page 37)).	☐
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 an FM 452 (Page 37)).	☐
3	Save the diagnostic DB (**File > Save**).	☐

Implementing the functions

Step	Action	✓
1	Integrate the required functions in your user program.	☐

Downloading blocks to the CPU

Step	Action	✓
1	Select the blocks in SIMATIC Manager and then download them with **PLC > Download to CPU.**	☐

9 Machine and cam data

9.1 Machine data and cam data

General information

This chapter is relevant if you want to write the parameters directly to the module in the user program, and without using the parameter assignment interface.

All machine and cam data are saved in the parameter DB. You must enter the number of the parameter DB in the associated channel DB in each case.

You can write to the parameter DB with "Export" and read the parameter DB with "Import" on the parameter assignment interface.

Sequence when writing machine data and cam data

Always edit machine data and cam data in the following order:

1. Write machine data
2. Enable machine data
3. Writing cam data

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

9.2 Writing and enabling machine data

Writing and enabling machine data

Machine data are used to adapt the FM 452 to the axis and encoder.

Machine data are stored in the parameter DB at addresses 3.1 to 104.0.

Initial parameter assignment

If the module does not yet contain any machine data (checkback signal PARA = 0), proceed as follows for initial parameter assignment without parameter assignment interface:

1. Enter the new values in the parameter DB.
2. Download the parameter DB to the CPU.
3. Set the following trigger bit in the channel DB:
   – Write machine data (MDWR_EN)
4. Call FC CAM_CTRL in the cyclic user program.

Changing machine data

To change existing machine data (checkback signal PARA = 1) by means of the user program, proceed as follows:

1. Enter the new values in the parameter DB.
2. Set the trigger bits at the channel DB:
   – Write machine data (MDWR_EN)
   – Enable machine data (MD_EN)
3. Call FC CAM_CTRL in the cyclic user program.

4. Check to see if the modified machine data were successfully transmitted and activated by evaluating the done bit (_D ending) and error bit (_ERR ending) assigned to each job:
   – "Write machine data" job completed (MDWR_D)
   – "Enable machine data" job completed (MD_D)
   – Error during "Write machine data" job (MDWR_ERR)
   – Error during "Enable machine data" job (MD_ERR)

   A job was completed without errors if done bit = 1 and error bit = 0 (refer to the section entitled FC CAM_CTRL (FC 1) (Page 40)).

   Set the done and error bits of a job to 0 after evaluation.

   Note

   If any parameters relevant for synchronization were modified, the synchronization settings are deleted when you enable the machine data. This operation also resets your settings, and deletes all machine data and cam data from the module.

   Parameters relevant to synchronization:
   - 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 retriggering
   - Direction adaptation
   - Scope
   - Software limit switch start and end

   ---

5. Always rewrite the cam data of the assigned cams, regardless if they have been changed or not:
   – Write cam data n, n = 1...8 (CAM1WR_EN...CAM8WR_EN).
6. Check to see if the cam data were transmitted successfully by evaluating the done bit (_D ending) and error bit (_ERR ending) assigned to each job:
   – "Write cam data n" job completed, n = 1...8 (CAM1WR_D...CAM8WR_D).
   – Error in "Write cam data n" job, n = 1...8 (CAM1WR_ERR...CAM8WR_ERR).

   A job was completed without errors if done bit = 1 and error bit = 0 (refer to the section entitled FC CAM_CTRL (FC 1) (Page 40)).

   Set the done and error bits of a job to 0 after evaluation.

9.3 Read machine data

Read machine parameters

To read current machine parameters from the module, proceed as follows:

1. Set the following trigger bit in the channel DB:
   – Read machine parameters (MDRD_EN)
2. Call FC CAM_CTRL in the cyclic user program.

   The current machine parameters is then written to the parameter DB on the CPU.

Extract from the channel DB

Address	Name	Type	Start value	Comment
35.0	MDWR_EN	BOOL	FALSE	1 = write machine parameters
35.1	MD_EN	BOOL	FALSE	1 = activate machine parameters
37.1	MDRD_EN	BOOL	FALSE	1 = read machine parameters

9.4 Writing cam data

Writing cam data

Cam data defines the type and function principle of the cams and their assignment to the tracks.

Cam data is stored in the parameter DB, starting at address 108.0. This data is grouped in packets, each consisting of 16 cams.

Cam data is active immediately after having been written.

To write cam parameters without using the programming interface, proceed as follows:

1. Enter the new values in the parameter DB.
2. Download the parameter DB to the CPU.
3. Set the trigger bits at the channel DB (CAM1WR_EN to CAM8WR_EN)
4. Call FC CAM_CTRL in the cyclic user program.

9.5 Reading cam data

Reading cam data

To read actual cam data from the module:

1. Set the following trigger bit in the channel DB:
   – Read cam data (CAM1RD_EN to CAM8RD_EN)
2. Call FC CAM_CTRL in the cyclic user program.

   This saves the actual cam data to the parameter DB on the CPU.

Extract from the channel DB

Address	Name	Type	Start value	Comment
35.3	CAM1WR_EN	BOOL	FALSE	1 = write cam 1 data (cams 0 to 15)
35.4	CAM2WR_EN	BOOL	FALSE	1 = write cam 2 data (cams 16 to 31)
35.5	CAM3WR_EN	BOOL	FALSE	1 = write cam 3 data (cams 32 to 47)
35.6	CAM4WR_EN	BOOL	FALSE	1 = write cam 4 data (cams 48 to 63)
35.7	CAM5WR_EN	BOOL	FALSE	1 = write cam 5 data (cams 64 to 79)
36.0	CAM6WR_EN	BOOL	FALSE	1 = write cam 6 data (cams 80 to 95)
36.1	CAM7WR_EN	BOOL	FALSE	1 = write cam 7 data (cams 96 to 111)
36.2	CAM8WR_EN	BOOL	FALSE	1 = write cam 8 data (cams 112 to 127)
37.2	CAM1RD_EN	BOOL	FALSE	1 = read cam 1 data (cams 0 to 15)
37.3	CAM2RD_EN	BOOL	FALSE	1 = read cam 2 data (cams 16 to 31)
37.4	CAM3RD_EN	BOOL	FALSE	1 = read cam 3 data (cams 32 to 47)
37.5	CAM4RD_EN	BOOL	FALSE	1 = read cam 4 data (cams 48 to 63)
37.6	CAM5RD_EN	BOOL	FALSE	1 = read cam 5 data (cams 64 to 79)
37.7	CAM6RD_EN	BOOL	FALSE	1 = read cam 6 data (cams 80 to 95)
38.0	CAM7RD_EN	BOOL	FALSE	1 = read cam 7 data (cams 96 to 111)
38.1	CAM8RD_EN	BOOL	FALSE	1 = read cam 8 data (cams 112 to 127)

9.6 System of units

Selecting a system of units

You can choose a specific system of units for the input and output of data in the parameter assignment interface of the cam controller (default: mm).

You can also select the following physical units:

- mm, inches, degrees, and pulse.

Note

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

If you change the system of units using the machine data, the values are **not** recalculated automatically.

If the system of units is changed from or to "pulses", the cam processing is deactivated, and the axis is no longer synchronized.

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 5 = pulses 6 = 10^{-3} degrees

Default system of units

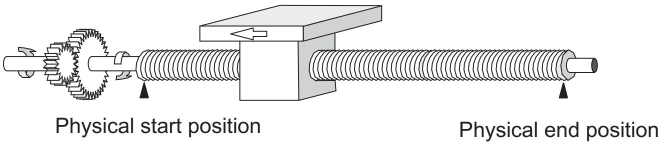
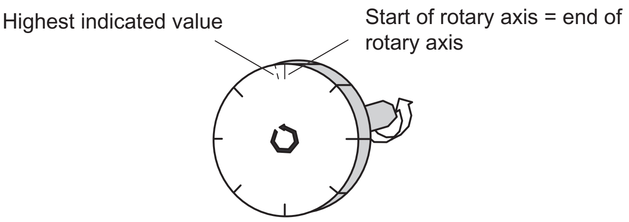
This manual always specifies limits using **mm as the system of units** To define the limits in another system of units, convert the values as shown below:

To convert		you calculate
mm → inch		Limit (inches) = 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
mm → pulse		Limit (pulses) = limit (mm) × 1000

1) The number of post-decimal places affects the number of pre-decimal places for the maximum value. Four post-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 post-decimal places, which means the maximum entry you can make is 1,000,000.000 mm.

9.7 Machine data of the axis

Axis type

Address	Name	Type	Initial value	Comment
12.0	AXIS_TYPE	DINT	L#0	**Axis type** 0 = linear axis 1 = rotary axis
A **linear axis** has a limited physical traversing range.  A **rotary axis** is not restricted in its motion range by mechanical end stops. 				

End of rotary axis

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

The "end of rotary axis" value is theoretically the highest actual value of the axis. However, the theoretical maximum value is never indicated, because it also represents the physical start position of the rotary axis (= zero).

The highest rotary axis value displayed is:

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

Example: End of rotary axis = 1000 mm

The displayed value jumps:

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

Rotary axis with absolute encoders

The rotary range (0 to end of rotary axis) of an axis with absolute encoder must correspond exactly to the total resolution of the absolute encoder.

$$\text{Rotary axis end}[\mu m] = \text{Number of revolutions(encoder)} * \frac{\text{steps(encoder)[pulse]}}{\text{Revolution}} * \text{RES}[\frac{\mu m}{\text{Pulse}}]$$

Reference point coordinate

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

Incremental encoder and initiator

The "Retrigger reference point" function and a synchronization event defined by the "Type of reference point retriggering" assign the reference point coordinate to this event.

Absolute encoder (SSI)

An assigned axis with absolute encoder is always synchronized, provided no error is detected (after transmission of the first error-free SSI frame). For more information, refer to the absolute encoder adjustment description (see chapter "Determining the correct absolute encoder adjustment (Page 82)"), which explains the interaction of absolute encoder adjustment with the other data.

Linear axis

The value of the reference point coordinate must always be within the working range (including the software limit switch start and end).

Rotary axis

The value of the reference point coordinate must be greater than or equal 0 and less than the "end of rotary axis" value (0 ≤ reference point coordinate < "End of rotary axis").

Retrigger reference point:

Address	Name	Type	Initial value	Comment
52.0	RETR_TYPE	DINT	L#0	Type of reference point retriggering Ranges: 0 = reference point switch and zero mark direction + 1 = reference point switch and zero mark direction - 6 = only reference point switch 7 = only zero mark
By setting the "Type of reference point retriggering", you define the conditions for synchronizing the axis when working with an incremental encoder or initiator (see chapter "Execute "Retrigger reference point" (Page 122)").				

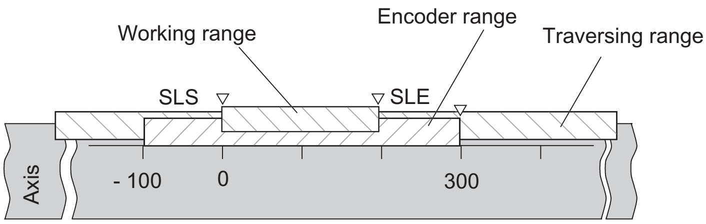
Software limit switch start and end

Address	Name	Type	Initial value	Comment
64.0	SSW_STRT	DINT	L# -100 000 000	**Software limit switch start**
68.0	SSW_END	DINT	L# 100 000 000	**Software limit switch end** Range: - 1 000 000 000 µm to 1 000 000 000 µm

This axis data is only of significance to a linear axis.

The software limit switches are enabled after the FM 452 is synchronized. The range set by the software limit switches represents the **working range**. The FM 452 can monitor the working range limits.

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



Absolute encoder

The FM 452 is synchronized after it has received a complete frame without errors. It monitors the software limit switches as of this time. The absolute encoder used must cover at least the working range (from software limit switch start to software limit switch end, including the limits).

Incremental encoder and initiator

The axis is not synchronized after every restart of the FM 452. The assigned software limit switches are not monitored unless the module has completed a synchronization cycle.

Relationship: working range, encoder range, traversing range:

- The "working range" is defined by your task-specific software limit switch settings.
- The "encoder range" represents the effective encoder range. For linear axes, the module imposes this range symmetrically across the working range, i.e., it shifts the encoder range in order to equalize the distances between the software limit switches and the ends of the encoder range (see the figure above).
- The "traversing range" represents the range of values the FM 452 is capable of processing. It is dependent on the resolution.

The following applies: working range ≥ encoder range ≥ traversing range

Hysteresis

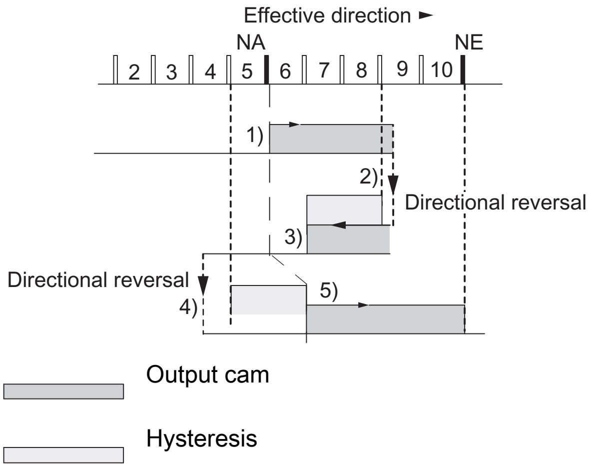
Address	Name	Type	Initial value	Comment
80.0	HYS	DINT	L#0	**Hysteresis** Ranges: 0 to 65 535 [Imp] x resolution [μm/Imp]
The range of values is determined by the resolution: Maximum input value: - to linear axes: max. input value < ¼ of the working range - to rotary axes: max. input value < ¼ of the rotary axis range				

Position-based cams and hysteresis

A position-based cam is activated when the following conditions have been met:

- The detected actual value lies within the position-based cam.
- Hysteresis is not active.

Switching points may vary, depending on the position of the direction reversal.



NA	Cam start
NE	Cam end
1)	Cam is activated, as the cam start is crossed in the positive effective direction.
2)	The cam remains active due to the hysteresis, irrespective of a direction reversal.
3)	At the end of hysteresis, the cam is deactivated.
4)	Cam remains deactivated after a direction reversal.
5)	The cam is activated again at the end of the hysteresis.

Figure 9-1 Enabling position-based cams with hysteresis

Note

When the direction is reversed, the hysteresis cannot reduce a cam actuation time shorter than the hysteresis.

Time-based cam with hysteresis

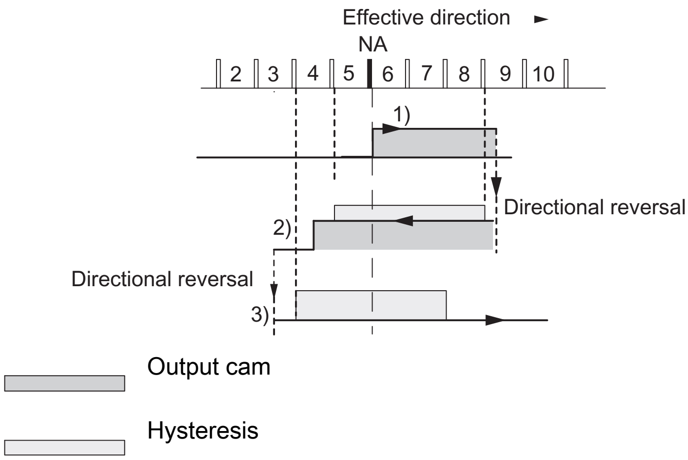
A time-based cam is activated when the following conditions have been met:

- A cam start is overrun in the effective direction.
- Hysteresis is not active.

Note

The hysteresis will hide a time-based cam if its range between the reversal point and the cam start is less than the hysteresis.

The figure illustrates a time-based cam that is **not** activated again.



1)	Cam is activated, as the cam start is crossed in the positive effective direction.
2)	After the direction reversal, the cam remains active until the activation time has expired, regardless of the hysteresis.
3)	Cam start is hidden by the hysteresis, the cam is not activated.

Figure 9-2 Activation of a time-based cam with hysteresis

Simulation velocity

Address	Name	Type	Initial value	Comment
84.0	SIM_SPD	DINT	L#0	**Simulation velocity** The simulation velocity depends on the resolution. 0 = standstill $5 * 10^8$ = highest setting supported by the module Within this range, the simulation velocity depends on the resolution: 1000 * resolution ≤ simulation velocity ≤ $3 * 10^7 *$ resolution

This machine parameter determines simulation velocity (see chapter "Influence of settings on the switching characteristics of time-based cams (Page 105)"). The actual simulation velocity V_{sim} can deviate from the setting at $V_{sim, V}$, and is calculated according to the following formula:

$$V_{Sim} = \frac{6*10exp7*RES}{\text{Integer}\left(\frac{6*10exp7*RES}{V_{Sim, V}}\right)}$$

Meaning of this formula:

- $V_{sim:}$ Simulation velocity set by FM 452. Unit: $^{\mu m}/_{min}$
- $V_{sim, V:}$ Default simulation velocity set in machine data. Unit: $^{\mu m}/_{min}$
- RES: Resolution derived from encoder data. Unit: $^{\mu m}/_{pulse}$
- Integer ( ): Of this expression, only the decimal integer is included in the further calculation. In all calculations, this expression must be within the range from 2 to 65536.

The actual simulation velocity changes abruptly as a result of correlations (see the formula).

Minimum edge interval

Address	Name	Type	Initial value	Comment
4.0	EDGEDIST	DINT	L#0	**Minimum edge interval** Range: 0 ... 1 000 000 000 µm

This machine parameter defines a range after detection of the start of a measurement for edge detection. The measurement is discarded if the end of the measurement is within this range.

The start of the measurement is not reported unless the "minimum edge interval" has been traveled.

9.8 Determining the correct absolute encoder adjustment

Definition

The absolute encoder adjustment and reference point coordinate maps the encoder's range of values unambiguously to the axis coordinate system.

Address	Name	Type	Start value	Comment
48.0	ENC_ADJ	DINT	L#0	Absolute encoder adjustment Range: 0 to 2^{25} -1
The "Absolute encoder adjustment" determines the encoder value that represents the reference point coordinate on the axis. The value must be less than the total number of steps of the absolute encoder.				

Procedure: Determining the correct absolute encoder adjustment

After you completed the basic program, you need to define a balanced correlation between the encoder and the coordinate system. The procedure shown in the next section is based on the use of the programming interface.

1. Move the axis to a known and physically unambiguous, reproducible point.

   This could be the "end software limit switch".
2. Call the "set reference point" function, using the coordinate position defined in step 1.

   FM 452 now calculates an encoder value for the reference point coordinate entered in the channel DB (REFPT in channel DB), namely the absolute encoder adjustment. You can view this value at the service screen of the programming interface.
3. Enter the value from the service screen in the "Absolute encoder adjustment" field on the "Axis" tab of the programming interface.
4. Save your parameter assignment to the corresponding parameter DB using the export function.
5. Close the programming interface by selecting File and Exit.
6. In HW Config, download the data to the CPU.
7. Restart the CPU to apply the data.

Note

This adjustment is made once during commissioning. During its startup, the programmed FM 452 will be synchronized after it has received a complete, faultless frame from the encoder.

Data in the channel DB

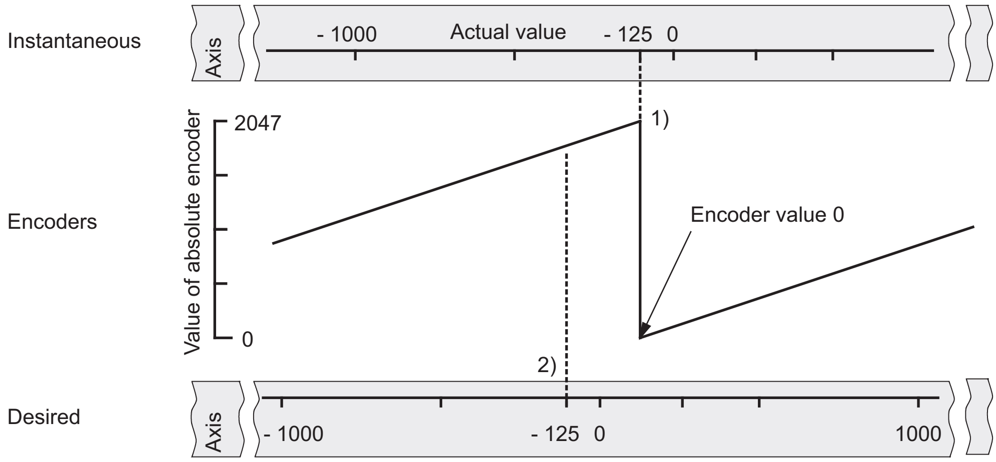
Address	Name	Type	Start value	Comment
98.0	REFPT	DINT	L#0	Reference point coordinate Range: -1 000 000 000 µm to +1 000 000 000 µm

9.9 Example: Adjusting the absolute encoder

Example of absolute encoder adjustment

For this example, let us presume:

- reference point coordinate = -125 mm
- working range of SSW_STRT = -1,000 mm to SSW_END = 1,000 mm
- absolute encoder adjustment = 0
- encoder range = 2048 increments (= pulses) , at a resolution of 1 mm/pulse
- A precise mechanical adjustment of the absolute encoder is not possible, and it does not provide a function for setting a selective actual value.



