SIEMENS

SIMATIC

S7-Technology

Function Manual

03/2008
A5E00251798-06

Safety Guidelines

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 corresponding 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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

⚠WARNING
This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the 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
Automation and Drives
Postfach 48 48
90327 NÜRNBERG
GERMANY

A5E00251798-06
℗ 03/2008

Preface

Purpose of this manuals

This manual gives you a complete overview of the optional software package "S7-Technology". The programming model, the individual technological objects and the individual function blocks according to PLCopen are explained.

It is designed for STEP 7 programmers and persons who work in the configuration, commissioning and automation system service with Motion Control application sector.

Required basic knowledge

To understand this manual you require a general knowledge in the automation technology and motion control field.

Users should be familiar in operating computers and programming devices on a Windows 2000 Professional or XP operating system platform. Adequate knowledge of the STEP 7 standard software is essential, because the optional software package "S7-Technology" is based on this software. The appropriate knowledge base is found in the “Programming with STEP 7“ manual.

Range of validity of this manual

This manual applies to the optional software package "S7-Technology" V 4.1 or higher. In the chapter "What's New in S7 Technology V 4.1?" you will find the differences in functionality and specifications of this version in comparison to the "S7 Technology" optional software package version V1.0.

Further Support

If you have any technical questions, please get in touch with your Siemens representative or responsible agent.

You will find your contact person at:

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

You will find a guide to the technical documentation offered for the individual SIMATIC Products and Systems here at:

http://www.siemens.com/simatic-tech-doku-portal

The online catalog and order system is found under:

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

Training Centers

Siemens offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Telephone: +49 (911) 895-3200.

Internet: http://www.sitrain.com

Table of contents

What's new in S7-Technology 1

What's new in S7-Technology V4.1

Flexible machine concept

Do you want to remove, either temporarily or permanently, configured technology objects from the cyclic processing of the integrated technology, or temporarily or permanently deactivate and remove DP slaves at the DP(DRIVE)?

The new technology functions "MC_ActivateTO" and "MC_ActivateDPSlave" are used to activate / deactivate technology objects and DP slaves at the DP(DRIVE). Alternatively, you can configure activation / deactivation in S7T Config.

Cam track

Use the new "Cam track" technology object if you want to use several output cams of the same type on one axis.

The cam track is configured in S7T Config and activated by calling the "MC_CamTrack" technology function. You can read the switching state of the individual output cams by calling the "MC_ReadCamTrackData" technology function. You can edit the parameters of the cam track in the user program by calling the "MC_WriteCamTrackData" technology function.

New functions of the Technology CPU CPU 31xT (firmware V2.6)

- Isochronous mode on the PROFIBUS DP (interface X1 MPI/DP)
- Updating the firmware online (via networks)
- Reset of the CPU to delivery state

For further information, please refer to the "CPU 31xT" Equipment Manual.

Older CPUs can be upgraded to the firmware version V2.6.

New STARTER functionality

The STARTER functionality integrated in S7-Technology corresponds to the new STARTER Version V4.1.

Brake control (vertical axes)

Control the integrated brake control of the drives with the new version of the "MC_Power" technology function. The brake control of the following drives is supported:

- SIMODRIVE 611U universal
- SINAMICS S120
- MASTERDRIVES Motion Control Plus

Read first and second derivation of a cam point

If a cam disk is generated and interpolated in the Technology CPU, the result of the interpolation cannot be checked in S7T Config. Use the new mode values *3* and *4* at the input parameter *Mode* of the "MC_GetCamPoint" technology function to read the first and second derivation of a cam point. You can determine the velocity and acceleration at the cam point with the read values.

Set the extrapolation for actual value coupling

Set the actual value filter and the extrapolation for the actual value coupling in the new dialogs of S7T Config.

Fine resolution of the torque reduction

SINAMICS and SIMODRIVE drives now support a fine resolution of the torque reduction. You can use the fine resolution of the torque reduction together with the technology functions "MC_SetTorqueLimit" and "MC_MoveToEndPos".

Maintenance of the torque limit after removal of the enable

With the current firmware of the integrated technology V4.1.x, the torque limit is retained even after removal of the enable by the "MC-Power" technology function.

Application and benefits 2

2.1 Compact and integrated

The Technology CPU integrates Motion Control functions in a SIMATIC CPU in order to combine the functionality of a SIMATIC CPU S7-300 with PLCopen-compliant Motion Control functions.

The Technology CPU is integrated to 100 % into the SIMATIC and TIA world and demonstrates its high performance when used in coupled motion sequences.

2.2 The world of "SIMATIC"

The Technology CPU is a standard SIMATIC CPU with integrated Motion Control functionality. This feature allows users to simply copy S7-300 programs from existing projects to new projects.

The Technology CPU is programmed based on the SIMATIC programming languages LAD, FBD or STL. PLC and Motion Control functions are edited in a single application program. It is not required to learn any additional programming language. You can always fall back on your SIMATIC S7 knowledge.

The Technology can be configured comfortably in STEP 7. There you can set all necessary parameters such as the mechanical data, drive selection data, controller settings, defaults, monitoring functions, output cams, measuring inputs, cam disks etc.

2.3 Integration of the PLCopen-compliant motion control functions in STEP 7

The PLCopen-compliant Motion Control functions of your Technology CPU let you directly utilize your Motion Control knowledge. Motion Control features functions which are compliant with PLCopen specifications in terms of interfaces, functionality and sequences, and which facilitate engineering, commissioning and service.

Thanks to the standardized interface, you can almost seamlessly implement the function blocks required to initiate your Motion Control commands in the application program.

2.4 Supported components and systems

Valid for Integrated Technology with firmware version V4.1.x

SIMATIC Technology CPU / software

Function	Product	Order no.
SIMATIC Technology CPU	CPU 315T-2DP (working memory 128 KB)	6ES7 315-6TG10-0AB0
SIMATIC Technology CPU	CPU 315T-2DP (working memory 256 KB)	6ES7 315-6TH13-0AB0
SIMATIC Technology CPU	CPU 317T-2DP (working memory 512 KB)	6ES7 317-6TJ10-0AB0
SIMATIC Technology CPU	CPU 317T-2DP (working memory 1024 KB)	6ES7 317-6TK13-0AB0
SIMATIC Technology CPU	MICROBOX 420-T	6ES7 675-3AG30-0AB0
Micro Memory Card (for CPU 31xT-2DP)	MMC 4 MB or 8 MB (8 MB recommended)	6ES7 953-8Lx11-0AA0
"S7-Technology" options package	SIMATIC S7-Technology V4.1	6ES7 864-1CC30-0YX0
STEP 7	STEP 7 V5.4 + SP2	6ES7 810-4CC07-0Yxx

The following software product can be used in addition to the "S7-Technology" options package:

Function	Product	Order no.
SCOUT CamTool	SCOUT CamTool V2.1	6AU1 810-0FA21-0XA0

Components on PROFIBUS DP(DRIVE)

The Technology CPU supports the components listed below for technological tasks in isochronous mode on DP(DRIVE) (status at the time this manual was released for printing):

Product	Order no.
SIMODRIVE	
SIMODRIVE 611U universal	6SN1 118-xNH00-0AAx
SIMODRIVE 611U universal HR	6SN1 114-0NB0x-0AAx
Options module Motion Control with PROFIBUS DP (for SIMODRIVE 611U)	6SN1 114-0NB01-0AA0
SIMODRIVE POSMO CA	6SN2 703-3AAx
SIMODRIVE POSMO CD	6SN2 703-2AAx
SIMODRIVE POSMO SI	6SN2 4x
SIMODRIVE sensor single-turn / synchro-flange	6FX2 001-5FP12
SIMODRIVE sensor single-turn / clamping flange	6FX2 001-5QP12
SIMODRIVE sensor multiturn / synchro-flange	6FX2 001-5FP24
SIMODRIVE sensor multiturn / clamping flange	6FX2 001-5QP24
MICROMASTER 4	
COMBIMASTER 411	6SE6 401-0PB00-0AA0
MICROMASTER 420	6SE6 400-1PB00-0AA0
MICROMASTER 430	6SE6 400-1PB00-0AA0
MICROMASTER 440	6SE6 400-1PB00-0AA0
MASTERDRIVES with communications module CBP2	
Motion Control	6SE7 090-0xx84-0FF5
Motion Control Plus	6SE7 090-0xx84-0FF5
Vector Control CUVC	6SE7 090-0xx84-0FF5
Vector Control Plus	6SE7 090-0xx84-0FF5
Note the order supplement "Gxx" when ordering communications module CBP2.	
SINAMICS	
SINAMICS S120 CU310 DP (firmware up to and including V2.5)	6SL3 040-0xA00-0xxx
SINAMICS S120 CU320 (firmware up to and including V2.5)	6SL3 040-0xA00-0xxx
SINAMICS G120 CU240S DP	6SL3 24x-0BAxx-xPAx
SINAMICS G120 CU240S DP F	6SL3 244-0BA21-1PA0
SINAMICS G120D CU240D DP	6SL3 544-0FA20-1PA0
SINAMICS G120D CU240D DP F	6SL3 544-0FA21-1PA0
Terminal Module TM15 *	6SL3 055-0AA00-3FA0
Terminal Module TM17 High Feature *	6SL3 055-0AA00-3HA0
SINUMERIK	
ADI4	6FC5 211-0BA01-0AAx
SIMATIC IM	
PROFIBUS module IM 174	6ES7 174-0AA00-0AA0

Product	Order no.
SIMATIC ET 200M **	
IM 153-2	6ES7 153-2AA01-0XB0
IM 153-2	6ES7 153-2AA02-0XB0
IM 153-2	6ES7 153-2BA00-0XB0
IM 153-2	6ES7 153-2BA01-0XB0
SM 331 AI8x14Bit	6ES7 331-7HF00-0AB0
SM 331 AI8x14Bit	6ES7 331-7HF01-0AB0
SM 332 AO4x16Bit	6ES7 332-7ND01-0AB0
SM 332 AO4x16Bit	6ES7 332-7ND02-0AB0
SM 321 DI16xDC24V	6ES7 321-1BH10-0AA0
SM 321 DI16xDC24V, Alarm	6ES7 321-7BH01-0AB0
SM 322 DO16xDC24V/0.5A	6ES7 322-1BH10-0AA0
SIMATIC ET 200S **	
IM 151-1 High Feature	6ES7 151-1BA00-0AB0
IM 151-1 High Feature	6ES7 151-1BA01-0AB0
IM 151-1 High Feature	6ES7 151-1BA02-0AB0
2AI I 2/4WIRE HS	6ES7 134-4MB02-0AB0
2AI I 2WIRE HS	6ES7 134-4GB51-0AB0
2AI I 2WIRE HS (Only with IM 151-1 High Feature 6ES7 151-1BA02-0AB0)	6ES7 134-4GB52-0AB0
2AI I 4WIRE HS	6ES7 134-4GB61-0AB0
2AI U HF	6ES7 134-4LB02-0AB0
2AI U HS	6ES7 134-4FB51-0AB0
2AO I HF	6ES7 135-4MB01-0AB0
2AO I HF	6ES7 135-4MB02-0AB0
2AO U HF	6ES7 135-4LB01-0AB0
2AO U HF	6ES7 135-4LB02-0AB0
2AO U HS (Only with IM 151-1 High Feature 6ES7 151-1BA02-0AB0)	6ES7 135-4FB52-0AB0
2DI DC24V HF	6ES7 131-4BB00-0AB0
2DI DC24V HF	6ES7 131-4BB01-0AB0
4DI UC24..48V	6ES7 131-4CD00-0AB0
4DI DC24V HF	6ES7 131-4BD00-0AB0
4DI DC24V HF	6ES7 131-4BD01-0AB0
8DI DC24V	6ES7 131-4BF00-0AA0
2DO DC24V/0,5A HF	6ES7 132-4BB00-0AB0
2DO DC24V/0,5A HF	6ES7 132-4BB01-0AB0
2DO DC24V/2A HF	6ES7 132-4BB30-0AB0
2DO DC24V/2A HF	6ES7 132-4BB31-0AB0
4DO DC24V/0,5A ST	6ES7 132-4BD00-0AA0
4DO DC24V/0,5A ST	6ES7 132-4BD01-0AA0
4DO DC24V/2A ST	6ES7 132-4BD30-0AA0
4DO DC24V/2A ST	6ES7 132-4BD31-0AA0

Product	Order no.
4DO DC24V/2A ST	6ES7 132-4BD32-0AA0
8DO DC24V 0.5A	6ES7 132-4BF00-0AA0
PM-E DC24..48V	6ES7 138-4CA50-0AB0
PM-E DC24..48V/ AC24..230V	6ES7 138-4CB00-0AB0
PM-E DC24..48V/ AC24..230V	6ES7 138-4CB10-0AB0
PM-E DC24..48V/ AC24..230V	6ES7 138-4CB11-0AB0
PM-E DC24..48V	6ES7 138-4CA00-0AA0
PM-E DC24..48V	6ES7 138-4CA01-0AA0
* For additional high-speed output cams, hardware limit switches and measuring inputs ** For analog absolute encoders, analog outputs for hydraulic axes, additional output cams and hardware limit switches.	

The components listed in this section can also be found in a hardware profile of HW Config. For this, select the profile "SIMATIC Technology CPU" in HW Config.

You must have installed the most recent version of S7 Technology in order to obtain a complete selection list in the profile.

DP-V0 slaves on DP(DRIVE)

In addition to ET 200M and ET 200S, the system supports the operation of additional I/O as DP-V0 slave on DP(DRIVE), however, with certain restraints:

- Interrupts are not possible
- DP-V0 slaves can not be operated consistently in isochronous mode on DP(DRIVE)
- Higher response times are to be expected

POSMO A positioning drive

The POSMO A positioning drive can be operated on the DP/MPI segment of CPU 317T-2 DP. You can integrate the positioning drive using the special function blocks of the "Posmo A Library." The Posmo A Library does not support DP(DRIVE).

Product	Order no.
SIMODRIVE POSMO A	6SN21x

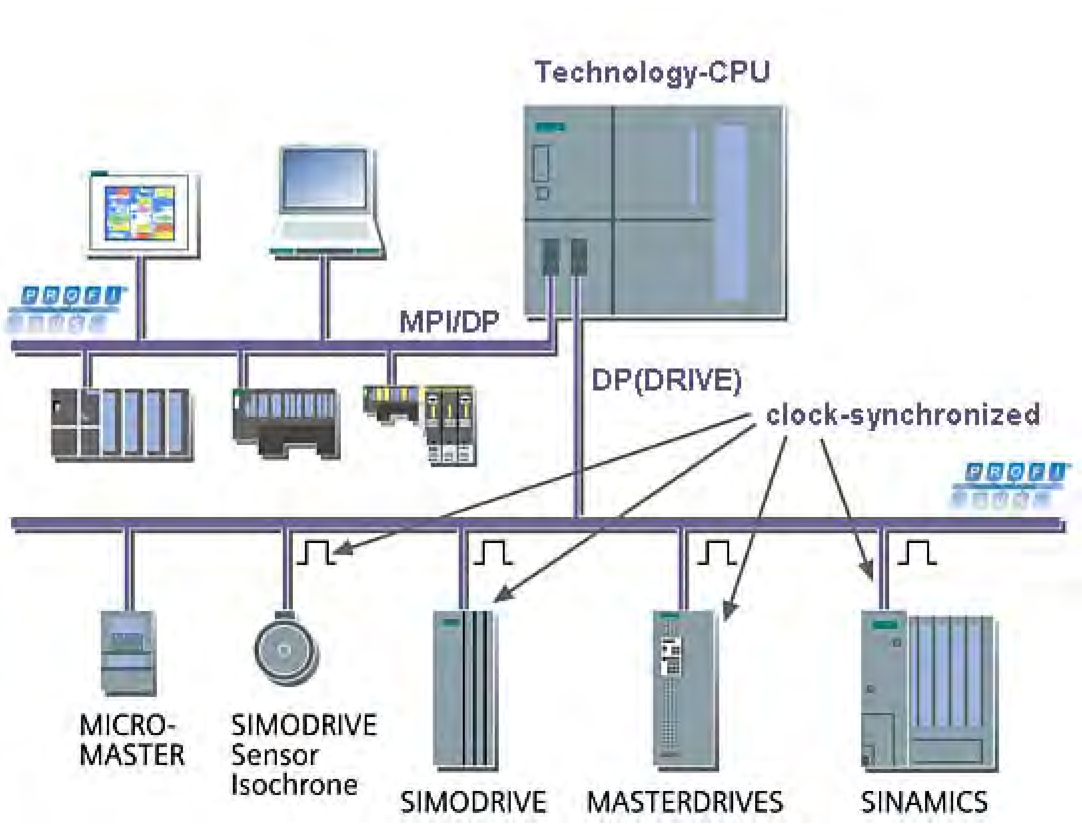
Hardware and software requirements

For information on hardware and software requirements for implementation of the "S7-Technology" options package, refer to the readme.wri file on your product CD.

2.5 Components and what they are used for

2.5.1 Hardware components

The diagram below shows the hardware components of a Motion Control solution with Technology CPU:



Technology CPU

The control unit of the Technology CPU performs the tasks known from a standard CPU of the S7-300 family. The integrated technology controls, evaluates and monitors all hardware components at DP(DRIVE) which are required by the Motion Control tasks. CPU 31xT features 4 integrated digital inputs and 8 integrated digital outputs. MICROBOX T features 8 integrated digital outputs.

MPI/DP

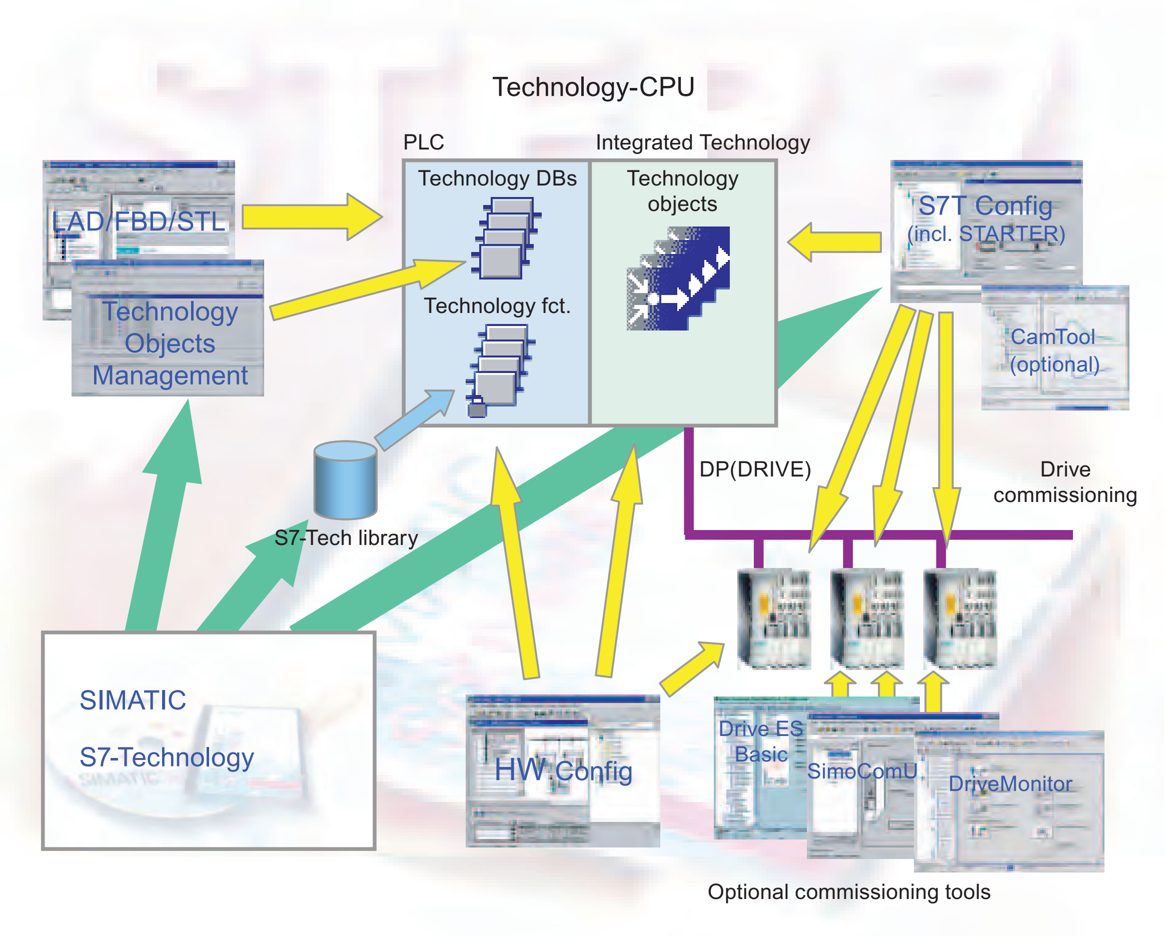
The MPI/DP interface is used to connect additional SIMATIC components, such as a PG, OP, S7 controllers and distributed I/O. Its operation as DP interface allows the configuration of large, extended networks.

DP(DRIVE)

The Technology CPU operates the DP(DRIVE) PROFIBUS interface in isochronous mode. All hardware components addressed by the integrated technology must be available at the DP(DRIVE). That includes components of the MICROMASTER, SIMODRIVE, MASTERDRIVES, SINAMICS, and SIMODRIVE Sensor family.

2.5.2 Configuration tools

All Motion Control applications are configured and programmed in STEP 7. The figure below shows the tools used to configure your MC application.



SIMATIC STEP 7

STEP 7 represents the global platform for configuring and programming the Technology CPU. All configuration tools required are called in SIMATIC Manager of STEP 7.

SIMATIC S7-Technology

SIMATIC S7-Technology is an options package used to configure Motion Control functionality of your Technology CPU. SIMATIC S7-Technology is fully integrated in STEP 7 after installation. Tools included in its installation:

- Technology Objects Management
- S7-Tech Library
- S7T Config, including STARTER

LAD/FBD/STL

You program your application and Motion Control jobs in the LAD/FBD/STL block editor. You read the actual values of your Motion Control application from the user program to analyze information and errors.

Technology Objects Management

The "Technology Objects Management" tool is used to create and delete the Technology DBs, and to set their parameters. You also call the "Technology Objects Management" tool to rename Technology DBs or to assign different block numbers.

S7-Tech Library

The S7-Tech Library is compliant with technology functions to PLCopen standard and is called in your user program as function block. You use these to control your Motion Control commands.

S7T Config

You configure the technology technology objects required to implement your motion control task in S7T Config.

The STARTER functionality for the drives from the MICROMASTER and SINAMICS families is integrated in S7T Config.

STARTER functionality in S7T Config

Parameterize the COMBIMASTER, MICROMASTER and SINAMICS drives with the STARTER functionality that is integrated in S7T Config. The parameterization of these drives can be called up in the SIMATIC Manager and in the Navigator of S7T Config.

The drives mentioned above can also be parameterized as stand-alone drives. Insert the desired drive as a "single drive device" in S7T Config to this purpose. The parameterization is carried out by means of PROFIBUS or the serial interface COM1, depending on the available communication interface.

CamTool (optional)

You can purchase the optional SCOUT CamTool which provides an easy-to-use graphical interface for creating cam disks.

HW Config

HW Config is used to configure the hardware of your Technology CPU and the subnets at the DP/MPI and DP(DRIVE) interfaces.

Drive ES Basic / SimoCom U / DriveMonitor

The optional tool packages Drive ES Basic and/or the stand-alone tools SimoCom U (SIMODRIVE) or DriveMonitor (MASTERDRIVE) are available to support drive commissioning.

Technology data blocks

Technology DBs can be called in the user program to read the actual data of technology objects, that is the actual values and states of an axis, or error information.

Technology functions

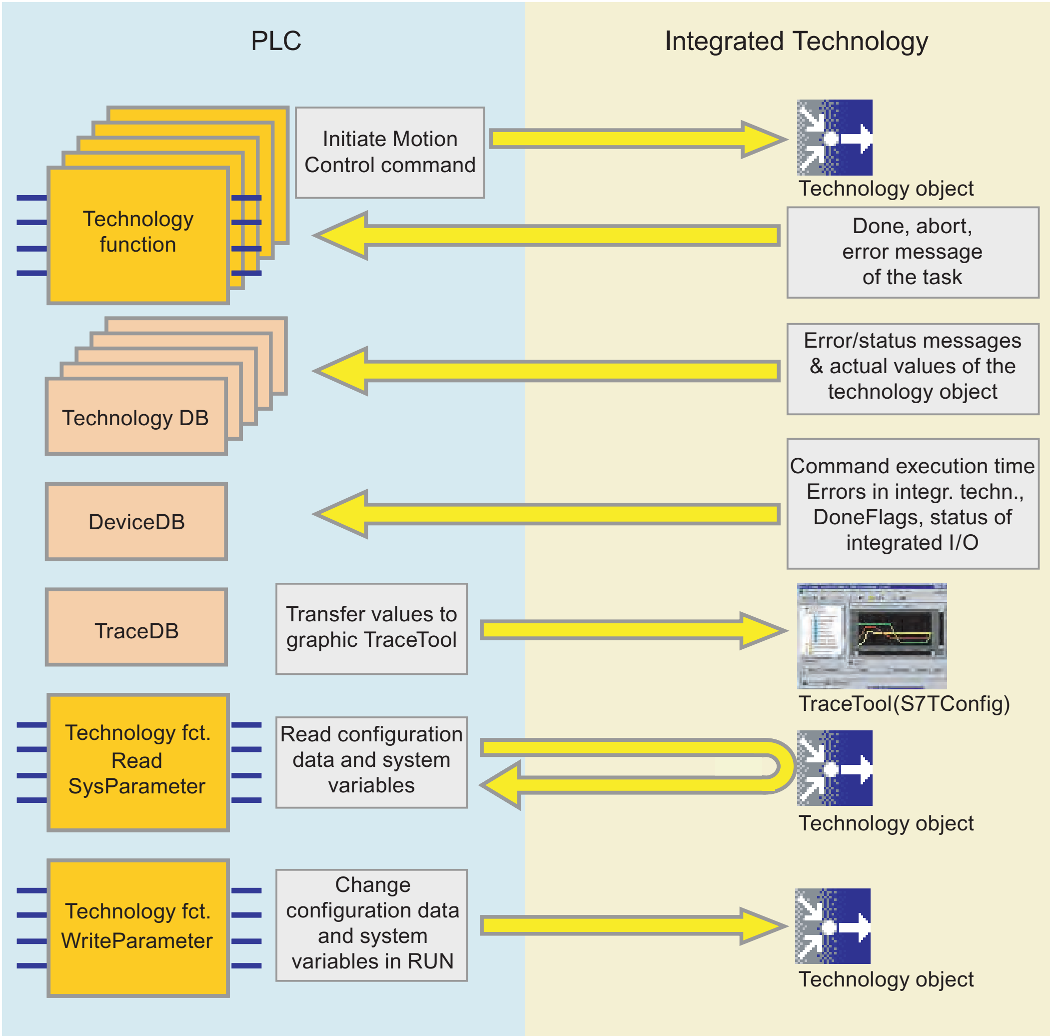
The PLCopen-compliant technology functions are called in the user program of the controller. Technology functions form the command interface to the technology objects. The user program can monitor the status of Motion Control commands by evaluating the output parameters of the technology function.

Technology Objects

The physical drives are mapped to technology objects and their properties parameterized at the technology objects. Each technology object is mapped to the STEP 7 user program as Technology DB to indicate its status. Technology objects may be interconnected and be logically linked to hardware components. All technology objects such as axes, cam disks, output cams, measuring inputs or external encoders are configured in S7T Config.

2.5.3 Technology functions and Technology DBs

Technology functions and Technology DBs form the user interface to the integrated Technology. The figure below shows the corresponding tasks handled by these components:



Technology functions

All Motion Control jobs are initiated in the Technology CPU by means of technology functions. The integrated technology processes the jobs according to the sequence initiated by the control system.

Motion Control commands address the technology objects you configured in S7T Config. The technology objects are addressed by setting the number of the corresponding Technology DB.

The command is initiated by a signal transition (positive edge) at input parameter *Execute* or *Enable*. The output parameters of the technology functions return information about job done and job cancel messages, and error messages if job initiation failed.

Technology DB

The integrated technology writes the actual values of the technology object to the allocated technology DB. Status and error messages generated in the course of job processing will also be written to the technology DB.

The technology DBs are always **non-retentive** and are write protected in the AS, irrespective of any object properties set at the DBs.

DB - MCDevice

The status of integrated technology is mapped to the Technology DB "MCDevice". "MCDevice" contains information about the maximum and average job processing time of Motion Control commands, and about any errors of integrated technology.

In addition, the "MCDevice" technology DB makes it possible to display the status of the integrated I/O and 32 completed messages (*DoneFlag*) of some Technology functions. Which completed messages are displayed you can choose with the input parameter *DoneFlag* of the Technology function.

DB - Trace

S7T Config features the graphical TraceTool which can be used to analyze system parameters and actual values.

In addition to the system parameters of integrated technology, the tool supports logging of up to 8 variables of the S7 user program, (two DINT values, two DWORD values, and four REAL values). The Trace Technology DB interfaces the control system with the integrated technology.

Technology functions ReadSysParameter / WriteParameter

It may be necessary in runtime to temporarily modify the configuration data and system variables of integrated technology. Using the Technology function "MC_ReadSysParameter" you can read the configuration data and the system variables, using "MC_WriteParameter" you can overwrite. Changes made in the parameters are **not retentive**.

Technology Objects 3

3.1 Axes - Basics

3.1.1 Axis technologies

Configure the "Axis" technology object as "speed-controlled axis", "positioning axis" or "synchronization axis." The various axis technologies differ in terms of axis functionality.

Function	Speed-controlled axis	Positioning axis	Synchronization axis
Change data record	X	X	X
Speed or velocity preset	X	X	X
Motion with torque reduction	X	X	X
Positioning	-	X	X
Move to fixed end stop	-	X	X
Homing	-	X	X
Advanced functions			
Measuring inputs	-	X	X
Output cams	-	X	X
Synchronous operation (gear, cam)	-	-	X
Superimposing synchronization (gear / cam)	-	-	X

3.1.2 Special modes of operation

Following mode with disabled power unit

The position and speed controllers are inactive in following mode with disabled power unit. The axis can neither initiate a motion, nor dynamic braking. The actual position and speed values will be updated. The axis position can also be monitored in following mode with disabled power unit if external functions are active. This mode is only supported for real axes.

Following mode with enabled power unit

The position control is inactive, and speed control is active in following mode with enabled power unit. The actual position and speed values will be updated. The axis position can also be monitored in following mode with enabled power unit if external functions are active at the axis. This mode is only supported for real axes.

Simulation mode

Simulation mode is used to test the programmed sequences in the controller and the interaction between different axes based on trace recordings without moving the axis. This mode is only supported for real axes.

During simulation mode all drives must be connected and be operating without faults. An axis is simulated internally by setting the actual values equal to the setpoint values. The following error in simulation mode always equals zero.

3.1.3 Axis types

You can select two different types of axes for your configuration. The axis type (linear or rotary, each also as modulo axis) is in essence determined by mechanical conditions, and by the physical units used to calculate axis-specific variables such as the position or speed.

- **Linear axes**

  Linear axes are usually configured where the axis traversing range is mechanically limited. The position profile is linear within the traversing range. Basic physical units of the motions are length units such as millimeter.

- **Rotary axes**

  Rotary axes are usually configured for the rotary motion range. Their motion range is not limited mechanically. Basic units for these motions are rotational units such as degrees. Usually, rotary axes are also configured for operation as modulo axes.

Linear and rotary axes can be configured for operation as modulo axes. The position of such a modulo axis is defined (setpoint and actual value) within a range which is derived from the modulo start value as the low limit, and by the modulo start value plus the modulo length as the high limit.

A rotary axis with a motion range from 0 ° to 360 °" has a modulo start value of 0 °, and a modulo length of 360 °.

The axis position is reset to 0 ° when it reaches its high limit (360 °.) When it passes the low limit (0 °), the axis position is set to the modulo start value plus the modulo length (360 °.) The position profile is linear within the modulo length.

Note

If linear or rotary axes are not configured for operation as unidirectional modulo axes, the position value increments continuously.

The variables representing the setpoint and actual position are of the data type REAL, with a resolution of 23 binary digits (mantissa.) Positioning precision of the axis therefore deteriorates proportionally to the increase of the position value. For this reason you should preferably use modulo axes for infinite axes, or reset the position to zero at appropriate times.

3.1.4 Hydraulic axes

Valid for Integrated Technology with firmware V3.2.x or higher

Integrated Technology with firmware V3.2.x or higher supports the control of hydraulic axes. The Technology CPU supports the control of hydraulic drives / hydraulic valves by way of +/- 10 V analog signals. The axis returns a manipulated motion variable as percentage, that is, -100% is equivalent to -10 V, and +100% is equivalent to +10 V.

The configuration and programming is similar to that of electrical axes.