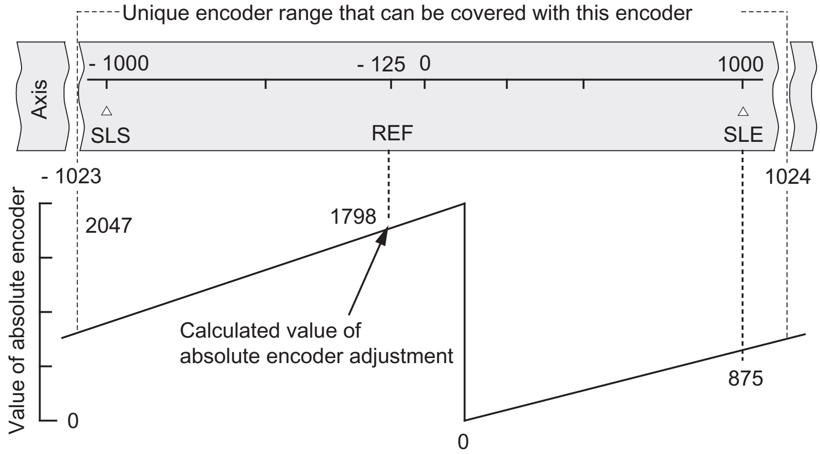
(1) Assignment of the coordinate system to encoder values, based on the absolute encoder adjustment. Encoder value 0 is equivalent to actual value -125.

(2) Required assignment of the coordinate system to the encoder. The coordinate value should be -125 at this position.

Result of "Set reference point"

The Set reference point operation creates the following relationship between the encoder and coordinate system:

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



The encoder returns 2048 unambiguous values. The working range is determined by the software limit switches. However, due to the set resolution of 1 mm/pulse, the encoder's working range extends beyond the set software limit switch range.

At the set resolution, the working range is already covered by 2001 values. In the example, this produces a "remainder" of 47 pulses which symmetrically enclose the working range.

Alternative: Mechanical adjustment of an encoder

To create a proper relationship between the coordinate system and the encoder:

1. Move the axis to a reproducible position (for example, the start software limit switch).
2. Set this coordinate value at the machine parameters as reference point coordinate.
3. Read the encoder value indicated at this position in the service screen form of the programming interface.
4. Set this value as absolute encoder adjustment at the machine parameters.

The parameters will thus always return the correct actual value.

As an alternative to steps 3 and 4, you can also zero the encoder by means of a "Reset" signal (if this exists), and then set the value "0" as the absolute encoder adjustment in the machine data.

9.10 Machine data of the encoder

Definition

The encoder returns position data to the module for evaluation and conversion to an actual value based on the resolution.

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

Data in the parameter DB:

Encoder type and frame length

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 = listen in 6 = 24 V initiator positive direction 7 = 24 V initiator negative direction 8 = SSI 13-bit (right-justified) 9 = SSI 25-bit (right-justified) 10 = listen in (right-justified)
At the "frame length" you define the clock frame output by FM 452. If you select "listen in", you disable the cycle clock of FM 452. The FM 452 can then listen in on other SSI frames with a 13 or 25 bit frame length. The transmission rate is determined by the cycle clock rate of the master module.				

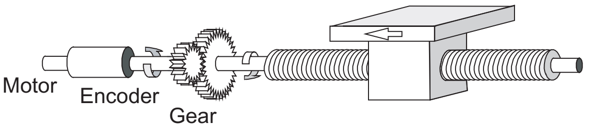
Distance per encoder revolution

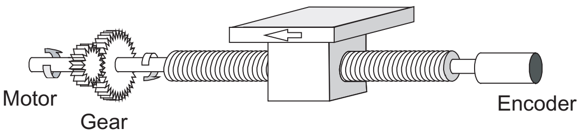
Address	Name	Type	Initial value	Comment
24.0	DISP_REV	DINT	L#80000	Distance per encoder revolution Range of values: 1 μm to 1 000 000 000 μm

Using the "Distance per encoder revolution" machine parameter you inform the FM 452 of the distance covered by the drive system 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 make allowances for all transmission elements, such as couplings or gears.

The chapter "Resolution (Page 91)" describes the correlation between the "Distance per encoder revolution" and "Increments per encoder revolution" machine data.





Increments per encoder revolution

Address	Name	Type	Initial value	Comment
32.0	INC_REV	DINT	L#500	Increments per encoder revolution Range of values: 1 to 2^{25} Note: This entry is irrelevant when using pulse units.

The "Increments per encoder revolution" machine data element specifies the number of increments output by an encoder per revolution. FM 452 calculates the resolution based on this value and the "Distance per encoder revolution" machine data element.

- Incremental encoder

  Any value within the range of values can be input. The module evaluates the increments in four operations (see chapter "Incremental encoder (Page 139)").
- Initiator

  Any value within the range of values is allowed.
- Absolute encoder

Limits differ between the various encoder types:

Encoder type	Message frame length / type	Range of values	Can be used as linear axis
Single-turn encoder	13-bit half fir tree	64 ... 8192 in powers of 2	
Single-turn encoder	13-bit right-justified	64 ... 8192 all values	X
Single-turn encoder	25-bit right-justified	64 ... 2^{25} all values	X
Multiturn encoder	25-bit fir tree	64 ... 8192 in powers of 2	
Multiturn encoder	25-bit right-justified	64 ... 2^{24} all values	
Listen in	Fir tree	64 ... 8192 in powers of 2	
Listen in	Right-justified	64 ... 2^{25} all values	X
Special setting: Multiturn encoder in single-turn mode	25-bit half fir tree	64 ... 8192 in powers of 2	

Note

The number of encoder pulses is calculated by multiplying the "increments per encoder revolution" by the "number of revolutions" (see chapter "Resolution (Page 91)").

Number of encoder revolutions

Address	Name	Type	Initial value	Comment
36.0	NO_REV	DINT	L#1024	Number of encoder revolutions Range of values: 1 (single-turn encoder) 2 to 2^{19} (multiturn encoder)
The "number of encoder revolutions" machine data element is only required for absolute encoders. It is used to define the maximum number of revolutions of this encoder. For more information about absolute encoders, refer the "Absolute encoders (Page 143)" chapter in this manual. **Single-turn encoder** Only the value "1" is possible. **Multiturn encoder** Multiturn encoder / listen in (fir tree): 2 ... 4096 in powers of 2 Multiturn encoder / listen in (right-justified): 2 ... 2^{19} all values, but with the following restriction: Increments/encoder revolution * number of encoder revolutions ≤ 2^{25}. **Linear scale** You can also interconnect a linear scale. To do so, enter the value "1". **Total number of encoder steps** The total number of steps is not a machine data element. Total number of steps = increments per encoder revolution * number of revolutions				

Baud rate

<table>
<tr><th>Address</th><th>Name</th><th>Type</th><th>Initial value</th><th>Comment</th></tr>
<tr><td>40.0</td><td>BAUD RATE</td><td>DINT</td><td>L#0</td><td>Baud rate
Range of values:
0 = 125 kHz
1 = 250 kHz
2 = 500 kHz
3 = 1000 kHz</td></tr>
<tr><td colspan="5">The "BAUDRATE" machine parameter defines the data transmission speed between the SSI encoder and FM 452.
This entry has no significance for incremental encoders.
The maximum cable length depends on the transmission rate:
• 125 kHz → 320 m
• 250 kHz → 160 m
• 500 kHz → 63 m
• 1000 kHz → 20 m</td></tr>
</table>

Count direction

<table>
<tr><th>Address</th><th>Name</th><th>Type</th><th>Initial value</th><th>Comment</th></tr>
<tr><td>59.0</td><td>CNT_DIR</td><td>BOOL</td><td>FALSE</td><td>Count direction
0 = normal
1 = inverted</td></tr>
<tr><td colspan="5">The "count direction" machine parameter is used to adapt the position feedback direction to the direction of axis movement. Also, take the rotation directions of all transmission elements into account (e.g., for example, couplings and gears).
• Normal = ascending count pulses (incremental encoder) or encoder values (absolute encoder) correspond to ascending actual position values
• Inverted = ascending count pulses (incremental encoder) or encoder values (absolute encoder) correspond to descending actual position values
It is not allowed to implement a lead time in combination with an absolute encoder (SSI) and inverted count direction.</td></tr>
</table>

Monitoring

Address	Name	Type	Initial value	Comment
				Monitoring functions
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 its monitoring function is enabled, FM 452 monitors the A, /A, B, /B, N, and /N signals of an incremental encoder. The monitoring function detects:

- Wire break
- Short-circuit on cables

  For incremental encoders without zero mark, you must take one of the following measures:
  - Disable the wire-break monitoring function
  - Interconnect the N and /N signals externally (see chapter "Incremental encoder (Page 139)")
- Edge interval of the count pulses
- Encoder supply failure

Frame error

Frame error monitoring for absolute encoders (SSI) cannot be disabled. It monitors the frame:

- Start and stop bit errors
- Monitoring of the monostable time of the connected encoder

Missing pulses (incremental encoder)

An incremental encoder must return a consistent number of increments between two successive zero marks. FM 452 checks whether the zero mark of an incremental encoder coincides with the correct encoder value. Disable missing pulse monitoring at encoders without zero mark. Also disable wire-break monitoring, or interconnect the zero mark inputs N and /N externally.

9.11 Resolution

Definition

The resolution reflects the precision of cam processing. It also determines the maximum possible traversing range.

The resolution (RES) is calculated as follows:

	Incremental encoder	Absolute encoder/ proximity switch
Input values	• Distance per encoder revolution • Increments per encoder revolution • Pulse evaluation: 4 operations • 1 increment = 4 pulses	• Distance per encoder revolution • Increments per encoder revolution • 1 increment = 1 pulse
Calculation	RES = (distance/encoder revolution) / (pulses/encoder revolution)	

Note

The resolution of the physical unit pulses is always 1.

All position values are rounded to the integer multiple of the resolution. This allows you to distinguish between the entered and used values.

Range of values of the resolution

Convert the range of values for the resolution according to the defined physical units. The resolution must be kept within this range when setting the "distance per encoder revolution" and "increments per encoder revolution" values.

Range of values for the resolution derived from the physical units:

Physical units system	Specified in...	Range of values of the resolution
mm	10^{-3} mm	0.1 * 10^{-3} mm to 1000 x 10^{-3} mm/pulse
inch	10^{-4} inch	0.1 * 10^{-4} inches to 1000 x 10^{-4} inches/pulse
degrees	10^{-4} degrees 10^{-3} degrees 10^{-2} degrees	0.1 * 10^{-4} degrees to 1000 x 10^{-4} degrees/pulse 0.1 * 10^{-3} degrees to 1000 x 10^{-3} degrees/pulse 0.1 * 10^{-2} degrees to 1000 x 10^{-2} degrees/pulse
Pulses	1 pulse	1

Example

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

  Resultant resolution (quadruple evaluation):

  Resolution

  = 1000 mm / 5000 increments

  = 0.2000 $^{mm}/_{increment}$

  = 0.2000 $^{mm}/_{4\ pulses}$

  = 0.0500 $^{mm}/_{pulse}$
- An SSI encoder has the following data:
  - Increments per encoder revolution: 4096
  - Distance per encoder revolution: 1000 mm
  - increment = 1 pulse

  Resultant resolution:

  Resolution

  = 1000 mm / 4096 increments

  = 0.2441 $^{mm}/_{increment}$

  = 0.2441 $^{mm}/_{pulse}$

Dependency of the traversing range on the resolution

The traversing range is limited by the number notation in FM 452. The number notation varies, depending on the resolution. Therefore, make sure that specifications are always within the valid limits.

The table below shows the maximum traversing range:

Resolution (RES) is in the range	Maximum traversing range
0.1 $^{μm}/_{pulse}$ ≤ RES < 1 $^{μm}/_{pulse}$	-10^8 μm to 10^8 μm (-100 m to +100 m)
1 $^{μm}/_{pulse}$ ≤ RES ≤ 1000 $^{μm}/_{pulse}$	-10^9 μm to 10^9 μm (-1000 m to +1000 m)

Dependency of the velocity on the resolution

The velocity indicated can vary within the following limits, depending on the resolution (specification relates to mm units):

- from 1 $^{\mu m}/_{min}$ to 90 $^{m}/_{min}$, at a resolution < 1 $^{\mu m}/_{pulse}$
- from 1 $^{\mu m}/_{min}$ to 900 $^{m}/_{min}$, at a resolution ≥ 1 $^{\mu m}/_{pulse}$

The velocity value is calculated and smoothed by the module at intervals of 4 ms.

Its minimum inaccuracy of one pulse/4 ms rules it out for closed-loop control.

9.12 Number of cams and track data

Scope

The scope determines the cam cycle time and the maximum number of programmable cams.

Scope	Cam cycle time
16 cams	20.48 µs
32 cams	40.96 µs
64 cams	81.92 µs
128 cams	163.84 µs

Number of cams in the parameter DB

Address	Name	Type	Start value	Comment
76.0	C_QTY	DINT	UDT3: L#0 UDT4: L#1 UDT5: L#2 UDT6: L#3	Scope: 0 = max. 16 cams 1 = max. 32 cams 2 = max. 64 cams 3 = max. 128 cams

Track data in the parameter DB

Activation of track outputs:

<table>
<tr><th>Address</th><th>Name</th><th>Type</th><th>Start value</th><th>Comment</th></tr>
<tr><td>90.0</td><td>TRACK_OUT</td><td>WORD</td><td>W#16#0</td><td>Activation of track outputs
Range:
0 = cam control system
1 = CPU
Bit number = track number</td></tr>
<tr><td colspan="5">The "activation of track outputs" machine parameter defines the type of control of track signals of tracks 0 to 15. Tracks can be controlled by:
• The cam control system: Track signals are activated and deactivated by FM 452 cam processing functions.
• CPU: The track signals represent the corresponding values of the track enables in the channel DB.
This allows for explicit activation of the track outputs in the user program.</td></tr>
</table>

Enable input:

Address	Name	Type	Start value	Comment
				Enable input
95.0	EN_IN_I3	BOOL	FALSE	1 = track signal track 3 AND enable input I3
				...
......	...	...	...	1 = track signal track 10 AND enable input I10
95.7	EN_IN_I10	BOOL	FALSE	
Track signal Q3 is activated if the following conditions are met: • The track is enabled with TRACK_EN. • The relevant external enable input I3 to I10 is set. • The track result is 1.				

Special tracks:

Address	Name	Type	Start value	Comment
				Special tracks
99.0	SPEC_TRC0	BOOL	FALSE	1 = track 0 is counter cam track
99.1	SPEC_TRC1	BOOL	FALSE	1 = track 1 is counter cam track
99.2	SPEC_TRC2	BOOL	FALSE	1 = track 2 is brake cam track
You can program tracks 0, 1 and 2 for use as special tracks.				

High count value of counter cam track

Address	Name	Type	Start value	Comment
100.0	CNT_LIM0	DINT	L#2	Upper count value of counter cam track (track 0)
104.0	CNT_LIM1	DINT	L#2	Upper count value of counter cam track (track 1)
				Range: 2 ... 65535
Use this machine parameter to define the high count value of the programmed counter cam track.				

9.13 Interrupt enable

Definition

You can define whether to generate process interrupts when cams 0 to 7 are activated and/or deactivated (see chapter "Basics of Programming an FM 452 (Page 37)").

- Cam on/off

  In the cam parameters, you can specify whether process interrupts are to be generated when cams 0 to 7 are activated and/or deactivated (see chapter "Cam parameters (Page 97)").
- Start measurement

  With "edge detection" setting, a rising edge at digital input I1 can trigger a process interrupt.
- Measurement completed

  Both with the "edge detection" and "length measurement" setting, a falling edge at digital input I1 can trigger a process interrupt.

Machine parameters for interrupt enable in the parameter DB

Address Absolute	Name	Type	Start value	Comment
3.1	PI_MEND	BOOL	FALSE	1 = enable process interrupt: End of measurement
3.2	PI_CAM	BOOL	FALSE	1 = enable process interrupt: Cam on/off
3.5	PI_MSTRT	BOOL	FALSE	1 = enable process interrupt: Start of measurement

Cam parameters for interrupt enable in the parameter DB

Address Relative	Name	Type	Start value	Comment
+0.4	PI_SW_ON	BOOL	FALSE	1 = process interrupt on activation
+0.5	PI_SW_OFF	BOOL	FALSE	1 = process interrupt on deactivation

9.14 Cam parameters

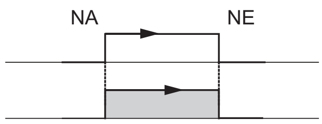
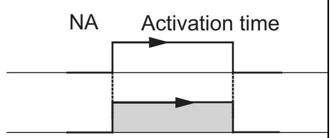
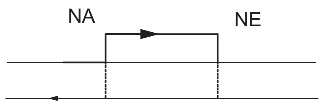
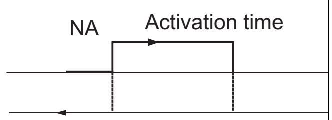
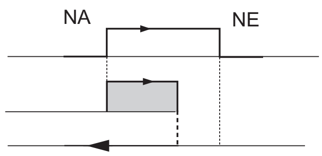
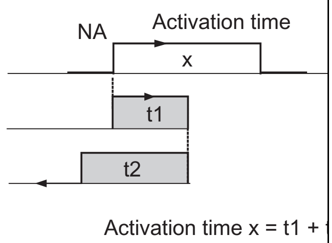
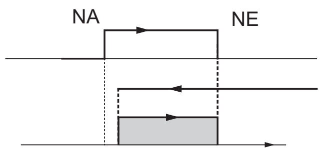
Definition

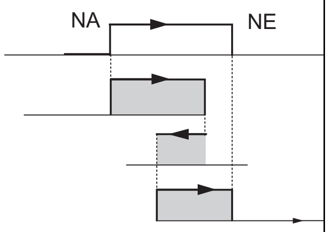
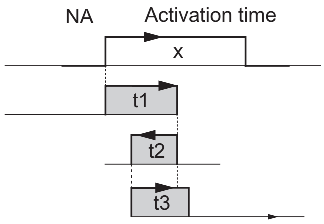
Cam data describe the properties of a cam, the assignment of each cam to a track, and the switching characteristic of the cam. The cam data listed below are set separately at each cam.

- The module interprets and processes only the cams with "valid" settings.
- Cams 0 to 7 support hardware interrupts.
- The number of assignable cams depends on the number of cams available.

Switching characteristics of cams depending on the effective direction

A positive effective direction is always assumed, with the exception of example 5.

No.	Description	Position-based cam	Time-based output cam
1	A cam is passed in the effective direction.		
2	A cam is passed in reversed effective direction.		
3	A cam is approached in effective direction; the motion direction of the axis is reversed while the cam is active.		
4	A cam is approached in reversed effective direction; the axis reverses its direction of movement on the cam in the effective direction.		Cam not switched

No.	Description	Position-based cam	Time-based output cam
5	A cam is approached from any direction, and exited in any direction; **both directions** are set as the effective direction.		 Activation time x = t1 + t2 + t3
▭ Assigned cam ▬ Switched cam NA = Cam start NE = Cam end			

Cam data in the parameter DB

Address Relative	Name	Type	Initial value	Comment
+0.0	CAMVALID	BOOL	FALSE	1 = cam valid
+0.1	EFFDIR_P	BOOL	TRUE	1 = positive effective direction (plus)
+0.2	EFFDIR_M	BOOL	TRUE	1 = negative effective direction (minus)
+0.3	CAM_TYPE	BOOL	FALSE	0 = position-based cam 1 = time-based cam
+0.4	PI_SW_ON	BOOL	FALSE	1 = hardware interrupt on activation
+0.5	PI_SW_OFF	BOOL	FALSE	1 = hardware interrupt on deactivation
+1.0	TRACK_NO	BYTE	B#16#0	**Track number** Range: 0 to 31

Effective direction

Two effective directions are supported:

positive: The cam is activated at the cam start, if the axis is moving in the direction of increasing actual values.

negative: The cam is activated at the cam end, if the axis is moving in the direction of decreasing actual values.