Special features of the hydraulic axis

- Inclusion of a valve profile in the calculation in order to compensate for non-linearity between the control signal and the hydraulic drive.
- An "Axis" technology can be used to control several hydraulic drives / valves.
- No cascaded speed or torque limiting function in the hydraulic drive.
- Additional friction compensation with offset superimposition

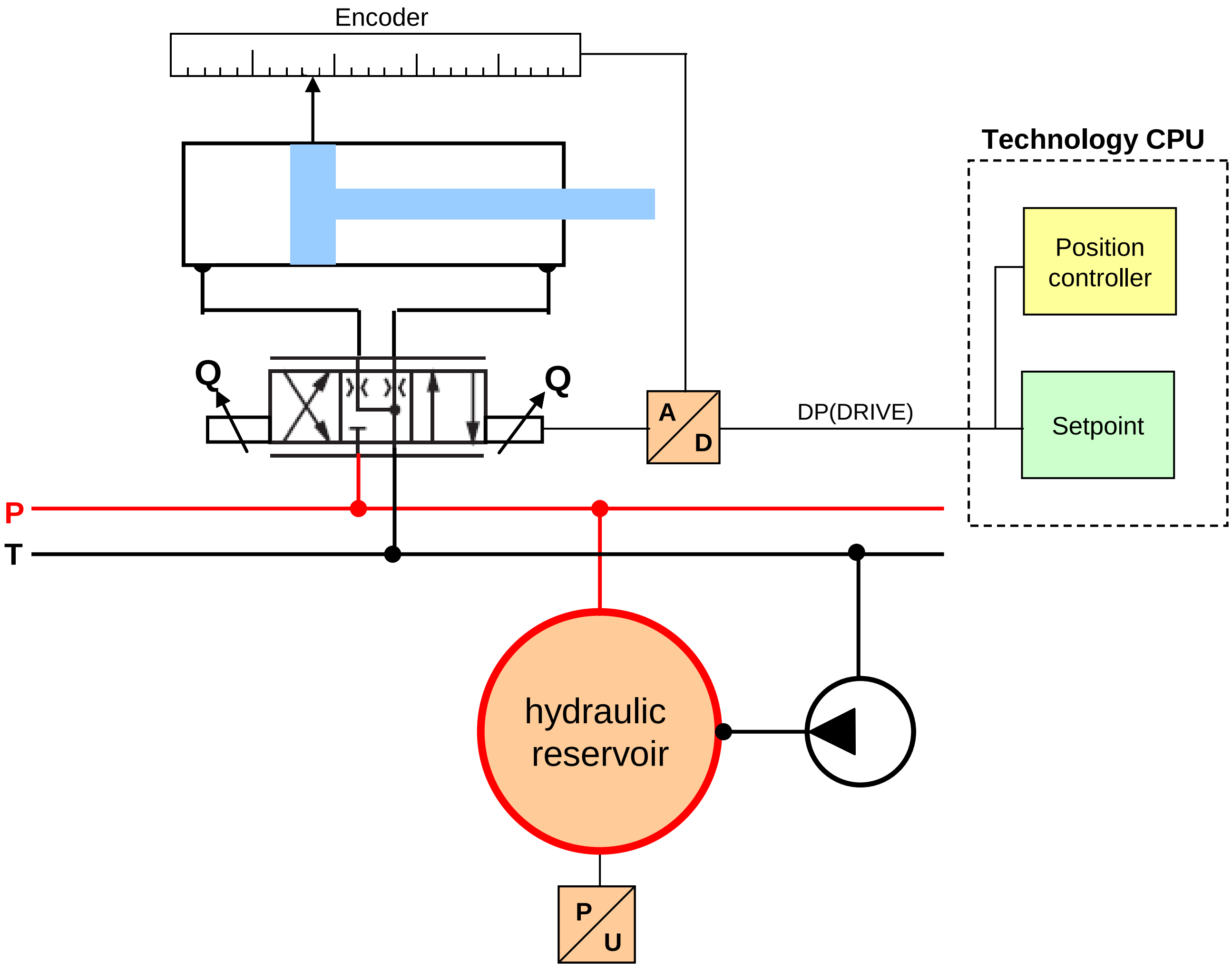
I/O supported for operation on DP(DRIVE)

- Drive interface IM 174/ADI4
- Analog output module of ET 200M or ET 200S for controlling the hydraulic drive
- Incremental encoders / rectangular TTL
- Absolute encoders / SSI
- Analog absolute encoders / sensor analog

Control variants supported by the Technology CPU:

Example of a control by way of proportional directional valves

Example of the control of proportional directional valves using the Technology CPU



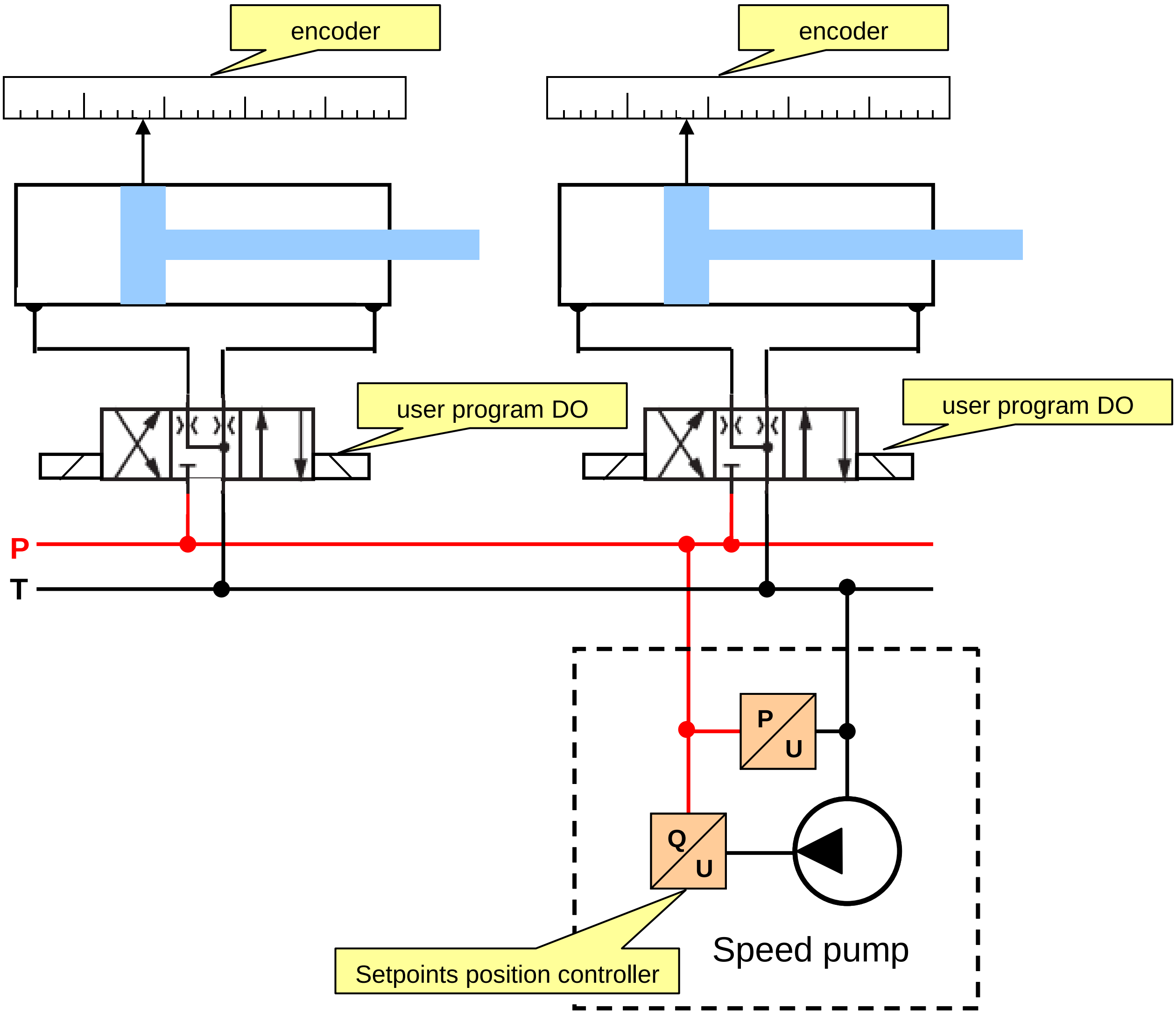
A hydraulic pump charges a pressure-regulated hydraulic reservoir. The hydraulic pump must be controlled by the user program.

In this example a 4/3 proportional directional valve controls the cylinder or the hydraulic drive. The Technology CPU controls the flow volume (Q) of the proportional directional valve by setting an analog output at DP(DRIVE). An encoder can be used to log the cylinder position.

Hydraulic axes can be configured for operation as speed-controlled axis, positioning axis or synchronization axis.

Example of the control of a variable speed pump using switching directional valves

Example of the control of switching directional valves and of a variable speed pump



A variable speed pump controls the flow volume (Q) for cylinder movement. The Technology CPU handles the control of the variable speed pump by setting an analog output at DP(DRIVE). The control of 4/3 switching directional valves must be handled in the user program by setting digital outputs. An encoder can be used to log the cylinder position.

This variable speed pump control only allows unidirectional control of the hydraulic by the Technology CPU, that is, as speed-controlled, position-controlled, or synchronization axis. The reversal of direction must be handled in the user program by setting the 4/3 switching directional valves and activating separate valve profiles.

The example shows that a single hydraulic axis (variable speed pump) can be used alternatively to control several cylinders or hydraulic drives. in this case, use a separate pair of valve profiles for each hydraulic drive of the shared axis.

3.1.5 Real and virtual axes

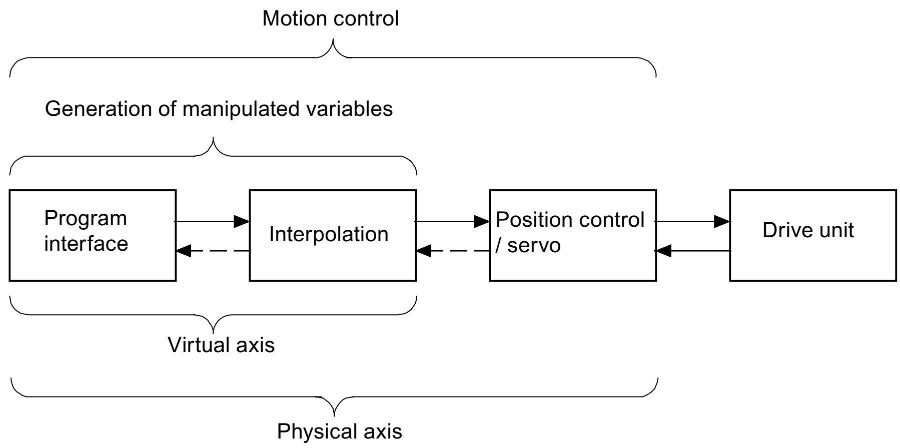
This documentation uses the term "Axes" for both real and virtual axes.

- **Real axis**

  This axis features motion control functions and a drive and encoder interface
- **Virtual axis**

  This axis type supports the generation of control variables, however, it has neither a closed-loop control, nor a drive interface, nor an encoder interface. The setpoints and actual values are always equivalent. A virtual axis is usually implemented as an auxiliary axis, for example, to generate the master setpoints for several real axes when operated as leading axis in a synchronized group.

Difference between a real and a virtual axis, based on the example of a position axis:



Note

Hydraulic axes are real axes with valve profile.

3.1.6 Difference between "Axis" and "Drive"

The "Axis" technology object interfaces the user program with the actual drive. It receives, executes and monitors Motion Control commands requested by the user program of the PLC.

The axis communicates via PROFIBUS with the drive which contains the speed and current controls.

Drives are configured and commissioned separately from the axis.

Function interface to the drive

The Technology CPU supports operation of digital drives (SIMODRIVE 611 universal, MASTERDRIVES MC, for example) by way of DP(DRIVE) interface, or of analog drives by way of IM 174 / ADI4, or of stepper drives by way of IM 174. A defined message frame interfaces the technology object and the drive component and must be selected and configured separately for each component according to the functionality required.

These message frames are used to transfer data such as control words, status signals or encoder information between the control system (technology object) and the drive component (drive, for example.)

Note

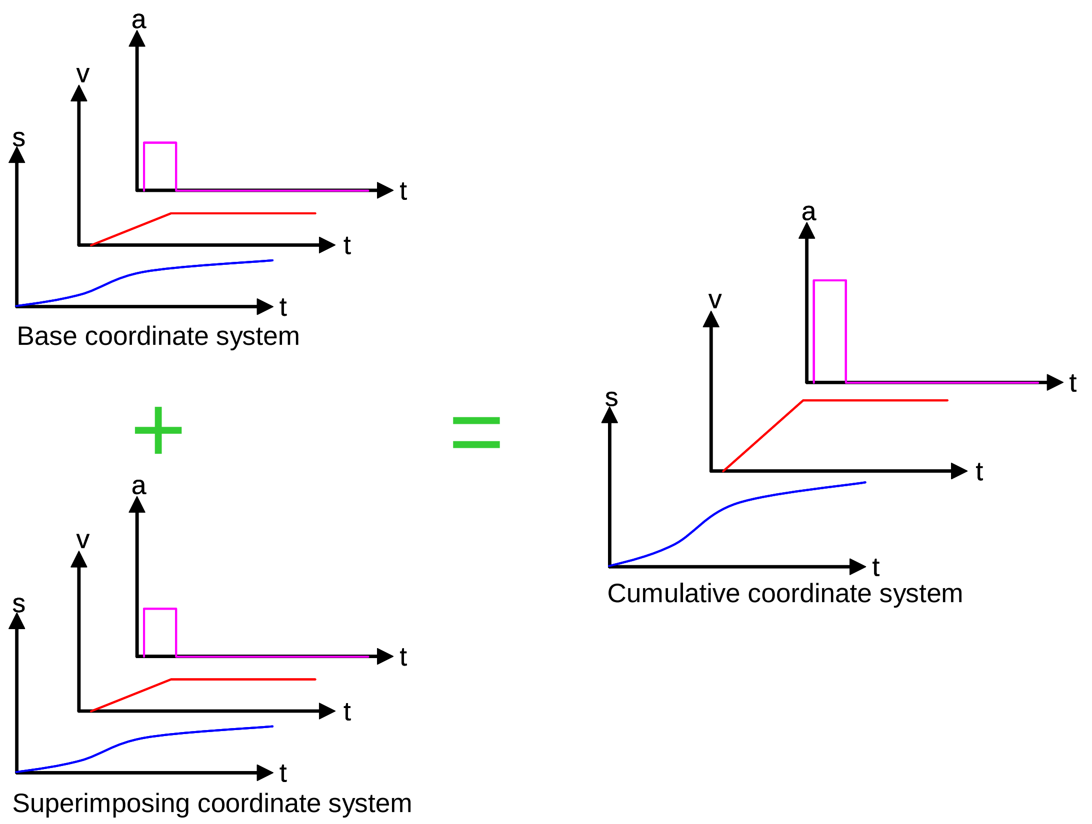
An axis can only execute functions which the connected drive actually supports. The supported functions such as operation with SIMODRIVE 611U or MASTERDRIVES MC are described in the drive documentation (for further information please also refer to the corresponding product descriptions.)

3.1.7 Base coordinate system - superimposing coordinate system

We distinguish between two different types of motion in the Technology CPU in terms of their effect:

- Basic motion
- Superimposed motion

The Technology CPU calculates motions based on a base coordinate system and a superimposing coordinate system. Superimposing motions return the following scenario:



All coordinates (time/distance, time/velocity and time/acceleration) are added at a given time to return the cumulative coordinate system.

Basic motion commands:

- "MC_MoveAbsolute"
- "MC_MoveRelative"
- "MC_MoveAdditive"
- "MC_MoveVelocity"
- "MC_MoveToEndPos"
- "MC_GearIn"
- "MC_GearOut"
- "MC_CamIn"
- "MC_CamOut"
- "MC_Phasing"

These commands affect the base coordinate system.

Commands for superimposing motions:

- "MC_MoveSuperImposed"
- "MC_GearInSuperImposed"
- "MC_GearOutSuperImposed"
- "MC_CamInSuperImposed"
- "MC_CamOutSuperImposed"
- "MC_PhasingSuperImposed"

These commands affect the superimposing coordinate system.

The "MC_Home" command may have an effect on both coordinate systems, depending on the value at input parameter *Mode*.

The vectors of the coordinate systems are indicated as system variables at the axis.

Values of the base coordinate system at the axis:

- *basicmotion.positionbasicmotion.velocitybasicmotion.acceleration*

Values of the superimposing coordinate system at the axis:

- *superimposedmotion.positionsuperimposedmotion.velocity*
- *superimposedmotion.acceleration*

Cumulative coordinates:

- *positioningstate.commandposition*: Position setpoint (total)
- *motionstatedata.commandvelocity*: Velocity setpoint (total)
- *motionstatedata.commandacceleration*: Acceleration setpoint (total)

Only one basic motion and one superimposing motion may be active at the axis at any given time. A superimposing may also be active without active basic motion.

3.1.8 Homing

The position entries and displays for position-controlled axes refer to the axis coordinate system. The axis coordinate system must be synchronized with the real, physical position of the axis.

Note

Note the following:

- Absolute encoders must be adjusted **only once**.
- The homing position of axes with incremental is retained when the CPU goes into STOP, and is lost after POWER OFF
- A homed axis must be homed again after its initialization (*Restart*) when operated with incremental measuring system.
- Motion commands with relative position definitions ("MC_MoveRelative", for example) can be executed both in homed / non-homed state.
- Motion commands with absolute positioning at a non-homed axis:
  - Executable if "no" was selected from the drop-down list in S7T Config, **Axis > Homing** dialog box, "Active homing" tab
  - Not executable if "yes" was selected from the drop-down list in S7T Config, **Axis > Homing** dialog box, "Active homing" tab

---

Incremental encoders are synchronized using the MC_Home technology function. In active homing, either the reference point coordinate, or the reference point coordinate minus the reference point offset will be set at a defined mechanical position of the axis.

Homing modes

- **Active homing**

  In active homing mode, the "MC_Home" technology function performs the required reference point approach. Any active motions will be canceled for this operation. Homing modes available in S7T Config:

  – Reference cam and encoder zero mark
  – Encoder zero mark only
  – External zero mark only

- **Passive homing**

  In passive homing mode, the "MC_Home" technology function does **not** technology function does execute a reference point approach. Active motion commands will not be affected. The "reference point approach" required must be implemented by means of external functions or in the user program.
  Homing modes available for configuration in S7T Config:

  – Reference cam and encoder zero mark
  – Encoder zero mark only
  – External zero mark only
  – Default setting

- **Direct homing**

  The axis position is set without making allowances for reference cams or zero marks. The axis must be at a standstill in order to allow precise assignment of the reference point to a mechanical position.

- **Correct position value**

  An offset value is subtracted from the current axis position. Current motions and homing operations are not affected.

- **Correct the internal axis coordinate system**

  An offset value is subtracted from the current position of the base or superimposed coordinate system. Current motions and homing operations are not affected.

Note

Device-specific properties

The reference cam can be connected either to the integrated inputs of the CPU or to a slave on DP(DRIVE) for homing with reference cam and encoder zero mark.
In homing mode with "external zero mark only", the external zero mark must be connected to the input device where the encoder is also registered, for example, at the drive or at IM 174/ADI4.

For further information on device-specific conditions and additional parameter settings, refer to the supplementary information on SIMODRIVE 611U, MASTERDRIVE-MC, SINAMICS S120 or IM 174/ADI4 on your product CD-ROM and to the manuals.

3.1.9 Absolute encoder adjustment

The "MC_Home" (*Mode* = 5) and "*"(* MC_ExternalEncoder= *Mode6*MC_ExternalEncoder) technology functions are used to activate the absolute encoder offset in the calculation for axes and external encoders.

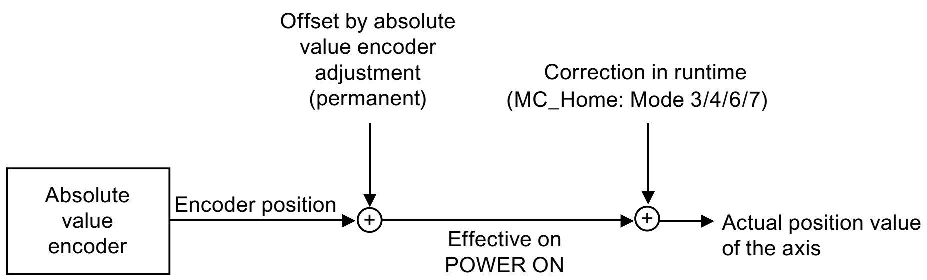
The actual position of an axis with absolute encoder is set to the required value. This shifts the absolute phase of the absolute encoder by an offset value. This offset is stored permanently and remains active until the next time the absolute encoder is adjusted. This function must be executed once in the controller commissioning phase.

The offset is deleted if the Technology CPU changes from STOP to RUN and the technology object is invalid. This is, for example, the case if the technology CPU is started without an MMC or Compact flash card.

To adjust the absolute encoder:

1. Disable the software limit switches, as you can not adjust the absolute encoder while these are active.
2. Move the axis to the relevant reference position, and then adjust the absolute encoder ("MC_Home" or "MC_ExternalEncoder" technology function.)
3. Enable the software limit switches again (if necessary)

Note that the result of absolute encoder adjustment only offsets the encoder value. The offset of absolute encoder adjustment and the absolute encoder value are decisive for the position after POWER OFF or restart ("MC_Reset", *Restart = TRUE*.) The current actual position is also affected during operation by the modulo settings of the axis and by positioning or position correction commands.



3.1.10 Data record changeover

Introduction

You can create several axis data records to change controller parameters, or to toggle from motor encoder mode to machine encoder mode while the system is in run, for example.

The axes listed below support multiple data records:

- Speed-controlled axes
- Positioning axes
- Synchronization axes

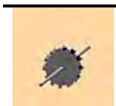
Note

Virtual axes only have one data record.

Configuration

Create the date record in the axis configuration in S7T Config. Program the data record using the Axis Wizard. Call the "MC_ChangeDataset" technology function to change the data record in RUN..

3.2 "Speed-controlled axis" technology object



Use the "Speed-controlled axis" technology object if you only want to preset, control and monitor the speed of an axis when the position is irrelevant.

Operating modes supported by speed-controlled axes: Technology object - Speed-controlled axis Speed-controlled axis - Technology object

- Speed control (can be set using technology function "MC_Power")
- Following mode (can be set using technology function "MC_Power")
- Simulation mode (can be set using technology function "MC_Power")

Functions supported for use with speed-controlled axes:

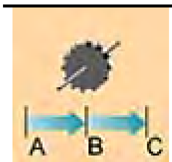
- Speed preset
- Motion with torque reduction

Technology functions

Technology functions supported by the technology object:

MC_Power	MC_Reset	MC_Stop
MC_Halt	MC_MoveVelocity	MC_ReadSysParameter
MC_WriteParameter	MC_SetTorqueLimit	MC_ChangeDataset

3.3 "Positioning axis" technology object



Use this technology object to move the axis to a defined position and to control and monitor that position.

The positioning axis contains all functions of the speed-controlled axis.

Operating modes supported by positioning axes: Technology object - Positioning axis Positioning axis - Technology object

- Speed control (can be set using technology function "MC_MoveVelocity")
- Position-controlled (can be set using technology function "MC_Power")
- Following mode (can be set using technology function "MC_Power")
- Simulation mode (can be set using technology function "MC_Power")

Functions supported for use with positioning axes:

- Traversing at preset speed
- Motion with torque reduction
- Positioning
- Move to fixed end stop
- Homing

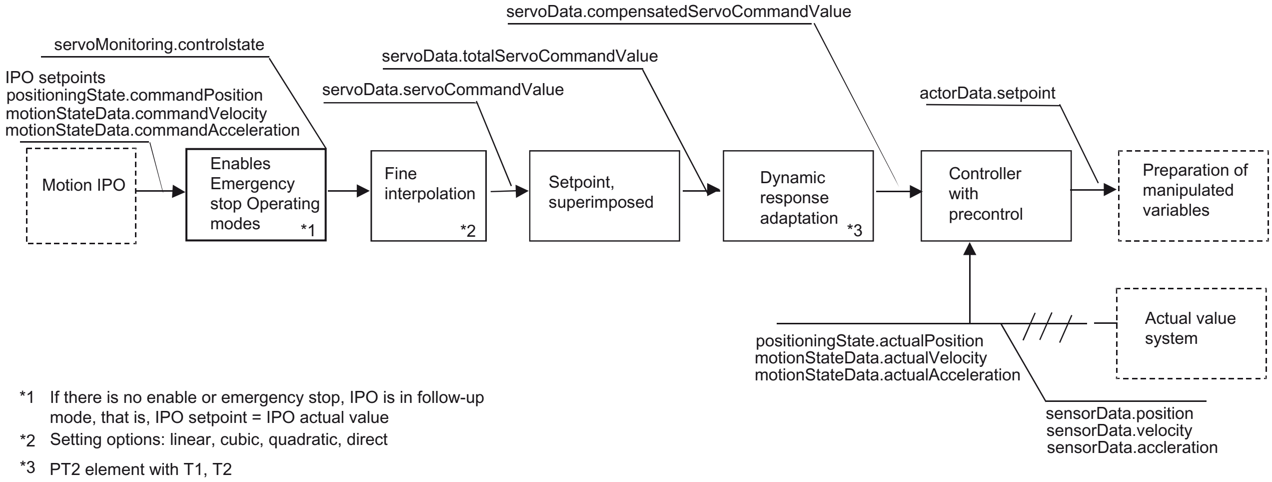
Technology functions

Technology functions supported by the technology object:

MC_Power	MC_Reset	MC_Home
MC_Stop	MC_Halt	MC_MoveAbsolute
MC_MoveRelative	MC_MoveAdditive	MC_MoveSuperImposed
MC_MoveVelocity	MC_MoveToEndPos	MC_ReadSysParameter
MC_WriteParameter	MC_SetTorqueLimit	MC_ChangeDataset

3.3.1 Position Control

The following figure shows the block diagram of the positioning axis with position control.



When position control is active, controllers, monitoring, and compensation are active. The monitoring functions are disabled in certain modes, for example, the position-related monitoring functions for torque or pressure limiting.

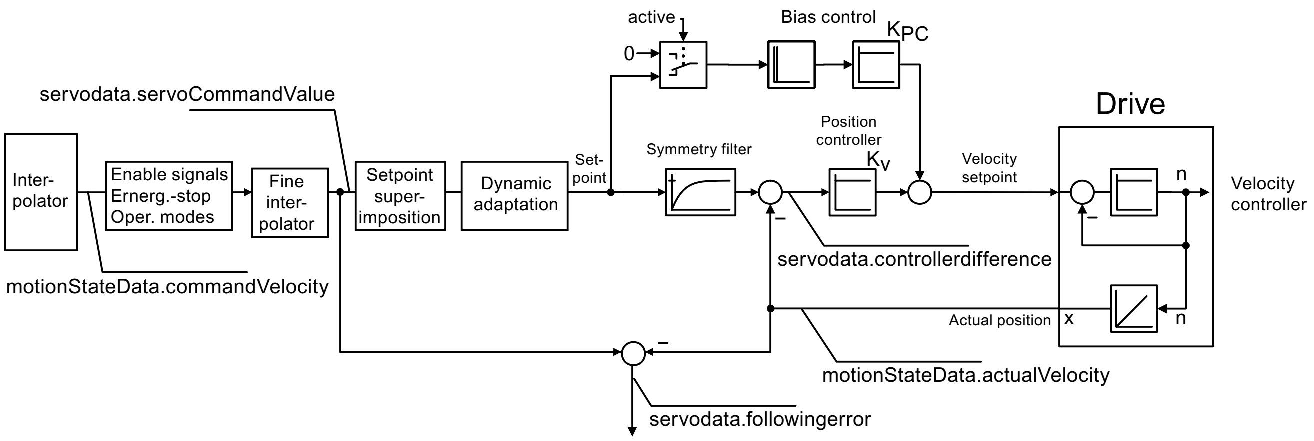
All compensation functions can be enabled/disabled.

Encoder systems, actual value calculation, and monitoring functions are active on the actual value side when position control is not activated. Compensation functions are not taken into account.

The *servomonitoring.controlstate* system variable indicates whether the position controller is active.

Control loop structures

S7T Config provides a P-action controller with or without pre-control function, and a PID controller.

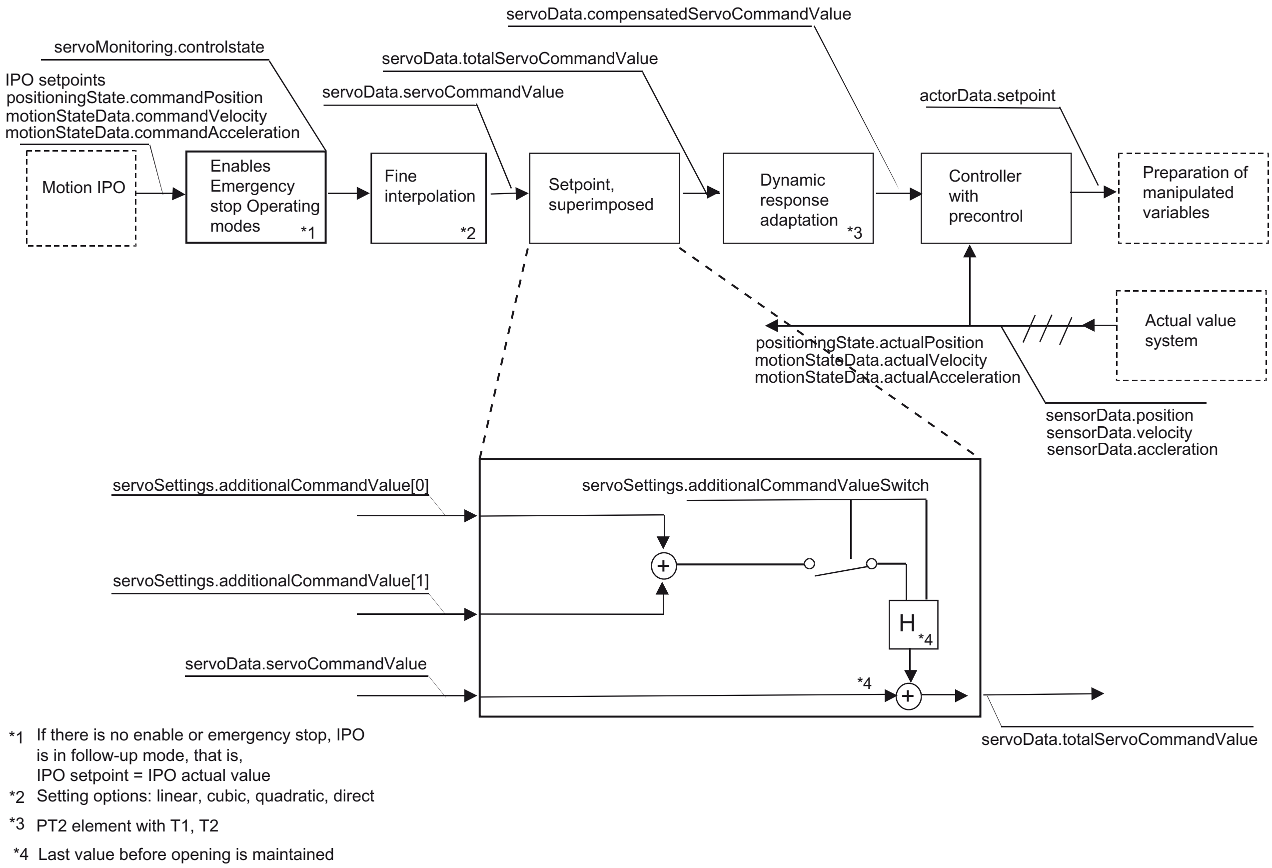


When the mode is changed from speed-controlled to position-controlled mode while the axis is in motion, the equivalent time of the position controller is required to set the setpoint. The equivalent time is set during configuration in *dynamicData.positionTimeConstant* (electrical axis) or in *dynamicQFData.positionTimeConstant* (hydraulic functionality).

Quantization of the controller error

You use the *commandValueQuantizationFilter* configuration data element to quantize the controller error.

Controllers with pre-control



Note

Controllers featuring a precontrol function should be used as shown below:

- P-action controller with pre-control for the electric axis
- DSC to improve the control quality with greater kv values for digitally coupled drives (only with P-action controllers with pre-control)
- PID controller at hydraulic axes. The actual value can optionally be set directly at the D-action element)

Balancing filters

Balancing filters are used to delay setpoints by the value of the process transition time, before these are compared with the actual value and multiplied by the position control gain, and are then set again at the process.

When precontrol is active, allowance can be made in the balancing filter for the process response prior to formation of the system deviation from the position setpoint and actual position.

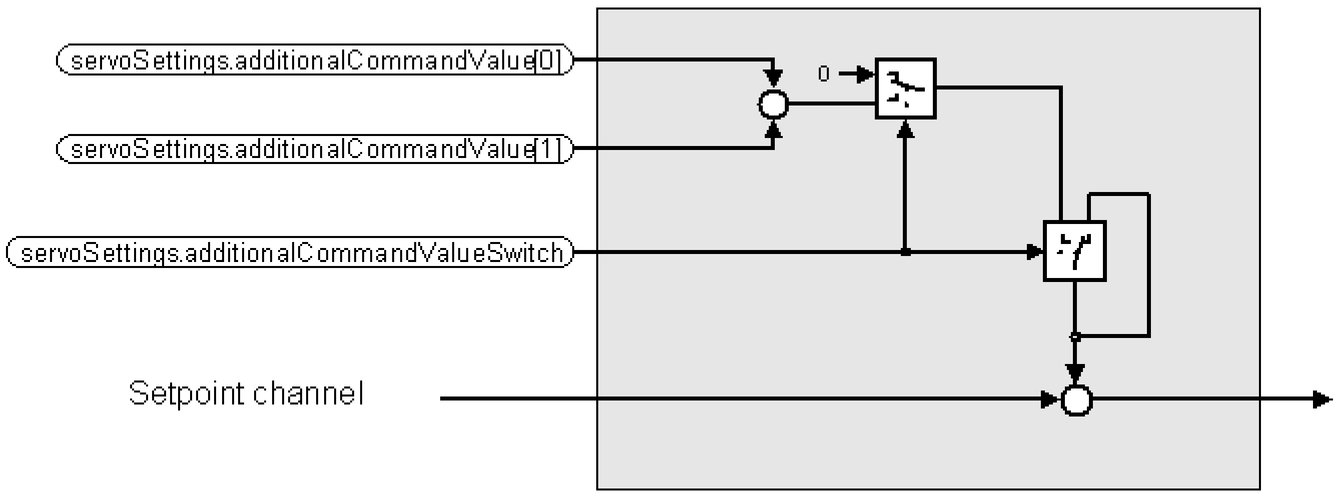
- A PT1 filter is used as a balancing filter with the configuration data element setting *balanceFilterMode = MODE_1*. Its time constant is set in *dynamicData.velocityTimeConstant* for an electrical axis and in *dynamicQFdata.qOutputTimeConstant* for a hydraulic axis.
- The following parameters are taken into consideration when the configuration date *balanceFilterMode = MODE_2* (V3.1 and higher) is set in the balancing filter:
  - The equivalent time of the speed control loop
  - The dead time determined by the system for the drive
  - A dead time that can be input in additive increments by the user
- The equivalent time of the speed control loop, including the dead time determined by the system with respect to the drive, plus an additive dead time that can be entered by the user are taken into account in the balance filter with the configuration data element *balanceFilterMode = MODE_2* (V3.1 or later).
- The equivalent time of the speed-control loop is included in settings with DSC .

We recommend you use the setting *balanceFilterMode = MODE_2* for the balancing filter.

3.3.1.1 Setpoint superimposition

Setpoint superimposition allows you to set additive setpoints at the setpoint channel.

You can tune the controller by setting additional signals of the function generator at this channel.



Note

An axis operated with setpoint superimposition cannot be switched back directly to normal position-controlled mode. You first have to reset the setpoint superimpositions to zero.

Superimposition has an effect on the position of the positioning axis. The superimposition also applies to the active position control and interpolator in following mode.

3.3.1.2 Dynamic response adaptation

The setpoint circuit of the control loop contains a configurable, dynamic PT2 filter with the time constants T1 and T2. This filter can be used to tune axis synchronization. This feature allows you to adapt the dynamic response of faster axes to the response of the slowest axis.

The equivalent time constant T_{LR} of the axis with the poorest dynamic response is selected as the resulting total time constant T_{Res}.

The dynamic response of the axis is derived from the equation:

$T_{Res} = T_1 + T_2 + T_{LR}$

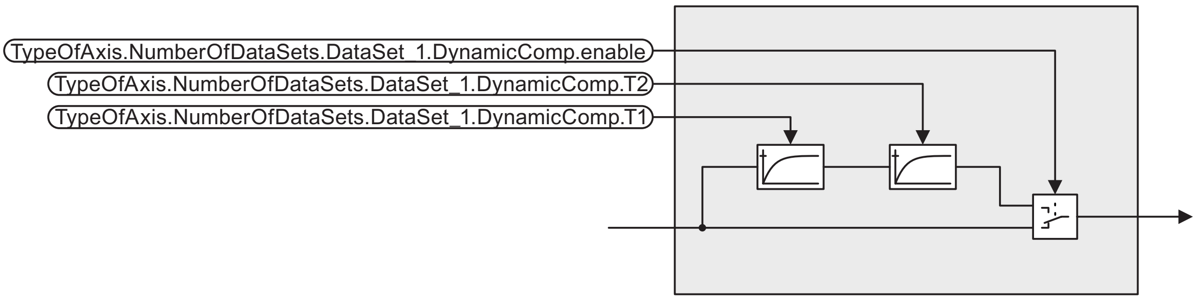
or the value of both time constants is derived from:

$T_1 + T_2 = T_{Res} - T_{LR}$

T_1	Additive time constant 1
T_2	Additive time constant 2
T_{Res}	(desired) resulting overall time constant of the axis
T_{LR}	Equivalent time constant of the closed position control loop

The time constants are set in the *NumberOfDataSets.DataSet_1.DynamicComp* configuration data element.

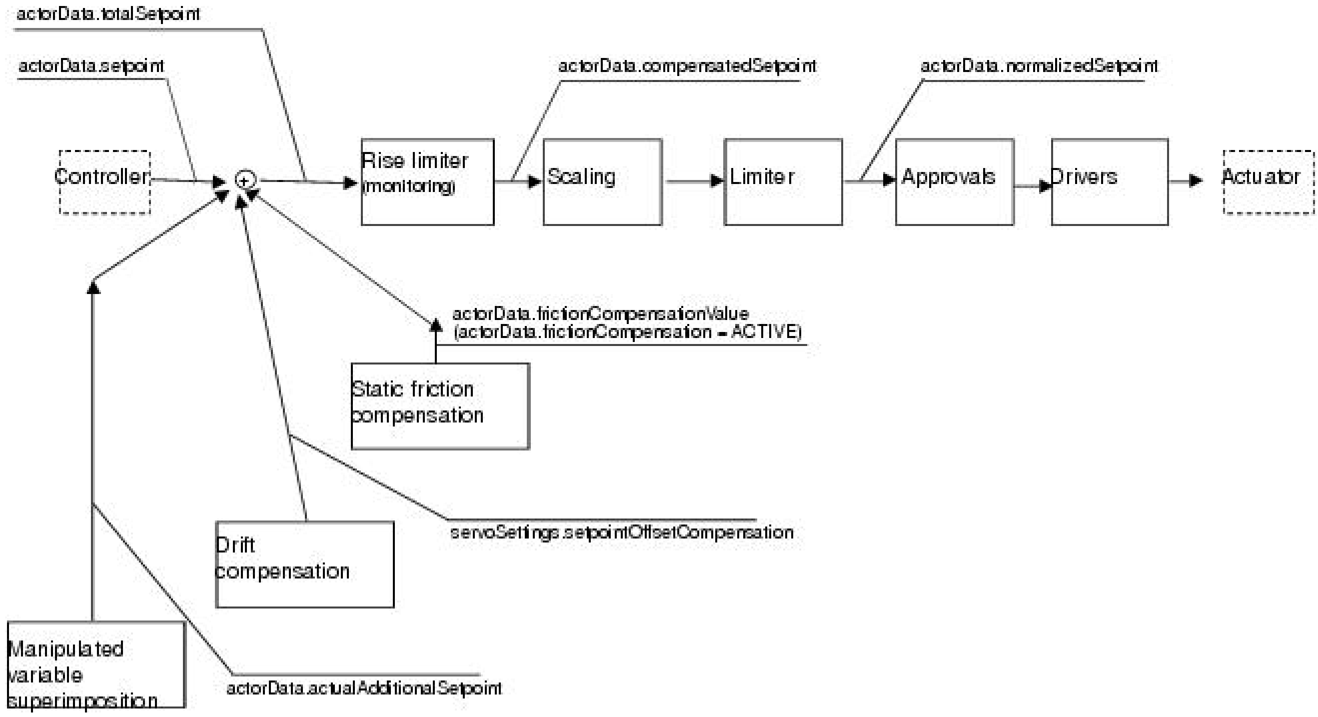
You can set PT2 response and thus prevent overshoot by wiring two PT1 elements in series.



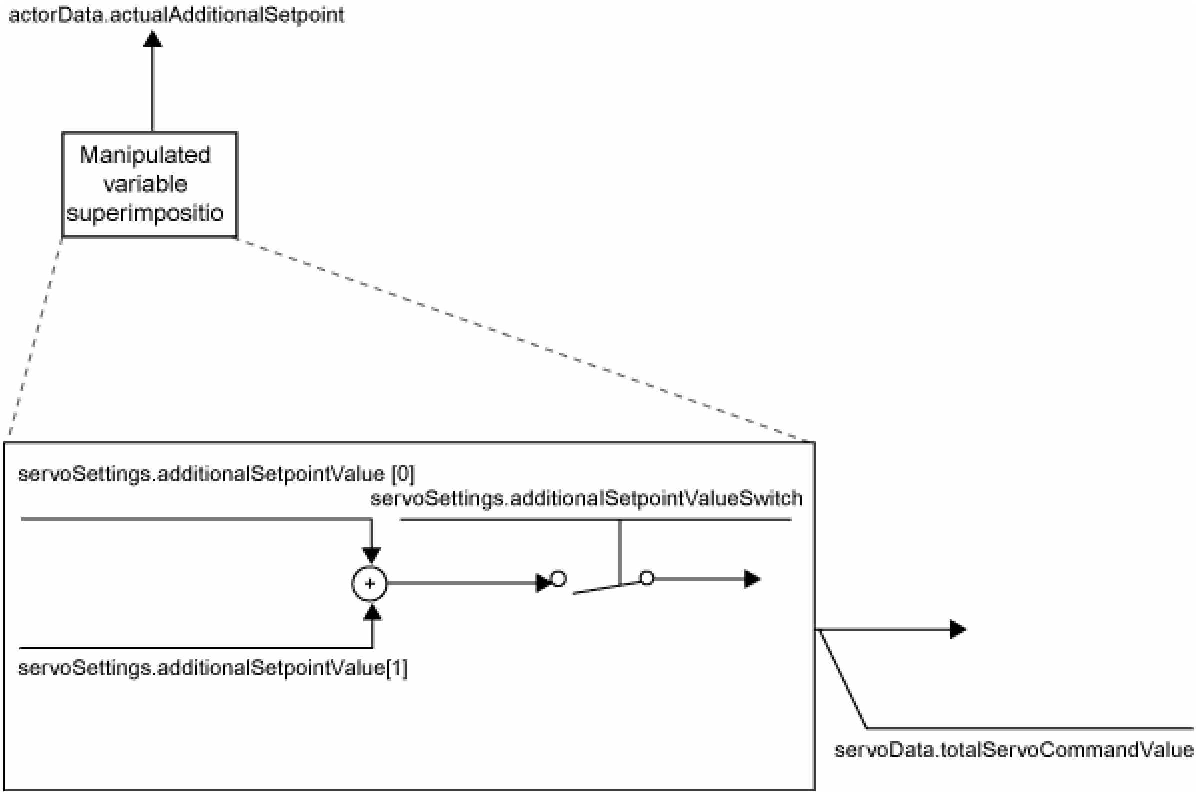
Note

Set T_1 unequal to T_2.

3.3.1.3 Preparation of manipulated variables for the electric axis



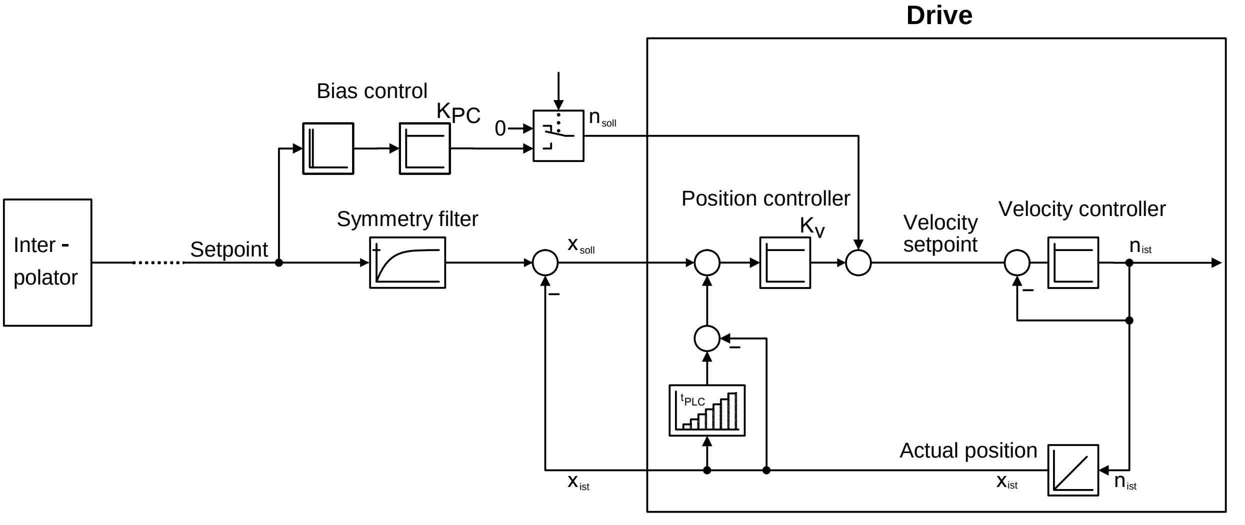
3.3.1.4 Manipulated variable superimposition



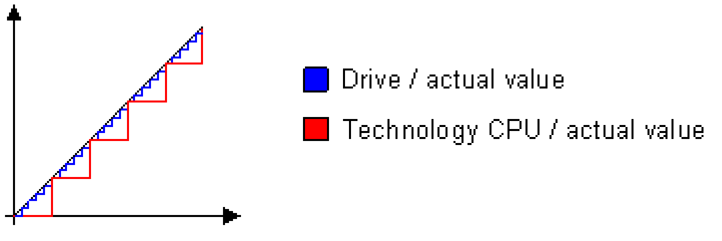
Superimposition of manipulated variables is set using the switch.

3.3.1.5 Dynamic Servo Control (DSC)

The position of position-controlled axes (positioning and synchronization axes) can be controlled either in the CPU or in the drive, provided the drive **supports dynamic Servo Control (DSC).**



For position control with DSC (Dynamic Servo Control), the system calculates both the position controller and the speed controller in the drive. This allows you to set significantly higher gain factors (Kv) for the position controller. The position and speed control parameters are calculated within the same cycle. Hence, the position control and its subordinate control loop, meaning the speed controller, are calculated at precisely the same speed.



The actual values changes are smaller compared to the clock than the actual value in the Technology CPU. The position controller can be corrected more exactly by having a more exact actual value.

This allows tuning of the controller's low-signal response, so that you can set significantly higher gain factors.

DSC allows you to set a higher K_v ratio for the sampling times. This increases the dynamics in the reference variable sequence and of disturbance response at highly dynamic drives. DSC is supported by all drives which support message frame 5, 6, or 105 and 106.

DSC is only useful for P-action position controllers.

Note

The position controller requires tuning. For further information please refer to "Optimizing the position controller - Overview (Page 853)."

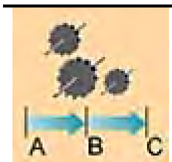
3.3.1.6 Speed-controlled operation of a position-controlled axis

The transition from position-controlled motion to speed-controlled motion and vice versa can be set both when the axis is at a standstill and when it is moving.

The dynamic response parameters and the maximum values for speed specification are derived from the settings for position-controlled axis operation.

The position-related monitoring functions are deactivated. The encoder limit frequency can be exceeded.

3.4 "Synchronization axis" technology object



A synchronization axis is a "following axis" which follows a "leading axis." Use the "synchronization axis" technology object to use the motion and position values of a leading axis as a master setpoint.

The synchronization axis contains all functions of the speed-controlled and positioning axes.

Modes of operation supported by synchronization axes: Technology object - Synchronous axis Synchronous axis - Technology object

- Speed control (can be set using technology function "MC_MoveVelocity")
- Position-controlled (can be set using technology function "MC_Power")
- Following mode (can be set using technology function "MC_Power")
- Simulation mode (can be set using technology function "MC_Power")

Functions supported for use with synchronization axes:

- Traversing at preset speed
- Motion with torque reduction
- Positioning
- Move to fixed end stop
- Homing
- Synchronous operation (gearing / camming)
- Superimposed synchronism (gearing / camming)

Technology functions

Technology functions supported by the technology object:

MC_Power	MC_Reset	MC_Home
MC_Stop	MC_Halt	MC_MoveAbsolute
MC_MoveRelative	MC_MoveAdditive	MC_MoveSuperImposed
MC_MoveVelocity	MC_MoveToEndPos	MC_GearIn
MC_CamIn	MC_GearOut	MC_CamOut
MC_Phasing	MC_ReadSysParameter	MC_WriteParameter
MC_SetTorqueLimit	MC_ChangeDataset	MC_GearInSuperimposed
MC_GearOutSuperimposed	MC_CamInSuperimposed	MC_CamOutSuperimposed
MC_PhasingSuperimposed		

3.4.1 Structure of the "Synchronization axis" technology object

The axis and synchronization object are separate objects. Both objects form a synchronization axis.

The "Axis" and synchronization objects are interactive in terms of their relevant operating states and effectiveness of commands.

If an active error is only pending at the synchronization object, the following axis is still capable of positioning, but not of synchronous operation. You can always avoid this effect by acknowledging all errors. Errors at the "Axis" object have an immediate effect on synchronization functionality.

When the axis reacts with a stop, the synchronous motion is stopped as well.

3.4.2 Synchronized group

The "Synchronization" technology object can be used to interconnect axes so that these form a synchronized group.

Example

A leading axis (master) generates a master setpoint. The synchronization object processes this value based on defined criteria, meaning that the gear ratio, scaling, offset, and the cam disk, and assigns it to the following axis as a control variable.

Note

The master setpoints and slave values are coupled without physical conversion in the relevant programmed unit. If, for example, the system operates with a linear leading axis (in mm units) and a slave rotary axis (in degree units), then one millimeter corresponds to one degree at a conversion ratio of 1:1.

Example of mechanical synchronism

Gear synchronism is given, for example, when two mechanically coupled rollers are driven by the same motor.

The camming model can be interpreted as a cam gear consisting of a mechanical cam disk and a sensing mechanism.

The technology functions used to synchronize and desynchronize components of a synchronized system correspond with the function of a mechanical coupling.

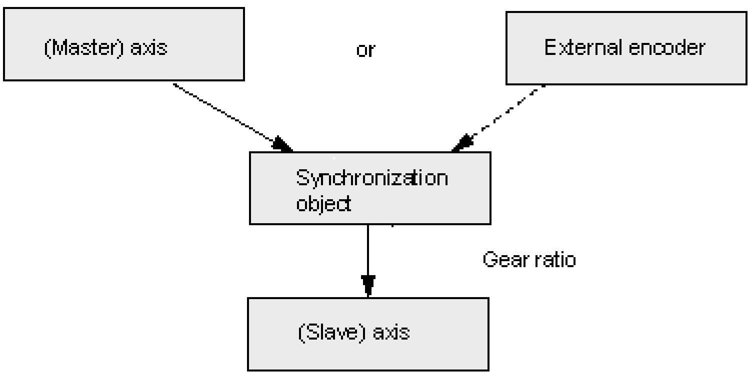
Synchronized group

A synchronized group consists at least of these elements:

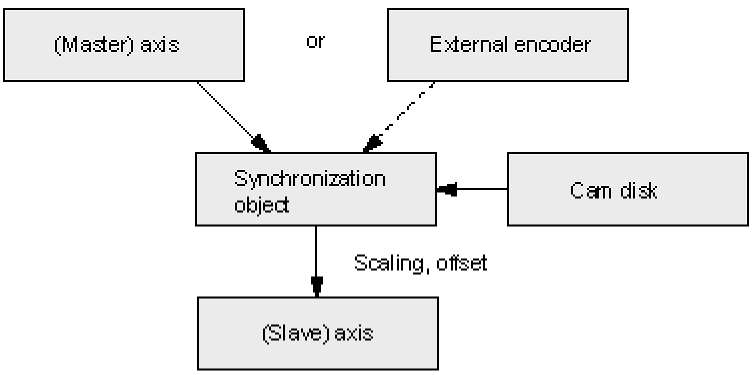
- A leading axis that returns the master setpoint

  Leading axes can be formed by real and virtual positioning and synchronization axes, or by external encoders
- A synchronous object
- A following axis

Objects in gearing mode



Objects in camming mode

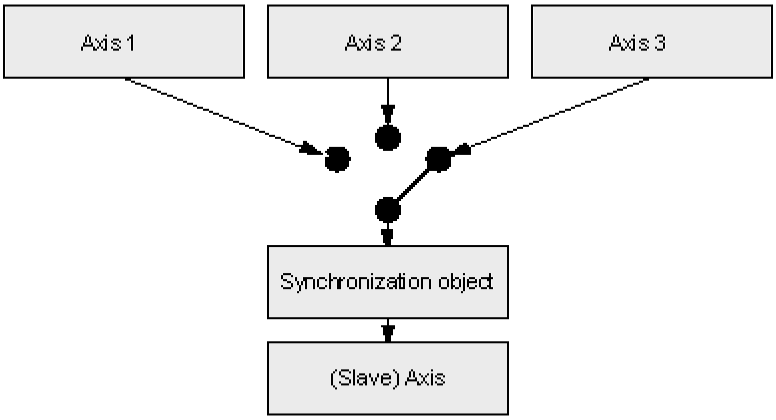


Master setpoint coupling

A following axis can be interconnected with multiple master setpoints using the synchronization object. However, only one master setpoint can be evaluated actively at any time. The master setpoint can be returned by positioning axes, by a synchronization axes (real or virtual), or by external encoders.

In order to change over to a different master setpoint, call the "MC_GearIn" (gearing) or "MC_CamIn" (camming) technology function again in the user program. Use the "MC_GearInSuperimposed" and "MC_CamInSuperimposed" technology functions for superimposing synchronization commands.

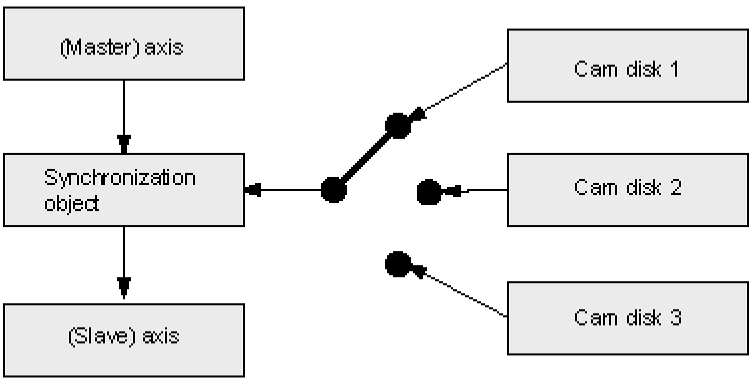
Example of a synchronization object with several master setpoints



Cam coupling

In the same way, you can use several cams disk for camming. You can use the "MC_CamIn" or "MC_CamInSuperimposed" technology functions in the user program for dynamic changeover of the cam disks.

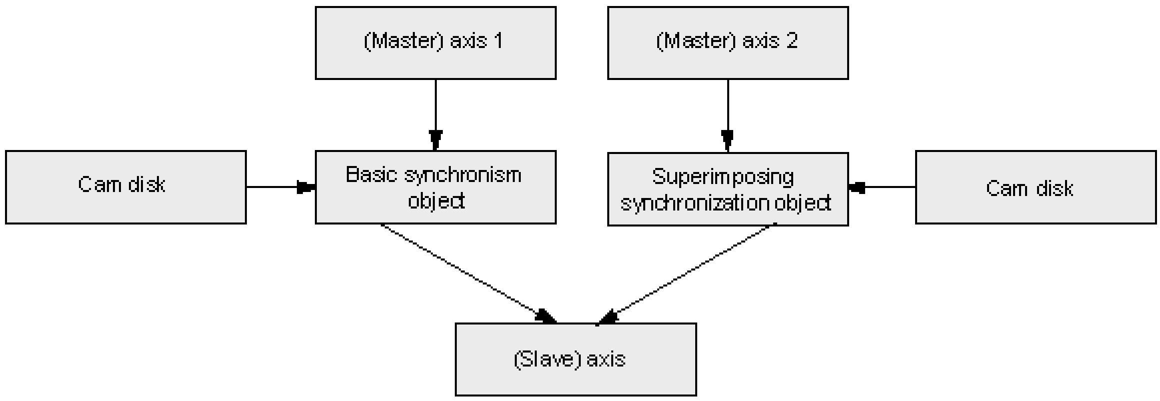
Example of camming with several cam disks:



Superimposing synchronism

Superimposing synchronism responds similar to superimposed positioning motions with regard to the basic motion on the axis (motion or synchronism).

The interconnection conditions for superimposing synchronism and basic synchronism are identical. Same as a basic synchronization object, you can logically link a superimposing synchronization object with several leading axes or cam disks.



Rules

The following rules apply to the interconnection:

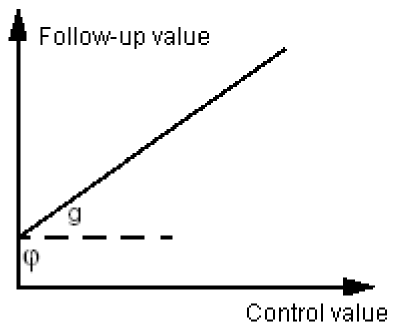
- Control values can be used at multiple instances A leading axis can output the master setpoint to several following axes Positioning axes, synchronization axes, virtual axes or external encoders can determine master setpoints.
- The synchronization object can be interconnected with several master setpoints and cam disks. Allowances must be made in the configuration of the synchronization object for all combinations required for operation.
- An axis may feature up to two synchronization objects (basic synchronism and superimposing synchronism.)

3.4.3 Gearing

Define the transformation ratio and gear ratio as a function of the ratio between two integers and an offset at the "MC_GearIn" and "MC_GearInSuperImposed" technology functions. You can modify the offset at later time using the "MC_Phasing" and "MC_PhasingSuperImposed" technology functions.

Response characteristic

The response characteristic of gearing can be expressed as a linear correlation between the master setpoint and the slave value.



Following value = g • master setpoint + φ
g = gear ratio (transmission ratio)
φ = offset

Direction

The gear ratio can be defined by a positive or negative value. Resultant response:

- **Positive gear ratio**:
  The leading and following axes move in the same direction.
- **Negative gear ratio**:
  The leading and following axes move in opposite direction.

Absolute or relative gearing

Absolute or relative gearing can be set at input parameter *Absolute*.

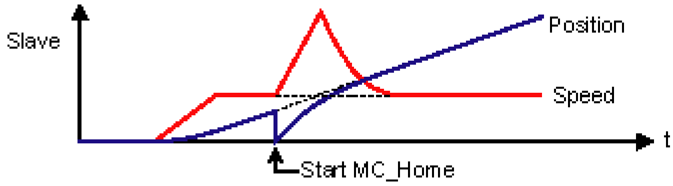
- **Absolute synchronism** establishes an absolute correlation between the leading and following axes.

  An offset between the leading and following axes is compensated for during synchronization based on the offset defined at the technology function.
- **Relative synchronism** establishes a relative correlation between the leading and following axes

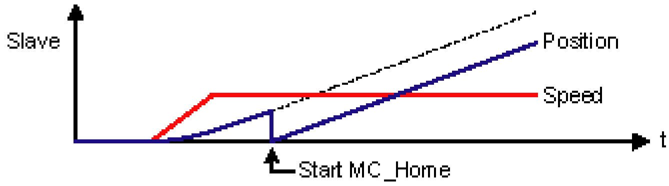
  The system does **not** compensate for any offset which may develop between the axes after the start of synchronization. The offset between the positions of the axes is set at random and is determined by the time of their synchronization.

Resetting axis positions during synchronous operation

Absolute coupling results in a compensating motion that eliminates the difference of position setpoints:



Relative coupling does not result in a compensating motion, meaning the offset changes:



You reset the axis position at the technology function "MC_Home" (*Mode = 3* or *4*).

3.4.4 Camming

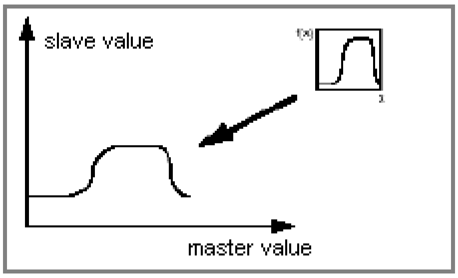
Camming is characterized by a **dynamic transmission ratio** between the leading and following axes, and by an offset. The transmission ratio is described by a **cam disk** (transfer function.)

Scaling and offset of the cam disk used is possible both at the leading axis and at the following axis. This enables individual adaptation of a cam disk in terms of its definition range and range of values.

Synchronization of the following axis to a position within the cam disk is also possible, as is a cam disk changeover at the end of a currently active cam disk.

Response characteristic

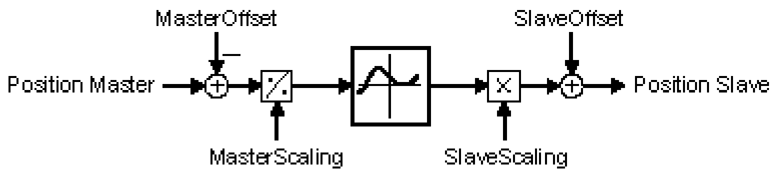
The response characteristics of camming are defined by a cam disk (y = f(x)).



Scaling and offset

The scaling and offset of the camming function can be set both at the leading axis and at the following axis using the camming technology function. The configured cam disk is not modified by the call of the "MC_CamIn" technology function.

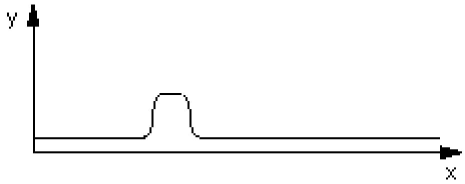
Transfer steps for camming



Non-cyclic or cyclic cam disk operation

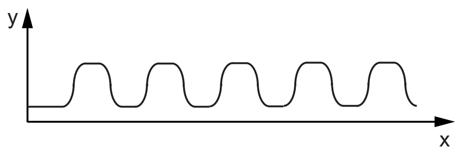
You can use input parameter *CyclicMode* of the camming technology function to set non-cyclic (*CyclicMode = 0*) or cyclic (*CyclicMode = 1*) camming mode.

- "Non-cyclic" mode means that the cam disk is only executed once. Camming is terminated when the end or start point of the cam is reached Example:

  (x = master setpoint; y = slave value)

- In cyclic cam disk mode, the leading axis are calculated as modulo function of the cam disk length As with gearing, the coupling can be terminated by a programmed cancellation command Example:

  (x = master setpoint; y = slave value)

Absolute and relative camming

Absolute and relative camming mode can be set at the *MasterAbsolute* and *SlaveAbsolute* input parameters of the "MC_CamIn" technology function.

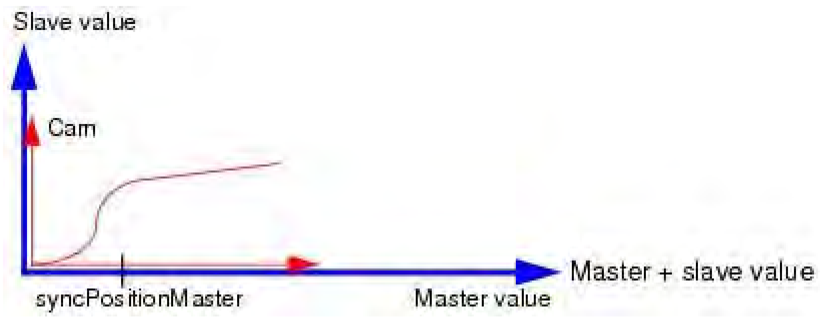
- If **absolute camming** is set, the master setpoints based on the cam disk definition range, and the slave values based on the range of values of the cam disk are interpreted as being absolute values.

  In absolute camming mode, the start and end values of modulo axes should be identical, because otherwise unwanted step jumps may occur.

- The cam disk is offset beginning at the start value of camming when operated in **relative camming** mode.

  The possible settings for absolute and relative synchronism are shown below.

Absolute synchronism with the leading axis and following axis

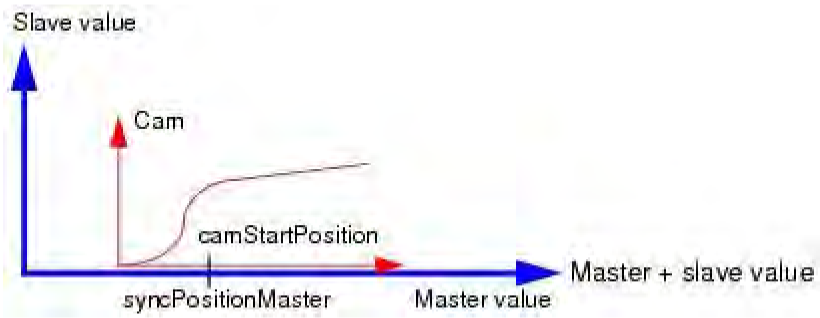


MasterAbsolute = 1; *SlaveAbsolute = 1*

In absolute synchronous operation, the effective cam disk precisely represents the configured cam disk. The cam disk is not shifted in the coordinate system "Slave value = f(master setpoint)."

During synchronization, the system approaches the cam disk point which is assigned to the actual master position. A synchronization motion is not initiated if the following axis is already positioned on a cam disk point at the start of camming.

Relative synchronism with the leading axis, and absolute with the following axis



MasterAbsolute = 0; *SlaveAbsolute = 1*

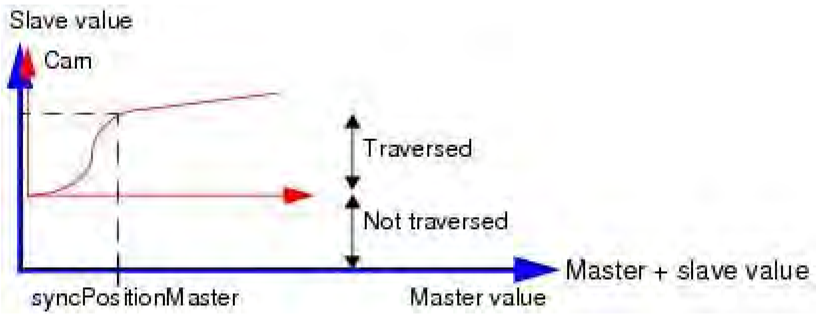
When operating with camming relative to the leading axis, the system evaluates *userdefault.cammingsettings.camstartpositionmaster* the system variable and sets this value as start position for camming.

You obtain the effective cam disk by shifting the configured cam disk along the abscissa in the "following value = f(master setpoint)" coordinate system, so that *userdefault.cammingsettings.camstartpositionmaster* coincides with the actual position of the following axis at the start of camming. As synchronism is absolute to the following axis, the axis is not shifted along the ordinate of the coordinate system.

During synchronization the following axis approaches the following value on the cam disk which is assigned to *userdefault.cammingsettings.camstartpositionmaster*.

A synchronization motion is not initiated if the following axis has already reached this position.

Absolute synchronism with the leading axis, and relative with the following axis



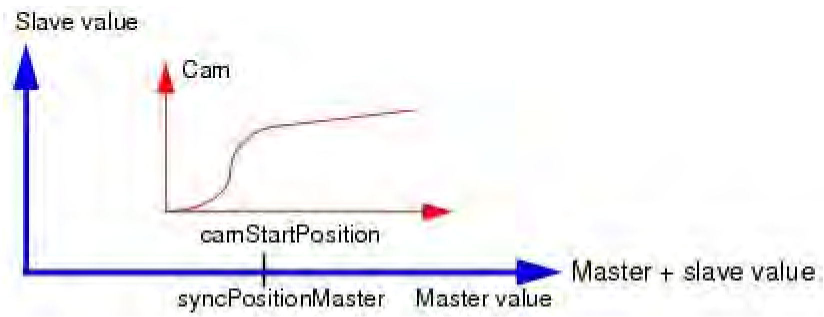
MasterAbsolute = 1; *SlaveAbsolute = 0*

You obtain the effective cam disk by shifting the configured cam disk along the ordinate in the "slave value = f(master setpoint)" coordinate, so that its start position coincides with the actual following axis position. As synchronism is absolute to the leading axis, the axis will not be shifted along the abscissa of the coordinate system.

During synchronization, the following axis moves by a distance defined by the difference between the slave value at the actual position of the leading axis set in the cam disk definition and the start position of the cam disk. The following axis is positioned on the point of the effective cam disk assigned to the actual position of the leading axis.

If the leading axis has reached the start position, or a certain cam disk point of which the slave value corresponds with that of the start position, the axis will not perform a synchronizing motion.

Relative synchronism with the leading axis and following axis



MasterAbsolute = 0; *SlaveAbsolute = 0*

When operating with camming relative to the master, the system evaluates system variable *userdefault.cammingsettings.camstartpositionmaster* and sets this value as start position for camming.

You obtain the effective cam disk by shifting the configured cam disk along the abscissa in the "slave value=f(master value)" coordinate system in order to shift *userdefault.cammingsettings.camstartpositionmaster* and the actual leading axis position. The start position of the cam disk must coincide with the start position of the active following axis.

During synchronization the following axis approaches the following value on the cam disk which is assigned to the *userdefault.cammingsettings.camstartpositionmaster* .

During synchronization, the following axis travels by the distance defined by the difference between the following values at *userdefault.cammingsettings.camstartpositionmaster* and at the cam disk start position. The following axis is positioned on the point of the effective cam disk assigned to the actual position of the leading axis.

A synchronization motion is not initiated if the following axis has already reached this position.

Correction of synchronous motions

Synchronous motions can be corrected by switching over the scaling and offset of the master setpoint and slave value.

Other options include:

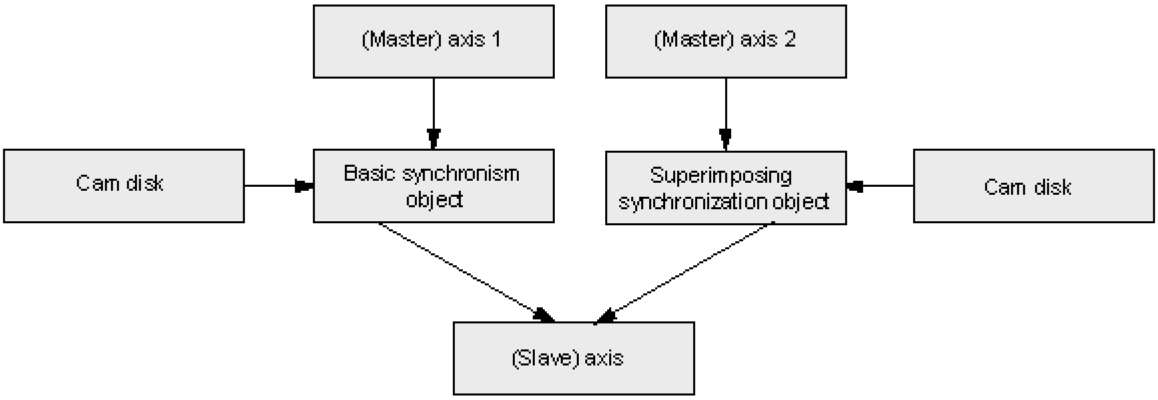
- Offset and scaling directly on the cam disk
- On-the-fly setting of the reference point on the master value source and the slave axis
- Superimposed positioning

Note

Any point of the cam disk can be set as the starting point for camming. This can be the starting point, the end point, or any point within the cam disk.

3.4.5 Superimposing synchronism

You can configure superimposing synchronism for camming and gearing. This is done by configuring an additional superimposing synchronization object at the synchronization axis. Cross-referencing synchronization objects is impossible.



Configuring axes with superimposed synchronism

A superimposing synchronization object is configured similar to a synchronization object.

Superimposed motion

Superimposing synchronism responds similar to superimposed positioning motions with regard to the basic motion on the axis (motion or synchronism).

Several simultaneous superimposing movements are not possible at the axis:

- Superimposing positioning motion
- Superimposing synchronism

Superimposing synchronism can be active without concurrently active basic motion or basic synchronism.

For further information on superimposing motions at an axis, refer to the corresponding descriptions of the technological functions.

Coordinates

The basic synchronization object is referenced to the base coordinates when the axis operates in absolute synchronism and the slave position is defined.

The superimposing synchronization object is referenced to the superimposing coordinates when the axis operates in absolute synchronism and the slave position is defined.

Synchronism with reference to base coordinates and superimposing synchronism with reference to superimposing coordinates do not influence each other.

The cumulative coordinate is derived from the base and superimposing coordinates.

Absolute and relative synchronism

You can program and execute relative or absolute superimposing synchronism in the synchronization function (with absolute reference to the superimposing coordinates!)

Monitoring

The output values of a synchronization object (and thus the motion element of superimposing axis synchronism) can be read from system variable *currentslavedata* at the synchronization object.

Synchronization monitoring / status

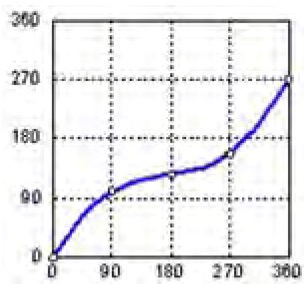
The variables and monitoring functions at the axis refer to the resultant synchronism.

Error messages (synchronization error at the synchronization axis) are reported to all interconnected synchronization objects.

3.4.6 Relative / absolute synchronism

The next examples show you the differences between relative and absolute synchronism.

In camming mode, the dependency of the following axis position (XF) on the position of the leading axis (XL) is described by a cam. The next examples always use the basic cam shown below:



The examples refer to "immediate synchronization." The actual positions of the participating axes are relevant to this use. For position-dependent synchronization, evaluate the "Synchronization position" instead of the "actual position."

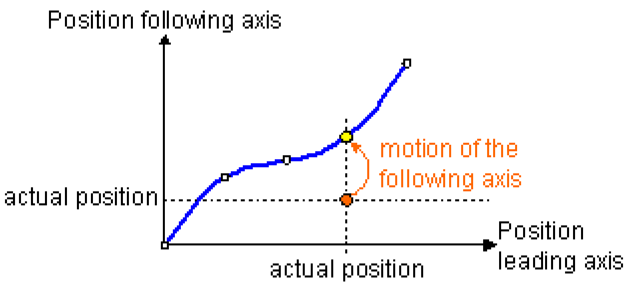
Absolute synchronism

Configure "absolute synchronism" at the "MC_CamIn" technology function using the following input parameters:

- *MasterAbsolute = TRUE*
- *SlaveAbsolute = TRUE*

Absolute synchronism is the simplest use case, as it uses the basic cam. The assignment is thus defined permanently by the cam. The following axis position is derived from a certain leading axis position precisely as you can read it from the cam.

During activation of synchronous operation, the following axis is moved from its current position to the position derived from the cam.