You can set both effective directions in parallel.

Track number

Define the active track for the cam by setting the track number.

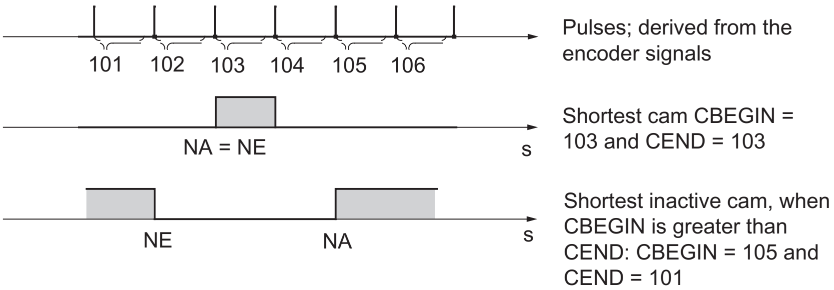
Note

Unused cams should always be set "invalid" (CAMVALID = FALSE).

Cam start (NA)/Cam end (NE) of position-based cams

Address Relative	Name	Type	Initial value	Comment Of position-based cams
+2.0 +6.0	CBEGIN CEND	DINT DINT	L#-100000000 L#100000000	Cam start (NA) Cam end (NE) Range: - 1 000 000 000 µm to 1 000 000 000 µm

Minimum length of a position-based output cam



Shortest cam with axis motion in positive direction

The inactive section of a cam must always have an interval of at least 4 pulses between the cam end (NE) and cam start (NA).

If NE = NA, the cam is activated for the duration of one pulse.

Cam start (NA)/Cam end (NE) for time-based cam

Address Relative	Name	Type	Initial value	Comment Of time-based cams
+2.0	CBEGIN	DINT	L#-100000000	Cam start (NA)
+6.0	CEND	DINT	L#100000000	Cam end (NE) Activation time Range: (0 to 13421) x 100 μs with up to 16 cams (0 to 26843) x 100 μs with up to 32 cams (0 to 53686) x 100 μs with up to 64 cams (0 to 65535) x 100 μs with up to 128 cams

With a time-based cam, you must specify a cam start and an activation time in place of the cam end. You can set a resolution of 100 μs for the activation time. The time runs starting with the activation of the cam.

Conditions of setting default times:

- 0 μs: A cam with 0 μs activation time is never activated
- 0 μs < t ≤ 400 μs: The FM 452 sets a minimum cam activation time of approx. 330 μs.
- t > 400 μs: FM 452 calculates the actual activation time t_{act} based on the default activation time t_{def} according to this formula:

$$t_{act} = \text{Integer}\left(\frac{t_{spec}}{\text{Cam cycle}}\right) * \text{Cam cycle time}$$

The maximum error is always less than one cam cycle.

Lead time

Address Relative	Name	Type	Initial value	Comment
+ 10.0	LTIME	INT	0	**Lead time** Range: (0 to 53686) x 100 µs with max. 16 cams (0 to 65535) x 100 µs with max. 32, 64 or 128 cams

You can compensate for any delays caused by the connected switchgear by setting a lead time. Define a lead time with a resolution of 100 µs. You can assign a lead time to each cam. The lead time applies to the cam start and cam end.

Anticipation distance

The anticipation distance of a cam is calculated continuously based on the current velocity and lead time. The entire cam is shifted in direction of the actual value by this distance. The assigned range is the "static range", and the range calculated based on the lead time is the "dynamic range".

Anticipation distance = [lead time] x [current velocity]

FM 452 calculates the anticipation distance of all cams within $^{1}/_{4}$ of the longest assigned lead time. A very high lead time setting for a cam reduces the lead of the calculation of dynamic adjustment.

Actual lead time

To calculate the actual lead time:

1. Determine the cam cycle time: This is the time FM 452 requires to complete processing of all cams, and depends on the number of cams assigned.
2. Calculate the actual lead time based on the following formula:

$$\text{Lead time}_{act} = \text{integer}\left(\frac{\text{Lead time}_{d}}{\text{Cam cycle time} * 4}\right) * \text{Cam cycle time} * 4$$

The identifiers have the following meanings:

Lead time $_{act}$ is the lead time set by the FM 452

Lead time $_{V}$ is your default setting.

Integer ( ) means that only the integer decimal values is included in the calculation of the parenthesis.

The maximum error of the lead time$_{act}$ is always < [cam cycle time] x 4.

Example:

The following values are defined:

Scope: maximum of 32 cams

Cam cycle time: 40.96 µs

Lead time$_{V}$ = 1000 µs

You obtain an actual lead time of 983 µs.

It is not allowed to implement a lead time in combination with an absolute encoder (SSI) and inverted count direction.

Note

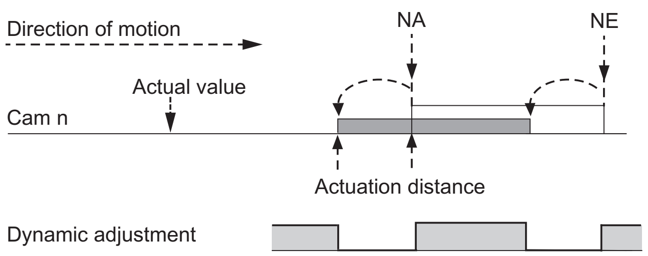
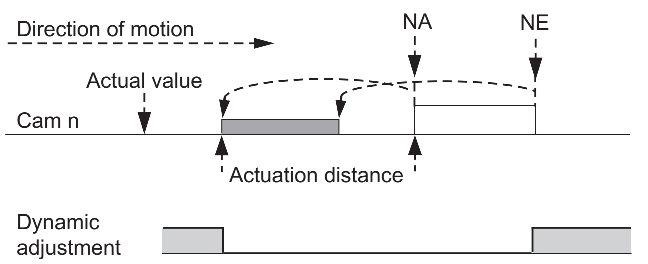
The actual lead time is always less than the assigned lead time. It can be 0, even though the assigned lead time is ≥100 µs.

The anticipation distance of a rotary axis must be less than the rotary axis range and the inactive part of the cam. This must be ensured for all velocities.

Dynamic cam adjustment

There are two distinct situations relating to the cam range:

1. The static and dynamic range of the cam overlap.
2. The static and dynamic range of the cam do not overlap.

Dynamic cam adjustment (different use cases)	
Dynamic adjustment	**Description**
	If the dynamic range of the cam overlaps its static range: • The cam is activated when its dynamic range is reached. At the same time, calculation of a new dynamic adjustment is disabled. • After the actual value has reaches the static range of the cam, the calculation of a new dynamic adjustment is re-enabled. A velocity change affects the cam end. • If the cam is deactivated at the end of the dynamic range, dynamic adjustment will be disabled again until the end of the static range of the cam.
	The following rule applies if the cam's dynamic and static ranges do not overlap: • The cam is activated when its dynamic range is reached. At the same time, calculation of a new dynamic adjustment is disabled. • At the end of the static range of the cam, dynamic adjustment is enabled again.
Dynamic range Static range A **new** dynamic adjustment is possible NA = Cam start NE = Cam end	

Note

When the direction of rotation changes, calculation of the dynamic adjustment is enabled again.

10 Settings

10.1 Influence of settings on the switching characteristics of time-based cams

Actual value changes

A time cam can be skipped by the following settings that change the actual value:

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

Activating a time-based cam

A time-based cam is always activated, regardless whether you skip its start position due to one of the settings listed above, provided the actual direction of movement of the axis matches the effective direction set at the cam. The programmed cam activation time starts.

Note

If the axis is at a standstill, the direction of movement is influenced by fluctuations of the actual value.

Set a hysteresis higher than the fluctuation of the actual value signal in order to suppress flutter when the axis is at a standstill.

This retains the last determined direction of movement while the axis is at a standstill.

WARNING

Injury to persons or damage to equipment can occur.

Any modification of the actual value at rotary axes may cause unwanted activation of time-based cams.

You should always set the "invalid" option at the time-based cams of a rotary axis if you want to influence the actual value using the settings mentioned earlier.

10.2 Modifying the "Set Actual Value/Set Actual Value on-the-fly/Cancel Set Actual Value" settings

Definition

Use the "Set actual value/Set actual value on-the-fly" settings to assign a new coordinate to the actual encoder value. This shifts the coordinate system by the value: ACT_{new} - $ACT_{current}$

Whereby:

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

Calculating new coordinates

Calculate all your default positions in the shifted coordinate system based on the following formula:

$Coordinate_{new} = Coordinate_{old} + (ACT_{new} - ACT_{current})$

Requirements

- The axis must be synchronized.
- With "Set actual value on-the-fly": Digital input I1 must be interconnected.

Programming steps

1. Enter the coordinate for the actual value, or for the actual value to set on-the-fly in the channel DB.
   - Linear axis:

     Select an actual value so that the software limit switches are still within the valid traversing range after the setting is called.

     The offset value derived from (ACT_{new} - $ACT_{current}$) must be less than or equal the valid traversing range (maximum 100, m or 1000 m).
   - Rotary axis:

     Rule for the specified actual value:

     0 ≤ actual value < end of rotary axis
2. Set the corresponding trigger bits at the channel DB.
3. Call FC CAM_CTRL.

   "Set actual values" is executed immediately.

   "Set actual value on-the-fly" is executed at the next positive edge at digital input I1. The FVAL_DONE bit is set.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
36.4	AVAL_EN	BOOL	FALSE	1 = set actual value
36.5	FVAL_EN	BOOL	FALSE	1 = set actual value on-the-fly
90.0	AVAL	DINT	L#0	Actual value coordinate
94.0	FVAL	DINT	L#0	Coordinate for on-the-fly actual value
25.5	FVAL_DONE	BOOL	FALSE	1 = set actual value on-the-fly executed

Effects of the setting

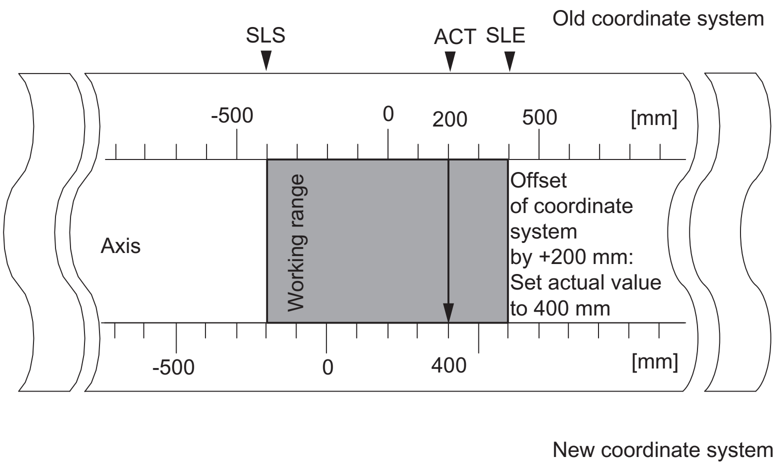
Based on the example of "set actual value" to 400 mm (at position 200 mm), you can see how this setting shifts the coordinate system. Resultant effects:

- The position of the working range is **not** shifted physically.
- The various points (such as the software limit switches) are assigned new coordinate values.
- The cams retain their coordinate values, and are therefore located at a different physical position.
- If the axis is synchronized and cam processing is enabled, the actual position value might skip cam edges or entire cams as a result of this setting.
- Status changes of the cam which would normally trigger an interrupt could be lost.

Note

For information on the switching characteristics of timing cams, refer to chapter "Influence of settings on the switching characteristics of time-based cams (Page 105)".

Table 10- 1 Shift of the coordinate system by "Set Actual Value" / "Set Actual Value on-the-Fly"

<table>
<tr><th>Set actual value</th><th>SLS
[mm]</th><th>ACT
[mm]</th><th>SLE
[mm]</th></tr>
<tr><td rowspan="2">
</td><td>-400</td><td>200</td><td>400</td></tr>
<tr><td>-200</td><td>400</td><td>600</td></tr>
</table>

Canceling the setting

The "Cancel set actual value" setting can be used to reset the coordinate shift caused by "Set actual value" or "Set actual value on-the-fly".

Once "set actual value on-the-fly" has been triggered, it can no longer be deleted before execution by a positive edge at input I1. However, it can be overwritten by a new "Set actual value on-the-fly" command.

Those settings will be reset at the next start of the module.

Parameter used in the channel DB

Address	Name	Type	Start value	Comment
35.2	AVALREM_EN	BOOL	FALSE	1: Cancel actual value setting

Possible causes of error

"Set actual value on-the-fly" and **"Retrigger reference point"** may not be executed simultaneously.

With the setting **"set actual value on-the-fly"**, an error can be reported if the setting means that a software limit switch would be exceeded at the rising edge on I1. This system error is reported by a diagnostics interrupt and written to the diagnostics buffer.

10.3 Execute "Set zero offset"

Definition

The "zero offset" setting lets you shift the zero point in the coordinate system by a defined value. The sign determines the offset direction.

Calculating a new coordinate

Calculate all values of the shifted coordinate system using the following equation:

$Coordinate_{new} = Coordinate_{old} - (ZPO_{new} - ZPO_{old})$

ZPO_{old} identifies any existing zero offset. If no zero offset was active prior to the call, set a 0 value at ZPO_{old}.

Using this equation, you can calculate the coordinates for the software limit switches, for example.

Programming steps

1. Enter the zero offset value in the channel DB.
   – Linear axis:

     The zero offset must be selected so that the software limit switches remain within the valid traversing range after the setting is called.
   – Rotary axis:

     Rule for zero offset:

     Value of zero offset ≤ end of the rotary axis.
2. Set the relevant trigger bit.

Data used in the channel DB

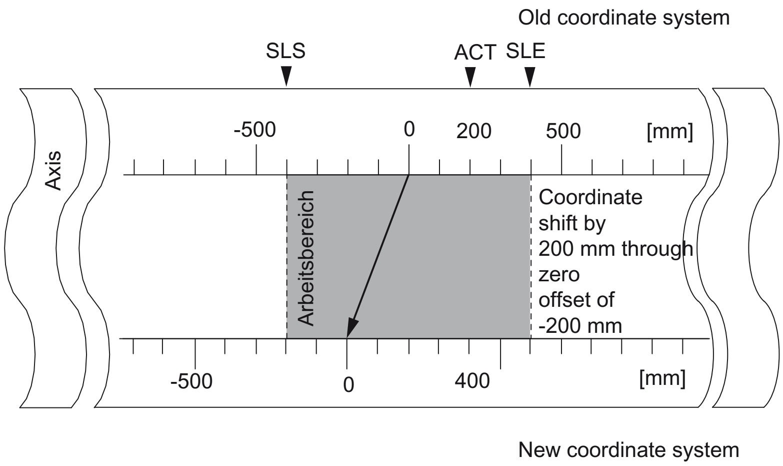
Address	Name	Type	Initial value	Comment
36.6	ZOFF_EN	BOOL	FALSE	1 = set zero offset
86.0	ZOFF	DINT	L#0	Zero offset

Effects on a linear axis

Based on the example of a zero offset of -200 mm, you can see that this setting shifts the coordinate system in positive direction. Resultant effects:

- The working range is **not** physically shifted.
- The various points (such as the software limit switches) are assigned new coordinate values.
- The cams retain their coordinate values, and are therefore located at a different physical position.
- If the axis is synchronized and cam processing is enabled, the actual position value might skip cam edges or entire cams as a result of this setting.
- Status changes of the cam which would normally trigger an interrupt could be lost.

Table 10- 2 Coordinate system shift as a result of zero offset

<table>
<tr><th>Zero offset</th><th>SLS
[mm]</th><th>ACT
[mm]</th><th>SLE
[mm]</th></tr>
<tr><td rowspan="2">
</td><td>-400</td><td>200</td><td>400</td></tr>
<tr><td>-200</td><td>400</td><td>600</td></tr>
</table>

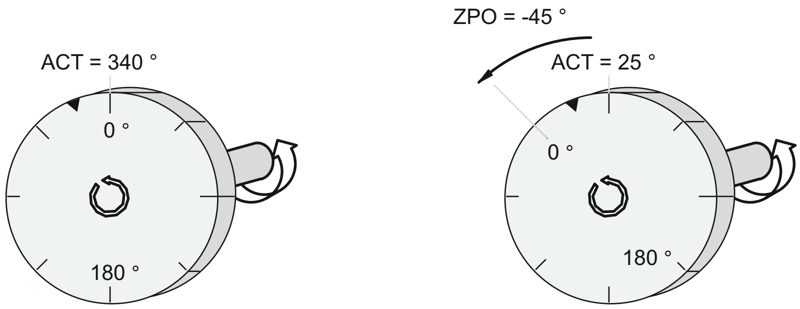
Note

For information on the switching characteristics of timing cams, refer to chapter "Influence of settings on the switching characteristics of time-based cams (Page 105)".

Effects on a rotary axis

Based on the example of a zero offset by -45°, you can see how this setting rotates the coordinate system:

Table 10- 3 Rotation of the coordinate system as a result of zero offset

<table>
<tr><th>Tool when ACT = 340°</th><th>Tool when ACT = 25°</th></tr>
<tr><td colspan="2">

The zero position rotates by -45 degrees. All axis points are assigned new coordinate values.</td></tr>
</table>

Including a **ZPO_{old} = 0**, the result is a new value of 385°.

As the actual value is restarted with 0 at the end of the rotary axis operating in positive directional rotation, the calculation returns an actual value of 25°:

$Coordinate_{new}$ = $Coordinate_{old}$ - (ZPO_{new} - ZPO_{old}) - end of rotary axis

The **end of rotary axis** value only needs to be subtracted if

$Coordinate_{old}$ - (ZPO_{new} - ZPO_{old}) is greater than the end of rotary axis.

Loss of synchronization

If synchronization is lost due to an error, or reset by means of "retrigger reference point", a zero offset is **remains** active.

Canceling the setting

A zero offset of 0 resets any existing zero offset.

10.4 Execute "Set reference point"

Definition

The "set reference point" setting is used to synchronize the axis. This setting shifts the working area. All shifts generated by a zero offset or "set actual value" function are retained.

Requirements

Cam processing must be disabled.

Programming steps

1. Enter the value for the reference point coordinate in the channel DB.
   – Linear axis:

     The reference-point coordinate may not exceed the range 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 relevant trigger bit.

Data used in the channel DB

Address	Name	Type	Start value	Comment
36.3	REFPT_EN	BOOL	FALSE	1 = set reference point coordinate
98.0	REFPT	DINT	L#0	Reference point coordinate
25.0	SYNC	BOOL	FALSE	1 = axis synchronized

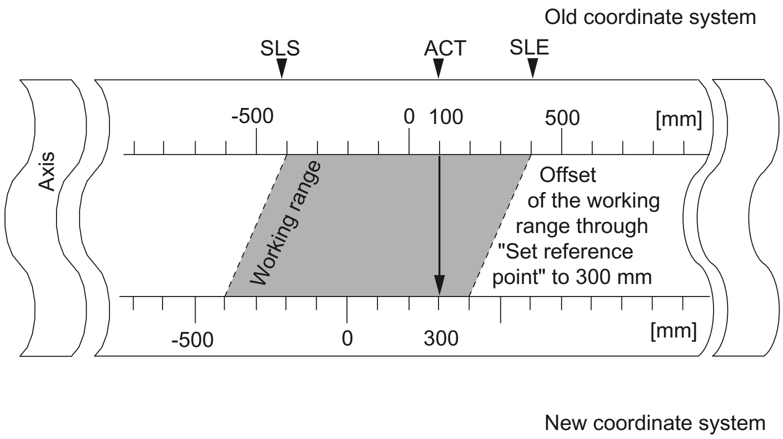
Effects of the setting

Based on the example "set reference point" to 300 mm, you can see how this setting shifts the working range of the axis.

This has the following effects:

- The actual position is set to the value of the reference point coordinate.
- The working range is physically shifted on the axis.
- The various points retain their original coordinates, but are now at new physical positions.
- The SYNC bit is set in the checkback signals.

Table 10- 4 Shifting the working range on the axis using "Set Reference Point"

<table>
<tr><th>Set reference point</th><th>SLS
[mm]</th><th>ACT
[mm]</th><th>SLE
[mm]</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>

Special features of absolute encoders

This setting is required for an absolute encoder adjustment (see chapter "Determining the correct absolute encoder adjustment (Page 82)").