Relative synchronism

A cam with an offset to the basic cam is applied in relative synchronism. We distinguish the following variants:

- The cam is shifted along the abscissa (*MasterAbsolute = FALSE; SlaveAbsolute = TRUE*)
- The cam is shifted along the ordinate (*MasterAbsolute = TRUE; SlaveAbsolute = FALSE*)
- The cam is shifted in any direction (*MasterAbsolute = FALSE; SlaveAbsolute = FALSE*)

Example 1 - Cam is shifted along the abscissa

Settings:

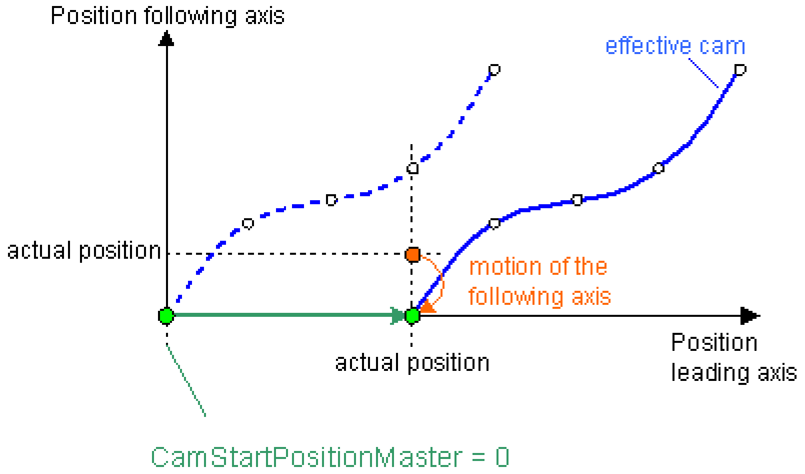
- *MasterAbsolute = FALSE*
- *SlaveAbsolute = TRUE*

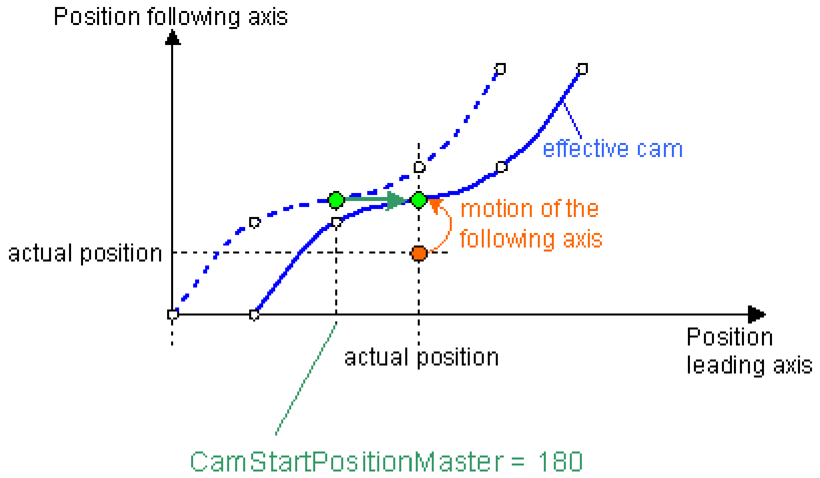
With these settings you produce an effective cam by shifting the basic cam along the abscissa.

The amount by which the cam is shifted along the abscissa depends on the position the leading axis had at the time camming was activated, and on the *userdefault.cammingsettings.camstartpositionmaster* system variable of the synchronization object. The cam is shifted so that the point of the basic cam on the leading axis coordinate *userdefault.cammingsettings.camstartpositionmaster* coincides with the actual leading axis position.

The first diagram applies to *userdefault.cammingsettings.camstartpositionmaster = 0,* and the second applies to *userdefault.cammingsettings.camstartpositionmaster = 180.*

The following axis moves from its current position to the position of the shifted cam.


Position following axis
effective cam
actual position
motion of the
following axis
Position
leading axis
actual position
CamStartPositionMaster = 0

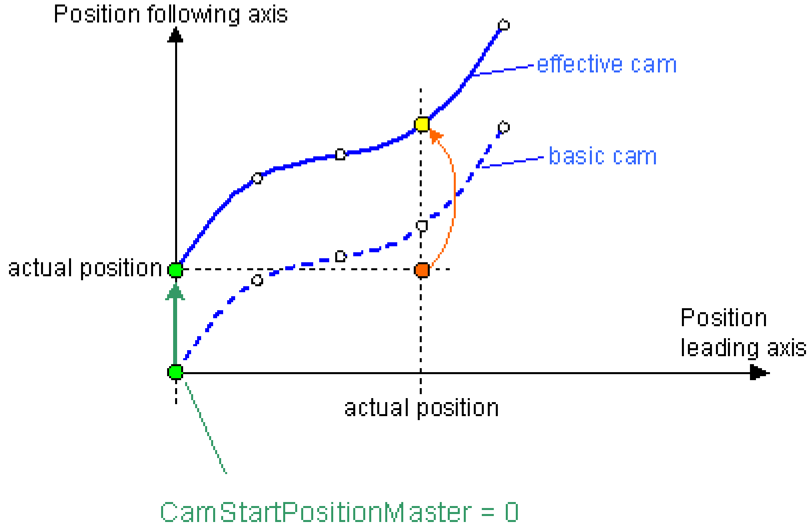

Position following axis
effective cam
motion of the
following axis
actual position
Position
leading axis
actual position
CamStartPositionMaster = 180

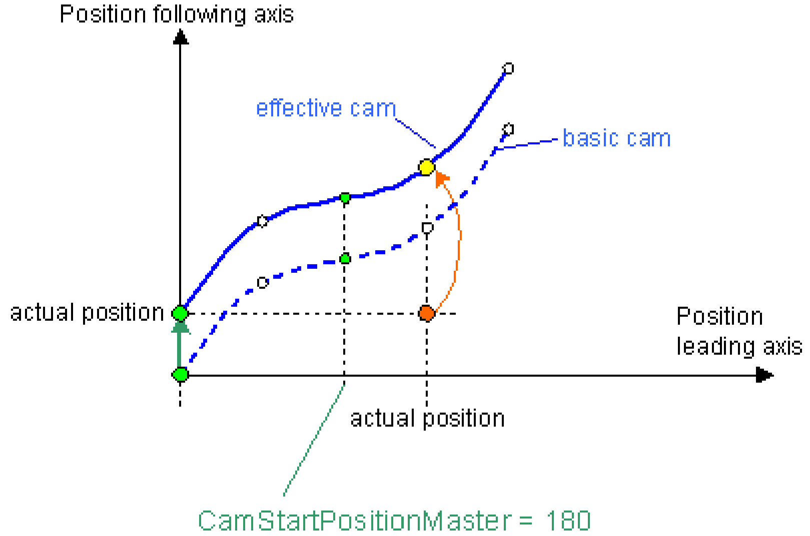
Example 2 - Cam shift along the ordinate

Settings:

- *MasterAbsolute = TRUE*
- *SlaveAbsolute = FALSE*

These settings produce an effective cam by shifting the basic cam along the ordinate.





The cam start position (smallest X value) is relevant instead of system variable *userdefault.cammingsettings.camstartpositionmaster* in this context. The cam is shifted so that its start point is shifted to the actual position of the following axis.

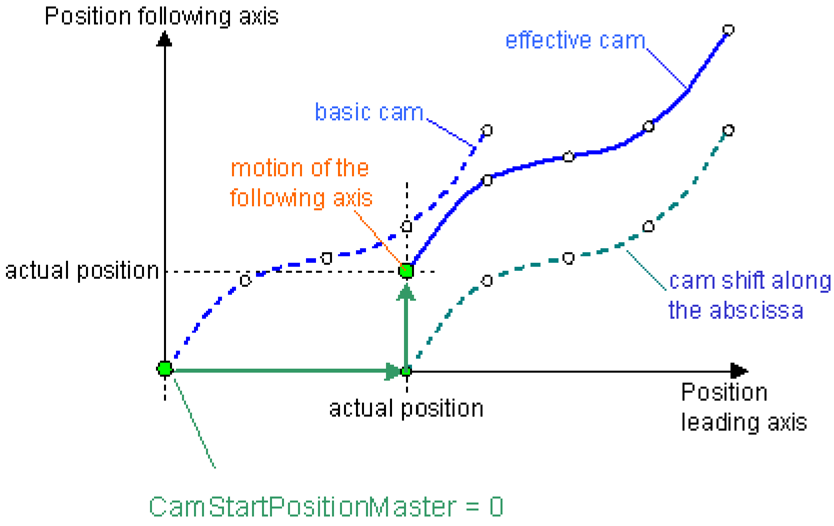
Example 3 - The cam is shifted along the ordinate and abscissa

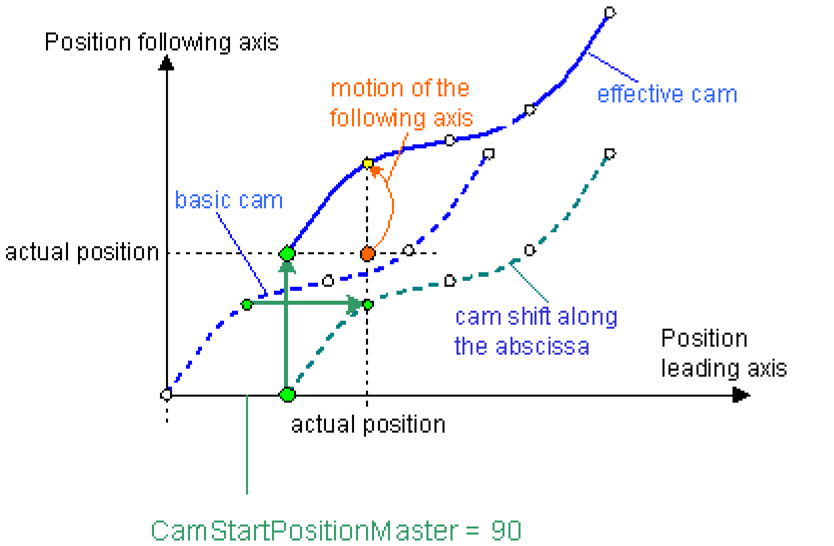
Settings:

- *MasterAbsolute = FALSE*
- *SlaveAbsolute = FALSE*

These settings produce an effective cam by superimposing both of the previously described functions.

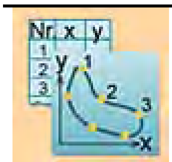
To illustrate the effect the basic cam can be shifted along the abscissa in an interim step. Subsequently the cam start point along the ordinate to the actual value of the following axis is shifted.





The following axis usually has to perform a compensating motion at the start of synchronism in order to reach the cam point which is assigned to the current leading axis position. The drawings show this as "motion of the following axis."

3.5 "Cam disk" technology object



Use the "Cam disk" technology object to implement complex motion sequences. A "cam disk" defines the dependency of a following axis position on the leading axis position.

A cam disk can also be used to define a valve profile for a hydraulic axis.

You configure the "Cam disk" technology object in S7T Config.

The cam profile can be defined either in S7T Config, or in the user program.

Definition of the cam in S7T Config

Use the CamEdit or Scout CamTool programs to define a cam based on interpolation points or polynomials.

Cam definition in the user program

1. Create the cam disk in S7T Config.
2. Delete and reset the cam disk using the "MC_CamClear" technology function.
3. Use technology function "MC_CamSectorAdd" to define the cam by setting interpolation points or polynomials.
4. Interpolate the cam disk using the "MC_CamInterpolate" technology function before you prepare the cam for operation.

Technology functions

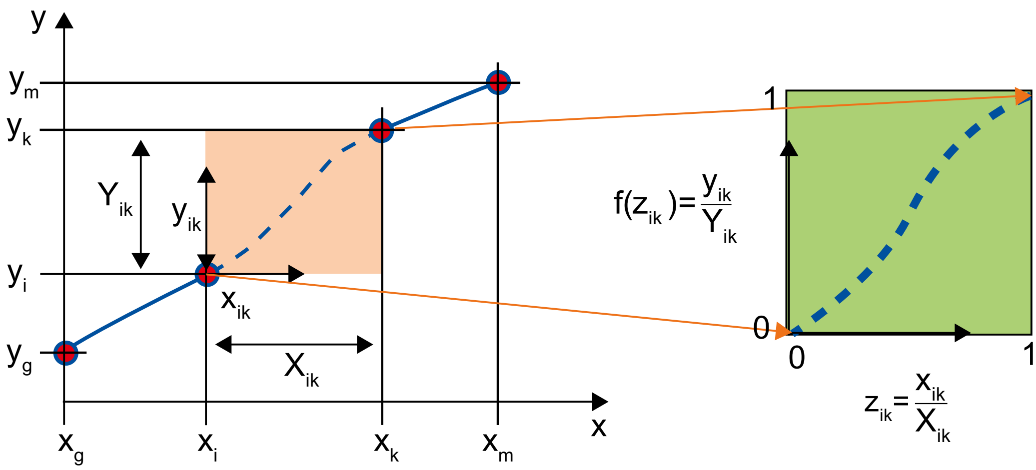
Technology functions supported by the technology object:

MC_Reset	MC_ReadSysParameter	MC_WriteParameter
MC_CamClear	MC_CamSectorAdd	MC_CamInterpolate
MC_GetCamPoint		

3.5.1 Scaling

When a cam disk is defined in S7T Config based on segments, the various cam segments can be provided in a standard form scaled to factor 1, meaning that the range of values and the defined range corresponds with the completed interval [0,1].

Mapping of a real cam disk segment to the scaled range:



The segments can also be entered in the real range.

Advantages of scaling

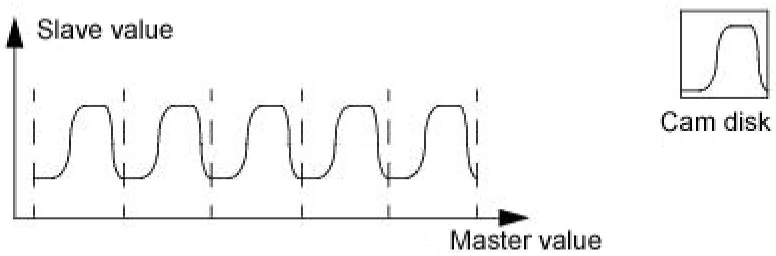
- Definite description of the motion for similar tasks
- Independence of real units and ranges of the leading and following axes.

3.5.2 Cam disk applications

A cam disk can be configured for operation in non-cyclic (single operation), cyclic relative (continuous) or cyclic absolute (return to interpolation point) mode.

Example of a cam disk operating in absolute cyclic mode

(x = master setpoint; y = slave value)

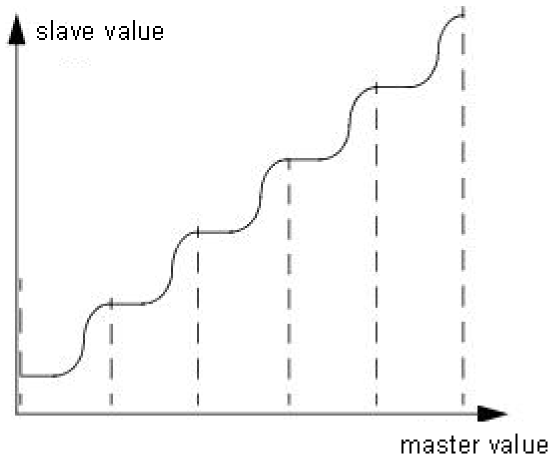


If the cam disk is operated in cyclic mode and absolute to the following axis, all successive start and end point values of the cam disk should match.

Any mismatch will cause a step response at the cam disk transition, and thus trigger synchronization monitoring.

Example of cam disk operation in relative cyclic mode

(x = master setpoint; y = slave value)

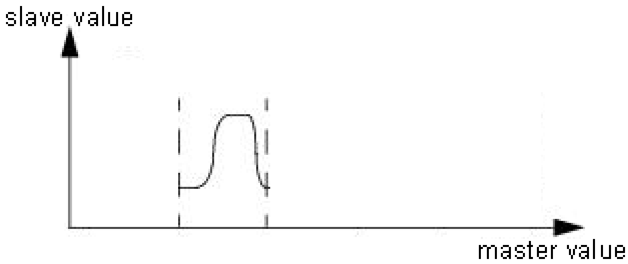


If the cam disk is operated in cyclic mode and relative to the following axis, the successive start and end point values of the cam disk do not have to match.

The system automatically sets the start point value at the end point at the cam disk transition. This results in no jumps occurring in the cam disk transition.

Example of cam disk operation in non-cyclic mode

(x = master setpoint; y = slave value)



Non-cyclic mode means that the cam disk is only executed once. Camming is terminated when the end or start point of the cam is reached

3.5.3 Interpolation of cams

Interpolation of a cam disk is a basic requirement for using it in synchronous operation or as valve profile.

Interpolation of the cam disk:

- In S7T Config - alongside with the download of the cam disk to the target system.
- By calling the "MC_CamInterpolate" technology function the user program.

During interpolation the system performs the following checks and corrections:

- The continuity in the definition range (range of leading axis values) and in the range of values (of the following axis) is checked. This check prevents redundant assignment of values to a definition value.
- Gaps between the interpolation points and segments are padded. These areas are padded according to the interpolation mode
- Missing edge areas are padded These areas are padded according to the interpolation mode and cam disk type.

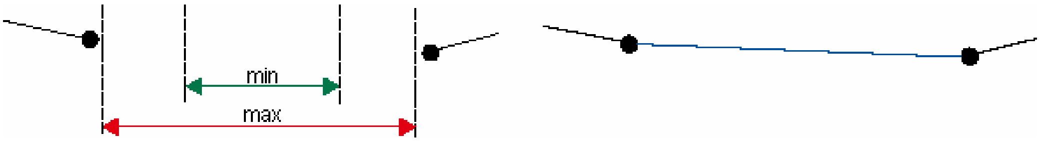
When interpolation is completed, the definition range of the cam disk is assigned an unambiguous range of values.

Continuity check

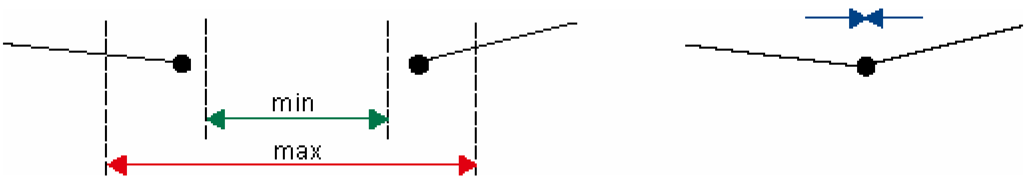
The system checks the continuity of a cam disk in the definition and value ranges, and corrects any discontinuity found.

During this process, the points of discontinuity are examined separately for the definition range / range of values, and are rated for one of the following corrective actions:

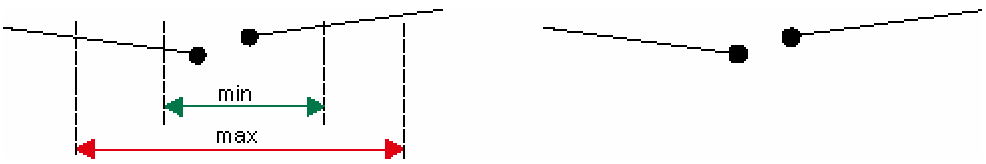
- If the absolute value of the gap between segments exceeds a maximum, a correction is made by interpolation between the two segments. This inserts a new segment according to the defined interpolation mode. The cam disk type is included in this action if a segment is inserted in the edge area.



- If the absolute value of the gap between segments is greater than the minimum and less than the maximum, a correction is made by joining the segment end points. The mean value of the spacing of the function is used for the correction. The shape of the segments is affected as a result.



- If the absolute value of the spacing between segments or interpolation points is less than the minimum value, a correction is not made. The discontinuity point is retained. When this discontinuity point is accessed, the right-hand edge point is output.



The maximum value can be set in the "Interpolation" tab when you interpolate using CamEdit. The "minimum value" corresponds with the entry "ignore gaps", and the "maximum" with the entry "join points".

In the case of an interpolation with the "MC_CamInterpolate" technology function, the minimum and maximum values amount to *1e-004.* each. Segment end points can therefore only be joined by interpolation using CamEdit.

The point of discontinuity is corrected according to the rating of the definition range /range of values.

Condition	Result
Deviation < minimum	Discontinuity retained
Minimum < deviation < maximum	Joins segment points
Deviation > maximum	Interpolation (new segment)

The correction is controlled for the definition range and the range of values each by the definition of the minimum and the maximum shape deviation.

- Function continuity can be achieved by linear interpolation
- Continuity in the derivatives can be achieved by spline interpolation

If the continuity condition can not be adhered to due to the selected interpolation method or the programmed geometry, a message is output to that effect.

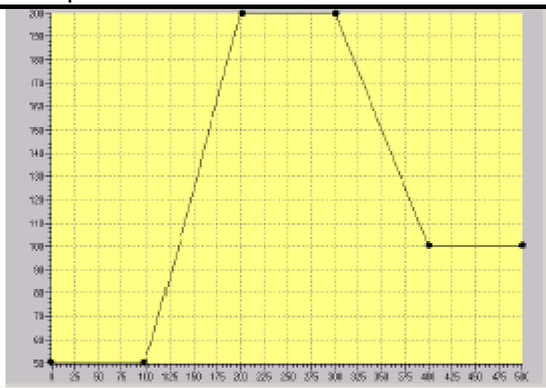
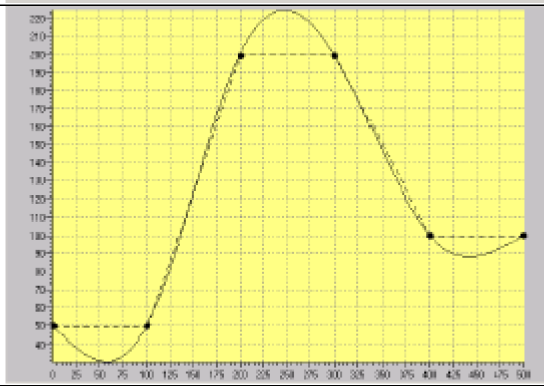
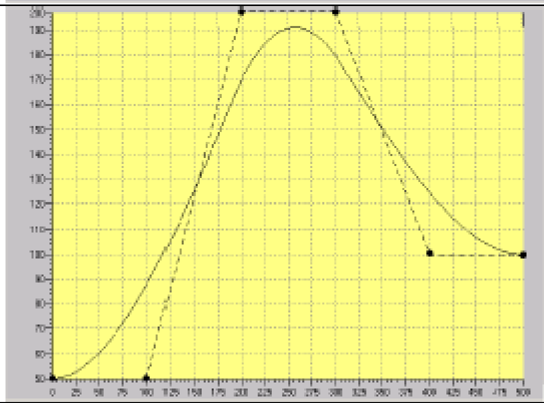
If an interpolation edge point lies within the programmed geometry, all geometry elements up to the edge points are rejected. If an interpolation edge point lies outside the programmed geometry, an end point is extrapolated according to the interpolation method used and taking into account the geometry characteristic.

Note

After interpolation, new polynomials or interpolation points can only be inserted after resetting the cam with ("MC_CamClear".)

3.5.4 Interpolation types

The interpolation type determines the rules of the interpolation of gaps. Interpolation types which can be used:

Interpolation type	Description	Example
Linear	Linear interpolation: Gaps in the cam profile are closed by a "linear" operation, meaning by means of a straight line.	
Cubic splines	Interpolation with cubic splines: The interpolated cam profile runs through the cam interpolation points or through the defined cam segments. The range of values at the cam disk may be (Y-min to Y-max) higher than the value at the start of interpolation when interpolation is completed.	
Bezier splines	Interpolation with Bezier splines: The interpolated cam profile runs along the cam interpolation points or cam segments. Interpolation does not change the range of cam disk values (Y-min to Y-max.)	

You can select the interpolation type using the "MC_CamInterpolate" technology function and CamEdit:

Interpolation type	"MC_CamInterpolate"	CamEdit - "Interpolation" tab
Linear	*Mode = 0*	"linear"
Cubic splines	*Mode = 1*	"cubic splines"
Bezier splines	*Mode = 2*	"Bezier splines"

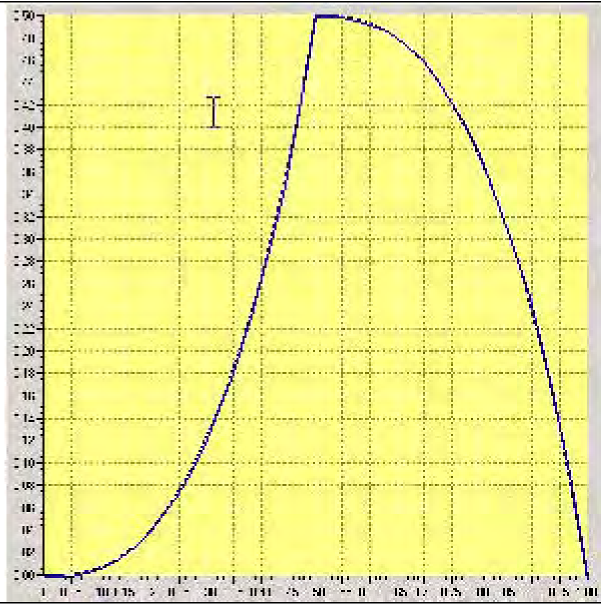
Cam type

The cam disk type describes its application and the resultant interpolation of the points of discontinuity at the edges of the cam disk:

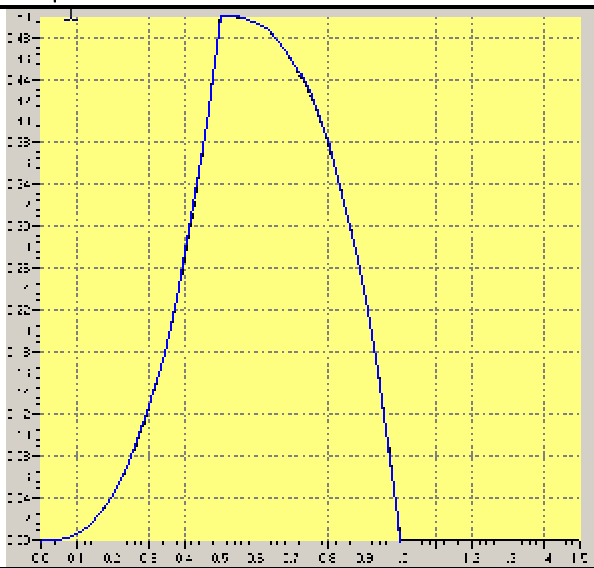
- not cyclic (discontinuity at the edge points)
  The technology CPU uses the cam disk as defined, including any discontinuity at the edges. This also applies when the cam disk is operated in cyclic mode. However, the acceleration limits and inertia of the mechanical system and drive play a decisive role.
- cyclic absolute (constant position and speed at the edge points)
  The cam disk is interpolated so that its position and velocity remain constant at the edges when operating in cyclic mode. The following axis values and velocities (cam gradients) are identical at the cam edge points.
- cyclic relative (constant speed at the edge points)
  The technology CPU converts the cam disk so that its velocity remains constant at the edges in cyclic mode. The values and velocities (cam gradients) at the cam edges are identical. The following axis values are not be calibrated.

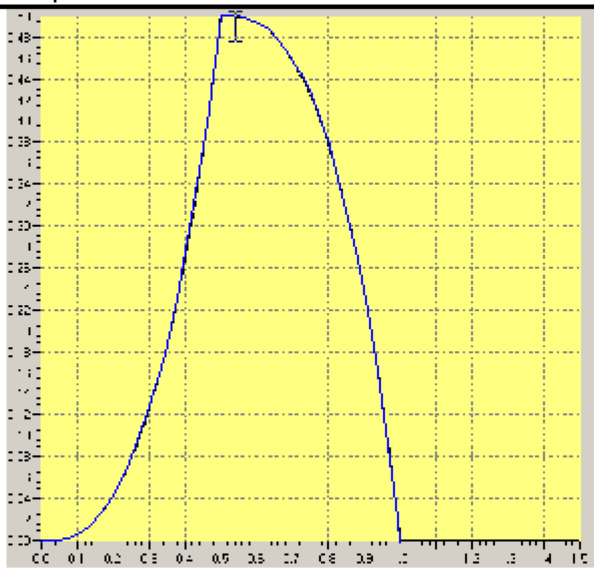
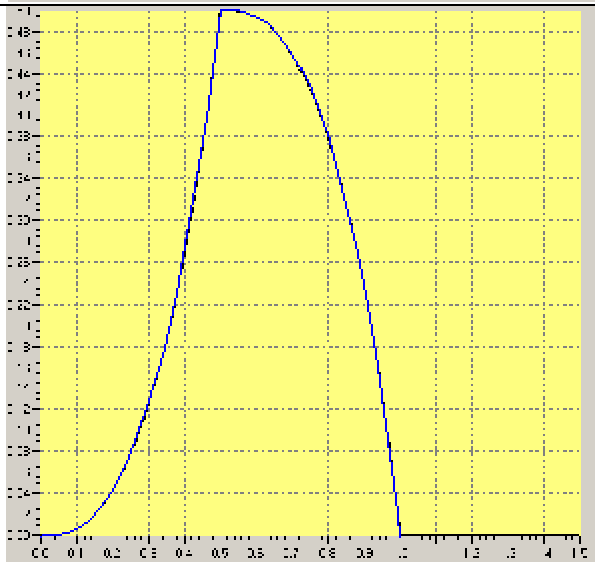
In "cyclic absolute" or "cyclic relative" mode, the Technology CPU can only calculate the cam disk if a sufficient distance lies between the first defined position of the cam disk and the end of the definition range. The same applies to the first defined position and the start of the definition range.

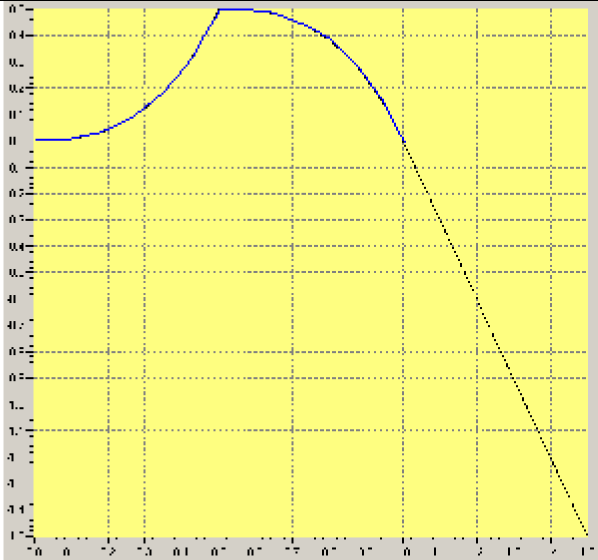
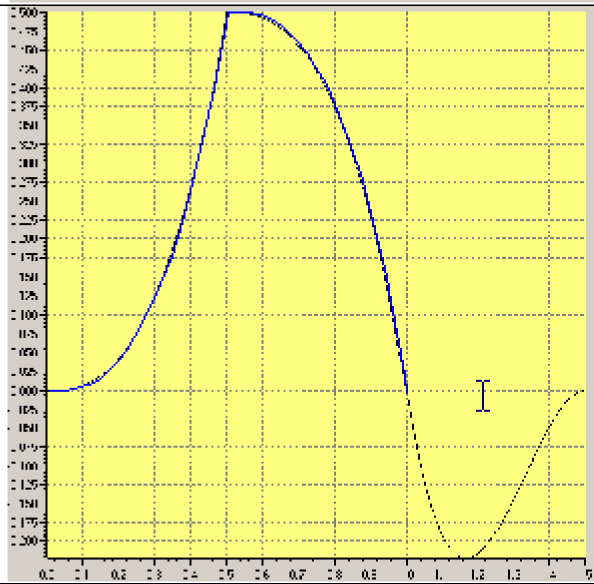
Example of the various interpolation modes based on a basic cam profile:

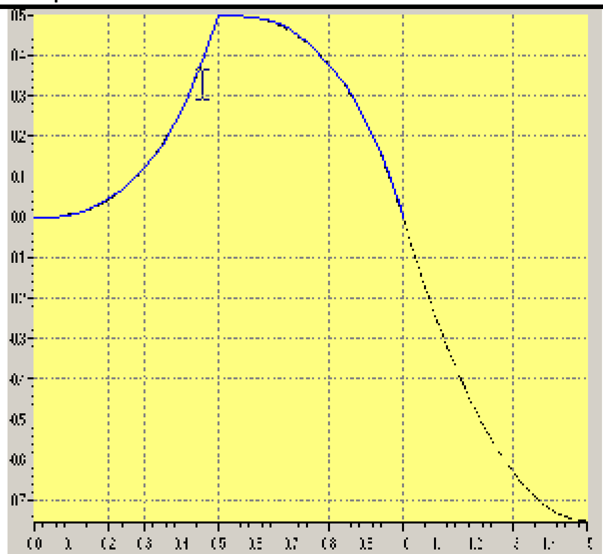
Basic polynomial	Polynom1: X-min = *0.0*, X-max = *0.5* Y-min = *0.0*, Y-max = *0.5* Polynom2: X-min=*0.5*, X-max=*1.0* Y-min=*0.0*, Y-max=*0.5*	

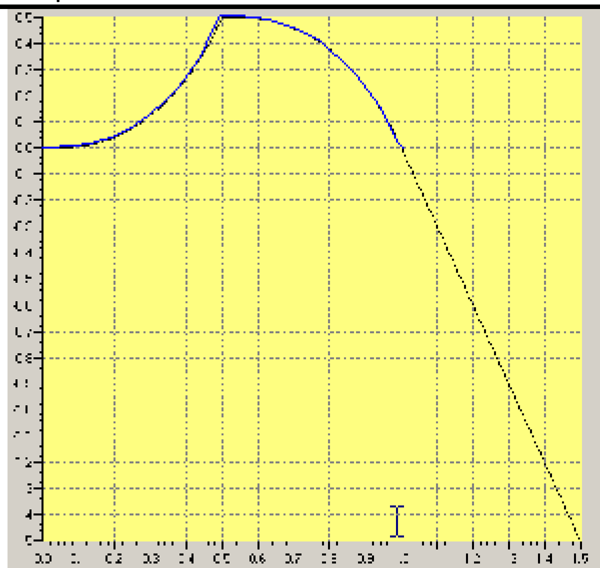
The basic cam consists of two polynomials. In order to be able to interpolate the cam according to its given interpolation mode (non-cyclic, absolute and relative cyclic mode), the end of the definition range (range of leading axis values) is extended from *1.00* to *1.50* in the examples shown in the next section. Within this extended range, the cam is interpolated according to the interpolation mode set.

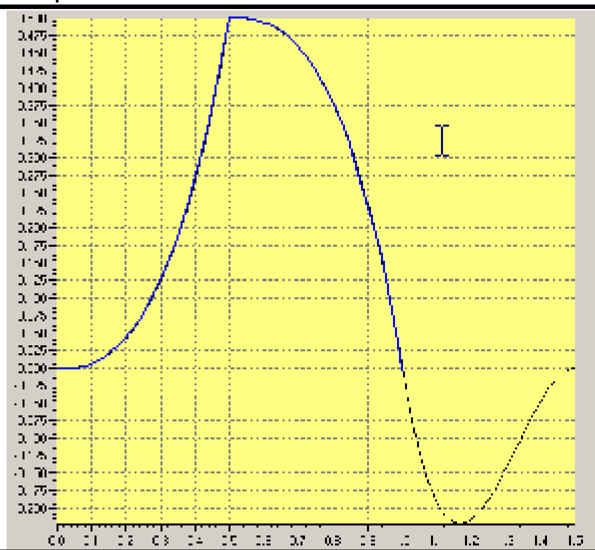
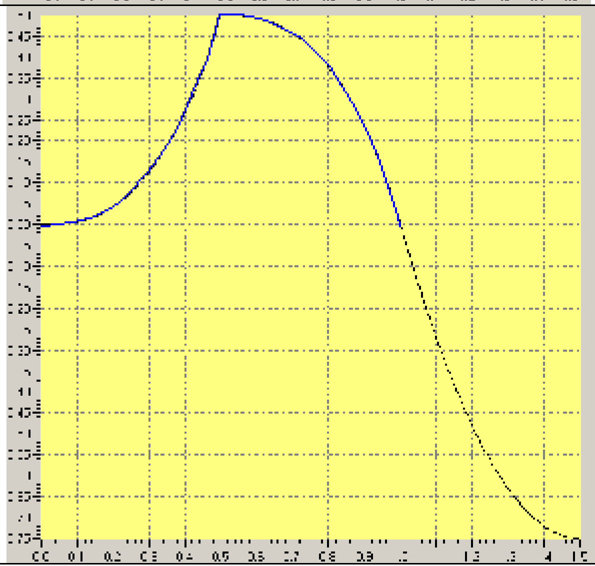
Interpolation type	Description	Interpolation result
Linear	Interpolation: linear, non-cyclic Input parameter "MC_CamInterpolate": *Mode = 0* *CamMode = 2* *StartPoint = 0.0* *EndPoint = 1.5* Result: The basic cam remains unchanged. Its end point is extended by "linear" interpolation along the abscissa.	

Interpolation type	Description	Interpolation result
	Interpolation: linear, cyclic absolute Input parameter "MC_CamInterpolate": *Mode = 0* *CamMode = 1* *StartPoint = 0.0* *EndPoint = 1.5* Result: The basic cam remains unchanged; its end point is extended relative to its start point by "linear" interpolation	
	Interpolation: linear, cyclic relative Input parameter "MC_CamInterpolate": *Mode = 0* *CamMode = 0* *StartPoint = 0.0* *EndPoint = 1.5* Result: The basic cam remains unchanged; it is extended along a linear coordinate, based on its initial gradient.	