10.5 Execute "Change cam edges"

Definition

The "change cam edges" setting can be used to change the cam start and, for positioning cams, the end of a specific positioning cam at runtime.

Requirements

The cam you want to change must be valid.

Programming steps

1. Enter the cam number in the channel DB.
2. At a positioning cam:

   Enter the cam start and cam end in the channel DB.

   For a timing cam:

   Enter the cam start value in the channel DB.
3. Set the relevant trigger bit.

Data used in the channel DB

Address	Name	Type	Start value	Comment
36.7	CH01CAM_EN	BOOL	FALSE	1 = write setting for cam edges (1 cam)
102.0	CAM_NO	INT	0	Cam number
104.0	CAM_START	DINT	L#0	Cam start
108.0	CAM_END	DINT	L#0	Cam end

Effects of the setting

FM 452 first shifts the on-triggering edge and then the off-triggering edge of the cam. This sequence does not depend on the direction in which the cam is shifted.

Special case:

The sequence described above may briefly generate an inverse cam if the new cam start is greater than the old cam end.

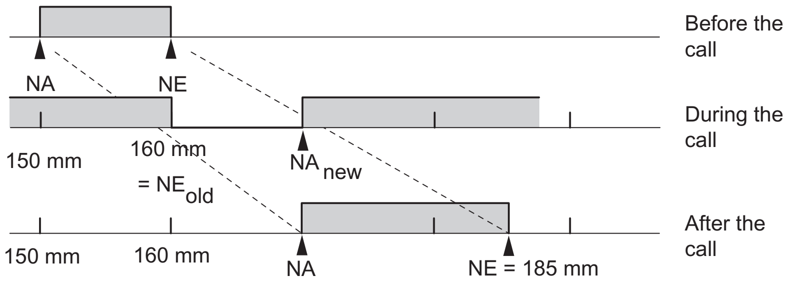


Figure 10-1 Step-by-step change of the cam edges

Note

If a process interrupt has been enabled for this cam, FM 452 can trigger one or two process interrupts when it detects the inverse cam, depending on the parameterization.

Changes to the on and/or off edge can cause skipping of a cam edge, or of the entire cam.

For information on the switching characteristics of timing cams, refer to chapter "Influence of settings on the switching characteristics of time-based cams (Page 105)".

Cam status changes that would normally trigger a process interrupt can be lost.

Reading modified values

You can read modified values by calling one of the jobs CAM1RD_EN to CAM8RD_EN.

Canceling the setting

The modified values are lost when you restart the module.

10.6 Perform "Fast Cam Parameter Change"

Definition

The "Fast cam parameter change" setting can be used to modify a group of up to 16 cams while the system is in RUN.

Requirements

The cams you want to modify must be valid.

Programming steps

1. Enter the number of cams to be modified in the channel DB.
2. Enter the number of the first cam to be modified in the channel DB.
3. Set the trigger bits for the required modifications.
4. Declare the new values at the channel DB.
5. Repeat steps 2 to 4 for each cam to be modified.
6. Set the relevant trigger bit in the channel DB.

Data used in the channel DB

Address Absolute	Name	Type	Start value	Comment
37.0	CH16CAM_EN	BOOL	FALSE	1 = write fast cam parameter change settings (16 cams)
176.0	C_QTY	BYTE	B#16#0	Number of cams to modify
177.0	DIS_CHECK	BOOL	FALSE	1 = disable data check

Address Relative	Name	Type	Start value	Comment
+0.0	CAM_NO	BYTE	B#16#0	Number of the cam to modify
+1.0	C_EFFDIR	BOOL	FALSE	1 = change the effective direction of the cam
+1.1	C_CBEGIN	BOOL	FALSE	1 = change the cam start to the value CBEGIN
+1.2	C_CEND	BOOL	FALSE	1 = change the cam end / activation time to the value CEND
+1.3	C_LTIME	BOOL	FALSE	1 = change the rate time to the value in LTIME
+1.4	CAM_OFF	BOOL	FALSE	1 = deactivate the cam during cam modification
+1.5	EFFDIR_P	BOOL	FALSE	1 = positive effective direction (plus)

Address Relative	Name	Type	Start value	Comment
+1.6	EFFDIR_M	BOOL	FALSE	1 = negative effective direction (minus)
+2.0	CBEGIN	DINT	L#0	New cam start
+6.0	CEND	DINT	L#0	New cam end / new activation time
+10.0	LTIME	INT	0	New rate time

Deactivating cams during modification

To maintain consistency, always deactivate the cam (CAM_OFF) when modifying its start and end settings.

Data validation by the module

Use the DIS_CHECK (channel DB) parameter to specify whether or not to disable the validation of transferred data by FM 452. If you disable data validation, you must ensure that only valid values are being transferred. Any input of invalid values without validation can lead to unexpected response of the module.

- FALSE: The module validates all data to be transferred.
- TRUE: Data validation with regard to the cam parameters is disabled. This allows for faster activation of the data to be changed on FM 452.

  Regardless of this setting, the module always checks whether

  – the axis is parameterized

  – the number of cams to be changed (C_QTY) is valid

  – the cam (cam number) to be changed is valid.

The data is only activated on the module after having been validated and found faultless.

Any faulty data is rejected.

Effects of the setting

Note

For information on the switching characteristics of timing cams, refer to chapter "Influence of settings on the switching characteristics of time-based cams (Page 105)".

Reading modified values

You can read the modified values by calling one of the jobs CAM1RD_EN to CAM8RD_EN.

Canceling the setting

The modified values are lost when you restart the module.

10.7 Executing "Length measurement" and "Edge detection"

Definition

The "length measurement" and "edge detection" let you determine the length of a part.

Length measurement and edge detection are active and remain active until you disable these functions, or select a different measuring method. If you select both measuring methods in parallel, FC CAM_CTRL enables length measurement.

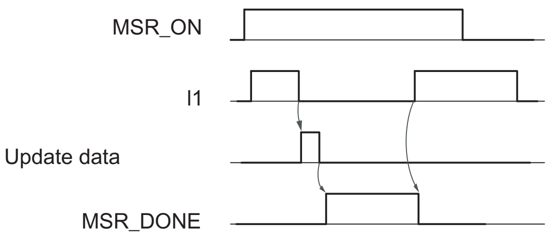
Requirements

A **bounce-free** switch must be connected to input I1.

Sequence of settings

Depending on the type of measurement, FM 452 updates the data on the module at a different time. FM 452 reports each update at a parameter on the checkback interface.

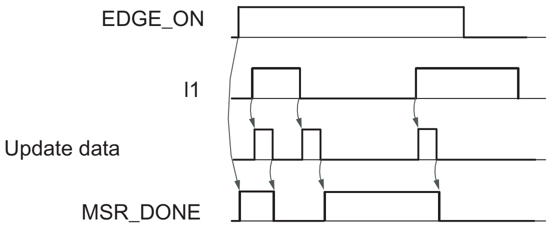
Sequence of the length measurement:



1. Set the function switch for "length measurement".
2. A positive edge at input I1 starts the length measurement.
3. The negative edge at input I1 stops the current measurement. FM 452 updates the start value, end value, and length data.
4. FM 452 reports the data update if parameter MSR_DONE is set. The parameter indicates completion of the measurement. The results of the measurement can be read out.
5. The next start of a measurement at the positive edge at I1 resets the MSR_DONE parameter.

FM 452 does not update the data if the setting is disabled during a length measurement. The MSR_DONE parameter remains reset.

Edge detection sequence:



1. Enter a value for the minimum edge interval in the parameter DB. Write and enable the machine parameters.
2. Set the "edge detection" function switch. This sets the MSR_DONE parameter.
3. The positive edge at input I1 starts edge detection. The measurement results are updated and can be read out; the start value of the measurement is entered; the end value and length assume the value -1.
4. After the update, FM 452 reports the change by resetting the MSR_DONE parameter.
5. The negative edge at input I1 stops the current measurement. FM 452 updates the data for the end value of the measurement and length.
6. After the update, FM 452 reports the change by setting the MSR_DONE parameter. The results of the measurement can be read out.
7. The next start of a measurement at the positive edge at I1 resets the MSR_DONE parameter.

FM 452 does not update the data if the setting is disabled during edge detection. The MSR_DONE parameter remains reset.

Data used in the channel DB

Address	Name	Type	Start value	Comment
25.1	MSR_DONE	BOOL	FALSE	1 = length measurement completed
34.0	EDGE_ON	BOOL	FALSE	1 = edge detection on
34.2	MSR_ON	BOOL	FALSE	1 = length measurement on
38.2	MSRRD_EN	BOOL	FALSE	1 = read measured values
112.0	BEG_VAL	DINT	L#0	Start value
116.0	END_VAL	DINT	L#0	End value
120.0	LEN_VAL	DINT	L#0	Length

Data used in the parameter DB

Address	Name	Type	Start value	Comment
4.0	EDGEDIST	DINT	L#0	Minimum edge interval for edge detection Range: 0 ... 1 000 000 000 μm

The minimum edge interval is used to define a range after the start of a measurement is detected during edge detection. The measurement will be discarded if the measuring operation ends within this range.

The start of the measurement is not reported unless the "minimum edge interval" has been traveled.

Conditions of length measurement

- The CPU program requires an interval of sufficient length between the off and on edges at input I1 in order to be able to evaluate the result of the measurement before a new measurement is triggered.
- The minimum interval between the positive and negative edges at input I1 and between the negative edge and the next positive edge at input I1 must be greater than 2 ms.

Process interrupts

The start and end of a measurement can be reported by means of process interrupt (see chapter "Basics of Programming an FM 452 (Page 37)").

Faulty measurement

The FM 452 returns the length value -1 in the event of a faulty length measurement/edge detection.

A "length measurement" or "edge detection" may only perform up to 126 zero transitions in one direction. The zero point represents the rotary axis transition from the end of rotary axis value to 0, or vice versa. The FM 452 reports a faulty "length measurement" or "edge detection" if it detects more than 126 zero transitions in one direction, regardless of whether or not it then detects any zero transitions in the opposite direction.

A length measurement is also considered faulty if:

- The length measured at a rotary axis is greater than 2^{31}
- The on and off edges are detected simultaneously by FM 452 (for example, caused by switch bounce).

Shift of the coordinate system during length measurement

Conditions under which a shift of the coordinate system will influence the measured length:

- You are using an incremental encoder or proximity switch, or operate the FM 452 in simulation mode.
- You are executing a "set reference point" or "retrigger reference point" function while a length measurement is active.

Example

To utilize the above mentioned influences on the measured length:

Your system always develops slip when you perform a length measurement.

The retrigger reference point function can be used to correct this slip in order to output correct length measurement values.

10.8 Execute "Retrigger reference point"

Definition

The "Retrigger reference point" setting can be used to synchronize the axis as a reaction to a recurring external event.

The setting remains active until you deactivate it.

Requirements

- You are using an incremental encoder or an initiator.
- The external event may represent the zero mark signal of an incremental encoder or reference point switch at input I2.

Programming steps

1. Set the value for the reference point coordinate at the parameter DB.
2. Set the type of "retrigger reference point" at the parameter DB.

   Options:

   – Only the zero mark of the encoder is evaluated (RETR_TYPE = 7).

   – Only the reference point switch is evaluated (RETR_TYPE = 6).

   – Evaluation of the zero mark signal

     in positive direction: evaluation of the first positive edge of the zero mark after passing the reference point switch in positive direction (RETR_TYPE = 0).

     In the negative direction: evaluation of the first negative edge of the zero mark after passing the reference point switch in negative direction (RETR_TYPE = 1).
3. Write and enable the machine parameters.
4. Set the function switch in the channel DB.

Data used in the channel DB

Address	Name	Type	Start value	Comment
34.3	REFTR_ON	BOOL	FALSE	1 = retrigger reference point
25.0	SYNC	BOOL	FALSE	1 = axis is synchronized

Data used in the parameter DB

Address	Name	Type	Start value	Comment
44.0	REFPT	DINT	L#0	Reference point coordinate
52.0	RETR_TYPE	DINT	L#0	Type of reference point retriggering

Effects of the setting

- FM 452 evaluates the zero mark and reference point switch depending on the direction of movement of the axis.
  – It evaluates the positive edges if the axis moves in positive direction.
  – It evaluates the negative edges if the axis moves in negative direction.
- It sets the actual position to the value of the reference point coordinate.
- The working range is physically shifted on the axis.
- The various points retain their original values, but are now at new physical positions.
- Cam status changes that would normally trigger a process interrupt can be lost.
- The SYNC bit is set in the checkback signals.

Note

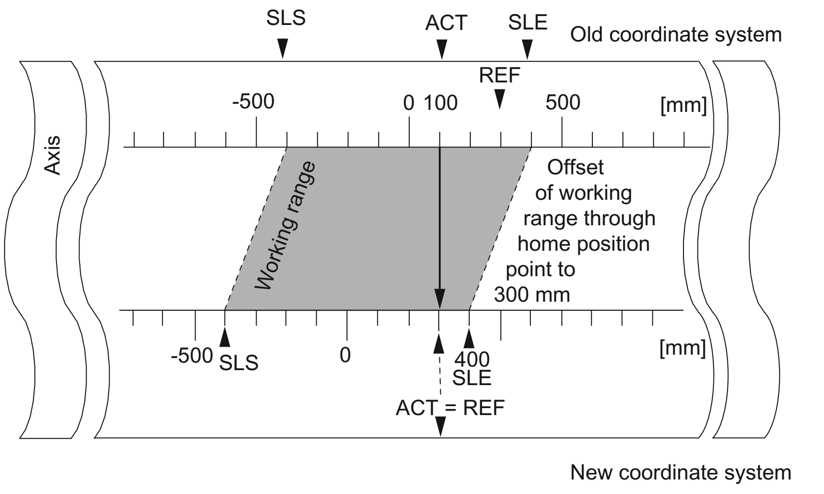
For information on the switching characteristics of timing cams, refer to chapter "Influence of settings on the switching characteristics of time-based cams (Page 105)".

Example

Rules for the example:

- The module evaluates the positive edges of the reference point switch and zero mark signals (axis moving in positive direction).
- Value of the reference point coordinate = 300 mm.
- No zero offset is active at the time of execution.

Table 10- 5 Shift of the axis working range by "retrigger reference point"

Retrigger reference point	SLS [mm]	REF [mm]	ACT [mm]	SLE [mm]
	-400	300	100	400
	-400	300	300	400

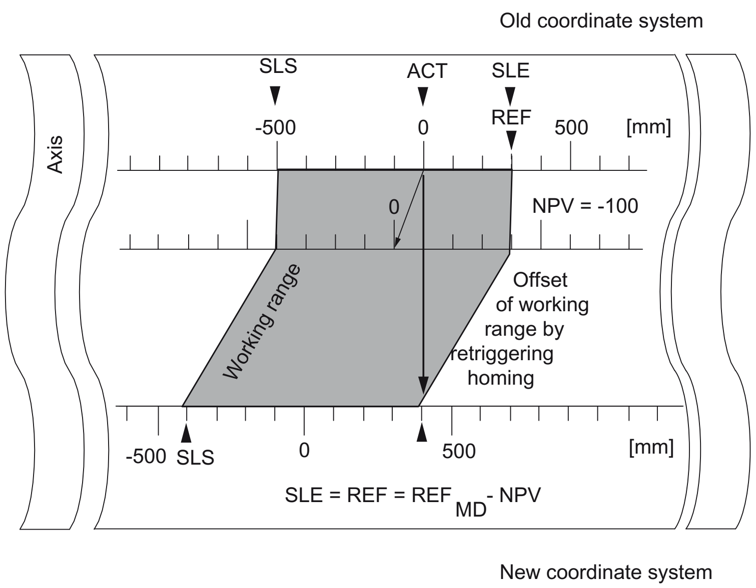
Inclusion of a zero offset

Any active zero offset is included in the retrigger reference point setting. The reference point coordinate setting is thus calculated according to the formula:

Ref = Ref_{MD} - Zero offset

Ref_{MD} is the value of the reference point coordinate stored in the machine parameters.

Table 10- 6 Shift of the axis working range by "Retrigger reference point" when zero offset is active

Retrigger reference point	SLS [mm]	REF [mm]	ACT [mm]	SLE [mm]
	-500	300	0	300
	-400	400	100	400
	-400	400	400	400

10.9 Execute "Disable software limit switch"

Definition

Use the "Disable software limit switches" function to disable monitoring of the software limit switches at a linear axis.

The setting remains active until you cancel it. This re-enables the originally programmed software limit switches.

Programming steps

Set the function switch at the channel DB.

Data used in the channel DB

Address	Name	Type	Start value	Comment
34.4	SSW_OFF	BOOL	FALSE	1 = software limit switch disabled

Data used in the parameter DB

Address	Name	Type	Start value	Comment
64.0	SSW_STRT	DINT	L#-1000000000	Start software limit switch
68.0	SSW_END	DINT	L#1000000000	End software limit switch

Effects of the setting

- simulation
  - Simulation mode stops when the axis passes a software limit switch.
  - You can resume simulation mode by enabling software limit switch monitoring. The axis moves in the defined direction.
- Zero offset when monitoring is disabled

  With a zero offset setting, and software limit switches operating within traversing range limits, the actual value may still be out of the permissible number range.
- Cams not operating within the programmed range of the software limit switches can be activated.

CAUTION

Risk of material damage!

If you restrict the traversing range using the software limit switches as a safety measure, deactivating the limit switches can result in serious damage to equipment.

In the planning and engineering phases for your plant, you should be certain that the drive is capable of covering the entire physical range.

10.10 "Simulation"

Definition

The "Simulation" setting allows you to activate the cam control system without connected encoders.

Programming steps

1. Set the simulation velocity at the parameter DB.
2. Write and enable the machine parameters.
3. Set either a positive or negative simulation direction at the channel DB.
4. Set the function switch in the channel DB.

Data used in the channel DB

Address	Name	Type	Start value	Comment
15.2	DIR_M	BOOL	FALSE	1 = simulation in negative direction
15.3	DIR_P	BOOL	FALSE	1 = simulation in positive direction
34.1	SIM_ON	BOOL	FALSE	1 = simulation on

Data used in the parameter DB

Address	Name	Type	Start value	Comment
84.0	SIM_SPD	DINT	L#0	Simulation velocity

Effects when simulation mode is activated

- Encoder signals will be ignored.
- The encoder input monitoring functions are disabled.
- All encoder errors reported will be reset.
- FM 452 simulates axis motion at a constant simulation velocity.
- Cam processing is disabled while simulation mode is active. However, you can then enable these operations again. Synchronism will be retained in this case.
- Starting at the current actual value, the actual position value changes dynamically based on the simulation velocity and direction.

Effects when simulation mode is deactivated

- Cam processing will be disabled.
- The synchronization of an incremental encoder or proximity switch will be cleared. The actual value is reset to the value of the reference point coordinate.
- The module reports the actual position value which corresponds with the absolute encoder value. The module then evaluates the encoder signals as defined at the machine parameters.

Limit values

The min./max. simulation velocity depends on the resolution (see chapter "Machine data of the axis (Page 75)").

Velocity

The module may operate with differences between the online velocity and offline settings (see chapter "Machine data of the axis (Page 75)").

10.11 Read "count values of counter cam tracks"

Definition

The "count values of counter cam tracks" is used to read the actual count values.

Programming steps

1. Set the counter cam tracks and the high limit of count values at the machine parameters.
2. Write and enable machine data.
3. Enable the count function.
4. The module sets the count value to its high limit.
5. The count value decrements by the count of 1 at each positive edge of the track result signal.
6. Set the trigger bit at the channel DB to read the count values.
7. The module writes both count values to the channel DB. The module outputs a 0 value for tracks not programmed for operation as count cam track.
8. The module sets track flag bit = 1 at the counter cam track when the count value = 0.
9. It resets the track flag bit = 0 at the next negative edge of the track result signal, and resets the counter to its high limit.