Interpolation type	Description	Interpolation result
Cubic splines	Interpolation: cubic splines, non-cyclic Input parameter "MC_CamInterpolate": *Mode = 1* *CamMode = 2* *StartPoint = 0.0* *EndPoint = 1.5* Result: The basic cam remains unchanged. The cam is extended within the extended definition range according to the interpolation mode	
	Interpolation: cubic splines, cyclic absolute Input parameter "MC_CamInterpolate": *Mode = 1CamMode = 1StartPoint = 0.0EndPoint = 1.5* Result: The basic cam remains unchanged. The cam is extended within the extended definition range according to the interpolation mode; the interpolated cam operates at constant position and velocity	

Interpolation type	Description	Interpolation result
	Interpolation: cubic splines, cyclic relative Input parameter "MC_CamInterpolate": *Mode = 1CamMode = 0Sta* *rtPoint = 0.0EndPoint = 1.5* Result: The basic cam remains unchanged. The cam is extended within the definition range according to the interpolation mode; the interpolated cam operates at constant velocity	

Interpolation type	Description	Interpolation result
Bezier splines	Interpolation: Bezier splines, non-cyclic Input parameter "MC_CamInterpolate": *Mode = 2CamMode =* 2*Sta* *rtPoint = 0.0EndPoint = 1.5* Result: The basic cam remains unchanged. The cam is extended within the extended definition range according to the interpolation mode	

Interpolation type	Description	Interpolation result
	Interpolation: Bezier splines, cyclic absolute Input parameter "MC_CamInterpolate": *Mode = 2CamMode = 1StartPoint = 0.0EndPoint = 1.5* Result: The basic cam remains unchanged. The cam is extended within the extended definition range according to the interpolation mode; the interpolated cam operates at constant position and velocity	
	Interpolation: Bezier splines, cyclic relative Input parameter "MC_CamInterpolate": *Mode = 2CamMode = 0StartPoint = 0.0EndPoint = 1.5* Result: The basic cam remains unchanged. The cam is extended within the definition range according to the interpolation mode; the interpolated cam operates at constant velocity	

Conditions:

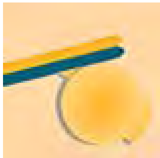
- Any different values defined at the "MC_CamInterpolate" technology function for *StartPoint* and *EndPoint* determine the range of the master values. Any interpolation points or segments set outside the defined range will be truncated. The limits of the interpolation cams are not maintained if the value *0.0* is set at *StartPoint* and *EndPoint*.
- In order to to interpolate the cam disk if *CamMode = 0* or *CamMode = 1* , you should always set values at *StartPoint* or *EndPoint* that cover a range greater than the range of the interpolated cam. Interpolation according to default parameters is not possible if this condition is not met. The technology function "MC_CamInterpolate" outputs the warning *0027* . S7T Config outputs an alarm if the cam was interpolated in CamEdit.

Example:

The "MC_CamSectorAdd" technology function is used to define a polynomial in the definition range from *0* to *0.9*. In order to be able to interpolate the cam disk in the edge area according to given interpolation rules, you should at least define the range *0* to *1.0* at "MC_CamInterpolate."

- "Linear interpolation" (*Mode = 0*) does not make any sense if the cam disk is to be interpolated in "cyclic relative" (*CamMode = 0*) or "cyclic absolute" (*CamMode = 1*) mode. Instead, select "cubic splines" (*Mode = 1*) or "Bezier splines" (*Mode = 2*.)
- If you define several segments by polynomials at technology function "MC_CamSectorAdd" you should always make sure that the start point of a polynomial matches the end point of the partner polynomial, or that a sufficient definition gap exists between the various polynomials Interpolation will fail otherwise The cam disk may show any step response after its interpolation Any input of partially overlapping segments prevents compliance with interpolation rules and warning *0027* will be output at the "MC_CamInterpolate" technology function. The cam disk may produce a step at this position
- If the cam disk is to be interpolated in "non-cyclic" mode (*CamMode = 2)* the definition range of the "MC_CamSectorAdd" (X-min to X-max) should correspond with the definition range of the "MC_CamInterpolate" technology function (*StartPoint* to *EndPoint*.)
- Rule for cyclic relative interpolation:
  The velocity remains constant at the cam transition
  V(X-min) = V(X-max)
- Rule for cyclic absolute interpolation:
  The velocity and position remain constant the cam transition
  V(X-min) = V(X-max),
  Y(X-min) = Y(X-max)

3.6 "Output cam" technology object



Use the "Output cam" technology object to generate control signals based on the axis position. You can evaluate the control signals in the user program, or set these at digital outputs. I/O supported:

- Digital outputs of integrated technology
- Digital outputs of a standard slave on DP(DRIVE)
- Digital outputs of SINAMICS Terminal Modules TM15 or TM17

The Technology CPU distinguishes between the output cams listed below:

- Position-based cam (Page 83)

  Position-based cams are activated between the starting position and the end position. The position-based cam is deactivated outside this range.
- Switching output cam (Page 85)

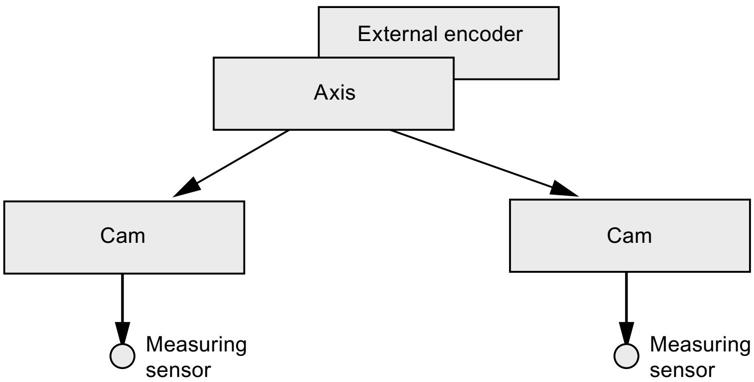
  Switching cams are set when the starting position is reached, and must be reset by the user program.
- Time-based cam (Page 86)

  Time-based cams are set for a defined period when the starting position is reached.

Interconnections

The "Output cam" technology object can be interconnected with the technology objects listed below:

- Positioning axes
- Synchronization axes
- External encoders



Reference to the actual or setpoint position

The switching positions of output cams can be referenced to the setpoint or actual value (setpoint output cam, actual value output cam.)

In order to enhance switching accuracy, you can compute the output cams within the position controller cycle, or use the integrated outputs of the Technology CPU for high-speed output cam operation.

Technology functions

Technology functions supported by the technology object:

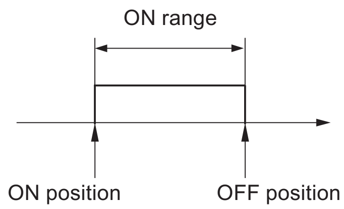
MC_Reset	MC_CamSwitch	MC_CamSwitchTime
MC_ReadSysParameter	MC_WriteParameter	

3.6.1 Position-based cam

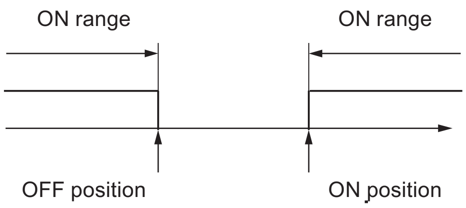
Cam activation range

The cam activation range of position-based cams is defined by the distance between the starting and end positions in **positive** count direction.

Starting position smaller than the end position:



Starting position greater than the end position:



The cam output is inverted when the starting position is greater that the end position

Switching characteristics

The switching characteristic of position-based cams is determined by the starting and end positions.

At the **"Output cam"** technology object an "effective direction" can be defined additionally at the technology object.

Switching action based on	Active position-based cam	Inactive position-based cam
Starting position, end position	• If the output cam was enabled with "MC_CamSwitch", or the cam track with "MC_CamTrack" • If the position is within the cam activation range • if the position value is shifted into the output cam activation range	• If the axis position is outside the starting or end area • when the position value is shifted out of the cam activation range *
Effective direction (not active at the cam track)	• when the position lies within the cam activation range and the effective direction = motion direction	• If the axis position is outside the starting or end area • if the motion direction does not match the configured effective direction • when the position value is shifted out of the cam activation range *
Hysteresis	when the cam moves out of the hysteresis range	when the cam moves out of the hysteresis range

* The position value of the interconnected object may change abruptly, for example, as a result of its referencing, or due to the shift of its coordinate system.

Inverted output

The inverted output function interchanges the cam activation range and the cam deactivation range.

3.6.2 Switching output cam

Switching characteristics

The switching action of switching output cams is determined by the starting position and by the direction of movement.

Action:	Condition:
Switching output cam activation	• The output cam is enabled by calling the "MC_CamSwitch" technology function (positive edge at input parameter *Execute*) • The direction of motion corresponds with the effective direction (input parameter *Direction*) • The starting position (input parameter *OnPosition*) is passed
Switching output cam deactivation	The output cam is deactivated by calling the "MC_CamSwitch" technology function (input parameter *Mode = 1)*

Note

The switching output cam is not activated if the cam starting position is not passed, for example, as a result of actual value settings.

Inverted switching action

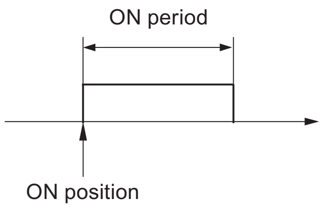
The output cam is **activated** if it is disabled by the "MC_CamSwitch" technology function.

Note

The output cam is **deactivated** at the cam starting position if the motion direction = effective direction or if the output cam was activated by the "MC_CamSwitch" technology function.

3.6.3 Time-based cam

Switching characteristics



The switching action of time-based cams is determined by the starting position and the cam activation time.

At the **"Output cam"** technology object an effective direction can be defined additionally at the technology object.

Switching action based on	Time-based cam is activated	Time-based cam is deactivated
Starting position	At the starting position	-
Cam activation time	-	on expiration of the programmed time
Effective direction (not active at the cam track)	At the starting position, if the motion direction = effective direction	on expiration of the programmed time

Note

Time-based cams can not be retriggered.

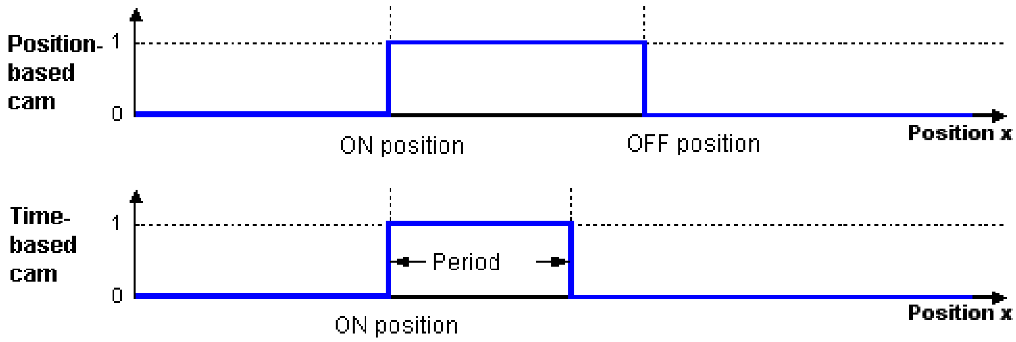
Inverted output

The time-based cam is activated for inverted output, and is deactivated at the cam deactivation position for the duration of a specified time.

3.6.4 Effective direction and behavior

Reaction

The diagram below shows the ON and OFF switching action of output cams, without hysteresis, derivative time, and without cam deactivation time.



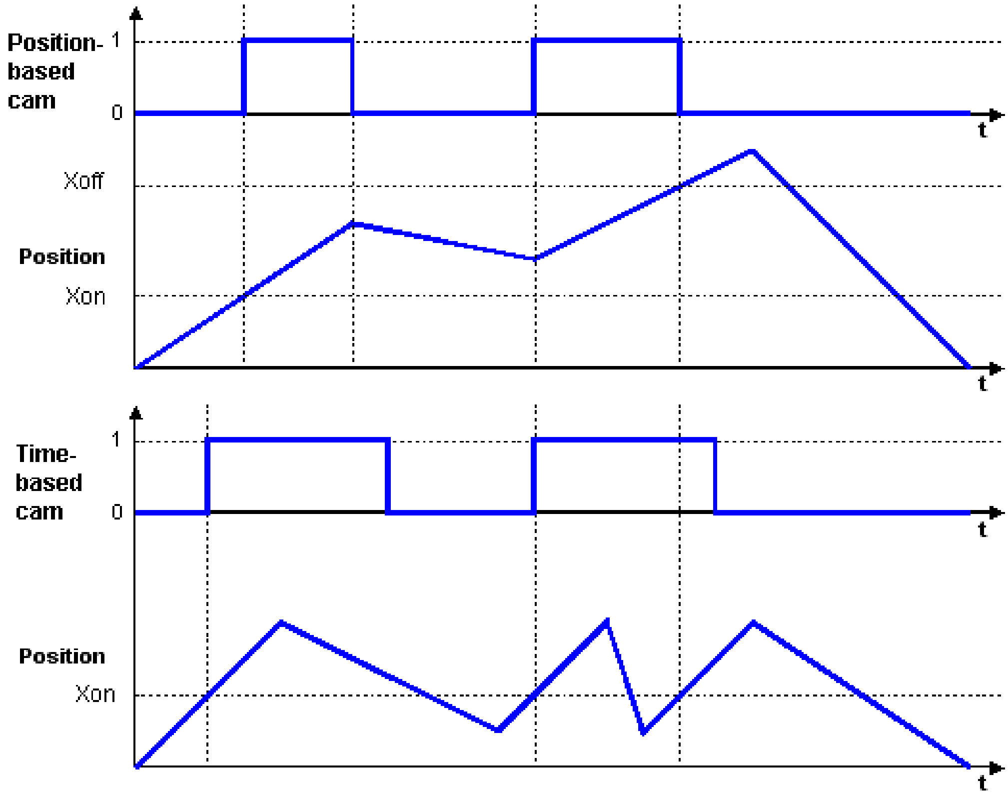
The switching action depends only on the position (position setpoint or actual position).

Effective direction

You can define a default effective direction when you activate output cams. The output cam is only activated if the motion corresponds with the set effective direction.
There are two options:

Effective direction	Behavior
Positive	The output cam is activated only in positive direction of movement.
Positive and negative	The output cam is activated independent of the direction of movement.
Negative	The output cam is activated only in negative direction of movement.
Current rotational direction	With this setting, the output cam is only activated in the currently set rotational direction. Standstill is a positive direction of rotation.

The diagram below shows the effects of the effective direction (= *Positiv)*:



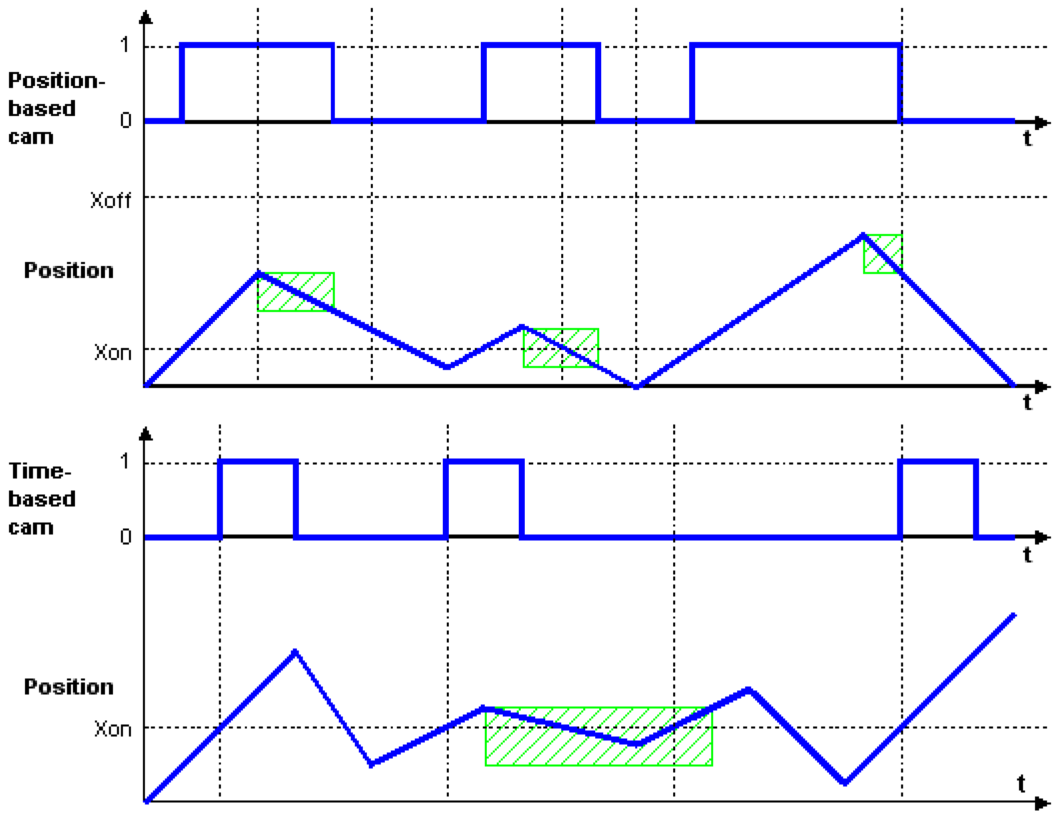
3.6.5 Hysteresis

Mechanical backlash may cause fluctuation of the actual position value which may lead to unwanted transitions of the output cam's switching state. This effect can be prevented by setting a hysteresis.

Hysteresis conditions

- The hysteresis is always active when the axis reverses its direction.
- The direction of motion is not redefined within the hysteresis.
- Within the hysteresis, the switching state of position-based cams is not changed.
- This current switching state will be set if modified output cam switching conditions are detected after the cam has moved out of the hysteresis window.

The diagram below shows the results of the effective direction (= positive)



The height of the green hatched area corresponds with the hysteresis.

Hysteresis range

The high limit of the hysteresis is set to a quarter of a modulo axis range and a quarter of a non-modulo axis range. The working range is defined by the software limit switches. The working range cannot be limited if a SW limit switch does not exist. You receive an error message if this maximum setting is exceeded.

- **Hysteresis of position-based cams**

  The hysteresis is enabled immediately after a reversal is detected. Position-based cams are switched off when they move out of the hysteresis window.

- **Hysteresis of time-based cams**

  The deactivation response of a time-based cam is determined by the cam activation time and not by the hysteresis.

The function is set at the *Hysteresis* input parameter of the "MC_CamSwitch" or "MC_CamSwitchTime" technology functions.

3.6.6 Time-based offset of cam switching points

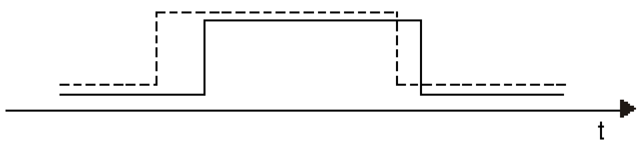
You can define a time-based offset, meaning a derivative action time, in order to compensate for mechanical reaction times, the rise times at digital outputs, the reaction time of connected switchgear, and communication times on DP(DRIVE). The time-based offset represents the cumulative delay times.

The time-based offset can be defined by setting positive and negative values. A negative cam activation time must be entered if the output cam is to be switched before it has reached its programmed start position.

The derivative action time is set at input parameter *Delay* of the "MC_CamSwitch" or. "MC_CamSwitchTime" technology functions. To maximize switching accuracy:

1. Determine the system delay time, based on several measurements at constant speed.
2. Define the derived mean value as time-based offset at input parameter *Delay*.

If suitable compensation times are used, the result is a switching accuracy of +/- 70 μs for high-speed output cams interconnected with the integrated outputs of the Technology CPU. The absolute switching accuracy (distance or angle) is derived from the set switching accuracy and current velocity.



_______ Output cam activation at input parameter *Delay* = *0* (no time-based offset)
\- - - - - - Output cam activation at input parameter *Delay* < *0*

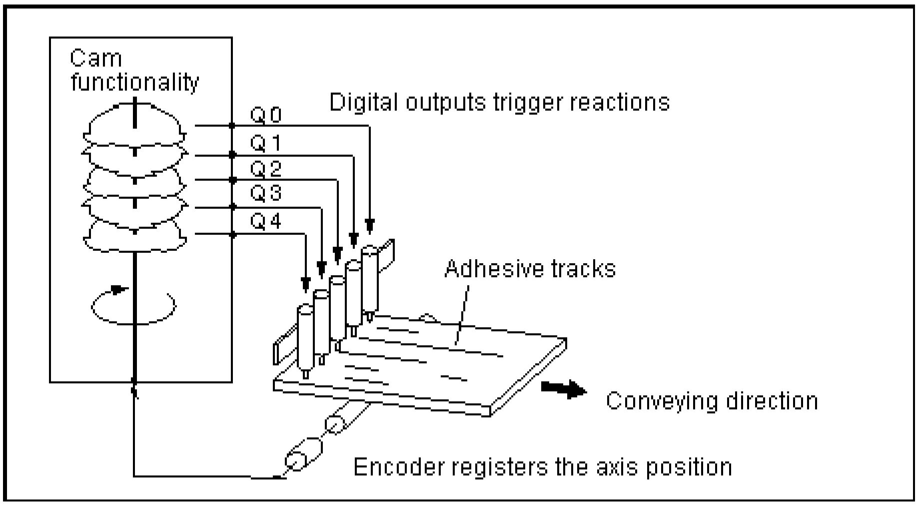
Note

Note:

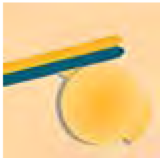
- The integrated technology calculates the dynamic switching position as a function of the axis velocity and defined time-based offset. The reference velocity is derived from the axis velocity at the time of activation of the output cam. Subsequent velocity changes are ignored in the calculation of output cam switching points.
- The offset of the calculated switching position may exceed the length of modulo axes.
- Long derivative action times exceeding one modulo cycle may lead to heavy fluctuation of the switching position of actual value output cams (actual value profile.) Use setpoint output cams if required.
- The dynamic derivative action of modulo axes can be greater than one modulo length. The system, however, does not collect switching operations. One switching operation is active in the system at any given point in time. A switching operation is completed when the output cam is switched off.

3.6.7 Example of an electronic cam control

Lines of glue are applied to a wooden board. The output cams are assigned to an external encoder. The output cams assigned to defined outputs are switched on and off at defined positions.



3.7 "Cam track" technology object



Use the "Cam track" technology object if you want to use several output cams of the same type on one axis or an external encoder. Up to 32 individual output cams can be used in a cam track. The switching signals of the individual output cams can be evaluated in the user program. The switching states of all the individual output cams can be switched as a total result to digital outputs. The following IO devices can be used:

- Digital outputs of integrated technology
- Digital outputs of a standard slave on DP(DRIVE)
- Digital outputs of SINAMICS Terminal Modules TM15 or TM17

The following output cam types can be selected for a cam track:

- Position-based cam (Page 95)

  Position-based cams are activated between the starting position and the end position. The position-based cam is deactivated outside this range.
- Time-based cam (Page 97)

  Time-based cams are set for a defined period when the starting position is reached.
- Time-based output cam with maximum ON length

  In the case of time-based output cams with maximum ON length, the ON period of the output cam is limited by the maximum ON length in addition to the duration. The output cam is switched off when the ON period or the maximum ON length is exceeded.

The selected cam type applies to all the output cams of a cam track. Position-based cams and time-based cams have the same switching behavior in the cam track as that of the "Output cam" technology object.

Interconnections

The "Cam track" technology object can be interconnected with the technology objects listed below:

- Positioning axes
- Synchronization axes
- External encoders

Reference to the actual or setpoint position

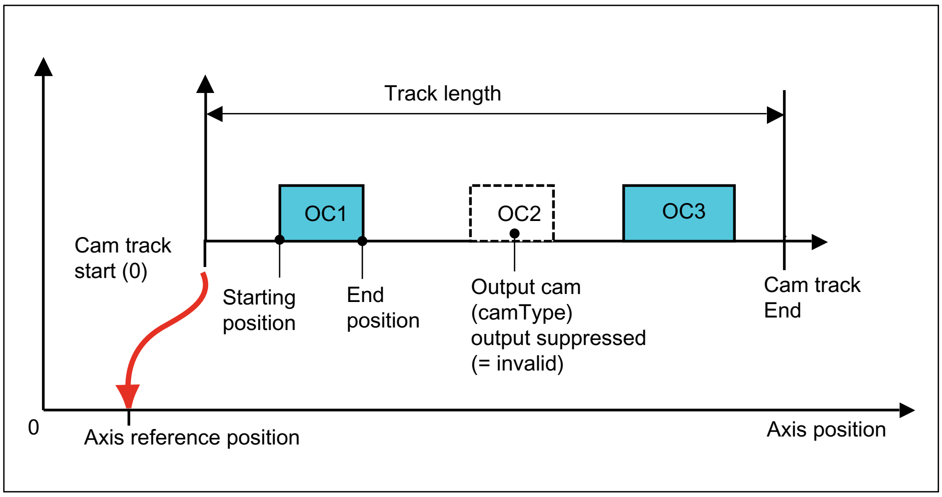
The switching positions of output cams can be referenced to the setpoint or actual value (setpoint output cam, actual value output cam.)

In order to enhance switching accuracy, you can compute the output cams within the position controller cycle, or use the integrated outputs of the Technology CPU for high-speed output cam operation.

Table 3-1 Comparison of the "Output cam" and "Cam track" technology objects

Feature	Output cam	Cam track
Supported cam types		
Position-based cam	X	X
Time-based cam	X	X
Time-based output cam with maximum ON length	-	X
Uni-directional output cam	X	-
Interconnection of cam outputs		
Several output cams of the same cam type to one digital output	Via AND / OR in S7T Config	X
Several output cams of different cam types to one digital output	Via AND / OR in S7T Config	-
Output cam properties		
Hysteresis	X	X
Effective direction	X	-
Type of activation / deactivation	Immediately	Adjustable
Derivative times (one time for switching ON / OFF)	X	X
Use	Cyclic	Cyclic or non-cyclic
Status of the output cam or of the individual output cam mapped in the technology DB	X	X
Enable output cam by using the technology function	MC_CamSwitch MC_CamSwitchTime	MC_CamTrack
Performance comparison		
Performance	Performance depending on the number of output cams.	From 5 individual output cams higher performance

Reference quantities of the cam track

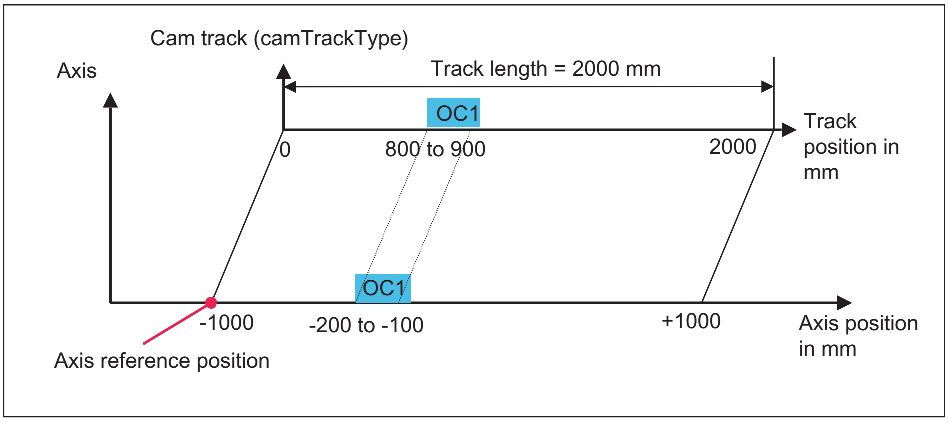


The cam track used as an example is defined with three individual output cams (*N1* ... *N3*).

The starting and end positions of the individual output cams are specified in the pre-assignment in S7T Config. The positions of the individual refer to the respective cam track beginning.

In the example the output cam *N2* is parameterized as invalid. The output cam *N2* is not taken into account as either a hardware output cam or as a software output cam. The validity of an output cam can be parameterized in S7T Config in the pre-assignment.

The track length determines the part of the cam track to be used. The cam track is applied and used at the axis reference position. The track length to be used is determined when calling up the "MC_CamTrack" technology function.



The cam track is always created in the positive direction on the axis reference position in accordance with your track length.

Technology functions

Technology functions supported by the technology object:

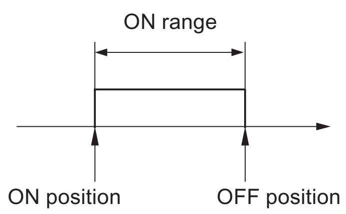
MC_Reset	MC_CamTrack	MC_ReadCamTrackData
MC_WriteCamTrackData	MC_ReadSysParameter	MC_WriteParameter

3.7.1 Position-based cam

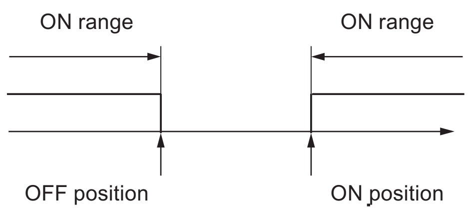
Cam activation range

The cam activation range of position-based cams is defined by the distance between the starting and end positions in **positive** count direction.

Starting position smaller than the end position:



Starting position greater than the end position:



The cam output is inverted when the starting position is greater that the end position

Switching characteristics

The switching characteristic of position-based cams is determined by the starting and end positions.

At the **"Output cam"** technology object an "effective direction" can be defined additionally at the technology object.

Switching action based on	Active position-based cam	Inactive position-based cam
Starting position, end position	• If the output cam was enabled with "MC_CamSwitch", or the cam track with "MC_CamTrack" • If the position is within the cam activation range • if the position value is shifted into the output cam activation range	• If the axis position is outside the starting or end area • when the position value is shifted out of the cam activation range *
Effective direction (not active at the cam track)	• when the position lies within the cam activation range and the effective direction = motion direction	• If the axis position is outside the starting or end area • if the motion direction does not match the configured effective direction • when the position value is shifted out of the cam activation range *
Hysteresis	when the cam moves out of the hysteresis range	when the cam moves out of the hysteresis range

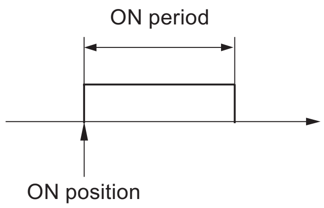
* The position value of the interconnected object may change abruptly, for example, as a result of its referencing, or due to the shift of its coordinate system.

Inverted output

The inverted output function interchanges the cam activation range and the cam deactivation range.

3.7.2 Time-based cam

Switching characteristics



The switching action of time-based cams is determined by the starting position and the cam activation time.

At the **"Output cam"** technology object an effective direction can be defined additionally at the technology object.

Switching action based on	Time-based cam is activated	Time-based cam is deactivated
Starting position	At the starting position	-
Cam activation time	-	on expiration of the programmed time
Effective direction (not active at the cam track)	At the starting position, if the motion direction = effective direction	on expiration of the programmed time

Note

Time-based cams can not be retriggered.

Inverted output

The time-based cam is activated for inverted output, and is deactivated at the cam deactivation position for the duration of a specified time.

3.7.3 Time-based output cam with maximum ON length

Switching characteristics

In the case of time-based cams with a maximum ON length, a maximum ON length is taken into consideration in addition to the time-based cam function.

The maximum ON length is defined in S7T Config in the pre-assignment of the cam track.

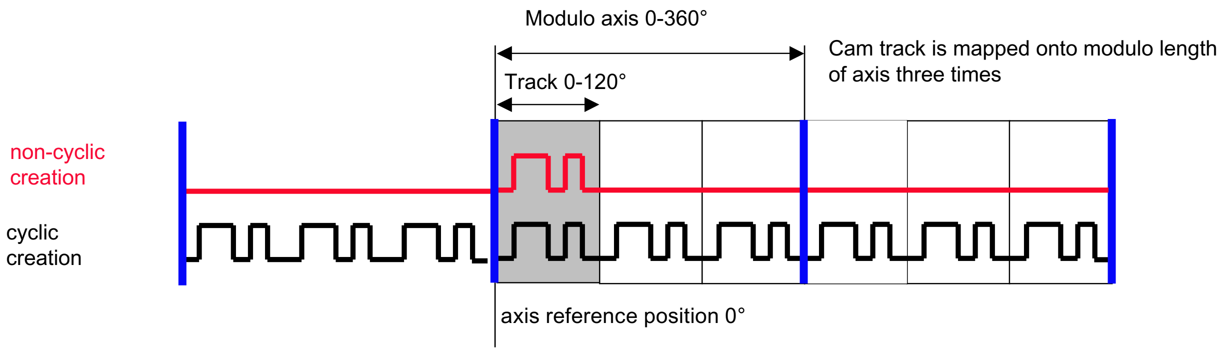
A time-based cam with a maximum ON length switches off when the configured time has expired, or the maximum ON length has been exceeded (depending on which event occurs first).

The maximum ON length is effective in both traversing directions of the axis, and the cam track's switch-on position is the reference position.

3.7.4 Cyclic and non-cyclic creation of the cam track

You can create a cam track in cyclic or non-cyclic mode on the axis reference position. To do so, use the input parameter *CyclicMode* of the technology "MC_CamTrack".

The following example shows the different effect during cyclic and non-cyclic creation of the cam track on the axis reference position. A modulo axis (0-360°) with a cam track (cam track length 120°) is used in the example. The cam track is created at the axis reference position 0°.



Non-cyclic creation of the cam track at the axis reference position.

The cam track is created once at the axis reference position. With modulo axes, the cam track is created in the modulo cycle, by activating it with the technology function"MC_CamTrack".

The cam track becomes active if the axis reference position is crossed in a positive direction.

The selection in the drop-down list box "Activate non-cyclic activated cam track outside the track range" in the cam track configuration of S7T Config defines the deactivation of the cam track.

- Selection "Yes" (default setting)

  The cam track remains active even after the cam track is left (cam track length). When the cam track is entered again, the cams switch again. Use *Mode = 1* the technology function "MC_CamTrack", to deactivate the cam track.

- Selection "No"

  The cam track is deactivated when the cam track length is left.

Cyclic creation of the cam track at the axis reference position

The cam track is repeatedly created at the axis reference position according to its track length. The cam track is active in all axis ranges. Use *Mode = 1* the technology function "MC_CamTrack", to deactivate the cam track.

Note

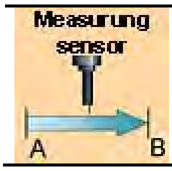
An unfavorable selection of the cam track length and the modulo length can lead to undesired acyclic overrides. **Avoid** the following constellations:

- Cam track projects beyond the modulo length, or cam track length > modulo length (with cyclic and non-cyclic creation of the cam track)
- Modulo length and cam track length are not in an integer relation to each other (with cyclic creation of the cam track)

Tip

With cyclic creation of the cam track, select a cam track length that is twice as large as the modulo length if you want to switch the cams only in every second modulo cycle.

3.8 "Measuring input" technology object



Use the "Measuring input" technology object for precise and fast logging of actual position values.

A measuring input can be assigned in S7T Config to an axis or to an individual external encoder. The sensor is connected to a digital input of the drive component, or to a TM15/TM17 High Feature module.

The measurement of actual position values is initiated by calling the Motion Control command "MC_MeasuringInput" in the user program of the CPU. The operating range of the measuring input can be limited accordingly by setting a start and end value.

The actual position value is recorded and saved at the positive or negative edge of the measuring input signal. The measuring function can be triggered at both edges, provided the component used supports this mode.

Technology functions

Technology functions supported by the technology object:

MC_Reset	MC_MeasuringInput	MC_ReadSysParameter
MC_WriteParameter		

3.8.1 Interconnection and connection of a measuring input

- In S7T Config, the "Measuring input" technology object can only be inserted for the "Axis" and "External encoder" technology objects.
- Each "Measuring input" technology object is only assigned to a single axis or to a single external encoder
- The measuring input must be located either at the corresponding drive component, or at a TM15/TM17 High Feature module. The measuring inputs are indicated specifically at the corresponding hardware, or have to be configured there as measuring inputs. The number of available measuring inputs depends on the hardware used.

  Other digital inputs at DP(DRIVE) can **not** be used as measuring inputs.

  The digital measuring input used for the "Measuring input" technology object at the TM15/TM17 High Feature module is configured in S7T Config
- Several "Measuring input" technology objects can be configured for a positioning axis, synchronization axis or an external encoder.

  The "Measuring input" technology objects may **not** be active simultaneously.
- Measuring inputs may not be configured for use with virtual axes.
- Measuring inputs can not be configured for operation with speed-controlled axes.