Data used in the channel DB

Address	Name	Type	Start value	Comment
15.5	CNTC0_EN	BOOL	FALSE	1 = enable count function of counter cam track 0
15.6	CNTC1_EN	BOOL	FALSE	1 = enable count function of counter cam track 1
38.3	CNTTRC_EN	BOOL	FALSE	1 = read count values of counter cam tracks
124.0	CNT_TRC0	INT	0	Actual count value of counter output cam track 0
126.0	CNT_TRC1	INT	0	Actual count value of counter output cam track 1

Data used in the parameter DB

Address	Name	Type	Start value	Comment
99.0	SPEC_TRC0	BOOL	FALSE	1 = track 0 is counter cam track
99.1	SPEC_TRC1	BOOL	FALSE	1 = track 1 is counter cam track
100.0	CNT_LIM0	DINT	L#2	Upper count value for counter output cam track 0
104.0	CNT_LIM1	DINT	L#2	Upper count value for counter output cam track 1

10.12 Read "position and track data"

Definition

The "position and track data" function can be used to read the actual position value, the velocity, and the track flag bits. The track flag bits are recorded before being logically linked to machine and channel data.

The algorithm implemented in FM 452 calculates velocity changes greater than 1 pulse / 4 ms. The indicated velocity includes this inaccuracy, and is thus unsuitable in particular for closed-loop control. The internal velocity value used for dynamic cam control offers a higher precision.

Programming steps

1. Set the trigger bit at the channel DB.
2. The data will be saved to the channel DB.

Data used in the channel DB

Address	Name	Type	Start value	Comment
38.4	ACTPOS_EN	BOOL	FALSE	1 = read position and track data
128.0	ACTPOS	DINT	L#0	Actual position
132.0	ACTSPD	DINT	L#0	Actual velocity
136.0	TRACK_ID	DWORD	DW#16#0	Track flag bits of tracks 0 to 31

10.13 Read "Encoder data"

Definition

The "encoder data" setting an be used to read actual encoder data, and the value for absolute encoder adjustment.

Requirements

The value for absolute encoder adjustment can be read after "set reference point" is configured (see chapter "Determining the correct absolute encoder adjustment (Page 82)").

Programming steps

1. Set the trigger bit at the channel DB.
2. The data is stored in the channel DB.

Data used in the channel DB

Address	Name	Type	Start value	Comment
38.5	ENCVAL_EN	BOOL	FALSE	1 = read encoder values
140.0	ENCVAL	DINT	L#0	Encoder value / counter value (internal representation)
144.0	ZEROVAL	DINT	L#0	Counter value at the last zero mark (internal representation)
148.0	ENC_ADJ	DINT	L#0	Absolute encoder adjustment

10.14 Read "Cam and track data"

Definition

The "cam and track data" setting can be used to read the current cam and track flag bits and the position. The track flag bits are detected before they are logically linked to machine and channel data.

Sequence of settings

1. Enter type ID = 1 at the FM_TYPE parameter of the channel DB. This allows you to read 24 bytes of cam and track data.

   If you enter type ID = 0, only the cam flag bits (16 bytes) will be read.
2. The data will be saved in the channel DB.

Data used in the channel DB

Address	Name	Type	Initial value	Comment
12.0	FM_TYPE	BOOL	FALSE	1 = FM 452
38.6	CAMOUT_EN	BOOL	FALSE	1 = read cam and track data
152.0	CAM_00_31	DWORD	DW#16#0	Cam flag bits for cams 0 to 31
156.0	CAM_32_63	DWORD	DW#16#0	Cam flag bits for cams 32 to 63
160.0	CAM_64_95	DWORD	DW#16#0	Cam flag bits for cams 64 to 95
164.0	CAM_96_127	DWORD	DW#16#0	Cam flag bits for cams 96 to 127
168.0	TRACK_ID1	DWORD	DW#16#0	Track flag bits of tracks 0 to 31
172.0	ACTPOS1	DINT	L#0	Current position

10.15 Setting "Control signals for the cam controller"

Definition

The "control signals for the cam control system" setting can be used to enable cam processing and the tracks.

Programming steps

1. Set the required bits at the channel DB
2. The data is transferred to the module at every call of FC CAM_CTRL.

Data used in the channel DB

Address	Name	Type	Start value	Comment
15.4	CAM_EN	BOOL	FALSE	1 = enable cam processing
16.0	TRACK_EN	WORD	W#16#0	Enable cam tracks 0 to 15 Bit 0 = track 0

Effects

Cam processing is started or stopped depending on the enable status.

The identifier bits of enabled tracks are transferred to the track signals and digital outputs.

10.16 Querying "Return signals for the cam controller"

Definition

The "return signals for the cam controller" setting informs you about the current state of the cam control and track signals. Consistency between the reported position and track signals is not guaranteed.

Programming steps

The data will be saved to the channel DB at each call of FC CAM_CTRL.

Data used in the channel DB

Address	Name	Type	Start value	Comment
23.4	CAM_ACT	BOOL	FALSE	1 = cam processing busy
26.0	ACT_POS	DINT	L#0	Current position of axis
30.0	TRACK_OUT	DWORD	DW#16#0	Current signals of tracks 0 to 31 Bit 0 = track 0

10.17 Querying the "return signals for diagnostics"

Definition

The "checkback signals for diagnostics" setting is used to report diagnostics events.

Programming steps

1. The module sets the DIAG bit in the checkback interface each time it writes a new entry to the diagnostics buffer. Error events belonging to any of the error classes listed in Appendix "Data blocks / error lists (Page 177)" are logged to the diagnostics buffer.
2. The module sets the DATA_ERR bit in the checkback interface when it detects faulty data in a write job. The cause of the error is logged to the diagnostics buffer.
3. The data is stored in the channel DB.
4. If the diagnostics buffer is read by FC DIAG or by the error evaluation of the programming interface, the module sets the DIAG bit back to 0.

Data used in the channel DB

Address	Name	Type	Start value	Comment
22.2	DIAG	BOOL	FALSE	1 = diagnostics buffer modified
22.4	DATA_ERR	BOOL	FALSE	1 = data error

Encoders 11

11.1 Incremental encoder

Connectable incremental encoders

The module supports incremental encoders outputting two pulses with 90° phase shift, and with or without zero mark signal:

- Encoders with asymmetrical 24 V output signals
  - Limit frequency = 50 kHz
  - cable length max. 100 m
- Encoders with symmetrical output signals and 5 V differential interface conforming to RS422
  - Limit frequency = 1 MHz
  - With 5 V supply voltage: cable length max. 32 m
  - With 24 V supply voltage: cable length max. 100 m

Note

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

Signal shapes

The diagram below shows the signal shapes of encoders with asymmetrical and symmetrical output signals.

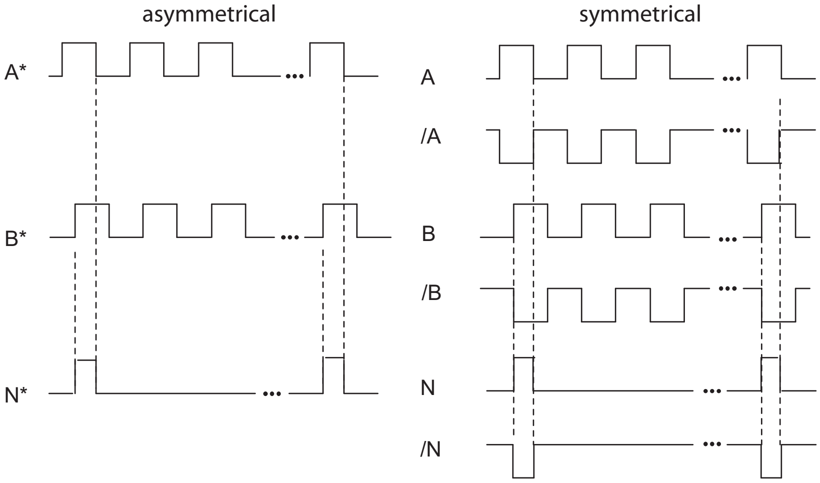


Figure 11-1 Signal shapes of incremental encoders

Signal evaluation

Increments

An increment identifies a signal period of signals A and B of an encoder. This value is specified in the technical data and/or on the rating plate of the encoder.

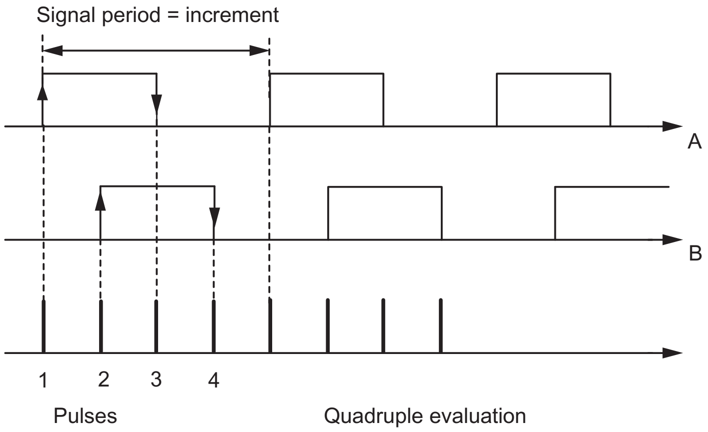


Figure 11-2 Increments and pulses

Pulses

FM 452 evaluates all 4 edges of the A and B (see the diagram) signals in each increment (quadruple evaluation).

1 increment (encoder default) = 4 pulses (FM evaluation)

Reaction times

FM 452 reaction times for connected incremental encoders:

Minimum reaction time = cam cycle time + switching time of the connected switching elements

Maximum reaction time = 2 x cam cycle time + switching time of the connected switching elements

Example

Example of the min./max. reaction time with a load of 16 cams:

- Cam cycle time: approx. 20 μs
- Response time of the hardware: approx. 150 μs

Minimum reaction time = 20 μs + 150 μs = 170 μs

Maximum reaction time = 2 x 20 μs + 150 μs = 190 μs

Note

You can compensate the reaction time with appropriate parameter settings for the cams or using dynamic adjustment.

Flat gain

The flat gain is equivalent to the difference between the min./max. reaction time. For incremental encoders this is:

Flat gain = cam cycle time

Note

If the switching time of the hardware on the FM 452 and the switching time of the connected switching elements can be ignored, then reliable switching of the cam is guaranteed if the cam is longer than the distance traveled within the cam cycle time.

11.2 Proximity switches

Definition

Proximity switches are simple switches which output pulse-shaped signals, and do not return a directional signal. You define the direction based on the machine data for selecting the proximity switch.

CAUTION
Risk of material damage! Incorrect direction settings may cause serious errors in the system (for example, faulty control of aggregates). Check the direction settings in the commissioning phase, and whenever you replace proximity switch.

Supported proximity switches

FM 452 supports the following proximity switches:

- Proximity switches with 24 V signal level (proximity switches) and limit frequency = 50 kHz
- max. 100 m cable length

Signal evaluation

The module counts the positive edges at signal A* of the proximity switches.

11.3 Absolute encoders

Single-turn and multiturn encoders

Absolute encoders are divided into the categories:

- Single-turn encoder

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

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

Supported absolute encoders

Absolute encoders with serial interface. Position data is transferred synchronously in accordance with the SSI protocol (**S**ynchronous**S**erial**I**nterface). FM 452 only supports the GRAY code. Due to the arrangement of the data bits in the transferred frames, the data formats "fir tree", "half fir tree" and "right-justified" are used.

Encoder type	Message frame length / type
Single-turn encoder	13-bit half fir tree
Single-turn encoder	13-bit right-justified
Single-turn encoder	25-bit right-justified
Multiturn encoder	25-bit fir tree
Multiturn encoder	25-bit right-justified
Listen in	Fir tree
Listen in	Right-justified
Special setting: Multiturn encoder in single-turn mode	25-bit half fir tree

Data Transmission

The data rate for data transmission depends on the cable length (see chapter "Technical data (Page 168)").

Evaluation of absolute encoder pulses

1 increment (encoder default) = 1 pulse (FM evaluation)

Listen in

"Listen in" means: An absolute encoder is operated in parallel on two modules (for example, FM 451 and FM 452). The FM 451 positioning module is the master and clocks the absolute encoder, while the FM 452 electronic cam control system acts as the slave listening in to the signals of the SSI frame.

Set "Increments/Encoder Revolution" and "Number of Revolutions" to the master setting. The baud rate is irrelevant. Depending on the encoder type, select "Listen in" or "Listen in Right-Justified" for "Frame length".

Wiring Listen in

The diagram below is based on the example of an FM 451 and FM 452 and shows how to wire the absolute encoder so that the FM 452 listens.

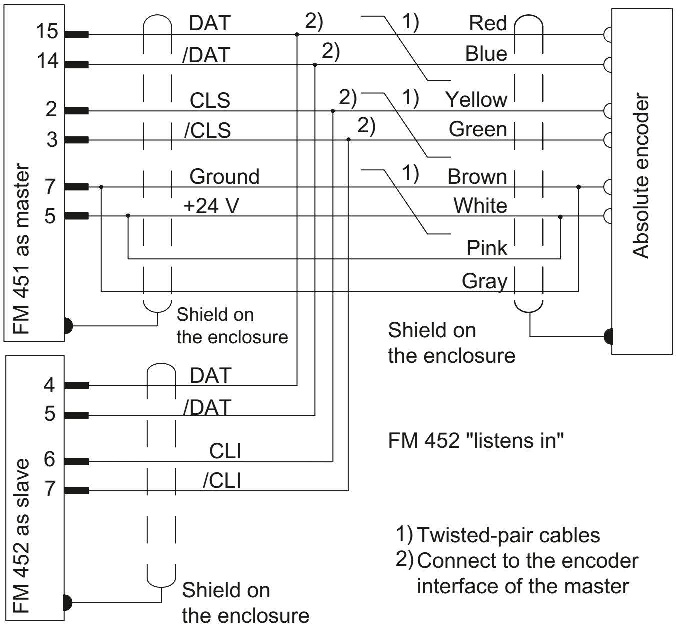


Figure 11-3 Connection of absolute encoders (SSI)

Note

If you want the FM 452 to listen in, you must connect the ground (M) of the encoder supply of the master (for example FM 451: Front connector, pin 48) and of the slave module (FM 452: Front connector, Pin 48) by means of low-impedance connection to CPU ground potential.

Response times

FM 452 provides the following response times for absolute encoders:

Minimum response time = frame cycle time + cam cycle time + switching time of the connected switching elements

Maximum response time = 2 x frame cycle time + monostable time + 2 x cam cycle time + switching time of the connected switching elements

For programmable absolute encoders:

Maximum response time = frame cycle time + monostable time + 2 x cam cycle time + switching time of the connected switching elements +1/max. step sequence rate

Monostable time

The following limit values apply to the monostable time:

- Minimum monostable time: > 15 μs
- Maximum monostable time: < 64 μs

Encoders with values outside of those specified limits are not permitted.

Frame cycle times

The frame cycle times depend on the transmission rate:

Baud rate	Frame cycle time for 13 bits	Frame cycle time for 25 bits
0.125 MHz	112 μs	208 μs
0.250 MHz	56 μs	104 μs
0.500 MHz	28 μs	52 μs
1.000 MHz	14 μs	26 μs

Example of response times

The following example shows how to calculate the minimum and maximum response time. A programmable encoder is not used in the example.

- Cam cycle time: approx. 20 µs for max. 16 cams
- Hardware switching time: Approx. 150 µs
- Frame cycle time: 26 µs at a transmission rate of 1 MHz (25-bit frame)
- Monostable time: 20 µs (depends on the encoder: typical 20 µs to 40 µs)

Maximum response time = 26 µs + 20 µs + 150 µs = 196 µs

Maximum response time = 2 x 26 µs + 20 µs + 2 x 20 µs + 150 µs = 262 µs

---

Note

You can compensate for the response time by programming the cams accordingly, or using dynamic adjustment.

Flat gain

The flat gain is equivalent to the difference between the min./max. response time.

With an absolute encoder it is as follows:

Flat gain = cam cycle time + frame cycle time + monostable time

With a programmable absolute encoder, it is as follows:

Flat gain = cam cycle time + frame cycle time + monostable time
+ 1/max. step sequence frequency

Note

If the switching time of the hardware on the FM 452 and the switching time of the connected switching elements can be ignored, then reliable switching of the cam is guaranteed if the cam is longer than the distance traveled within the cam cycle time.

Diagnosis 12

12.1 Options of error diagnostics

Overview

- With the programming device/PC, you can read the diagnostic buffer using the **Test > Error Evaluation** programming interface.
  - You will see the error class and error number along with plain text.
- You can evaluate errors in your program. The following options are available:
  - The return values (RET_VAL) of the linked FC as a group display for errors that occurred while the FC was being executed.
  - The error bits of the jobs as a group display for errors that occurred while executing a job.
  - The error bit DATA_ERR as a group display for an error detected by the FM 452 during a write job.
  - The error flag in JOB_ERR, for the cause of error in the communication between the FC and FM 452.
  - FC CAM_DIAG for reading out the diagnostic buffer of the FM 452. Here, you can find out the causes of errors in jobs and asynchronous events (operating errors, diagnostic errors).
  - Diagnostics interrupts for fast reaction to events.

12.2 Meaning of the error LEDs

Display

The status and error display indicate various error states. The LEDs are lit, even with errors that occur briefly, for at least 3 seconds.

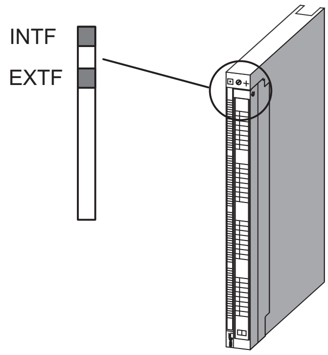


Figure 12-1 Status and Fault/Error Indicators of the FM 452

Display	Meaning	Notes
INTF (red) LED - ON	Group error for internal error	This LED indicates the following error states on the FM 452: • Hardware interrupt lost • Watchdog expired • FM 452 is not configured. • Incorrect FM 452 parameter assignment (only when parameters assigned with SDB)
EXTF (red) LED - ON	Group error for external errors	This LED indicates the following error states: • No external 24 V auxiliary supply • Front connector missing • Encoder wire break • Process error • Absolute encoder frame error • Incremental encoder pulse missing or zero mark signal missing
INTF EXTF	Module defective	All outputs are disabled. The module must be replaced.

12.3 Diagnostics interrupts

12.3.1 Enable diagnostics interrupts

Alarm Processing

The FM 452 can trigger hardware and diagnostic interrupts. Process those interrupts in an interrupt OB. If an interrupt is generated and the corresponding OB is not loaded, the CPU changes to STOP (refer to the manual *Programming with STEP* 7).

You enable the servicing of diagnostic interrupts as follows:

1. Select the module in HW Config
2. Use the **Edit > Object Properties > Basic Parameters** command to enable the diagnostic interrupt.
3. Save and compile the hardware configuration.
4. Download the hardware configuration to the CPU.

Overview of the Diagnostic Interrupts

Events and errors triggering a diagnostic interrupt:

- Process error
- Incorrect machine data (when programmed with SDB)
- Incorrect cam data (when programmed with SDB)
- Diagnostics errors

For detailed information on this error, refer to the appendix.

See also

Data and Structure of the Diagnostic DB (Page 186)

12.3.2 Reaction of FM 452 to errors with diagnostics interrupt

Reactions

- Cam processing will be disabled.
- The synchronization will be cleared by the following diagnostic interrupts:
  - Front connector missing, external power supply missing
  - Zero mark error detected, cable fault (5 V encoder signals)
  - Traversing range exceeded (indicated by a process/hardware error)
  - Set actual value cannot be executed (indicated by a process error).
- With one exception, control signals are no longer processed

  Exception:

  After the software limit switch has been passed, a reversal of direction is still possible in simulation mode.
- Function switch and job processing continues.

FM 452 detects an error ("incoming")

A diagnostic interrupt is an "incoming" event if at least one error is pending. If not all errors were cleared, the queued errors are reported once again as "incoming" events.

Sequence:

1. FM 452 detects one or several errors, and generates a diagnostics interrupt. The "INTF" or "EXTF" LED lights up. The error event is logged to the diagnostics buffer.
2. The CPU operating system calls OB82.
3. You can now evaluate the start information of OB82.
4. The OB82_MOD_ADDR parameter shows the interrupt triggering module.
5. For further information, call FC CAM_DIAG.

FM 452 detects a transition to error-free state ("outgoing")

A diagnostics interrupt is only registered "outgoing" if all errors on the module are cleared.

Sequence:

1. FM 452 detects that all errors have been cleared, and generates a diagnostics interrupt. The LED "INTF" or "EXTF" is no longer lit. The diagnostics buffer remains unchanged.
2. The CPU operating system calls OB82.
3. The OB82_MOD_ADDR parameter shows the interrupt triggering module.
4. Evaluate the OB82_MDL_DEFECT bit.

   When this bit is "0", no errors are present on the module. You can close the evaluation session at this point.

Diagnostics interrupt control by CPU states

- When the CPU is in the STOP state, diagnostic interrupts from the FM 452 are disabled.
- If none of the queued errors were cleared while the CPU was in STOP, the FM 452 reports all these errors as "incoming" event at the next CPU transition to RUN.
- If all existing errors have been eliminated in the CPU STOP state, then the error-free FM 452 state is not signaled with a diagnostic interrupt after the CPU changes to RUN.

Examples 13

13.1 Introduction

Example project folder

The FM 352/FM 452 software package you installed contains two example projects showing you several typical applications based on a number of selected functions.

The German example project for the FM 452 is located in the folder ...\STEP7\EXAMPLES\zEn19_02.

This folder contains several S7 programs of varying complexity and objectives.

13.2 Preconditions

Overview

Requirements:

- A completely wired S7 station, consisting of a power supply module, a CPU and an FM 452 module, version V5 or higher. The characteristics of earlier module versions may deviate from the description.
- You have correctly installed STEP 7 and the configuration package for the FMx52 on your programming device/PC.
- The PG is connected to the CPU.

13.3 Preparing the examples

Procedure

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

1. Open the **\STEP7\EXAMPLES\zEn19_02_FMx52___Prog** example project in SIMATIC Manager, then copy it under a suitable name to your project folder.
2. Insert a station in this project according to your hardware configuration.
3. Complete the hardware configuration with HW Config and save the configuration.
4. Select an example program and copy its block folder to your station.
5. Assign parameters for the FM 452 in HW Config using the instructions provided in the Manual FM 452 Getting Started, section FM 452 parameter assignment (http://support.automation.siemens.com/WW/view/en/1407404).
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 an FM 452 (Page 37)).
7. Download the hardware configuration to your CPU.
8. Download the blocks to your CPU.
9. To try out the next example, go to step 4.

13.4 Displaying the code of the examples

Display

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

Select the view with "Symbolic representation", "Symbol selection" and "Comment." If your screen provides sufficient space, you can also open the "Symbol information" view.

13.5 Testing the example

Procedure

After you have successfully completed all necessary entries, download the entire block folder 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.

1. From the variables table, select the "Symbol" and "Symbol Comment" views.
2. Open a variable table.
3. Open the variables table with the configured CPU, and monitor the variables cyclically.

This updates the variables dynamically when the CPU is in RUN mode.

All the examples require that the machine data and cam data were entered and saved using the parameter assignment interface. This allows you to work through the examples sequentially.

13.6 Reusing an example project

Restrictions

The code of the samples in neither optimized nor designed for all eventualities.

Error evaluation is not programmed in detail in the sample programs in order to avoid the programs becoming unwieldy.

13.7 Sample program 1 "Getting Started"

Objective

In this sample, you commission your cam controller after you have assigned parameters for it in the parameter assignment interface according to the "Getting Started" manual.

The sample extends the program shown in the "Integration in the User Program" chapter of the getting started by adding error evaluation.

Requirements

You have assigned the cam controller parameters as described in "Getting Started."

The address of your module is entered correctly at MOD_ADDR in the channel DB.

Startup

Enter the address of your module in channel DB at the MOD_ADDR address.

In the startup OB (OB 100), call FC CAM_INIT to reset all control and checkback signals as well as job management in the channel DB.

Cyclic mode

1. Open the variables table.
2. Go online to the configured CPU to monitor the variables.
3. Transfer the prepared control values.

   The module changes to simulation mode. You can see how the actual value (CAM.ACT_POS) and track signals (CAM.TRACK_OUT) change dynamically.
4. Now change the simulation direction, specify different reference point coordinates, then disable simulation etc. by modifying and transferring the control values.

Error evaluation

Generate a data error by entering a reference point coordinate greater than the end of the rotary axis (for example, 10000000). The CPU goes into STOP. (In a sample, this is the simplest method of indicating an error. You can, of course, program a more sophisticated method.)

Open the hardware configuration and double-click the FM 452. This opens the parameter assignment interface. Select Test > Error evaluation to view the cause of the error.

To clear the error:

1. Enter a valid control value.
2. Switch the CPU to STOP.
3. Switch the CPU to RUN mode.
4. Enable the control values. Control values enabled before the restart of the CPU restarts are reset by the initialization routine in OB100, and thus have no effect.

13.8 Sample Program 2 "Commissioning"

Target

In this example, you commission a cam control system without using the programming interface. You control and monitor the system using the variable tables (VATs).

Requirements

You have programmed the cam control system as described in the "Getting Started" Manual.

The module address is entered at the MOD_ADDR block parameter in the channel and diagnostics DBs.

The included channel DB already contains the DB number (3) of the parameter DB in the PARADBNO parameter.

PARADB included in this example contains default machine and cam parameters.

Startup

In the startup OB (OB100), call FC CAM_INIT to initialize the channel DB. Next, set the trigger bits for all jobs and control signals required after the module has completed its startup.

Cyclic mode

Open the tag tables VAT1 and VAT2, then go online to the configured CPU to monitor the tags.

At VAT1, you can view the changes in the actual position and the track signals. The module is in operation.

In VAT2, you can see the most important entries of the diagnostic buffer of the module. For information on error classes and numbers, refer to the appendix of the manual.

Edit the machine and cam data in DB PARADB, download the DB to the CPU, and then enable the control values in VAT1. This writes the new data to the module and activates them. Faulty data is indicated in VAT2. For information on machine and cam data, refer to the chapter "Machine and cam data (Page 67)".

Error evaluation

Try to generate further errors:

- Define a reference point coordinate which is greater than the end of the rotary axis.
- Switch off the external power supply.
- Delete PARADB from the CPU (online), and then try to write the machine parameters. The error evaluation is intentionally programmed so that the CPU goes to STOP mode. If you update VAT1 once again, the error code for this error is indicated in CAM.JOB_ERR.

See also

Data and Structure of the Diagnostic DB (Page 186)

13.9 Sample program 3 "OneModule"

Objective

In this example, you control a cam controller in a 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 CAM_CTRL call at the end of the user program
- Central error evaluation by CAM_DIAG at the end of the user program
- Evaluation of the diagnostic buffer in conjunction with DATA_ERR

Requirements

You have assigned the cam controller parameters as described in "Getting Started."

The module address is entered at the MOD_ADDR block parameter in the channel and diagnostic DBs.

The included channel DB already contains the DB number (3) of the parameter DB in the PARADBNO parameter.

PARADB included in this example contains default machine and cam data.

Startup

At the startup OB (OB100), set the startup flag (step 0) for the user program in the corresponding instance DB.

Operation

The CPU is in STOP.

- Open variables table VAT1, then transfer the control values.
- Restart the CPU (STOP > RUN).

You can see how the actual position (CAM.ACT_POS), the cam data (CAM.CAM_00_31) and the track signals (CAM.TRACK_OUT) change. You should also observe the step number of the step sequence (PROGDB.STEPNO).

When cam 4 is set (130 degrees), the cam 0 and 1 parameters are assigned the new values you defined in VAT1. You can view the change in the VAT.

Next, the program waits for an external event.

- Once again, transfer the prepared control values from the VAT (this time, PROGDB.SWITCH is evaluated).

The previous values will be restored in the cam data.

The step sequence is completed after this cycle, the step number = -2, and simulation is stopped.

If you want to repeat the entire sequence again, restart the CPU (STOP > RUN). (This procedure is, of course, only acceptable in the sample program.)

If you have not activated the PROGDB.SWITCH switch before a CPU STOP, the parameters of the cam in the parameter DB are not set back to the original values. In this case, once again download the parameter DB to the CPU.

Error evaluation

Any processing error will stop the step sequence and disable simulation mode. Step number -1 will be entered.

Try to generate cam errors the central error evaluation will save to the PROGDB.CAM_ERR bit as group error.

- Set cam positions in VAT1 which are higher than the end of the rotary axis.
- Define negative cam positions in VAT1.

User program (FB PROG)

The user program accesses data in the module-specific data blocks using the <blockname>.<symbolic name> syntax. The user program can thus operate exactly one module. The DB numbers defined at the call of the user program are merely passed on to supply FC CAM_CTRL and FC CAM_DIAG. With this type of programming, you can access DB data using symbolic names. Indirect addressing of several modules is part of sample program 5 "MultiModules."

The user program executes a step sequence as follows:

Step 0: Initialization of the cam controller. Sets the jobs and corresponding data to be executed at a restart of the module. The restart of the module may be triggered by a CPU restart, or by the return of a rack, for example.

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

Step 2: The program continuously reads the cam identifier bits and waits until cam 4 is set.

Step 3: Cams 0 and 1 receive new parameters. To let you view the change, the cam data are read before and after the change and indicated at VAT1.

Step 4: The program waits for the set jobs to be executed.

Step 5: The program waits for the "external" event "switch on" (CAM.SWITCH = 1) that you can set at the VAT.

Step 6: The incoming event resets cams 0 and 1 to the value read in the initialization step.

Step 7: The program waits for the set jobs to be executed.

FC CAM_CTRL and FC CAM_DIAG are called at the end of the step sequence. Output CAM_ERR will be set if the diagnostics function has detected an alarm indicating incorrect cam data.

13.10 Sample program 4 "Interrupts"

Objective

This sample contains a user program with the same task as in sample program 3 "OneModule". In this sample, you are shown how to evaluate a diagnostics interrupt for specific modules, and how to process this in the user program to produce a general module error.

Preconditions

You have programmed the cam controller as described in the "Getting Started."

The module address is entered at the MOD_ADDR block parameter in the channel and diagnostics DBs.

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

The included channel DBs already contain the DB number (3) at the PARADBNO parameter of the parameter DB.

PARADB included in this example contains default machine and cam data.

Startup

At the startup OB (OB100), the startup flag (step 0) for the user program is set in the instance DB.

Operation

As in sample program 3 "OneModule".

Error evaluation

As in sample program 3 "OneModule".

Generate diagnostics interrupts by disconnecting the auxiliary power supply to the module, or by removing the front connector. The module error MOD_ERR and the diagnostic error OB82_ERR are set to 1 and the step number becomes -1. When you eliminate the problem, the error flags are also reset. Cam processing and simulation, however, remain disabled.

User program (FB PROG)

The task is the same as in the sample program 3 "OneModule". However, the block was expanded by adding evaluation of the diagnostics event.

In this sample, no special measures have been taken for restarting after eliminating the error. We have left this up to you as part of the exercise.

Diagnostics interrupt (OB82)

Depending on the address of the module that triggered the interrupt (OB82_MDL_ADDR), the error ID in the corresponding instance DB of the user program is entered in the diagnostics interrupt.

13.11 Sample program 5 "MultiModules"

Objective

This sample contains the same user program as sample program 3 "OneModule", however it is used to operate two modules with different cam parameters. The same copy of the user program is used for both modules. Of course, each module has its own set of data blocks.

Requirements

You have inserted two FM 452 modules and configured these in HW Config.

You have assigned parameters for both cam controllers as described in the "Getting Started" manual.

The address of the relevant module has been entered in the block parameter MOD_ADDR in the channel DBs and diagnostic DBs.

The included channel DBs already contain the DB number (3 or 13) of the corresponding parameter DB in the PARADBNO parameter.

The PARADB and PARADB2 parameter DBs of the sample project contain default machine and cam data for both modules.

A set of variables tables is also prepared for each module.

Startup

At the startup OB (OB100), set the startup flag (step 0) for the user program at both instance DBs.

Operation

The CPU is in STOP.

- Open VAT1 and VAT11, and transfer their control values.
- Restart the CPU (STOP > RUN).

You can see how monitor the change of the actual positions, of cam data, and of the track signals of both modules.

Error evaluation

As in sample program 4 "Interrupts", but separately for each of the two modules..

User program (FB PROG)

The objectives and steps of the user program are as in example program 4 "Interrupts" and in example program 3 "One Module".

The user program is designed for the operation of more than one module, since it indirectly accesses module-specific DBs (channel DB, diagnostic DB, and parameter DB). In addition to transferring the DB numbers specified in the call to FC CAM_CTRL and FC CAM_DIAG, these are also deployed by the user program itself. 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.

Diagnostics interrupt (OB82)

Depending on the address of the module that triggered the interrupt (OB82_MDL_ADDR), the error ID in the corresponding instance DB of the user program is entered in the diagnostics interrupt.

Technical data A

A.1 General technical data

The following technical data are described in the Manual SIMATIC S7-400 Automation System Module Data (http://support.automation.siemens.com/WW/view/en/1117740):

- Standards and licenses
- Electromagnetic compatibility
- Transport and storage conditions
- Mechanical and climatic environmental conditions
- Specifications for insulation tests, protection class, and degree of protection

Observe installation guidelines

SIMATIC products meet the requirements if you follow the installation instructions described in the manuals when installing and operating the equipment.

A.2 Technical data

Technical data

Dimensions and weight	
Dimensions W x H x D (mm)	25 x 290 x 280
Weight	Approx. 650 g
Current, voltage and power	
Current consumption (from the backplane bus)	Max. 500 mA
Power loss	Typically 8.1 W
Current consumption of encoders, digital inputs and outputs from 1L+, 2L+ and 3L+ (no-load)	max. 40 mA (front connector: pin 3, 26, 36)
Supply of digital inputs and outputs	• Supply voltage: 24 V DC (permissible range: 20.4 V to 28.8 V) • Permissible potential difference between the input ground connection M (front connector: pin 48) – and the central grounding point (shield): 60 V AC; 75 V DC – Insulation test voltage: 500 V DC
Encoder supply	• Encoder power supply 5 V – 5.2 V ± 2 % – Max. 300 mA – Short-circuit proof • Encoder power supply 24 V – aux. voltage -1.5 V – Max. 300 mA – Short-circuit proof
Load voltage polarity reversal protection	No

Encoder inputs	
Position acquisition	• Incremental • Absolute
Signal voltages	• Symmetrical inputs: 5 V to RS422 • Asymmetrical inputs: 24 V/typ. 9 mA
Input frequency and cable length for symmetrical incremental encoder with 5 V supply	max. 1 MHz with 32 m shielded cable length
Input frequency and cable length for symmetrical incremental encoder with 24 V supply	max. 1 MHz with 100 m shielded cable length
Input frequency and cable length for asymmetrical incremental encoder with 24 V supply	• max. 50 KHz with 25 m shielded cable length • max. 25 KHz with 100 m shielded cable length
Data transfer rate and cable length for absolute encoders	• max. 125 KHz with 320 m shielded cable length • max. 250 KHz with 160 m shielded cable length • max. 500 KHz with 63 m shielded cable length • max. 1 MHz with 20 m shielded cable length
Listen mode with absolute encoders	Yes
Input signals	• Incremental: 2 pulse trains, 90° phase shift, 1 zero pulse • Absolute: Absolute value; SSI and Gray code • Proximity switch 24 V
Digital inputs	
Number of digital inputs	11
Number of simultaneously controllable digital inputs	11
Electrical isolation	No
Status display	Yes, green LED per input
Input voltage	• 0 signal: -30 ... 5 V • 1 signal: 11 ... 30 V
Input current	• 0 signal: ≤ 2 mA (quiescent current) • 1 signal: 9 mA
Input delay	• 0 → 1 signal: max. 200 μs • 1 → 0 signal: max. 200 μs
Connection of a 2-wire BERO	supported
Unshielded cable length	max. 32 m
Shielded cable length	max. 600 m
Switching rate	max. 500 Hz
Dielectric strength test	to VDE 0160

Digital outputs	
Number of outputs	16
Electrical isolation	No
Status display	Yes, green LED per channel
Output current	• 0 signal: 0.5 mA • High signal: 0.5 A (permissible range: 5 ... 600 mA) • Lamp load: 5 W
Output delay for output current 0.5 A	• 0 → 1 signal: max. 150 μs • 1 → 0 signal: max. 150 μs
Signal level for 1 signal	1L+, 2L+, 3L+: -0.8 V
Controlling of digital inputs	Yes
Control of a counter input	Only conditionally, see note
Short-circuit protection	Yes, electronically clocked
Limit of inductive cut-off voltage	1L+, 2L+, 3L+: -48 V
Switching rate	• resistive load: max. 500 Hz • inductive load: max. 0.5 Hz
Total current of the digital outputs	Simultaneity factor 100%: 8 A
Unshielded cable length	max. 100 m
Shielded cable length	max. 600 m
Dielectric strength test	to VDE 0160

Note

When the 24 V power supply is turned on using a mechanical contact, the FM 452 applies a pulse to the outputs. The pulse may assume a length of 50 μs within the permissible output current range. Make allowances for this factor when using FM 452 in combination with high-speed counters.

Connection diagrams B

B.1 Encoder types

Overview

The table below describes the encoders supported by FM 452. The wiring diagrams for these encoders are described in this section:'

Encoder type	Connecting cable	Remark
Incremental encoder Siemens 6FX 2001-2□□□□	4 x 2 x 0.25 + 2 x 1 mm^2	V_p = 5V, RS 422
Incremental encoder Siemens 6FX 2001-2□□□□	4 x 2 x 0.5 mm^2	V_p = 24V, RS 422
Incremental encoder Siemens 6FX 2001-4□□□□	4 x 2 x 0.5 mm^2	U_p=24V, HTL
Absolute encoder Siemens 6FX 2001-5□□□□	4 x 2 x 0.5 mm^2	U_p=24V, SSI

B.2 Connection Diagram for Incremental Encoder Siemens 6FX 2001-2 (Up=5V; RS 422)

Connection diagram

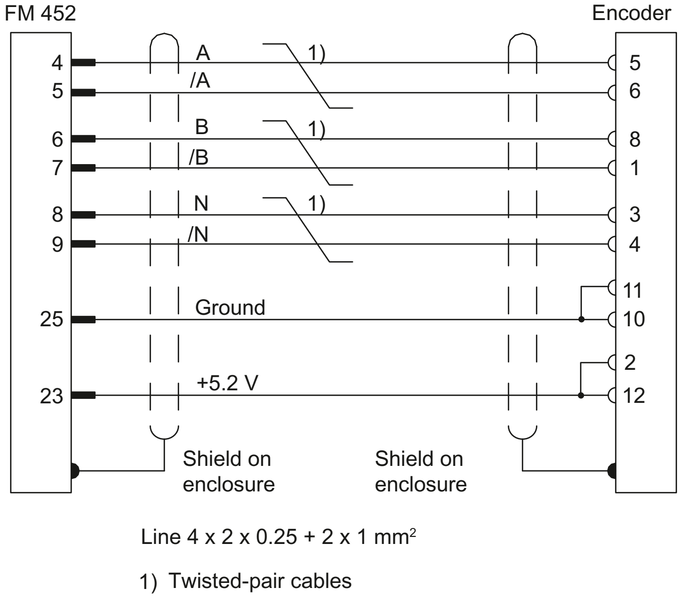


Figure B-1 Connection diagram for the incremental encoder Siemens 6FX 2001-2VVVV (Up=5 V: RS422)

Circular connector

12-pin socket, Siemens 6FX2003-0SU12

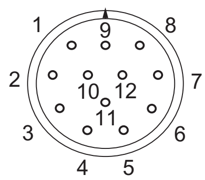


Figure B-2 Circular connector, terminal side (solder side)

B.3 Connection Diagram for Incremental Encoder Siemens 6FX 2001-2 (Up=24V; RS 422)

Connection diagram

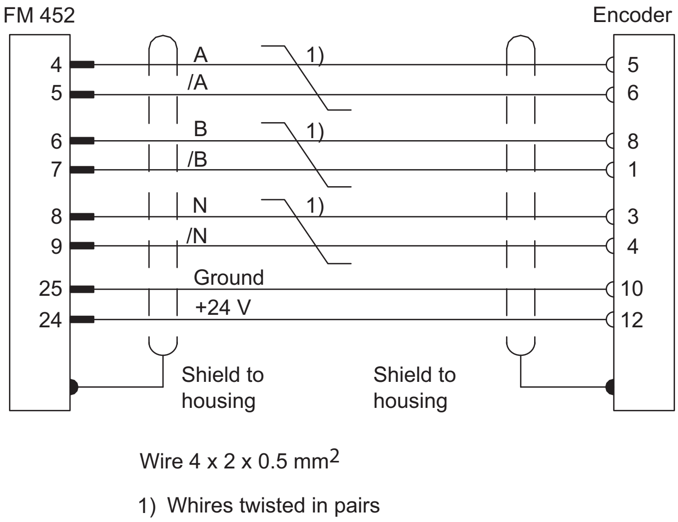


Figure B-3 Connection diagram for the incremental encoder Siemens 6FX 2001-2 (Up=24V; RS 422)

Circular connector

12-pin socket, Siemens 6FX2003-0SU12

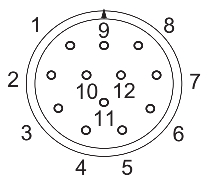


Figure B-4 Circular connector, terminal side (solder side)

B.4 Wiring Diagram of the Incremental Encoder Siemens 6FX 2001-4 (Up = 24 V; HTL)

Connection diagram

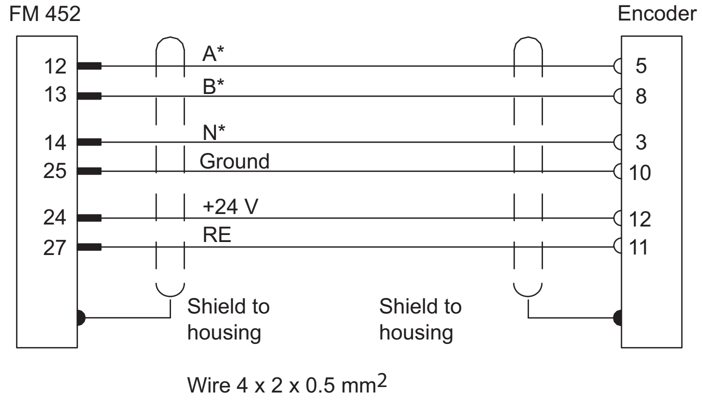


Figure B-5 Connection diagram for the incremental encoder Siemens 6FX 2001-4 (Up = 24 V; HTL)

Circular connector

12-pin socket, Siemens 6FX2003-0SU12

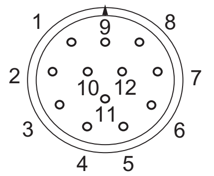


Figure B-6 Circular connector, terminal side (solder side)

Note

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

- Current sourcing: Connect RE (27) to ground (25).
- Current sinking: Connect RE (27) to +24 V (24).