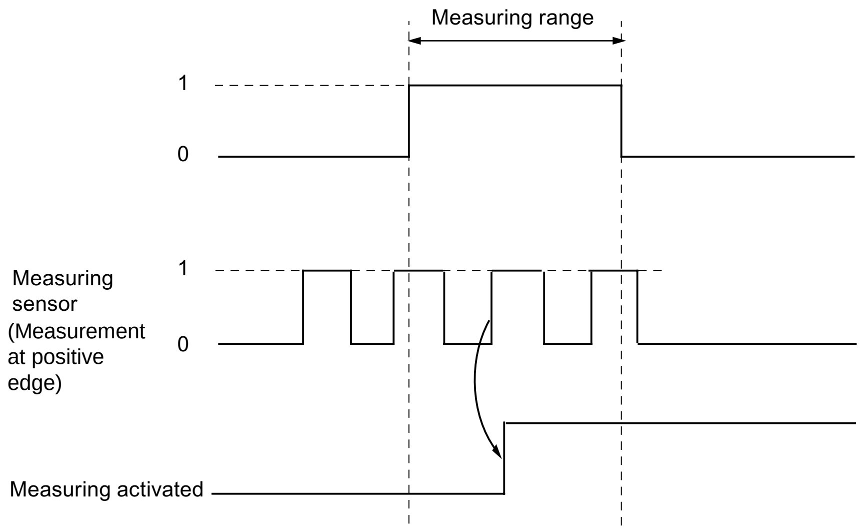
Activation / deactivation

Measuring inputs are activated / deactivated by calling the "MC_MeasuringInput" technology function.

3.8.2 Measuring range

The measuring command may apply to the entire range, or be limited to a start and end position (active range).

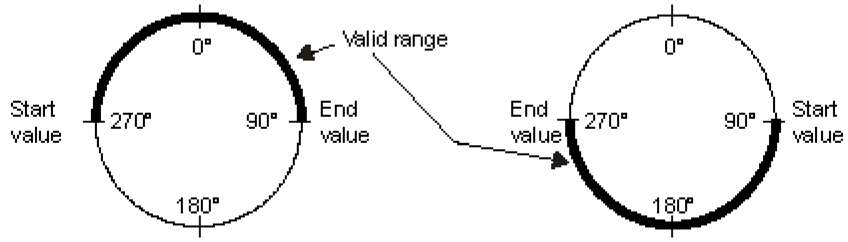
The measurement can be restricted to a defined range. The measurement is only triggered when the position lies within the measuring range.



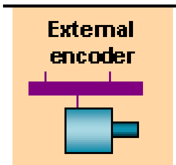
The measuring command is canceled and an error is indicated at the technology function and at the technology DB if a trigger signal is not recognized within the measuring range.

The order by which the start and end values are specified is irrelevant to non-modulo axes. If the initial value is greater than the end value, the values are interchanged.

If the start value is greater than the end value at a modulo axis, the valid range is extended from the start value beyond the modulo transition of the axis to the end value.



3.9 "External encoder" technology object



Use the "External encoder" technology object to return the position or angle of a mechanical component as a master setpoint to the Technology CPU.

The "External encoder" technology object returns the actual position and velocity values of a distance measuring system. The technology object does not control the effective drive component.

Technology functions

Technology functions supported by the technology object:

MC_Reset	MC_ExternalEncoder	MC_ReadSysParameter
MC_WriteParameter		

3.9.1 Interconnection and connection of an external encoder

Interconnection

The "External encoder" technology object can be interconnected with the following technology objects:

- "Synchronization axis" technology object as leading axis
- "Output cam" technology object as actual position value
- "Measuring input" technology object as actual position value

Wiring

Encoders supported by the "External encoders" technology object:

- Analog absolute encoder (sensor analog) wired to an analog input module of ET 200M or ET 200S
- Incremental encoder (rectangular TTL) wired to analog drive interface IM 174/ADI4
- Absolute encoder (SSI) wired to analog drive interface IM 174/ADI4
- Incremental or absolute encoders wired to the encoder input of a DP drive
- Absolute encoder SIMODRIVE Sensor Isochron (message frame 81)

Examples

- You can use the second encoder interface of a double-axis module can be used to wire an external encoder to SIMODRIVE 611U
- A second encoder can be wired to MASTERDRIVES MC using an encoder module
- Isochronous PROFIBUS encoders can be operated directly on DP(DRIVE)
- SINAMICS S120 supports the connection of a second encoder via SMC 30, SME 20, SME 25

3.9.2 Synchronization of the external encoder

The Technology CPU supports various homing modes. The reference position of the external encoder can be set at the input parameter *Mode* of the "MC_ExternalEncoder" technology function.

Homing with incremental encoders

- Direct homing (*Mode* = *2* and *4*)

  Setting the reference point. The value of the reference point coordinate is assigned to the current encoder position
- Passive homing / On-the-fly homing (*Mode* = *3* and *5*)

  The value of the reference point coordinate is assigned to the current encoder position as the encoder is moving:
  - after the encoder zero mark is reached (default setting)
  - on reaching the external zero mark (set in the expert list)
  - When the next encoder zero marker after the reference cam is reached (set in the expert list)

  The distance between the reference cam and the encoder zero mark can be monitored using the encoder zero mark monitoring function.

  Synchronization with incremental encoders can be set in the expert list.

Configuring 4

4.1 Fundamental procedure for configuration

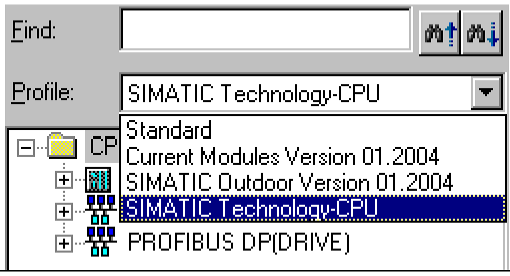
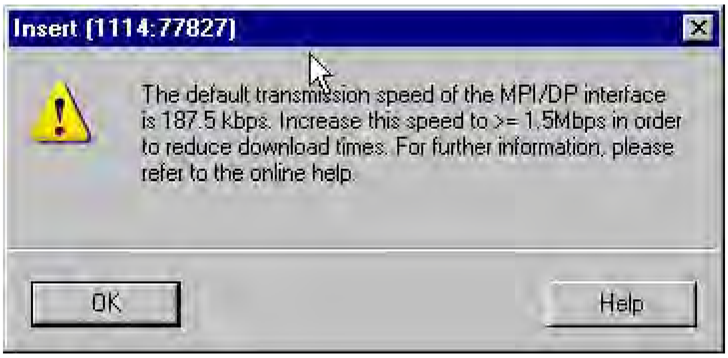
Steps to perform in order to implement motion control commands:

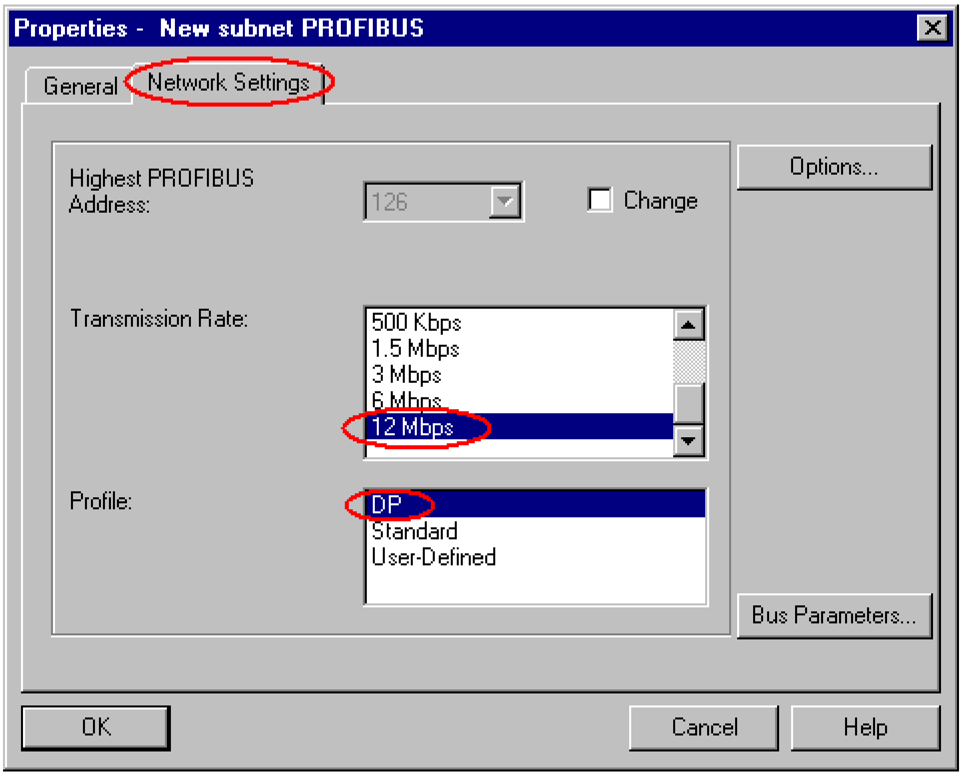
Step	Action	Tool
1.	Creating a project	STEP 7 - SIMATIC Manager
2.	Configure the Technology CPU (Page 106)	HW Config
	Configure the drives (Page 110)	
	Station > Save and compile menu command	
3.	Configuring axes (Page 126)	S7T Config
	Configure Technology Objects	
	Project > Save and compile all menu command	
4.	Creating and managing technology DBs (Page 119)	Technology Objects Management
5.	Programming function blocks	LAD/FBD/STL
6.	**PLC > Download user program to Memory Card** menu command (including system data)	STEP 7 - SIMATIC Manager

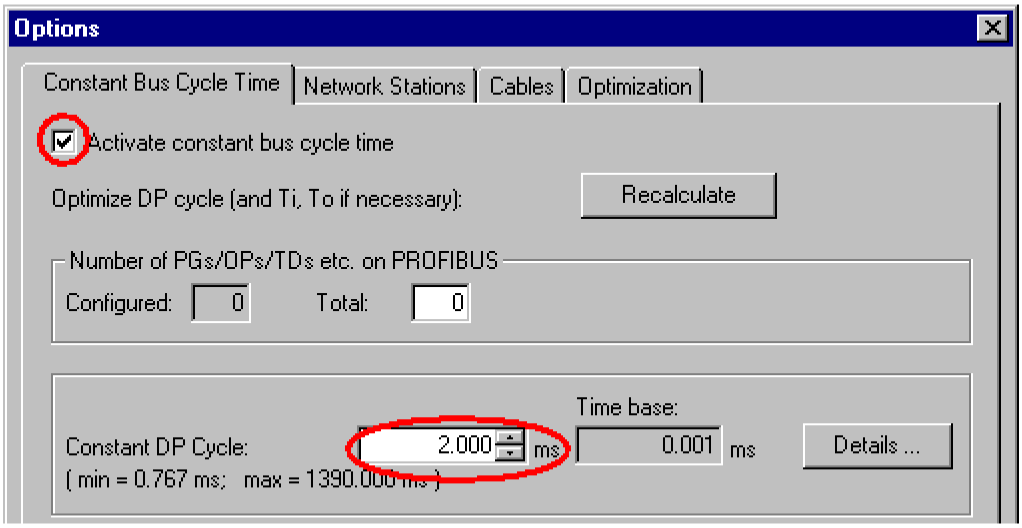
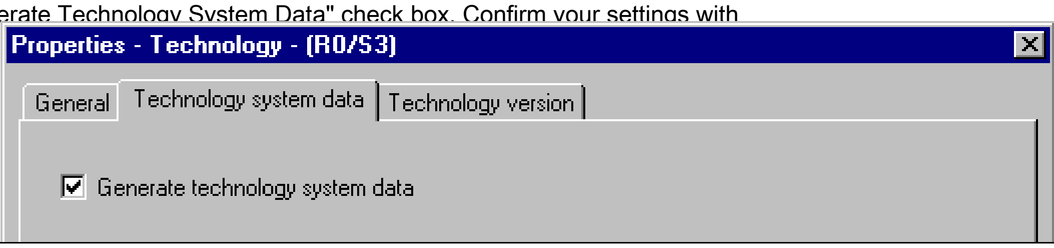
4.2 Configuring the technology CPU and drives in HW Config

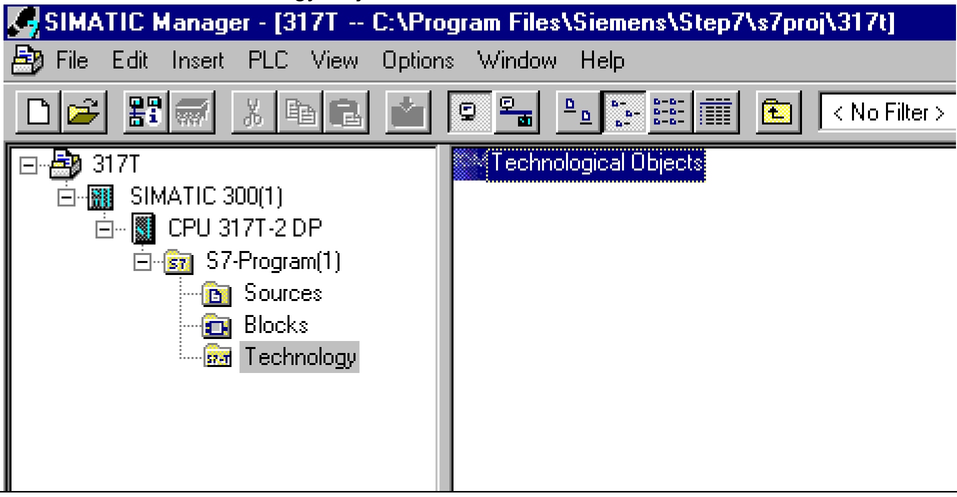
4.2.1 Configuring the Technology CPU in HW Config

To configure the Technology CPU:

Step	Description
1.	Create a new project SIMATIC Manager and add a SIMATIC 300 station.
2.	Open HW Config by selecting the "SIMATIC 300" station and double-clicking "Hardware".
3.	In the "Hardware catalog" view, select the "SIMATIC Technology CPU" hardware profile from the "Profile" drop-down list. 
4.	Copy a mounting rail from the "Hardware catalog" view to the station window of HW Config.
5.	Drag-and-drop a power supply module to the mounting rail (for example: "PS 307 5A".)
6.	Drag-and-drop the relevant Technology CPU from the HW catalog to the selected row containing the rail. This opens the message box shown below.  Minimize transfer times by increasing the transmission rate at the MPI/DP interface. For further information, refer to the message box help.
7.	Close the message box Set the PROFIBUS properties of DP(DRIVE) in the next dialog box.
8.	Click "New" to create a new "Subnet" (DP master system). Make sure that the Technology CPU is the only master station on DP(DRIVE). Set address *0* at the DP(DRIVE) in order to enhance operating performance.

Step	Description
9.	Select the "Network settings" tab from the properties dialog box of the PROFIBUS network  Select a transmission rate of 12 Mbps. Do not change the "DP" profile of the PROFIBUS network. To enhance performance of the DP(DRIVE) system: • Set the *0* value at "Highest PROFIBUS address". • Deactivate the "Enable cyclic broadcast of bus parameters" option. Click "Bus parameters," and then reset the check box.

Step	Description
10.	Click "Options". Set the "Activate constant bus cycle time" check box in the next dialog box.  As the value for the constant DP bus cycle time, enter the time required for isochronous operation of the drive components on DP(DRIVE). Observe the device-specific properties and the quantity framework of the components supported for operation on DP(DRIVE). Retain the defaults of all other options offered in this dialog box. **Note:** You can also make the settings using the properties dialog boxes of the drives. The equidistant DP cycle and the cycle time are then automatically activated or aligned.
11.	Close all dialog boxes of HW Config by clicking "OK".
12.	Increase the transmission rate at the MPI/DP interface (see step 6). Save and compile the hardware configuration data and download these to the target system.
13.	In HW Config, double-click the technology (slot 3), and then select the "Technology System Data" tab. Set the "Generate Technology System Data" check box. Confirm your settings with "OK". 

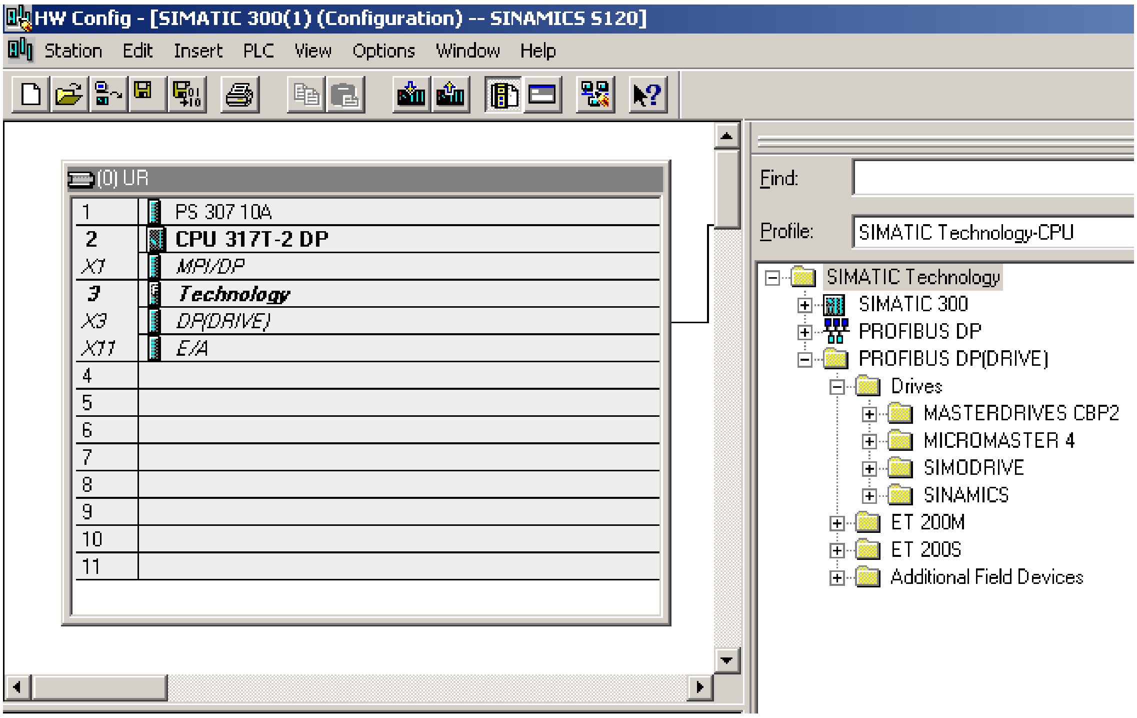
Step	Description
14.	Save and compile the hardware configuration data and download these to the target system. Adapt the PG / PC interface. Download the hardware configuration and firmware data of the integrated technology to the target system. Result: After you compiled the data, the project dialog box in SIMATIC Manager shows the "Technology" folder that contains the "Technology Objects". 

4.2.2 Configuring drives in HW Config

Basic procedure

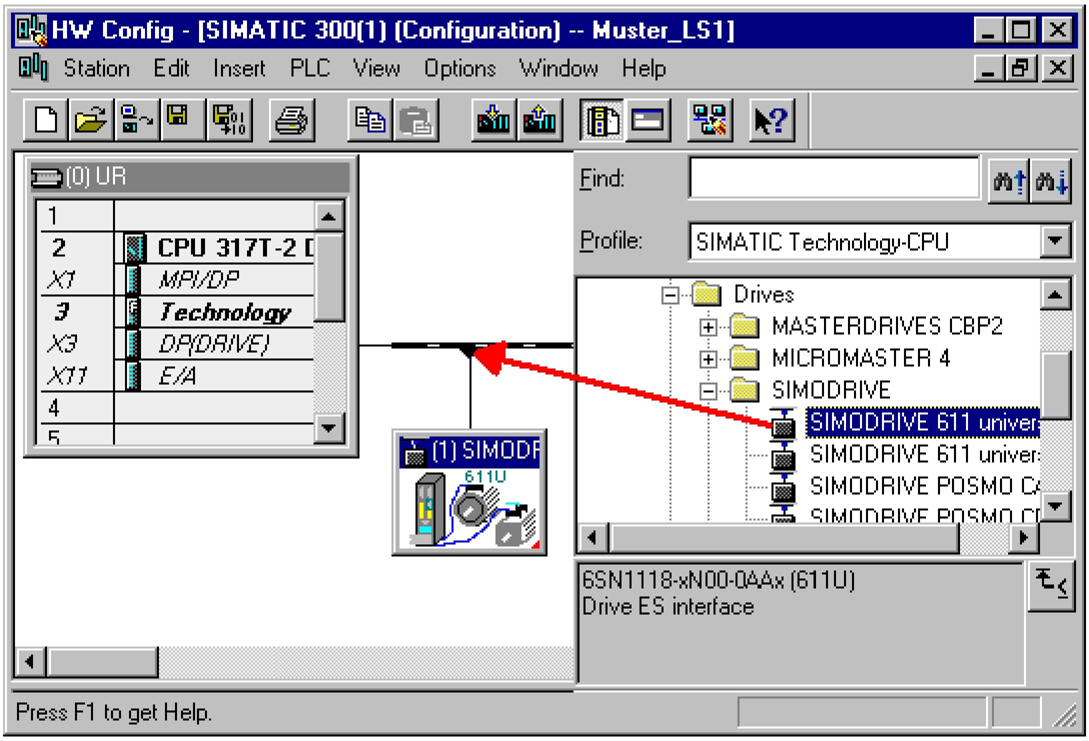
The next steps below describe the basic procedure of configuring the drives based on the example of a SIMODRIVE 611 universal.

For detailed information on the HW configuration of drives, refer to the product information and to the relevant drive documentation.

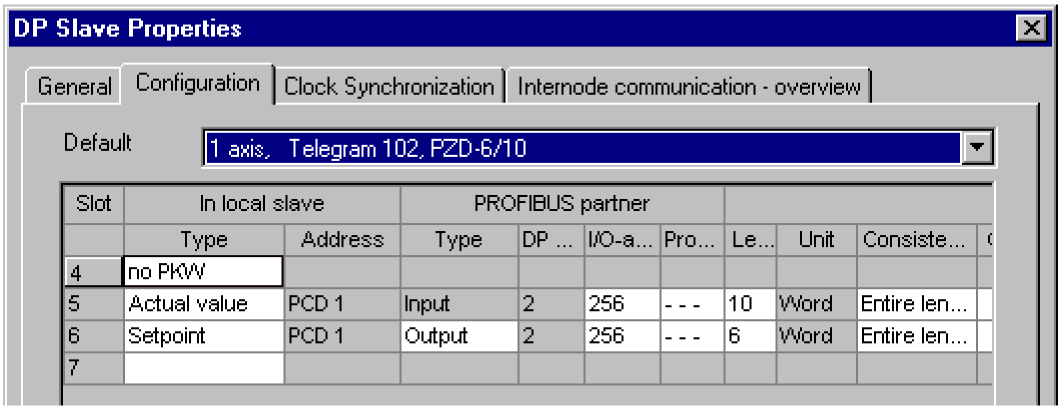
<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1.</td><td>Open the HW catalog, and then open the folder
"SIMATIC Technology > PROFIBUS DP(DRIVE) > Drives > SIMODRIVE".


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

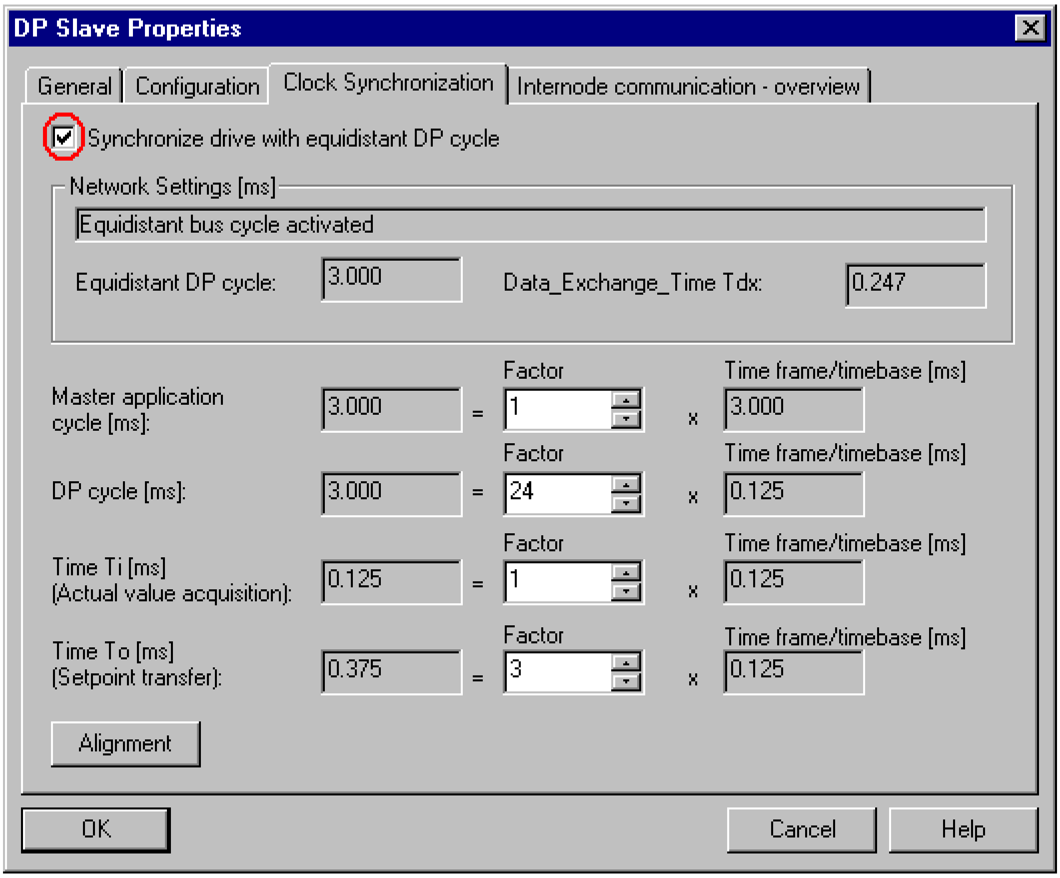
<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>2.</td><td>Select the required drive component (example: "SIMODRIVE 611 universal"). Drag-and-drop this component to the DP master system.



The "Properties – PROFIBUS interface SIMODRIVE 611 U DP2, DP3" dialog box opens.</td></tr>
<tr><td>3.</td><td>Set the relevant PROFIBUS address 4, for example. Confirm your entry with "OK".
The "Properties - Drive" dialog box opens.
Close the dialog box with "OK".</td></tr>
<tr><td>4.</td><td>Select the "Configuration" tab on the "DP Slave Properties" dialog box and select the required message frame.



For details on selecting message frames and supported functions, refer to "Message frame types and their functions (Page 113)".
Select for example the message frame 102 for the operation of one axis on SIMODRIVE611 universal as positioning or synchronization axis without DSC with torque reduction).</td></tr>
</table>

Step	Description
5.	Select the "Synchronous operation" tab and set the "Synchronize drive with equidistant DP cycle" check box.  As a result of the settings made in the PROFIBUS network configuration, the DP cycle time is activated and displayed on this dialog box The DP cycle coefficient is adapted accordingly. If you do not configure the cycle time in the network properties the coefficients must be adapted in this dialog box in order to obtain the relevant cycle time (example: 16 for a DP cycle of 2.00 ms). The DP cycle of the DP(DRIVE) interface is always an integer multiple of 0.5 ms and is mainly determined by the number and type of components to be operated on the network. Also observe the drive-specific help available in this dialog box when you define the DP cycle.
6.	For this example, set the "To" time coefficient to obtain a time of 0.5 ms. Do not change the default value of the "Ti" coefficient.
7.	Click "Alignment". This triggers the following settings:· • The constant bus cycle is activated • The DP cycle of the master system will be adapted to the drive properties • All drive components of the same family receive the same parameter configuration
8.	Close the dialog box with "OK".
9.	Save and compile the hardware configuration data and download these to the target system. If you have set the "Generate technology system data" check box in the technology properties, the system will generate the system data for the HW configuration and for the technology firmware.

Note

Drive components of different families must be calibrated manually. To do so, note down the set values and then transfer these to the corresponding dialog boxes.

4.2.3 Message frame types and their functions

The message frame used to communicate with the drive must be defined in HW Config. The message frame to be selected is determined by the axis function required, for example, torque limiting or DSC, or by the functionality of the drive used.

Note

When changing the message frame type of a drive component in HW Config, you also need to adapt the selection of the message frame in the interface configuration of the corresponding technology object in S7T Config.

Communication with digital drives via PROFIBUS DP is handled in accordance with PROFIdrive Version 3 specifications.

Technologies and message frame types supported

Table 4-1 Message frame types and functionality:

Message frame type	Short description / functionality
1	n-setpoint interface 16 bits
2	n-setpoint interface 32 bits, without encoder
3	n-setpoint interface 32 bits, with encoder 1
4	n-setpoint interface 32 bits, with encoder 1 and encoder 2
5	n-setpoint interface 32 bits, with DSC and encoder 1
6	n-setpoint interface 32 bits, with DSC, encoder 1 and encoder 2
101	n-setpoint interface with torque reduction
102	n-setpoint interface with encoder 1 and torque reduction
103	n-setpoint interface with encoder 1, encoder 2 and torque reduction
105	n-setpoint interface with DSC, encoder 1 and torque reduction
106	n-setpoint interface with DSC, encoder 1, encoder 2 and torque reduction

Table 4-2 Drives and message frame types:

Drive component	Technology Objects	Interface	Programmable standard message frame types
SIMODRIVE			
SIMODRIVE 611U universal	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	1 ... 6, 101, 102, 103, 105, 106 **
SIMODRIVE 611U universal HR	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	1 ... 6, 101, 102, 103, 105, 106 **
SIMODRIVE POSMO CA/CD	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	1 ... 6, 101, 102, 103, 105, 106
SIMODRIVE POSMO SI	Speed-controlled axis, positioning axis, synchronization axis, measuring input, output cam	Digital	1, 2, 3, 5, 101, 102, 105
SIMODRIVE sensor isochronous	External encoder	Digital	81
MICROMASTER 4			
COMBIMASTER 411	Speed-controlled axis	Digital *	1
MICROMASTER 420	Speed-controlled axis	Digital *	1
MICROMASTER 430	Speed-controlled axis	Digital *	1
MICROMASTER 440	Speed-controlled axis	Digital *	1
MASTERDRIVES			
Motion Control	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	1 ... 6
Motion Control Plus	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	1 ... 6
Vector Control CUVC	Speed-controlled axis	Digital *	1, 2
Vector Control Plus	Speed-controlled axis	Digital *	1, 2
SINAMICS			
SINAMICS S120	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	1 ... 6, 102, 103, 105, 106
TM15 terminal module	Measuring inputs Output cams	Digital	-
TM17 High Feature Terminal Module	Measuring inputs Output cams	Digital	-

Drive component	Technology Objects	Interface	Programmable standard message frame types
SINUMERIK			
ADI4	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	3
SIMATIC Distributed I/O			
PROFIBUS module IM 174	Speed-controlled axis, positioning axis, synchronization axis, external encoder, measuring input, output cam	Digital	3
ET 200M, ET 200S	Output cams	Digital	-
* The drive does not support isochronous operation on PROFIBUS. ** For further details refer to the supplementary descriptions of 611U			

4.3 Technology Objects Management

4.3.1 Starting Technology Objects Management

Use S7T Config to configure the technology objects for your motion control task.

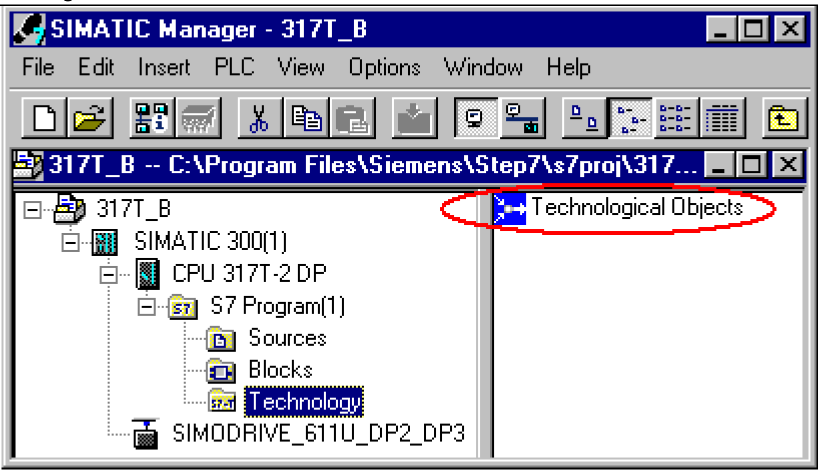
The technology DBs form the interface between the user program and the technology objects. Technology DBs are managed in "Technology Objects Management" (TOM).

"Technology Objects Management" can be used in parallel with SIMATIC Manager.

Prerequisites

- SIMATIC Manager is open
- The Technology CPU was configured in HW Config, and the configuration data were saved

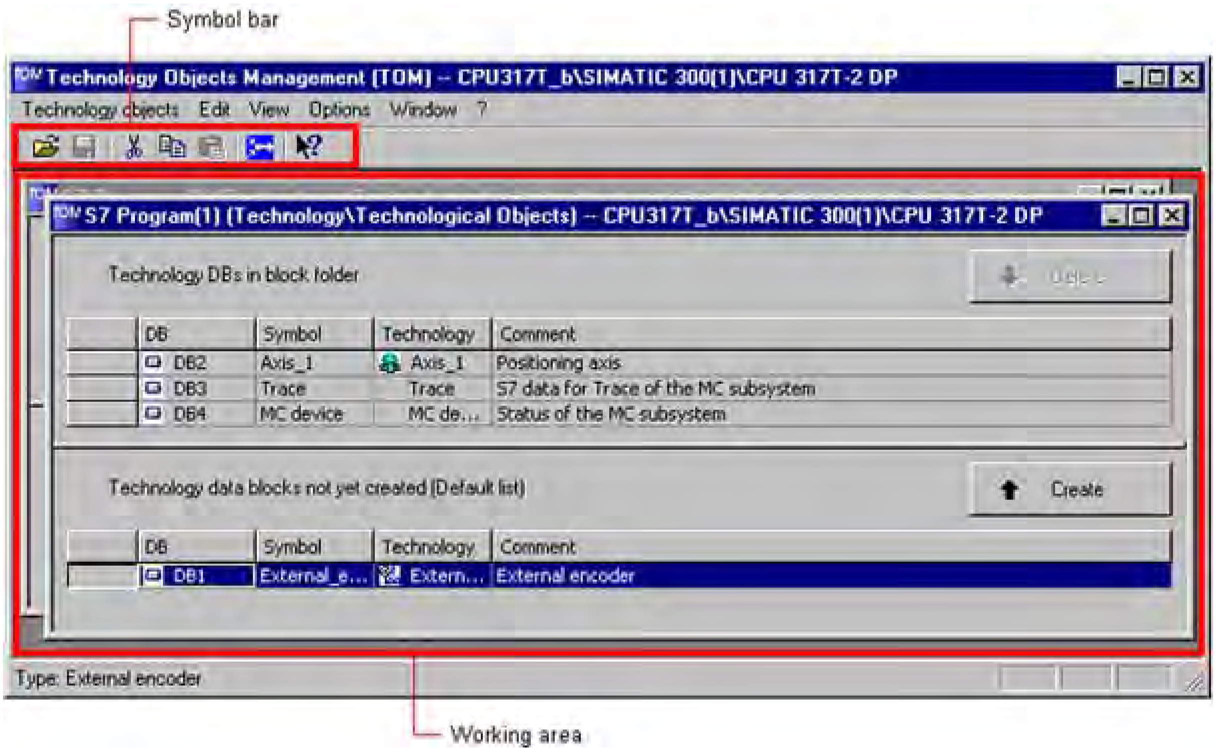
Starting Technology Objects Management

<table>
<tr><th>Step</th><th>Significance</th></tr>
<tr><td>1.</td><td>In SIMATIC Manager, select "Technology objects" from the "Technology" folder to run "Technology Objects Management".</td></tr>
<tr><td>2.</td><td>


Select "Technology Objects", and then select the Edit > Open object command in SIMATIC Manager.
You can also start S7T Config without running "Technology Objects Management".</td></tr>
<tr><td>3.</td><td>Select "Technology objects" from the "Technology“ folder in SIMATIC Manager, and then select Options > Configure technology.
Using the menu command in SIMATIC Manager to start it is useful if you do want not create any new technology objects, and therefore do not have to create any new technology DBs.</td></tr>
</table>

4.3.2 User interface of Technology Objects Management

Layout of "Technology Objects Management":



Toolbar

The toolbar contains the essential menu commands.

Work area

Each STEP 7 project is opened in a separate working window on your working area. Each window shows all existing technology DBs:

- The upper area of the window shows the technology DBs you already created.
- The bottom section of the working area lists the technology objects for which you have not yet generated any technology data blocks.

4.3.3 Using Technology Objects Management

"Technology Objects Management" is used to manage the technology DBs of a STEP 7 project.

You can open a separate window for each configured station containing a Technology CPU on the working area of "Technology Objects Management".

On the active window, you can:

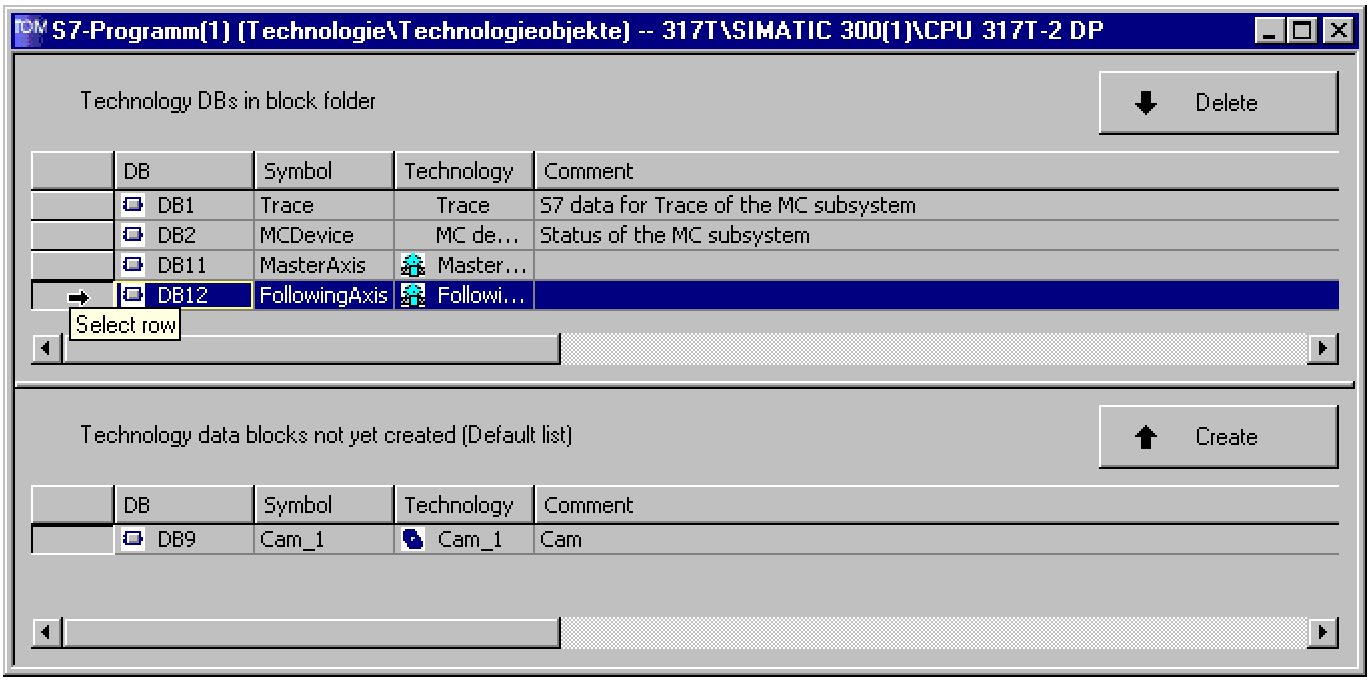
- Create, program or delete technology DBs
- Set IPO synchronous mode at the axis (**View > Expert mode** menu command)
- Define update cycles of the technology DBs (expert mode)
- Configure the start parameters of axis commands (properties "IPO synchronous" in expert mode)
- Starting S7T Config

Prerequisite

- The configuration of the Technology CPU in S7T Config is completed, saved and compiled.

Creating or deleting technology DBs

Select one or several technology DBs you want to create or delete:



Next, click "Create" or "Delete".

Note

For further information on using "Technology Objects Management", refer to the corresponding Online Help system.

Starting S7T Config

Start S7T Config by selecting the **Options > Configure technology** command.

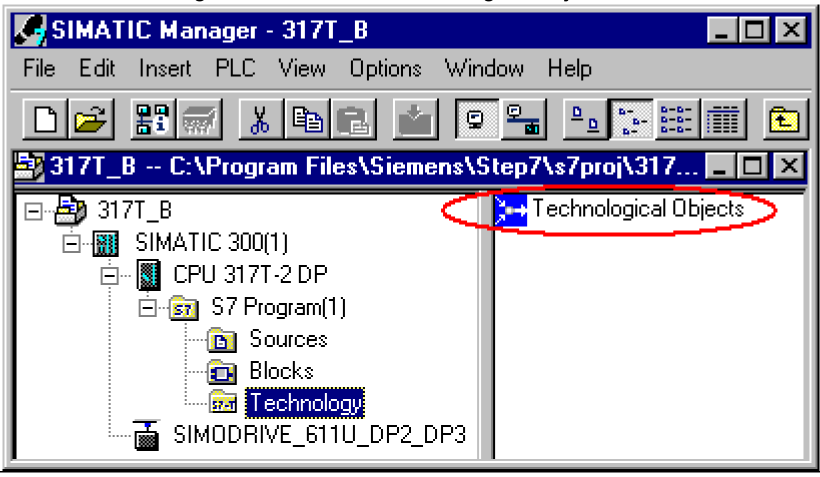
4.3.4 Creating and managing technology DBs

Technology objects (TOs) are accessed by means of technology DBs. You should therefore generate the technology DBs using the "Technology Objects Management" tool before you create the user program.

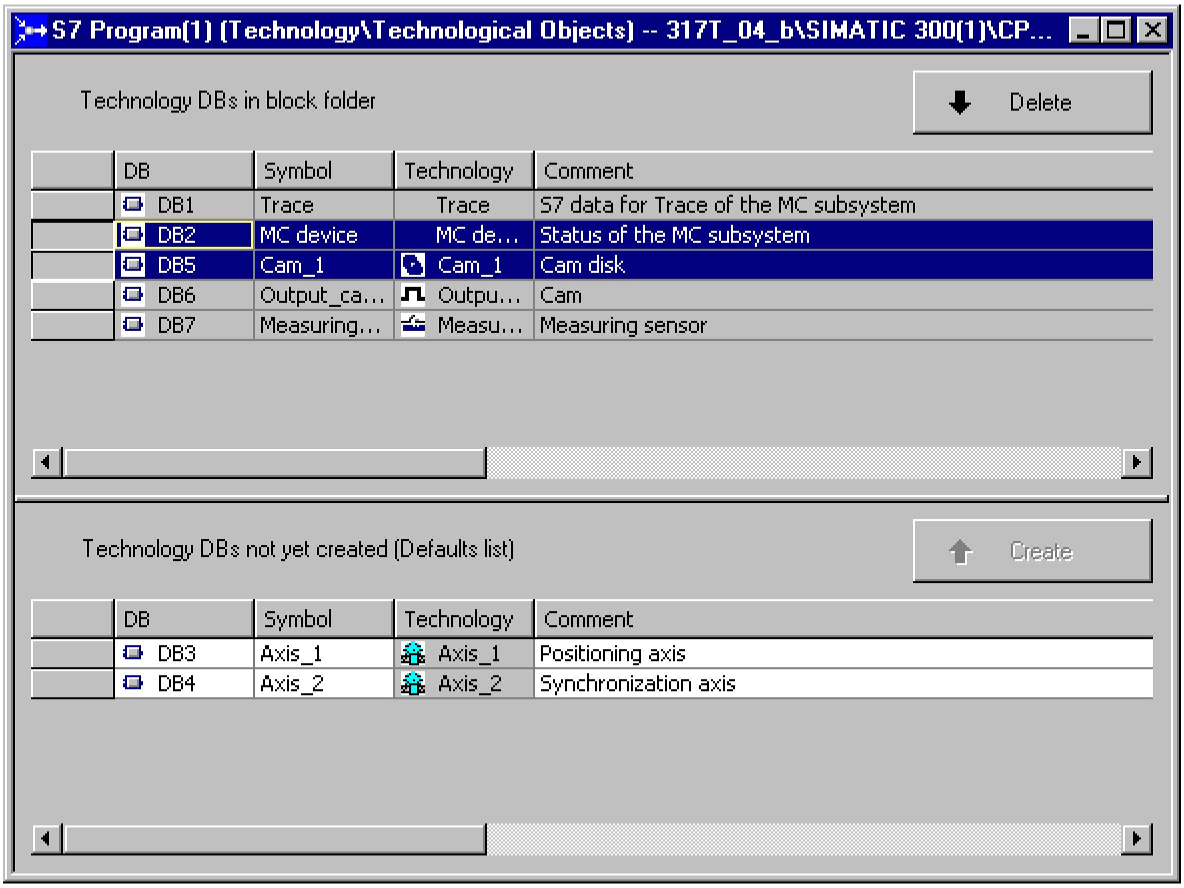
Prerequisite

- You concluded the configuration of the Technology CPU and drives in HW Config by executing the **Station > Save and compile** command.
- The configuration of the Technology CPU is successfully completed in S7T Config and saved.

Example: To rename a technology DB

Step	Description
1.	In SIMATIC manager, double-click "Technological objects". 

Step	Description
	The "Technology Objects Management" dialog opens. (Example):



Step	Description
	You selected the DBs you want to create.
2.	From the "Technology DBs in block folder" area, select the relevant line to rename its DB.
3.	Click "Delete"
4.	Edit the block number in the "Technology DBs not created yet" area.
5.	Select the line showing the DB you want to create.
6.	Click "Create"

Note

Delete any previously downloaded technology DB of this name from the CPU. You can then download the renamed technology DB with the new DB number to the CPU.

Note

Technology DBs copied in the block folder of SIMATIC Manager must be verified. The corresponding DBs are not valid as technology DBs under the following conditions:

- If marked in red in the dialog box mentioned above
  In this case, select "Technology DBs in block folder," and then click "Delete."
- If not listed in the previously mentioned dialog box
  In this case, delete the block from the block folder of the SIMATIC Manager.

4.4 Using S7T Config

4.4.1 Starting S7T Config

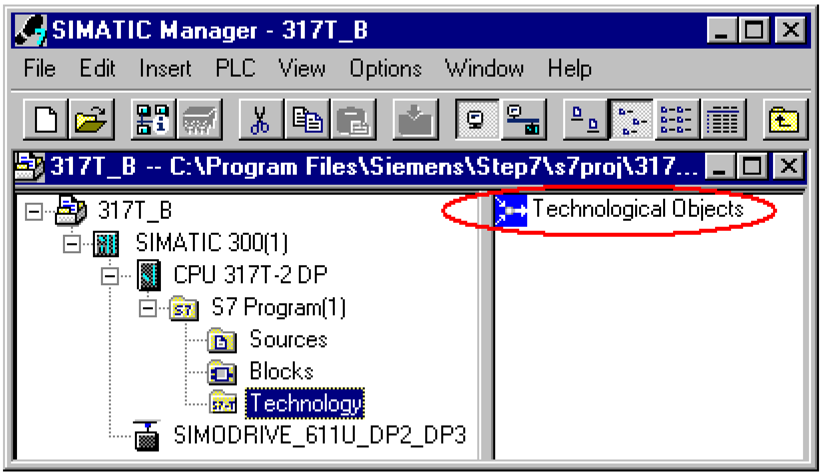
Prerequisites

- SIMATIC Manager is open
- The Technology CPU was configured in HW Config, and the configuration data were saved

Starting S7T Config

S7T Config included in your S7-Technology option package provides comfortable features for programming the technology objects of your motion control task.

Start S7T Config by selecting "Technological objects" from the "Technology" folder in SIMATIC Manager The object is only visible if the Technology CPU is configured in HW Config and if all configuration data were compiled. The drives and external encoders will be renamed at the start of S7T Config if their names contain any white spaces. These white spaces will be replaced with underscore characters.



Select "Technology objects", and then select **Edit > Open Object** in order to run "Technology Objects Management".

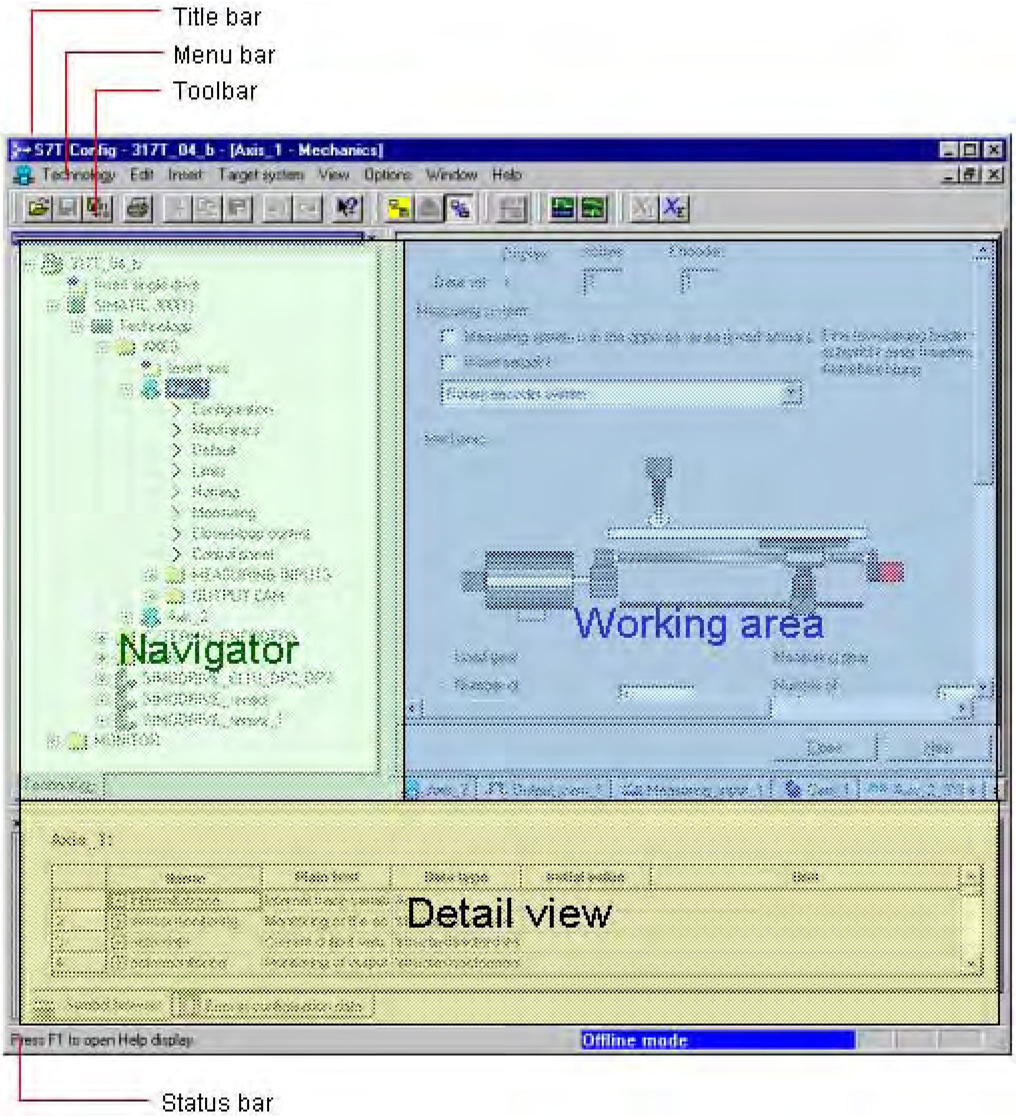
Select the **Options > Configure technology** in "Technology Objects Management" in order to open S7T Config and to configure the technology objects. S7T Config starts automatically if you have not yet created any technology objects.

You can also start S7T Config without running "Technology Objects Management". Select "Technology objects" from the "Technology“ folder in SIMATIC Manager, and then select **Options > Configure technology**. Using the menu command in SIMATIC Manager to start it is useful if you do want not create any new technology objects, and therefore do not have to create any new technology DBs.

4.4.2 The user interface of S7T Config

S7T Config is used to configure the technology objects for your motion control tasks.

S7T Config opens with this user interface:



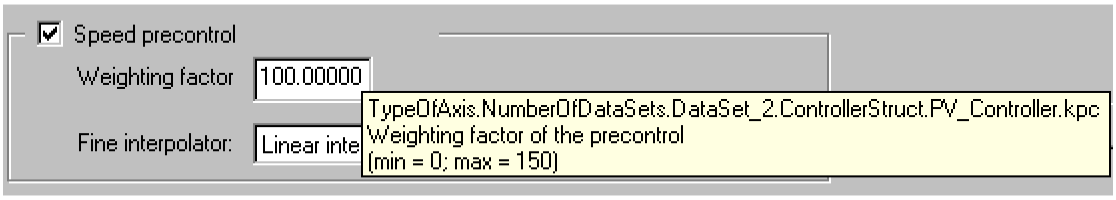
Navigator

The Navigator visualizes the project structure of the technology, and the assignment of the configured technology objects. Identical objects are indicated by the same icons. The system indicates online mode by highlighting the icons with a corresponding background color, or by displaying additional icons (connector icons) which characterize the interconnection of the PG/PC with the PLC. Please refer to the S7T Config online help for a detailed description. You can open the relevant configuration dialog box by double-clicking an object on the working area.

Work area

The technology object configuration dialog boxes, for example, for setting configuration data, defaults and limits, are displayed on the working area. The tabs indicate the technology objects for which you have opened the relevant dialog box. You can open dialog boxes for several technology objects.

All the modifiable parameters in the working area are assigned high and low limits which are indicated by a tooltip in S7T Config.



Detail View

The details view outputs information about the elements you selected in Navigator. The view shows different tabs, depending on the object you selected. The number of tabs depends on the active dialog boxes on the working area, and on the project status (offline or online).

To obtain information about an object in the detail view:

1. Select the relevant object in the Navigator. The detail view shows all tabs available for this object.
2. Click the tab in the detail view The tab content is shown

Title bar

The title bar contains the project name and the name of the active configuration dialog box of the selected technology object.

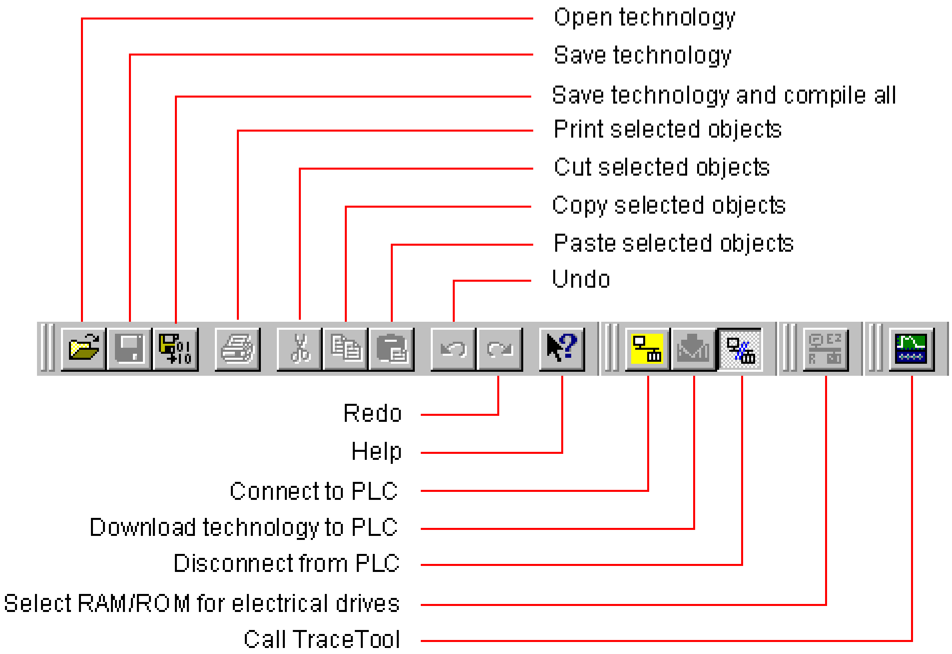
Menu bar

The menu bar contains the menus currently supported in S7T Config.

Toolbar

The toolbar icons provide quick access to frequently used and currently available menu commands by means of mouse click. If you position the cursor briefly on an icon, a tooltip for this function is displayed. The status bar displays additional information.

Grayed-out symbols indicate functions which cannot be executed in the current working state.



Status bar

The status bar shows context-sensitive information and the project status (offline or online mode).

Offline mode

Communication with the Technology CPU is down when S7T Config is operated in offline mode. In this mode, you can create new technology objects, configure drives using the integrated STARTER tool, and compile the technology data.

Online mode

S7T Config communication with the Technology CPU is up when operated in online mode. In this mode you can perform diagnostics operations, download new project data to the device, and edit parameters in the CPU.

The system checks technology data consistency when you change to online mode. This check verifies the availability or connection of the devices, drives and technology objects configured in the technology. If an inconsistency is found, or an error occurs (for example deviation of the existing configuration in the target system from the offline configuration in S7T Config), the corresponding objects are highlighted on a red background in the Navigator.

⚠ CAUTION
In order to ensure isochronous control of the drives, do not connect the programming device (PG/PC) to the DP(DRIVE) interface.

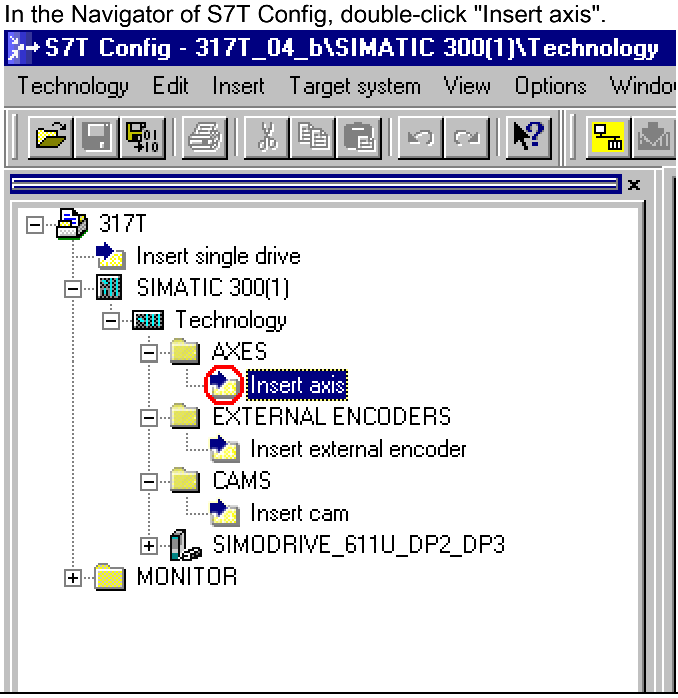
4.5 Configuring electrical axes

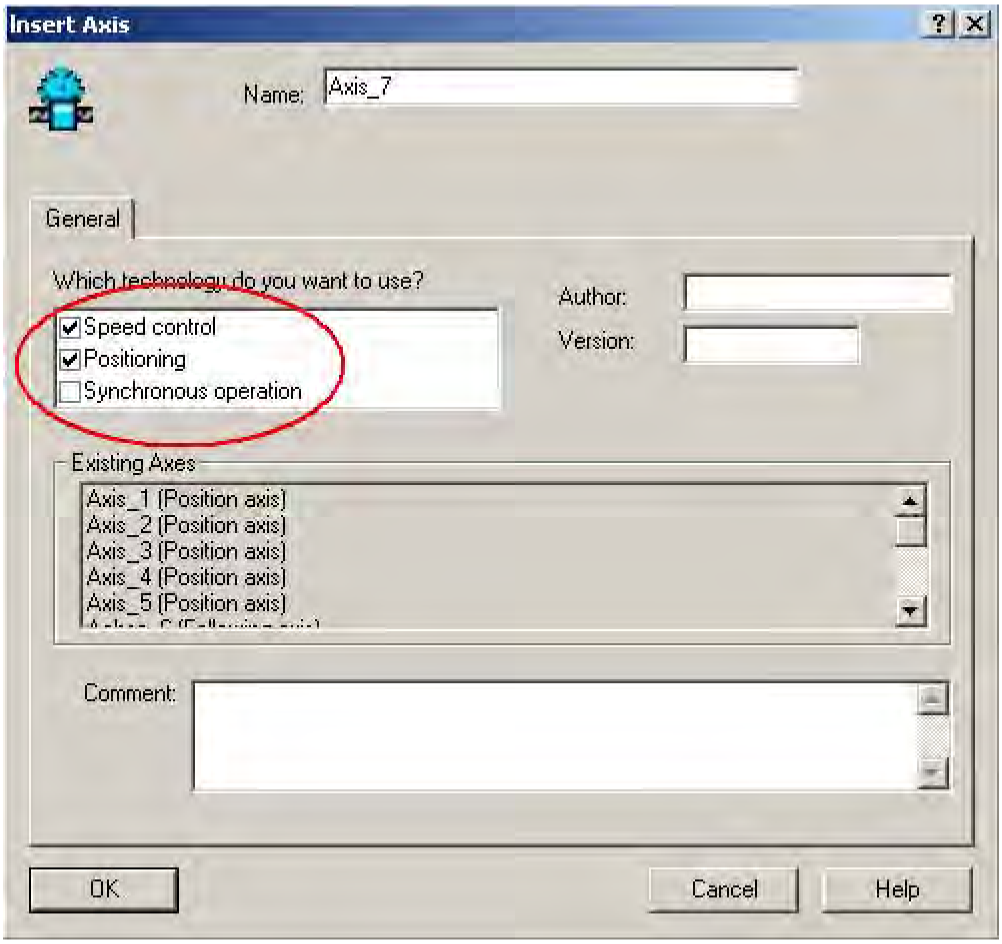
Prerequisite

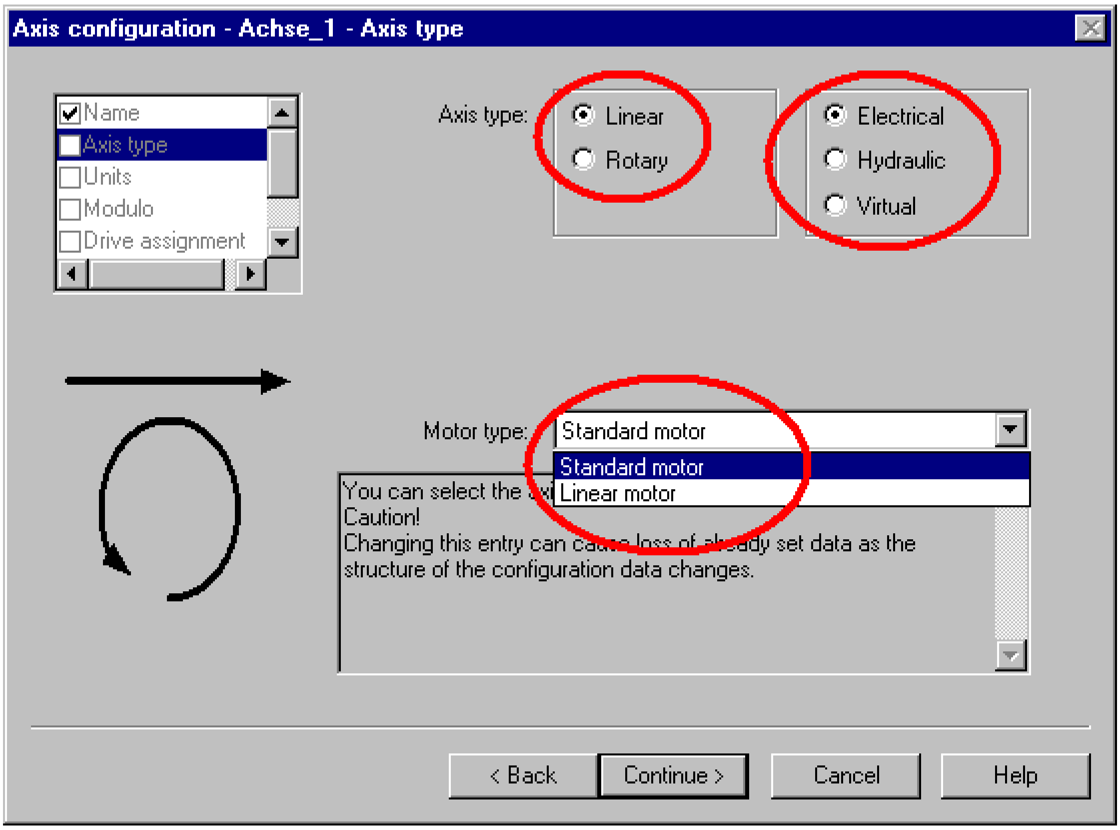
- The Technology CPU was configured in HW Config and configuration data were compiled.

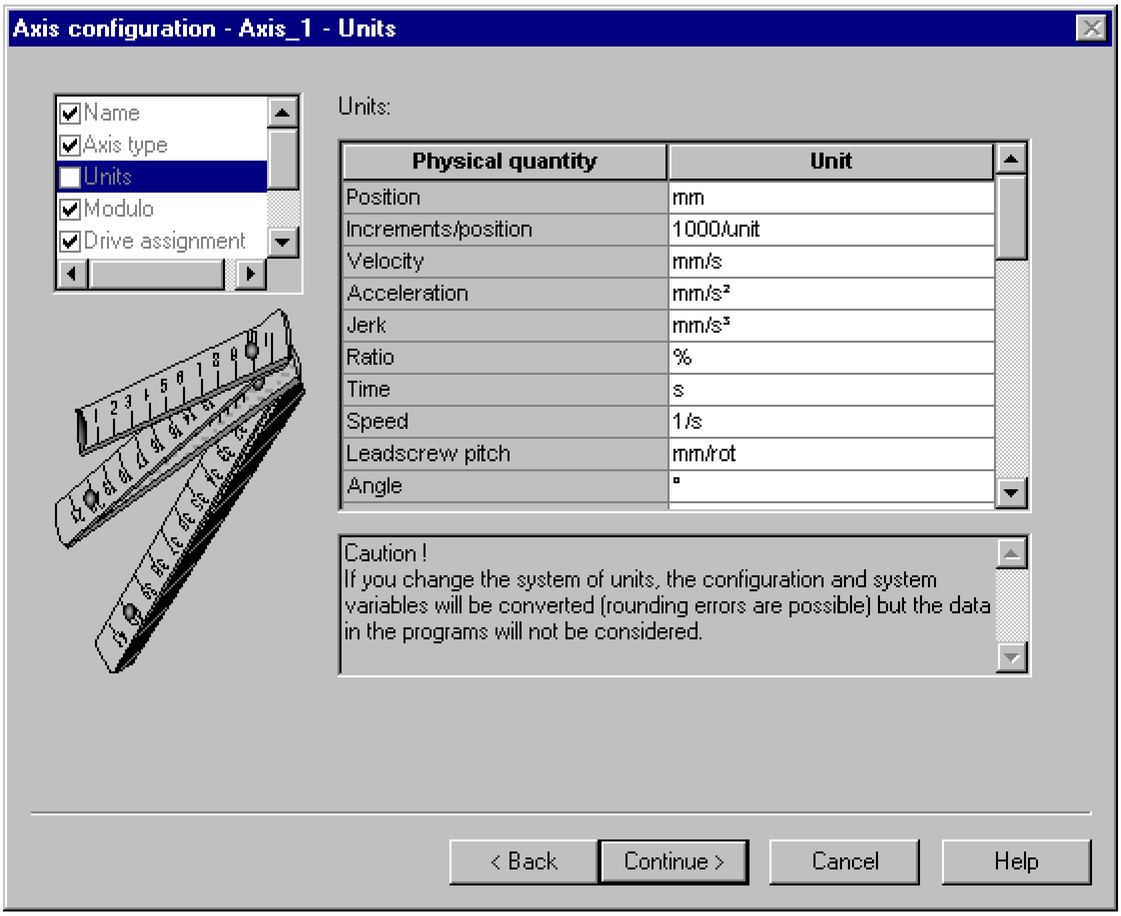
Inserting a new axis in S7T Config

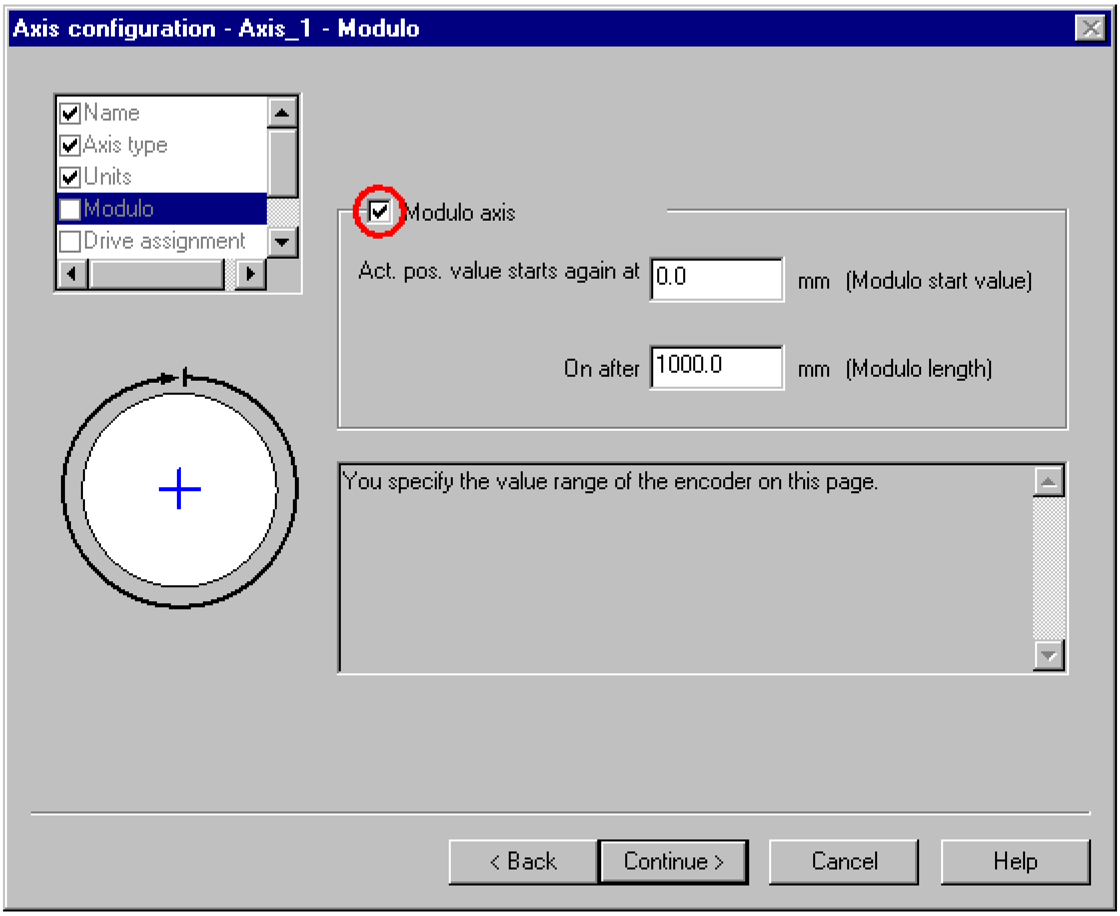
The steps below describe the basic procedure of inserting and configuring an axis, based on the example of a positioning axis.

Step	Description
1.	In the Navigator of S7T Config, double-click "Insert axis". 