B.5 Connection Diagram for Absolute Encoder Siemens 6FX 2001-5 (Up=24V; SSI)

Connection diagram

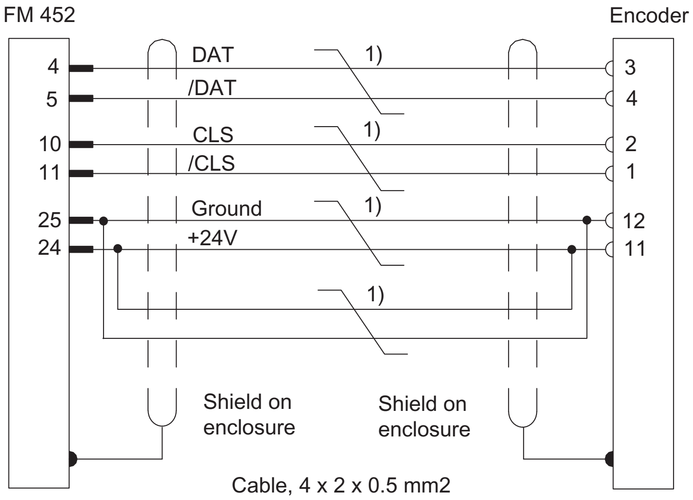


Figure B-7 Connection diagram for the absolute encoder Siemens 6FX 2001-5 (Up=24V; SSI)

Circular connector

12-pin socket, Siemens 6FX2003-0SU12

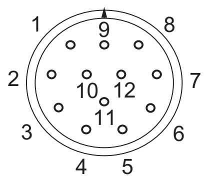


Figure B-8 Circular connector, terminal side (solder side)

Data blocks / error lists C

C.1 Content of the channel DB

Note

Do not modify any data not listed in this table.

Content of the channel DB

Address	Name	Type	Start value	Comment
Addresses/ version switch				
0.0	MOD_ADDR (enter!)	INT	0	Module address
2.0	CH_NO	INT	1	Channel number (always 1)
10.0	PARADBNO	INT	-1	Number of the parameter DB -1 = DB not available
12.0	FM_TYPE	BOOL	FALSE	0 = FM352 up to V4.0 1 = FM452 or FM352 V5.0 or higher
Control signals				
15.2	DIR_M	BOOL	FALSE	1 = simulation in negative direction
15.3	DIR_P	BOOL	FALSE	1 = simulation in positive direction
15.4	CAM_EN	BOOL	FALSE	1 = enable cam processing
15.5	CNTC0_EN	BOOL	FALSE	1 = enable count function of counter cam track 0
15.6	CNTC1_EN	BOOL	FALSE	1 = enable count function of counter cam track 1
16.0	TRACK_EN	WORD	W#16#0	Enable cam tracks 0 to 15 Bit 0 = track 0
Checkback signals				
22.2	DIAG	BOOL	FALSE	1 = diagnostic buffer changed
22.4	DATA_ERR	BOOL	FALSE	1 = data error
22.7	PARA	BOOL	FALSE	1 = module is programmed
23.4	CAM_ACT	BOOL	FALSE	1 = cam processing busy

Address	Name	Type	Start value	Comment
25.0	SYNC	BOOL	FALSE	1 = axis is synchronized
25.1	MSR_DONE	BOOL	FALSE	1= length measurement or edge detection completed
25.2	GO_M	BOOL	FALSE	1 = axis moving in negative direction
25.3	GO_P	BOOL	FALSE	1 = axis moving in positive direction
25.4	HYS	BOOL	FALSE	1 = axis within the hysteresis range
25.5	FVAL_DONE	BOOL	FALSE	1 = set actual value on-the-fly executed
26.0	ACT_POS	DINT	L#0	Current position of axis
30.0	TRACK_OUT	DWORD	DW#16#0	Current signals of tracks 0 to 31 Bit 0 = track 0
Function switches				
34.0	EDGE_ON	BOOL	FALSE	1 = edge detection on
34.1	SIM_ON	BOOL	FALSE	1 = simulation on
34.2	MSR_ON	BOOL	FALSE	1 = length measurement on
34.3	REFTR_ON	BOOL	FALSE	1 = retrigger reference point
34.4	SSW_OFF	BOOL	FALSE	1 = software limit switch deactivated
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	AVALREM_EN	BOOL	FALSE	1 = set actual value, cancel set actual value on-the-fly
35.3	CAM1WR_EN	BOOL	FALSE	1 = write cam data 1 (cams 0 to 15)
35.4	CAM2WR_EN	BOOL	FALSE	1 = write cam data 2 (cams 16 to 31)
35.5	CAM3WR_EN	BOOL	FALSE	1 = write cam data 3 (cams 32 to 47)
35.6	CAM4WR_EN	BOOL	FALSE	1 = write cam data 4 (cams 48 to 63)
35.7	CAM5WR_EN	BOOL	FALSE	1 = write cam data 5 (cams 64 to 79)

Address	Name	Type	Start value	Comment
36.0	CAM6WR_EN	BOOL	FALSE	1 = write cam data 6 (cams 80 to 95)
36.1	CAM7WR_EN	BOOL	FALSE	1 = write cam data 7 (cams 96 to 111)
36.2	CAM8WR_EN	BOOL	FALSE	1 = write cam data 8 (cams 112 to 127)
36.3	REFPT_EN	BOOL	FALSE	1 = set reference point coordinates
36.4	AVAL_EN	BOOL	FALSE	1 = set actual value
36.5	FVAL_EN	BOOL	FALSE	1 = set actual value on-the-fly
36.6	ZOFF_EN	BOOL	FALSE	1 = set zero offset
36.7	CH01CAM_EN	BOOL	FALSE	1 = write cam edge setting (1 cam)
37.0	CH16CAM_EN	BOOL	FALSE	1 = write fast cam parameter change settings (16 cams)
Trigger bits for read jobs				
37.1	MDRD_EN	BOOL	FALSE	1 = read machine data
37.2	CAM1RD_EN	BOOL	FALSE	1 = read cam data 1 (cams 0 to 15)
37.3	CAM2RD_EN	BOOL	FALSE	1 = read cam data 2 (cams 16 to 31)
37.4	CAM3RD_EN	BOOL	FALSE	1 = read cam data 3 (cams 32 to 47)
37.5	CAM4RD_EN	BOOL	FALSE	1 = read cam data 4 (cams 48 to 63)
37.6	CAM5RD_EN	BOOL	FALSE	1 = read cam data 5 (cams 64 to 79)
37.7	CAM6RD_EN	BOOL	FALSE	1 = read cam data 6 (cams 80 to 95)
38.0	CAM7RD_EN	BOOL	FALSE	1 = read cam data 7 (cams 96 to 111)
38.1	CAM8RD_EN	BOOL	FALSE	1 = read cam data 8 (cams 112 to 127)
38.2	MSRRD_EN	BOOL	FALSE	1 = read measured values
38.3	CNTTRC_EN	BOOL	FALSE	1 = read count values of counter cam tracks
38.4	ACTPOS_EN	BOOL	FALSE	1 = read position and track data
38.5	ENCVAL_EN	BOOL	FALSE	1 = read encoder values
38.6	CAMOUT_EN	BOOL	FALSE	1 = read cam and track data

Address	Name	Type	Start value	Comment
Done bits for function switches				
40.0	EDGE_D	BOOL	FALSE	1 = "activate edge detection" or "deactivate edge detection" completed
40.1	SIM_D	BOOL	FALSE	1 = "activate simulation" or "deactivate simulation" completed
40.2	MSR_D	BOOL	FALSE	1 = "activate length measurement" or "deactivate length measurement" completed
40.3	REFTR_D	BOOL	FALSE	1 = "Activate retrigger reference point" or "Deactivate retrigger reference point" completed
40.4	SSW_D	BOOL	FALSE	1 = "Activate software limit switch" or "Deactivate software limit switch" completed
Done bits for write jobs				
41.0	MDWR_D	BOOL	FALSE	1 = "Write machine data" job completed
41.1	MD_D	BOOL	FALSE	1 = "Activate machine data" job completed
41.2	AVALREM_D	BOOL	FALSE	1 = "cancel set actual value" or "cancel set actual value on-the-fly" completed
41.3	CAM1WR_D	BOOL	FALSE	1 = "write cam data 1" job completed
41.4	CAM2WR_D	BOOL	FALSE	1 = "write cam data 2" job completed
41.5	CAM3WR_D	BOOL	FALSE	1 = "write cam data 3" job completed
41.6	CAM4WR_D	BOOL	FALSE	1 = "write cam data 4" job completed
41.7	CAM5WR_D	BOOL	FALSE	1 = "write cam data 5" job completed
42.0	CAM6WR_D	BOOL	FALSE	1 = "write cam data 6" job completed
42.1	CAM7WR_D	BOOL	FALSE	1 = "write cam data 7" job completed
42.2	CAM8WR_D	BOOL	FALSE	1 = "write cam data 8" job completed
42.3	REFPT_D	BOOL	FALSE	1 = "set reference point" job completed
42.4	AVAL_D	BOOL	FALSE	1 = "set actual value" job completed
42.5	FVAL_D	BOOL	FALSE	1 = "Set actual value on-the-fly" job completed
42.6	ZOFF_D	BOOL	FALSE	1 = "set zero offset" job completed
42.7	CH01CAM_D	BOOL	FALSE	1 = "change 1 cam" job completed
43.0	CH16CAM_D	BOOL	FALSE	1 = "change 16 cams" completed (fast cam parameter change)
Done bits for read jobs				
43.1	MDRD_D	BOOL	FALSE	1 = "read machine data" job completed
43.2	CAM1RD_D	BOOL	FALSE	1 = "read cam data 1" job completed
43.3	CAM2RD_D	BOOL	FALSE	1 = "read cam data 2" job completed
43.4	CAM3RD_D	BOOL	FALSE	1 = "read cam data 3" job completed
43.5	CAM4RD_D	BOOL	FALSE	1 = "read cam data 4" job completed
43.6	CAM5RD_D	BOOL	FALSE	1 = "read cam data 5" job completed
43.7	CAM6RD_D	BOOL	FALSE	1 = "read cam data 6" job completed

Address	Name	Type	Start value	Comment
44.0	CAM7RD_D	BOOL	FALSE	1 = "read cam data 7" job completed
44.1	CAM8RD_D	BOOL	FALSE	1 = "read cam data 8" job completed
44.2	MSRRD_D	BOOL	FALSE	1 = "read measured values" job completed
44.3	CNTTRC_D	BOOL	FALSE	1 = "read count values of counter cam tracks" job completed
44.4	ACTPOS_D	BOOL	FALSE	1 = "read position and track data" job completed
44.5	ENCVAL_D	BOOL	FALSE	1 = "read actual encoder value" job completed
44.6	CAMOUT_D	BOOL	FALSE	1 = "read position and track data" job completed
Error bits for function switches				
46.0	EDGE_ERR	BOOL	FALSE	1 = error in "Activate edge detection" or "deactivate edge detection"
46.1	SIM_ERR	BOOL	FALSE	1 = error in "Activate simulation" or "Deactivate simulation"
46.2	MSR_ERR	BOOL	FALSE	1 = error in "Activate length measurement" or "Deactivate length measurement"
46.3	REFTR_ERR	BOOL	FALSE	1 = error in "Activate retrigger reference point" or "Deactivate retrigger reference point"
46.4	SSW_ERR	BOOL	FALSE	1 = error in "Activate software limit switch" or "Deactivate software limit switch"
Error bits for write jobs				
47.0	MDWR_ERR	BOOL	FALSE	1 = error in "Write machine data" job
47.1	MD_ERR	BOOL	FALSE	1 = error in "Activate machine data" job
47.2	AVALREM_ERR	BOOL	FALSE	1 = error in "Cancel set actual value" or "Cancel set actual value on-the-fly"
47.3	CAM1WR_ERR	BOOL	FALSE	1 = error in "Write cam data 1" job
47.4	CAM2WR_ERR	BOOL	FALSE	1 = error in "Write cam 2 data" job
47.5	CAM3WR_ERR	BOOL	FALSE	1 = error in "Write cam 3 data" job
47.6	CAM4WR_ERR	BOOL	FALSE	1 = error in "Write cam 4 data" job
47.7	CAM5WR_ERR	BOOL	FALSE	1 = error in "Write cam 5 data" job
48.0	CAM6WR_ERR	BOOL	FALSE	1 = error in "Write cam 6 data" job
48.1	CAM7WR_ERR	BOOL	FALSE	1 = error in "Write cam 7 data" job
48.2	CAM8WR_ERR	BOOL	FALSE	1 = error in "Write cam 8 data" job
48.3	REFPT_ERR	BOOL	FALSE	1 = error in "Set reference point" job
48.4	AVAL_ERR	BOOL	FALSE	1 = error in "Set actual value" job
48.5	FVAL_ERR	BOOL	FALSE	1 = error in "Set actual value on-the-fly" job
48.6	ZOFF_ERR	BOOL	FALSE	1 = error in "Set zero offset" job
48.7	CH01CAM_ERR	BOOL	FALSE	1 = error in "Change 1 cam" job
49.0	CH16CAM_ERR	BOOL	FALSE	1 = error in "Change 16 cams" (fast cam parameter change)

Address	Name	Type	Start value	Comment
Error bits for read jobs				
49.1	MDRD_ERR	BOOL	FALSE	1 = error in "Read machine data" job
49.2	CAM1RD_ERR	BOOL	FALSE	1 = error in "Read cam data 1" job
49.3	CAM2RD_ERR	BOOL	FALSE	1 = error in "Read cam data 2" job
49.4	CAM3RD_ERR	BOOL	FALSE	1 = error in "Read cam data 3" job
49.5	CAM4RD_ERR	BOOL	FALSE	1 = error in "Read cam data 4" job
49.6	CAM5RD_ERR	BOOL	FALSE	1 = error in "Read cam data 5" job
49.7	CAM6RD_ERR	BOOL	FALSE	1 = error in "Read cam data 6" job
50.0	CAM7RD_ERR	BOOL	FALSE	1 = error in "Read cam data 7" job
50.1	CAM8RD_ERR	BOOL	FALSE	1 = error in "Read cam data 8" job
50.2	MSRRD_ERR	BOOL	FALSE	1 = error in "Read measured values" job
50.3	CNTTRC_ERR	BOOL	FALSE	1 = error in "Read count values of counter cam tracks" job
50.4	ACTPOS_ERR	BOOL	FALSE	1 = error in "Read position and track data" job
50.5	ENCVAL_ERR	BOOL	FALSE	1 = error in "Read current encoder value" job
50.6	CAMOUT_ERR	BOOL	FALSE	1 = error in "Read cam and track data" job
Job management for FC CAM_CTRL				
52.0	JOB_ERR	INT	0	Communication errors
54.0	JOBBUSY	BOOL	FALSE	1 = at least one job is running
54.1	JOBRESET	BOOL	FALSE	1 = reset all error and done bits
Job management for FC CAM_MSRM				
56.0	JOB_ERR_M	INT	0	Communication errors
58.0	JOBBUSY_M	BOOL	FALSE	1 = job active
Parameter for "zero offset" job				
86.0	ZOFF	DINT	L#0	Zero offset
Parameter for "set actual value" job				
90.0	AVAL	DINT	L#0	Coordinate for "Set actual value"
Parameter for "set actual value on-the-fly" job				
94.0	FVAL	DINT	L#0	Coordinate for "Set actual value on-the-fly"
Parameter for "set reference point" job				
98.0	REFPT	DINT	L#0	Coordinate for "Set reference point"
Parameters for "change cam edges" job				
102.0	CAM_NO	INT	0	Cam number
104.0	CAM_START	DINT	L#0	Cam start
108.0	CAM_END	DINT	L#0	Cam end
Data for the "Length measurement/edge detection" job				
112.0	BEG_VAL	DINT	L#0	Initial value
116.0	END_VAL	DINT	L#0	End value
120.0	LEN_VAL	DINT	L#0	Length

Address	Name	Type	Start value	Comment
Data for the "Read count values" job				
124.0	CNT_TRC0	INT	0	Current count value of counter cam track 0
126.0	CNT_TRC1	INT	0	Current count value of counter cam track 1
Data for the "Read position and track data" job				
128.0	ACTPOS	DINT	L#0	Current position
132.0	ACTSPD	DINT	L#0	Current velocity
136.0	TRACK_ID	DWORD	DW#16#0	Track identifier bits of tracks 0 to 31
Data for the "Read encoder data" job				
140.0	ENCVAL	DINT	L#0	Encoder value
144.0	ZEROVAL	DINT	L#0	Count value at the last zero mark
148.0	ENC_ADJ	DINT	L#0	Absolute encoder adjustment
Data for the "Read cam and track data" job				
152.0	CAM_00_31	DWORD	DW#16#0	Cam identifier bits for cams 0 to 31
156.0	CAM_32_63	DWORD	DW#16#0	Cam identifier bits for cams 32 to 63
160.0	CAM_64_95	DWORD	DW#16#0	Cam identifier bits for cams 64 to 95
164.0	CAM_96_127	DWORD	DW#16#0	Cam identifier bits for cams 96 to 127
168.0	TRACK_ID1	DWORD	DW#16#0	Track identifier bits of tracks 0 to 31
172.0	ACTPOS1	DINT	L#0	Current position
Data for the "Fast cam parameter change" job				
176.0	C_QTY	BYTE	B#16#0	Number of cams to modify
177.0	DIS_CHECK	BOOL	FALSE	1 = disable data check
180.0	CAM	ARRAY [0...15] STRUCT		Note: The following structure must be completed for each cam to be modified
Relative address				
+0.0	CAM_NO	BYTE	B#16#0	Number of the cam to modify
+1.0	C_EFFDIR	BOOL	FALSE	1 = change the effective direction
+1.1	C_CBEGIN	BOOL	FALSE	1 = change the cam start to the value CBEGIN (new cam start)
+1.2	C_CEND	BOOL	FALSE	1 = change the cam end / activation time to the value CEND (new cam end)
+1.3	C_LTIME	BOOL	FALSE	1 = change the lead time to the LTIME value (new lead time)
+1.4	CAM_OFF	BOOL	FALSE	1 = deactivate the cam during the cam change
+1.5	EFFDIR_P	BOOL	FALSE	1 = new effective direction positive (plus)
+1.6	EFFDIR_M	BOOL	FALSE	1 = new effective direction negative (minus)
+2.0	CBEGIN	DINT	L#0	New cam start
+6.0	CEND	DINT	L#0	New cam end / new activation time
+10.0	LTIME	INT	0	New lead time

C.2 Content of the Parameter DB

Note

Do not modify any data not listed in this table.

Content of the Parameter DB

Address	Name	Type	Start value	Comment
Machine parameters				
3.1	PI_MEND	BOOL	FALSE	1: Enable process interrupt: End of measurement
3.2	PI_CAM	BOOL	FALSE	1: Enable process interrupt: Cam on/off
3.5	PI_MSTRT	BOOL	FALSE	1: Enable process interrupt: Start of measurement
4.0	EDGEDIST	DINT	L#0	Minimum edge interval for edge detection
8.0	UNITS	DINT	L#1	Physical units system
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 frame length
24.0	DISP_REV	DINT	L#80000	Distance per encoder revolution
32.0	INC_REV	DINT	L#500	Increments per encoder revolution
36.0	NO_REV	DINT	L#1024	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	RETR_TYPE	DINT	L#0	Type of retrigger reference point
56.0	CNT_DIR	DINT	L#0	Counting direction: 0: normal, 1: Inverted
63.0	MON_WIRE	BOOL	TRUE	1: Wire-break monitoring
63.1	MON_FRAME	BOOL	TRUE	1: Frame error monitoring
63.2	MON_PULSE	BOOL	TRUE	1: Missing pulse monitoring
64.0	SSW_STRT	DINT	L#-100000000	Software limit switch start
68.0	SSW_END	DINT	L#100000000	Software limit switch end
76.0	C_QTY	DINT	L#0	Scope: 0, 1, 2, 3 = max. 16, 32, 64, 128 cams
80.0	HYS	DINT	L#0	Hysteresis
84.0	SIM_SPD	DINT	L#0	Simulation velocity
90.0	TRACK_OUT	WORD	W#16#0	Control of track outputs: 0 = cam control system, 1 = CPU; Bit number = track number

Address	Name	Type	Start value	Comment
95.0	EN_IN_I3	BOOL	FALSE	Enable input I3
95.1	EN_IN_I4	BOOL	FALSE	Input enable
95.2	EN_IN_I5	BOOL	FALSE	Input I5 enable
95.3	EN_IN_I6	BOOL	FALSE	Input I6 enable
95.4	EN_IN_I7	BOOL	FALSE	Input I7 enable
95.5	EN_IN_I8	BOOL	FALSE	Input I8 enable
95.6	EN_IN_I9	BOOL	FALSE	Input I9 enable
95.7	EN_IN_I10	BOOL	FALSE	Input I10 enable
99.0	SPEC_TRC0	BOOL	FALSE	1 = track 0 is counter cam track
99.1	SPEC_TRC1	BOOL	FALSE	1 = track 1 is counter cam track
99.2	SPEC_TRC2	BOOL	FALSE	1 = track 2 is brake cam track
100.0	CNT_LIM0	DINT	L#2	Upper count value for counter cam track 0
104.0	CNT_LIM1	DINT	L#2	Upper count value for counter cam track 1
Parameters for cams 0 to 15 / 0 to 31 / 0 to 63 / 0 to 127				
108.0		STRUCT		(12 bytes length per element)
Relative address				
+0.0	CAMVALID	BOOL	FALSE	1: Cam valid
+0.1	EFFDIR_P	BOOL	TRUE	1: Positive effective direction (plus)
+0.2	EFFDIR_M	BOOL	TRUE	1: Negative effective direction (minus)
+0.3	CAM_TYPE	BOOL	FALSE	0: Distance cam, 1: Time-based output cam
+0.4	PI_SW_ON	BOOL	FALSE	1: Process interrupt on activation
+0.5	PI_SW_OFF	BOOL	FALSE	1: Process interrupt on deactivation
+1.0	TRACK_NO	BYTE	B#16#0	Track number
+2.0	CBEGIN	DINT	L#-100 000 000	Cam start
+6.0	CEND	DINT	L#100 000 000	Cam end/activation time
+10.0	LTIME	INT	0	Rate time

C.3 Data and Structure of the Diagnostic DB

Note

Do not modify any data not listed in this table.

Structure of the diagnostic DB

Address	Name	Type	Initial value	Comment
0.0	MOD_ADDR (enter!)	INT	0	Module address
256.0	JOB_ERR	INT	0	Communication errors
258.0	JOBBUSY	BOOL	FALSE	1 = job busy
258.1	DIAGRD_EN	BOOL	FALSE	1 = read diagnostic buffer unconditional
260.0	DIAG_CNT	INT	0	Number of valid entries in the list
262.0	DIAG[1]	STRUCT		Diagnostic data latest entry
272.0	DIAG[2]	STRUCT		Diagnostic data second entry
282.0	DIAG[3]	STRUCT		Diagnostic data third entry
292.0	DIAG[4]	STRUCT		Diagnostic data oldest entry

Structure of the Diagnostic Entry

Address	Name	Type	Initial value	Comment
+0.0	STATE	BOOL	FALSE	0 = event outgoing 1 = event incoming
+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 4: Data error 5: Machine data error 7: Cam data error 15: Messages 128: Diagnostic error
+4.0	FNO	INT	0	Error number 0 to 255
+6.0	CH_NO	INT	0	Channel number (always 1)
+8.0	CAMNO	INT	0	Cam numbers 0 to 127 with error class = cam data error

List of JOB_ERR Interrupts

JOB_ERR (hex)	JOB_ERR (dec)	JOB_ERR (int)	Meaning
80A0	32928	-32608	Negative acknowledgment when reading from module. Module removed during read operation or module defective.
80A1	32929	-32607	Negative acknowledgment when writing to module. Module removed during write operation or module defective.
80A2	32930	-32606	Protocol error at layer 2
80A3	32931	-32605	Protocol error involving user interface/user
80A4	32932	-32604	Communication problem on K bus
80B1	32945	-32591	Specified length wrong. Faulty FM_TYPE parameter setting at the channel DB for the module in use.
80B2	32946	-32590	The configured slot is empty.
80B3	32947	-32589	Actual module type does not match configured module type.
80C0	32960	-32576	The module has not yet prepared the data to be read.
80C1	32961	-32575	The data of a write job of the same type have not yet been processed on the module.
80C2	32962	-32574	The module is currently processing the maximum number of jobs.
80C3	32963	-32573	Required resources (memory etc.) currently in use.
80C4	32964	-32572	Communication errors
80C5	32965	-32571	Distributed I/Os not available.
80C6	32966	-32570	Priority class abort (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	DB number of the parameter DB too high. (write job)
853A	34106	-31430	Parameter DB does not exist. (write job)
8544	34116	-31420	Error in nth (n > 1) read access to a DB after error occurred. (write job)
8723	34595	-30941	Channel DB or parameter DB too short. The data cannot be written to the DB. (read job)
8730	34608	-30928	Parameter DB on the CPU write-protected. The data cannot be written to the DB (read job).
8732	34610	-30926	DB number of the parameter DB too high. (read job)
873A	34618	-30918	Parameter DB does not exist. (read job)
8745	34629	-30907	Error in nth (n > 1) write access to a DB after error occurred. (read job)
The errors 80A2 to 80A4 and 80Cx are temporary, i.e. these can be cleared after a waiting time without user intervention. Interrupts in the format 7xxx indicate temporary states in communication.			

See also

Machine data of the encoder (Page 86)

Cam parameters (Page 97)

Enable diagnostics interrupts (Page 149)

C.4 Error class 1: Process error

Meaning

Operating errors are detected asynchronous to operator input/commands.

No.	Meaning		Diagnostics interrupt
1	Software limit switch start passed		Yes
2	Software limit switch end passed		Yes
3	Traversing range start passed		Yes
4	Traversing range end passed		Yes
13	Set actual value on-the-fly cannot be executed		Yes
	Cause	After the actual value was set on-the-fly, the software limit switch is outside the traversing range. (-100 m to +100 m or -1000 m to +1000 m.) The shift resulting from set actual value / set actual value on-the-fly is more than ± 100 m or ± 1000 m.	
	Effect	Axis not synchronized	

C.5 Error class 4: Data error

Meaning

Data errors are detected synchronously to an operator input/control.

No.	Meaning		Diagnostic interrupt
10	**Zero offset error**		No
	Cause	The zero offset is greater than ±100 m or ±1000 m. The software limit switches are outside the traversing range (-100 m to +100 m or -1000 m to +1000 m) after the zero offset. Rotary axis: The absolute value of the zero offset is greater than the end of rotary axis.	
11	**Faulty actual value setting**		No
	Cause	Linear axis: The coordinate lies outside the current (possibly offset) software limit switches. Rotary axis: The coordinate is < 0 or greater than the end of rotary axis.	
12	**Reference point error**		No
	Cause	Linear axis: The coordinate lies outside the current (possibly offset) software limit switches. Rotary axis: The coordinate is < 0 or greater than the end of rotary axis.	
20	**Enable machine data not permitted**		No
	Cause	There are no new (error-free) machine data on the module.	
21	**Set actual value on-the-fly not permitted**		No
	Cause	An attempt was made to execute "set actual value on-the-fly" while "retrigger reference point" was active.	
27	**Illegal bit-coded setting**		No
	Cause	Unused and, in this case, unwritten bits are not 0. An attempt was made to select "length measurement" and "edge detection" at the same time.	
28	**Retrigger reference point is not permitted**		No
	Cause	An attempt was made to execute "retrigger reference point" while "set actual value on-the-fly" was active. An attempt was made to execute "retrigger reference point" with an SSI encoder.	
29	**Illegal bit-coded command**		No
	Cause	Unused and, in this case, unwritten bits are not 0.	
30	**Incorrect lead time**		No
31	**Invalid cam number**		No
	Cause	The cam is not valid. The cam number is not in the range 0 to 127.	

No.	Meaning		Diagnostic interrupt
32	**Incorrect cam start**		No
	Cause	The cam start is outside the traversing range (-100 m to +100 m or -1000 m to +1000 m). Rotary axis: The cam start is < 0 or greater than the end of rotary axis.	
33	**Invalid cam end / invalid activation time**		No
	Cause	The cam end is outside the traversing range (-100 m to +100 m or -1000 m to +1000 m). Rotary axis: The cam end is < 0 or greater than the end of rotary axis. The cam is not active at least for the duration of one pulse. With an inverse cam, there are not at least 4 pulses between the cam start and cam end.	
34	**Cancel set actual value not possible**		No
	Cause	The actual position value would be outside the operating range with an SSI encoder and linear axis after making the setting.	
35	**Incorrect actual value specified by "set actual value" / "set actual value on-the-fly"**		No
	Cause	The specified actual value is outside the permitted numeric range of ±100 m or ±1000 m. When this setting is applied, the software limit switches would be outside the traversing range (-100 m to +100 m or -1000 m to +1000 m). The offset derived from set actual value / set actual value on-the-fly would be more than ±100 m or ±1000 m.	
107	**Axis parameters not assigned**		No
	Cause	There are no machine data on the axis. There are no machine data activated on the axis.	
108	**Axis not synchronized**		No
	Cause	One of the settings "set actual value" or "set actual value on-the-fly" was initiated although the axis is not synchronized.	
109	**Output cam processing running**		No
110	**Invalid number of cams to be modified.**		No

C.6 Error class 5: Machine data error

Meaning

The diagnostic interrupt is only triggered if an error is detected in the system data block (SDB).

No.	Meaning		Diagnostic interrupt
5	Error in process interrupt setting		Yes
	Cause	You have attempted to select a process interrupt that the module does not support.	
6	Incorrect minimum edge distance		Yes
	Cause	You have entered a value < 0 or > 10^9 µm as the minimum edge distance.	
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 valid range from 1 to 10^9 µm or 1 to 10^8 µm (depending on the resolution).	
10	Incorrect encoder type		Yes
	Cause	The value for the encoder type lies outside of the permissible range of 1 to 10.	
11	Incorrect distance/encoder revolution		Yes
	Cause	The value for distance/encoder revolution is outside the valid range from 1 to 10^9 µm (regardless of the resolution).	
13	Incorrect number of increments/encoder revolution (see chapter "Machine data of the encoder (Page 86)")		Yes
14	Incorrect number of increments/encoder revolution (see chapter "Machine data of the encoder (Page 86)")		Yes
15	Incorrect baud rate		Yes
	Cause	You have specified a baud rate outside the permitted 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 < 0 or greater than the end of rotary axis.	
17	Incorrect absolute value adjustment		Yes
	Cause	SSI encoder: The value of the absolute encoder adjustment is not in the encoder range: [increments per encoder revolution] x [number of revolutions - 1].	
18	Incorrect type of reference point retrigger		Yes
	Cause	You have specified a value other than 0, 1, 6 and 7.	

No.	Meaning		Diagnostic interrupt
19	Incorrect direction adaptation		Yes
	Cause	You have specified a value other than 0 and 1.	
20	Hardware monitoring not possible		Yes
	Cause	You set frame error monitoring = FALSE at the parameter DB. The monitoring of missing pulses is not possible with the encoder used. Disable the MON_PULSE parameter.	
21	Incorrect software limit switch start		Yes
	Cause	Linear axis: The start software limit switch is outside the travel range (-100 m...+100 m or -1000 m...+1000 m, depending on the 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 travel range (-100 m...+100 m or -1000 m...+1000 m, depending on the resolution) or is less than the start software limit switch. The software limit switch end (including any existing zero offset) is greater than +100 m or +1000 m (depending on the resolution).	
144	Incorrect number of cams		Yes
	Cause	You have specified a value other than 0 to 3 for the number of cams.	
145	Incorrect hysteresis		Yes
	Cause	The hysteresis is outside the range 0 to 65535 * resolution. The hysteresis is greater than [¼ x operating range] or {¼ x rotary axis range].	
146	Incorrect simulation velocity		Yes
	Cause	The simulation velocity is outside the range 1000*RES to $3*10^7$*RES, or is greater than $5*10^8$ μm/min. The simulation velocity cannot be set internally.	
147	Incorrect track		Yes
	Cause	Activation of a track outside 0 to 15 (bits 0 to 15) was selected.	
148	Incorrect selection of enable inputs		Yes
	Cause	You wanted to enable a track outside 3 to 10 (bits 0 to 7) using an external signal.	
149	Incorrect special track selection.		Yes
	Cause	You wanted to define a track outside 0, 1 and 2 (bits 0, 1 and 2) as a special track	
150	Incorrect high count value track 0		Yes
	Cause	You have specified a count value < 2 or > 65535 as the high count value.	
151	Incorrect high count value track 1		Yes
	Cause	You have specified a count value < 2 or > 65535 as the high count value.	

No.	Meaning		Diagnostic interrupt
200	**Incorrect resolution**		Yes
	Cause	You have specified a resolution < 0.1 µm/pulse or >1000 µm/pulse. You have specified a distance/encoder revolution and a number of pulses/encoder revolution, that results in a resolution of < 0.1 or > 1000.	
201	**Position encoder does not match the operating range/rotary axis range**		Yes
	Cause	SSI position encoder and rotary axis: The encoder does not exactly cover the rotary axis range. Linear axis: The encoder does not cover at least the operating range (including software limit switches).	

C.7 Error class 7: Cam data error

Meaning

The diagnostic interrupt is only triggered if an error is detected in the system data block (SDB).

No.	Meaning		Diagnostic interrupt
1	**Invalid process interrupt**		Yes
	Cause	You attempted to define a process interrupt for a cam with a cam number > 7.	
2	**Incorrect track number**		Yes
	Cause	The track number is outside the range from 0 to 31.	
3	**Incorrect cam start**		Yes
	Cause	The cam start is outside the traversing range (-100 m to +100 m or -1000 m to +1000 m). Rotary axis: The cam start is < 0 and/or greater than the end of rotary axis.	
4	**Incorrect cam end**		Yes
	Cause	The cam end is outside the traversing range (-100 m to +100 m, or -1000 m to +1000 m). The cam length is not at least one pulse. Rotary axis: The cam end is < 0 and/or greater than the end of rotary axis. With an inverse cam, there are not at least 4 pulses between the cam start and cam end.	
5	**Incorrect activation time**		Yes
	Cause	The activation time is < 0 µs. The maximum value depends on the number of cams available (see chapter "Cam parameters (Page 97)").	
6	**Incorrect lead time**[1)]		Yes
	Cause	The lead time is < 0 µs. The maximum value depends on the number of cams available (see chapter "Cam parameters (Page 97)").	
50	**Too many cam sets**		Yes
	Cause	You wanted to enter more cam records than is possible with this number of cams.	
51	**Axis in operation**		Yes
	Cause	You attempted to enter cam records while the cam control system was active.	
52	**Axis parameters not assigned**		Yes
	Cause	You want to enter cam data although no machine data have been activated yet.	

1) The error message may also be issued if you have assigned the parameter "inverted" as the count direction in connection with an absolute encoder (SSI).

C.8 Error class 15: Messages

Meaning

No.	Meaning		Diagnostic Interrupt
1	**Start of parameter assignment**		No
	Cause	The module has detected a parameter assignment by a system data block.	
2	**End of parameter assignment**		No
	Cause	The module has processed the parameter assignment by a system data block error-free.	

C.9 Error class 128: Diagnostics errors

Meaning

No.	Meaning		Diagnostic interrupt
4	**External auxiliary voltage missing**		Yes
	Cause	• External auxiliary 24 V voltage is not connected or has failed • Short circuit (e.g. at the connected encoder)	
	Effect	See chapter "Reaction of FM 452 to errors with diagnostics interrupt (Page 150)." • Cam processing will be deactivated • Track outputs will be disabled • The synchronization of incremental encoders is canceled • FM 452 parameters not assigned (checkback signal PARA = 0).	
	Remedy	Make sure that the 24 V connection is correct (If the 24 V connection is correct, then the module is defective).	
5	**Front connector missing**		Yes
	Cause	Front connector not inserted	
	Effect	• External 24 V auxiliary voltage missing • Module not ready	
	Remedy	Insert the front connector	
51	**Watchdog expired**		Yes
	Cause	• Strong interference on FM 452 • Error in FM 452	
	Effect	• Module is reset • Provided that no faults were detected after the module was reset, the module is ready for operation. • The module signals the expired WATCHDOG with "incoming" and "outgoing"	
	Remedy	• Eliminate the interference • Contact the relevant sales department who will require details of the circumstances leading to the error. • Replace the FM 452	

No.	Meaning		Diagnostic interrupt
52	**Internal module power supply failed**		Yes
	Cause	Error in FM 452	
	Effect	• Module is reset • Provided that no faults were detected after the module was reset, the module is ready for operation.	
	Remedy	Replace the FM 452	
70	**Process interrupt lost**		Yes
	Cause	A process interrupt event detected by FM 452 cannot be reported, because the same event has not yet been processed by the user program/ CPU.	
	Effect	• Cam processing will be deactivated • Track outputs will be disabled • Synchronization of incremental encoders will be canceled	
	Remedy	• Link OB40 into the user program • Check the bus connection of the module • Deactivate process interrupt • Adapt your hardware and software to suit your process requirements (for example, faster CPU, optimize user program).	
144	**Encoder wire break**		Yes
	Cause	• Encoder cable sheared off or not inserted • Encoder has no transverse signals • Incorrect pin assignment • Cable too long • Encoder signals short-circuited	
	Effect	• Cam processing will be deactivated • Track outputs will be disabled • Synchronization of incremental encoders will be canceled	
	Remedy	• Check the encoder cable • Comply with encoder specifications • Monitoring can be temporarily disabled at the responsibility of the operator using the programming interface. • Observe the module's technical specifications	

No.	Meaning		Diagnostic interrupt
145	**Absolute encoder frame error**		Yes
	Cause	The frame traffic between FM 452 and the absolute encoder (SSI) is disturbed or interrupted: • Encoder cable sheared off or not inserted • Incorrect encoder type • Incorrect encoder setting (programmable encoders) • Incorrect entry of the frame length • Encoder returns incorrect values (encoder defective) • Interference on measuring system cable • Selected baud rate is too high	
	Effect	• Cam processing will be deactivated • Track outputs will be disabled • The last correct actual value remains unchanged until the end of the next correct SSI transfer	
	Remedy	• Check the encoder cable • Check the encoder • Check the frame traffic between the encoder and FM 452	
146	**Incremental encoder pulse error frame**		Yes
	Cause	• Encoder monitoring has detected missing pulses • Number of increments per encoder revolution incorrectly entered • Encoder defective: does not return the specified number of pulses • Faulty or missing zero marker • Interference on the encoder cable	
	Effect	• Cam processing will be deactivated • Track outputs will be disabled • Synchronization is canceled	
	Remedy	• Enter the correct number of increments/encoder revolutions. • Check the encoder and its cable • Comply with shielding and grounding regulations • Monitoring can be temporarily disabled at the responsibility of the operator using the programming interface.	

Index

A

B

C

D

E

M

N

O

P

R

S