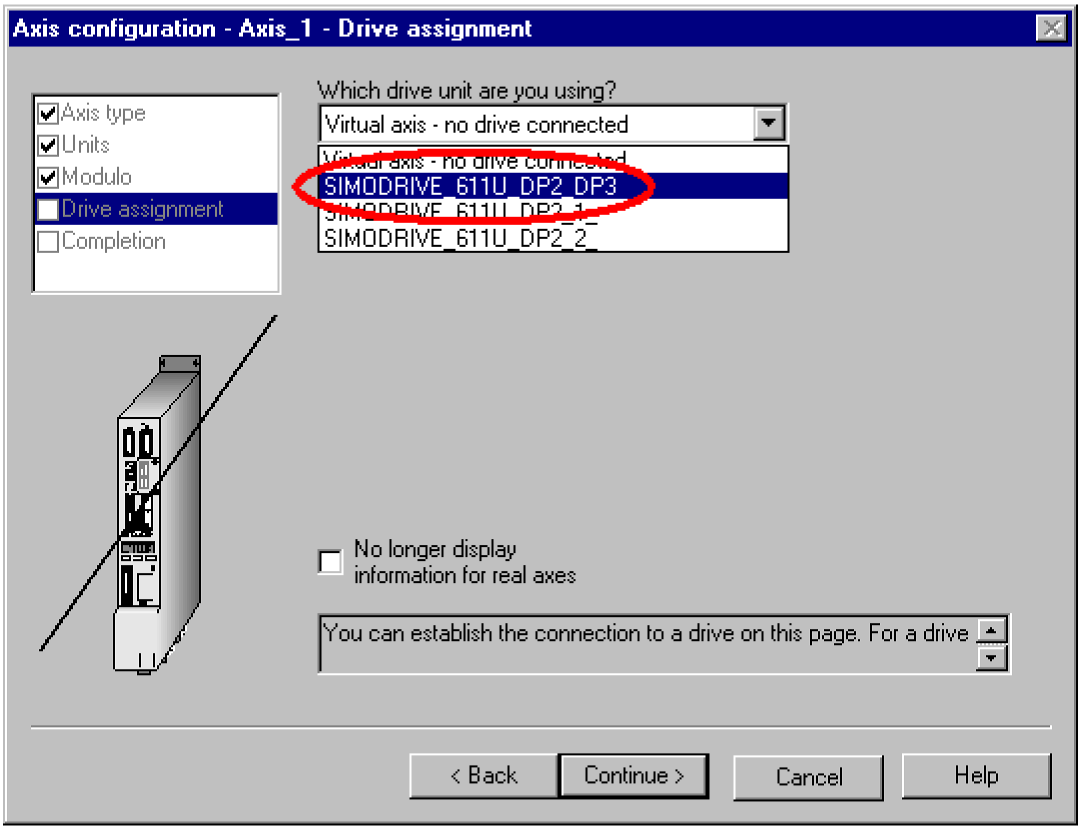
Step	Description
2.	Select the "Positioning" technology from the "General" tab, for example, for the configuration of a positioning axis. Disable the "Positioning" technology when you configure a speed-controlled axis.  You can edit the name, enter an author, define a version name, and enter a comment. "Technology Objects Management" applies the name of the technology object as symbolic name to the technology DB of the object in STEP 7.
3.	Click on the "OK" button. Result: The Axis Wizard appears. **Note** You define the axis technology (speed, positioning, synchronous operation) once when you configure a new axis. In order to change the axis technology at a later time, you must delete and recreate the axis.

Step	Description
4.	Select the axis type and the motor type from the next dialog box.  Click "Continue".

Step	Description
5.	Select the physical units for axis configuration and for the user program from the list in the next dialog box.  Note: The physical unit settings only apply to this axis. The list may also show physical units which are irrelevant to this axis due to the axis configuration. Example: The value "1000.0" with positioning unit in [mm] set at input parameter *Position* of the MC_MoveAbsolute" technology function is equivalent to a position value of 1000.0 mm. Caution: Numerical values set in the user programs (for motion commands, for example) are **not** converted into the new units if you change the unit settings. The values always refer to the configured unit! Click "Continue".

Step	Description
6.	If you intend to operate the axis in modulo mode, set the "Modulo axis" check box, and enter the required modulo length and start value.  Click "Continue".

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>7.</td><td>
On the next dialog box, select the drive component you want to connect to the axis.



The drop-down list shows all drives configured in HW Config. Drives already connected are no longer included in the drop-down list. The program only shows the "Synchronize Sinamics devices" button if a SINAMICS S120 drive is configured in HW Config.

Select the "virtual axis" drive component in order to allocate a logic control function to the axis in a synchronized group. The dialog box displays additional options if a real drive unit is selected.

Select the relevant "Axis" from the "Drive" box if the selected drive is a multiple axis module (for example, SIMODRIVE 611 universal double-axis module, IM 174/ADI4, SINAMICS S120).

The "log. HW addresses", "Input" and "Output" display boxes show the drive's logical addresses you set in the drive configuration in HW Config.

If you recreate the configuration, the default message frame type set in HW Config will be entered in the drop-down list box (see Message frame types and their functions (Page 113) for further information.) If the message frame set in HW Config is unknown or was edited by the user, so that it does not match any known standard message frames, the telegram wizard helps you to make the correct selection from the drop-down list.

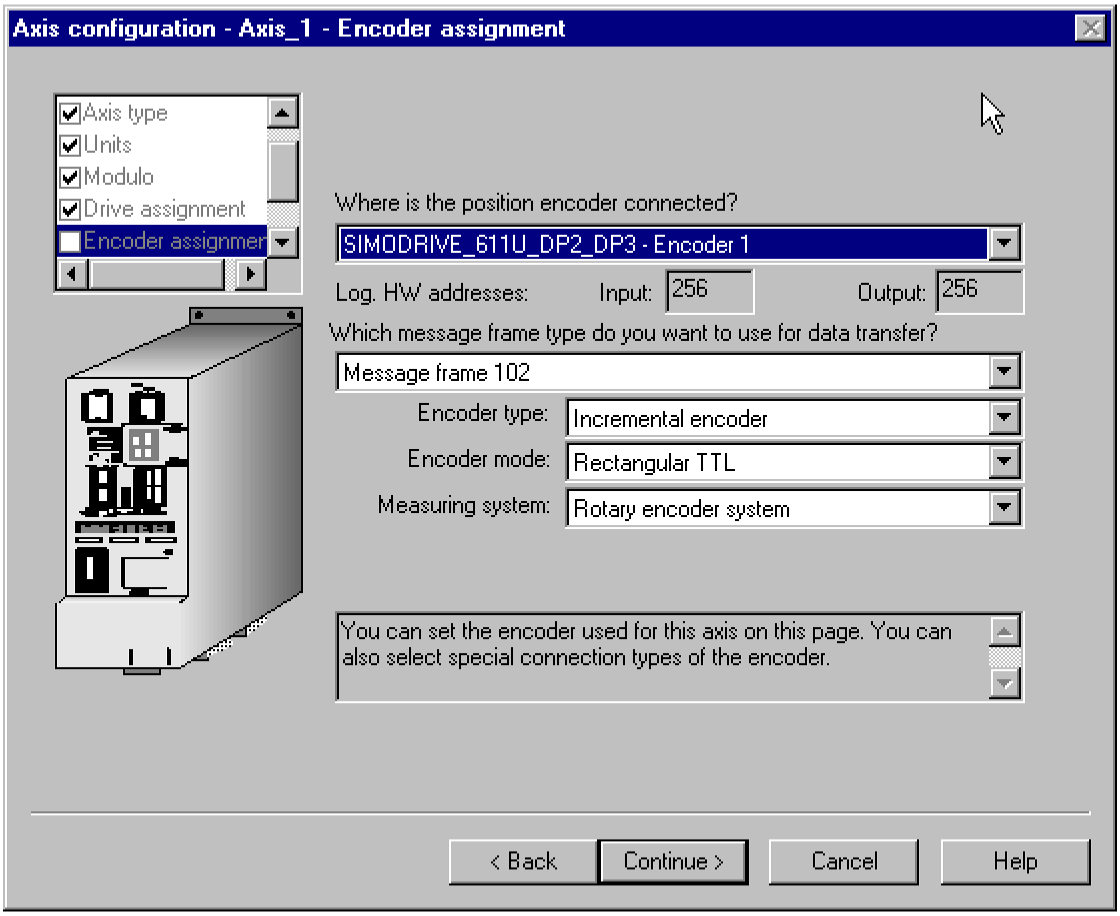
Enter the rated speed of the motor in the "Rated speed" input box. This value is relevant to speed assessment and must therefore correspond with the drive configuration.

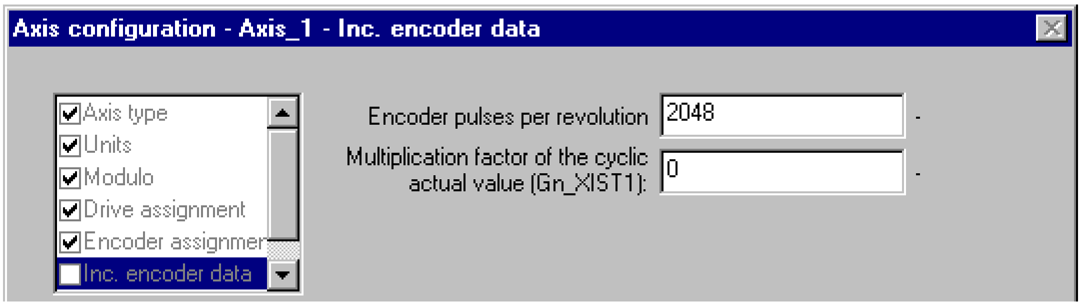
Click "Continue".

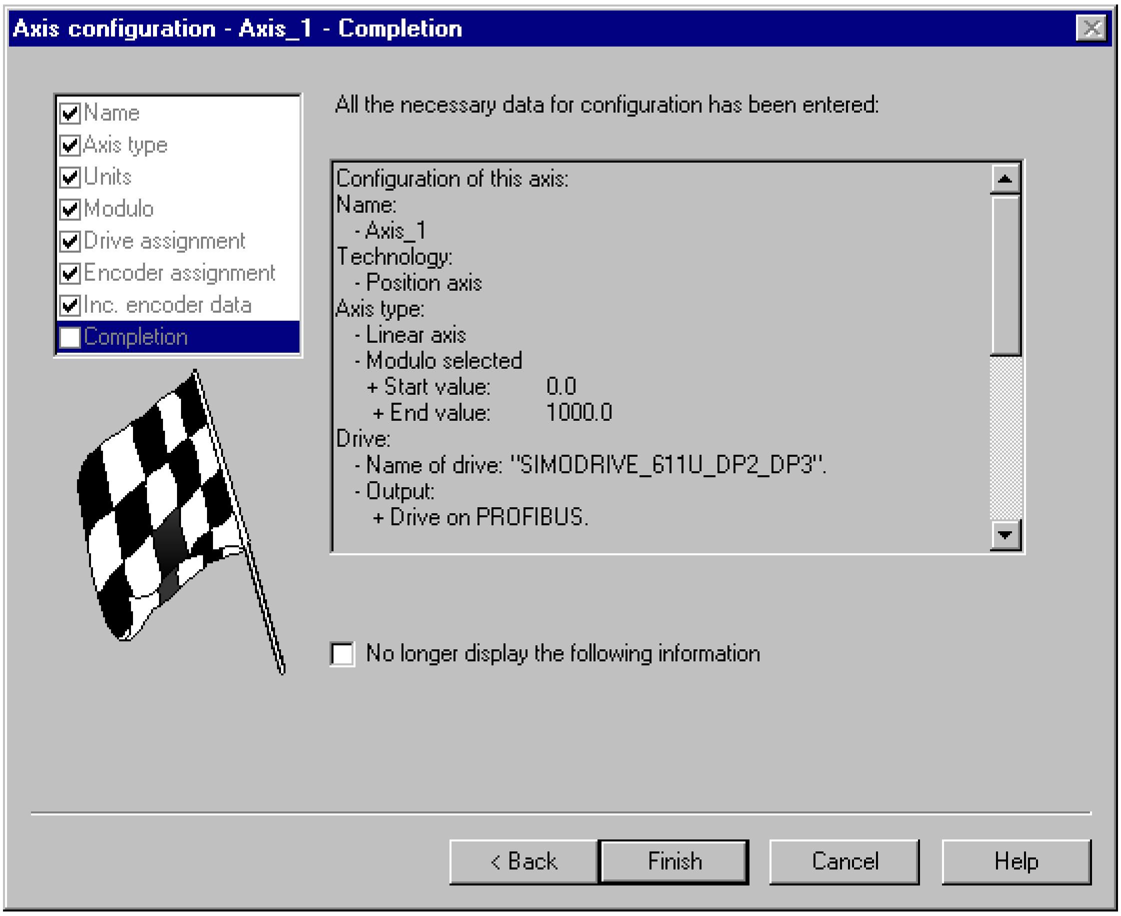
Result: The "Interaction axis - drive" dialog box opens. Confirm the dialog with "OK". The S7T Config Online Help opens. Close the Online Help and continue the axis configuration.

Note

- The message frame type you configured in S7T Config for the interface of the technology object for the drive component must match the message frame type of the component configured in HW Config.
- Ensure that the motor speed configured in the drive and the values entered in this dialog box are consistent. If a SINAMICS drive is used, this dialog box furthermore contains the "Data transfer from drive" button with which the values of the drive parameterization can be used.
</td></tr>
</table>

Step	Description
8.	Select the encoder of the drive from the next dialog box.  Select a default encoder which is to return its values to the CPU in the message frame configured in HW Config (usually, the encoder directly connected to the drive). You can select a different encoder for axis position detection from the list of configured encoders. Open the drop-down list to select the message frame type used to exchange encoder data. Default is the message frame configured in HW Config. Define the settings specified in the encoder documentation or in drive configuration data in the "Encoder type", "Encoder mode" and "Measuring system" drop-down lists. Click "Continue". **Note** Ensure that the encoder data configured in the drive and the values entered in this dialog box are consistent. If a SINAMICS drive is used, this dialog box furthermore contains the "Data transfer from drive" button with which the values of the drive parameterization can be used.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>9.</td><td>
In the next dialog box, define the encoder resolution and the multiplication factor of the cyclic actual value.
On the next dialog box, define the encoder resolution and the multiplication factor of the cyclic actual value.
For information on the encoder resolution, refer to the encoder documentation. Calibrate the multiplication factor of the cyclic actual value based on the values used in the configuration of the drive component.


Click "Continue" to continue the axis configuration.
</td></tr>
<tr><td>10.</td><td>
When completing the axis configuration, check the summary of your entries.


Exit the axis configuration dialog by clicking "Finish".
Result: The "Interaction axis - drive" dialog box opens. Confirm the dialog with "OK". You have successfully completed a positioning axis configuration with the help of the Axis Wizard.
</td></tr>
</table>

4.5.1 Adding a data record for data record changeover

Introduction

The "MC_ChangeDataset" technology function can be used to perform a data record changeover at an axis. You can create data records for speed-controlled axes, positioning axes and synchronization axes. Virtual axes always have only one data record.

The data record changeover function can be used, for example, to toggle between the motor encoder and the machine encoder, or to edit controller parameters.

Note

When editing data record parameters, note that certain parameters must be identical in the data records:

- Parameters changing the structure (different controller types such as PV or PID controllers, for example)
- Important activating parameters (following monitoring on/off, DSC on/off, for example)

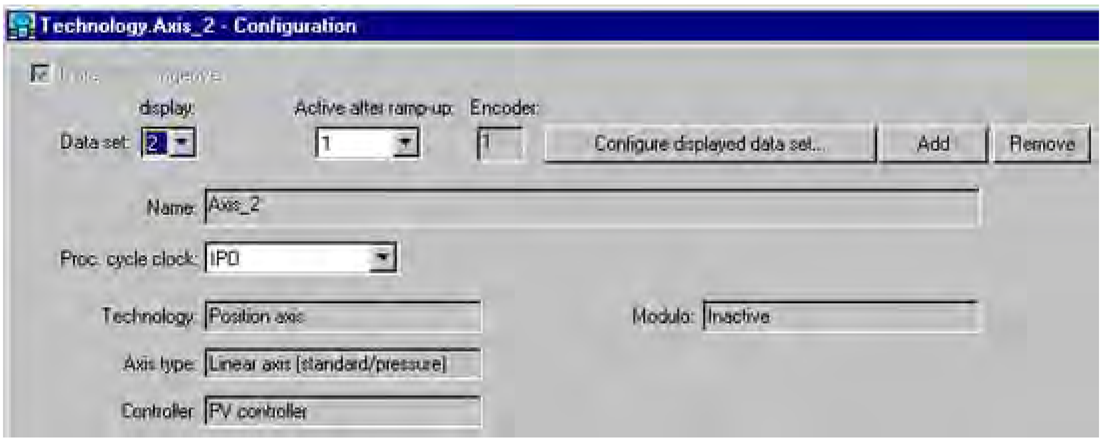
The consistency check reports any illegal technology parameter settings when you configure data records in S7T Config.

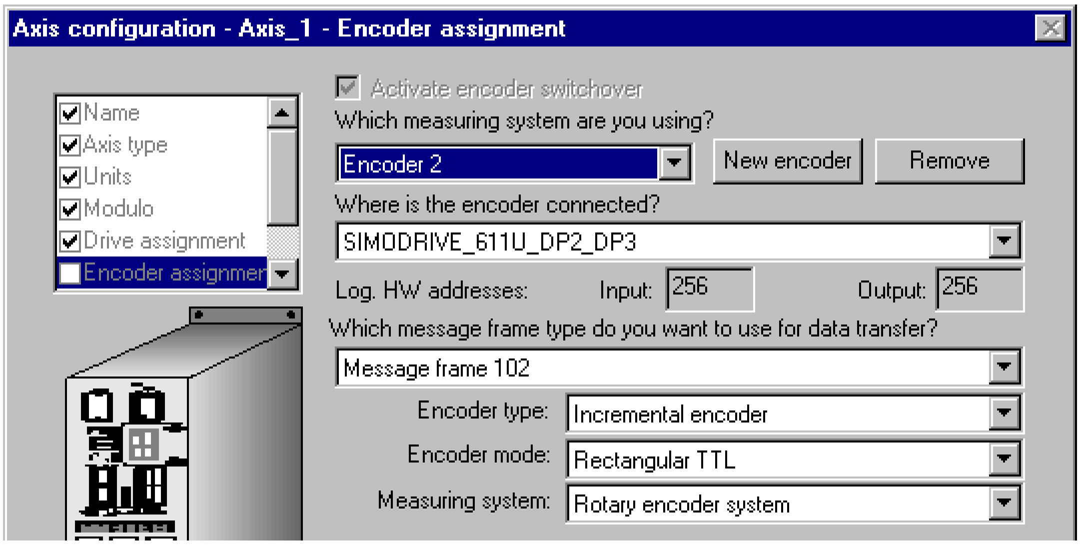
Prerequisite

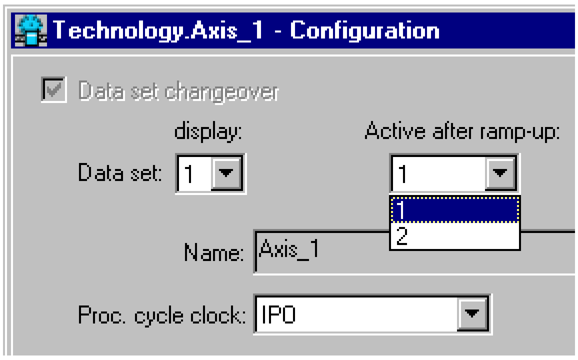
- The axis is configured.

Adding a new data record

To configure a new data record:

Step	Description
1.	In the Navigator of S7T Config, open the configuration dialog box of the axis to which you want to add a data record.
2.	Set the "data record changeover" check box. The check box is hidden if the axis already contains several data records.
3.	Click "Add". Result: A new data record is added to the axis. 
4.	Click "Configure active data set." Result: The Axis Wizard appears.
5.	Use the axis wizard to configure the data record. **Note**: The technology used is set once when you create the axis. You can not use a second data record to change the axis technology.
6.	You may also perform an encoder changeover using the data record changeover function. Do so by setting the "enable encoder changeover" check box on the "Axis configuration - Encoder assignment" dialog box. Ignore this step and the next otherwise. Click "new encoder" to create the second encoder. Result: A drop-down list opens from which you can select an encoder for this data record.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>7.</td><td>Select the connection and message frame type for the second encoder. In the final step, enter the encoder information ("Encoder type", "Encoder mode" and "Measuring system").
For the encoder configuration select settings from the documentation of the encoder or configuration settings of the drive components to which the encoder is connected.


Click "Next". Continue by editing the steps in the wizard to complete the configuration.</td></tr>
<tr><td>8.</td><td>In the axis configuration dialog box, select the data record to be loaded after CPU startup ("active after startup").


Result: The "Encoder" table shows an overview of the configured encoders of the axis.</td></tr>
</table>

Deleting the data record of an axis

You start to delete a data record by selecting it from the axis configuration dialog box. Next, click "Delete". The data record is deleted.

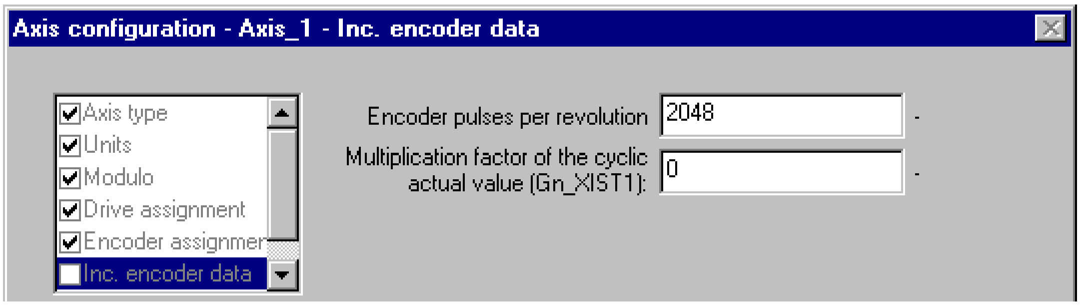
Selecting a data record

Select the data record to be edited from the "Display data record" drop-down list. This selection is available in all dialog boxes of the axis.

Select the data record which is used by default after startup from the "Data record active after startup".

4.5.2 Fine resolution

Encoder pulses can be evaluated in the drive at a higher resolution, depending on the encoder registration and type. The configuration of signal evaluation in the drive must be interconnected with the "Multiplication factor of the cyclic actual value" parameter in the axis configuration in S7T Config.



Example

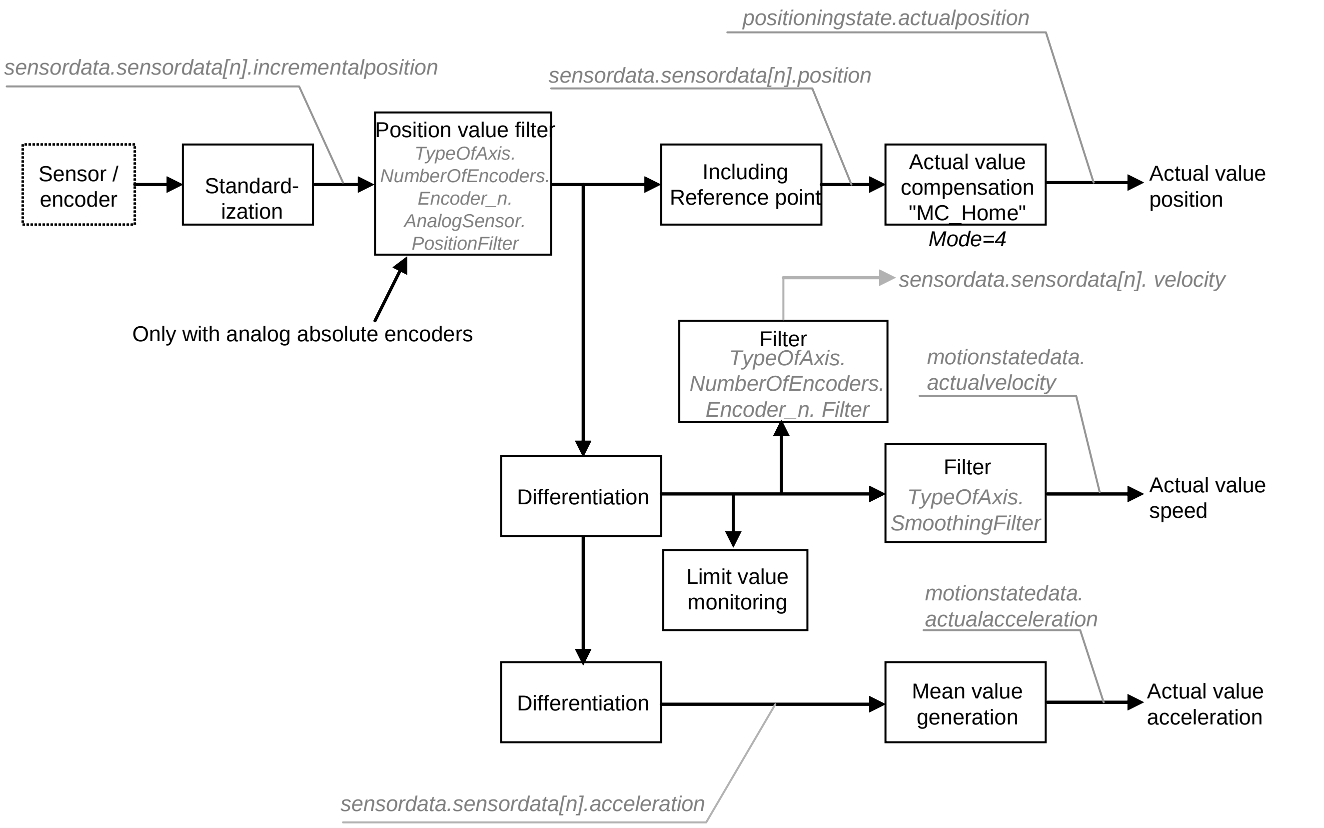
An incremental encoder is connected to SIMODRIVE 611U and returns 2048 p/rev (encoder resolution). By default, the drive multiplies this number of pulses by the factor of 2048 (2 to the power of 11). This factor must be entered at the "Multiplication factor of the cyclic actual value" parameter.

Default

The "Multiplication factor of the cyclic actual value" is set to *0* by default. The system automatically uses multiplication factor 2048 (2exp11).

4.5.3 Actual value logging

The diagram below shows the preparation of the actual value of an axis or external encoder in the Technology CPU.



Data are logged within the position control cycle. These data can be used to calculate further variables such as velocity and acceleration.

The system variables *sensordata.sensordata[n].incrementalposition* and *sensordata.sensordata[n].position* are calculated in the position control cycle clock. All other system variables are calculated in the corresponding execution cycle clock of the axis or external encoder. For external encoders, the system variables *sensordata.sensordata[n]....* are called *sensordata... .*

You use the *TypeOfAxis.NumberOfEncoders.Encoder_n.AnalogSensor.PositionFilter* configuration data element to activate or deactivate the actual position value. The filter is only available for analog absolute encoders:

Configuration data element	Significance	
TypeOfAxis.NumberOfEncoders.Encoder_n. AnalogSensor.PositionFilter.enable	*YES*	Actual position value activated
	NO	Actual position value deactivated
TypeOfAxis.NumberOfEncoders.Encoder_n. AnalogSensor.PositionFilter.timeConstant	Time constant for PT1 smoothing.	

The "sensor" velocity filter can be activated/deactivated by means of the *TypeOfAxis.NumberOfEncoders.Encoder_n.Filter* configuration data element:

Configuration data element	Significance	
TypeOfAxis.NumberOfEncoders.Encoder_n. Filter.enable	*YES*	Actual value filter activated
	NO	Actual value filter deactivated
TypeOfAxis.NumberOfEncoders.Encoder_n. Filter.timeConstant	Time constant for PT1 smoothing.	

(For external encoders, the configuration data element is called *TypeOfAxis.Encoder_1.Filter*... .

The actual velocity value filter can be activated/deactivated by means of the *TypeOfAxis.SmoothingFilter* configuration data element:

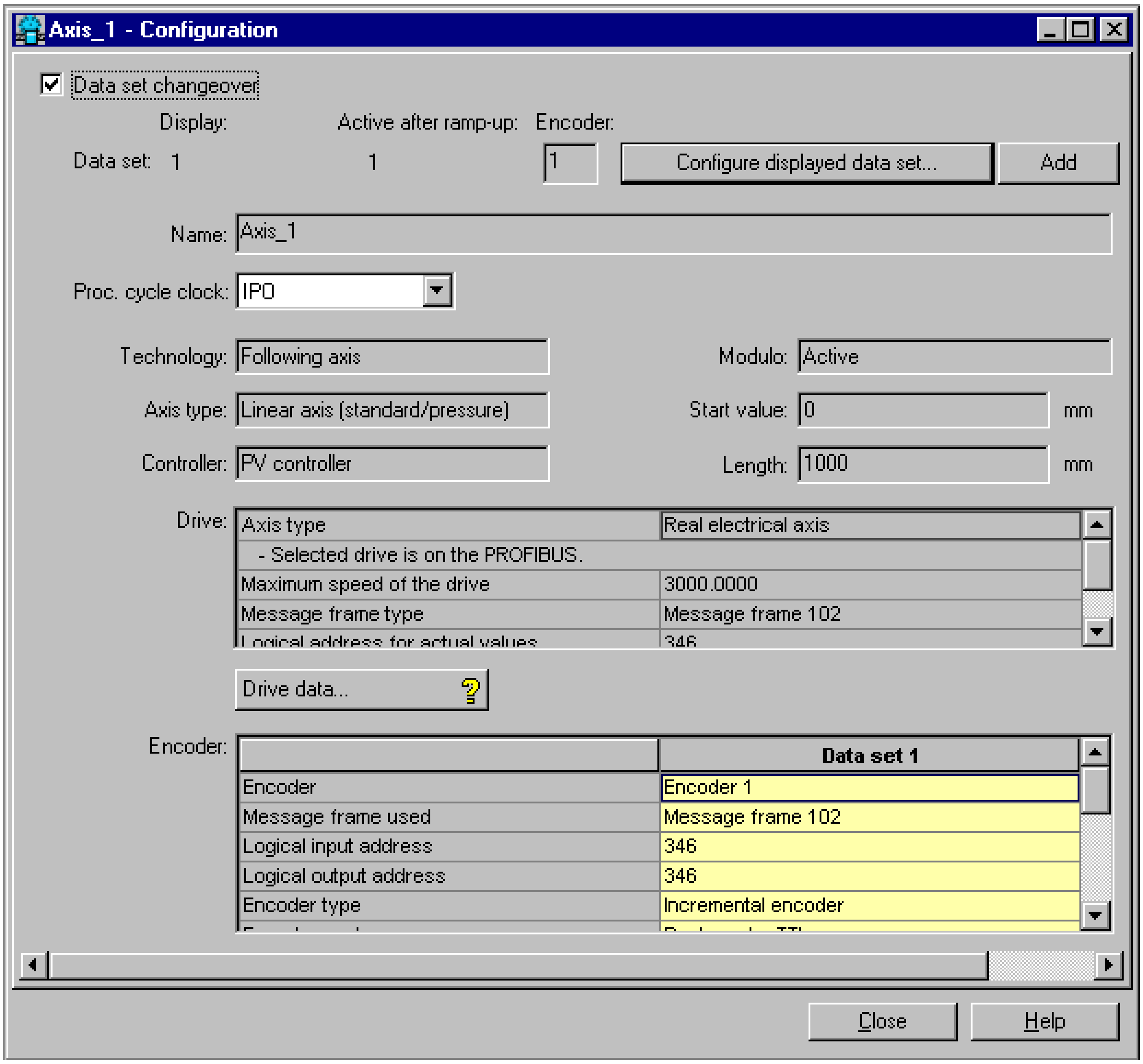
Configuration data element	Significance	
TypeOfAxis.SmoothingFilter.enable	*YES*	Actual value filter activated
	NO	Actual value filter deactivated
TypeOfAxis.SmoothingFilter.mode	You can set the filter calculation method at this configuration parameter:	
	DEFAULT_MODE	Averaging as a function of the ratio: Execution cycle of the axis / external encoder to the position control cycle Example: A *3 : 1* ratio is set between the execution cycle and the position control cycle. In this case, the mean value is formed in three position controller cycles.
	AVERAGING	Averaging using a time constant
	PT	PT1 smoothing using a time constant
TypeOfAxis.SmoothingFilter.timeConstant	Time constant for PT1 smoothing or "*AVERAGING*".	

4.5.4 Configuration

4.5.4.1 Configuration - Axis

The **Axis > Configuration** dialog box shows the axis and drive settings

You can edit these settings in a dialog box by clicking "Configure active data set".



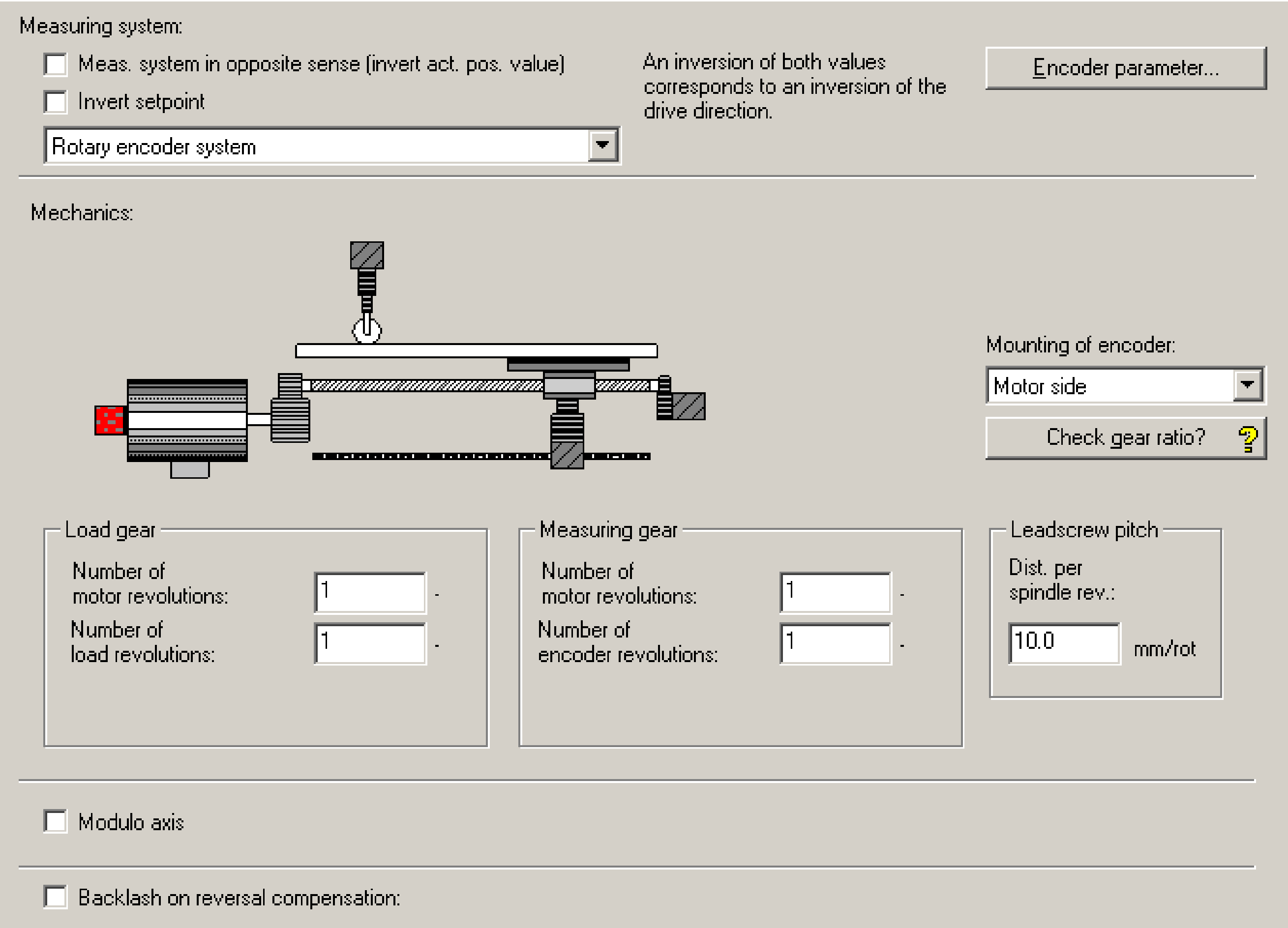
This dialog box also contains functions for managing the data records for a data record changeover. If the technology object supports multiple data records, you can:

- Create new data records
- Delete data records
- Define which data record to load for the technology object during CPU startup, and
- Reconfigure the selected data record.

4.5.5 Mechanical system

4.5.5.1 Mechanics - Electrical axis

The **Axis > Mechanics** dialog box is used to configure the mechanical parameters of the axis and of the encoder.



Settings for the axis:

- Drive direction settings
- Load gear settings
- Leadscrew pitch of a linear axis
- Modulo settings
- Backlash compensation settings

Settings for the encoder:

- Measuring gear settings
- Backlash compensation
- Encoder mounting type

Conditions of determining gear parameters

The configuration may return error messages indicating incompatible gear parameters. The selection of incompatible configuration data may lead to internal overflows in the integrated technology. Formulae used to determine suitable parameters:

f_1 = Numerator measuring gear x 360 x internal resolution x denominator load gear

f_2 = Denominator measuring gear x encoder resolution x actual value factor x numerator load gear

From f_1 and f_2, the greatest common divisor **k** must be determined and used in the following formula:

$f_{11} = (f_1/k) < 2^{32}$

$f_{22} = (f_2/k) < 2^{32}$

The results of f_{11} and f_{22} must be less than 2^{32}. If this is not the case, check whether an appropriate modification of the parameters in the f_1 and f_2 formulas will produce values for f_{11} and f_{22} that do not exceed the maximum permissible value.

Check the following value if f_{11} and f_{22} meets the requirements described above and a configuration error is detected nonetheless:

f_3 = Modulo length x internal resolution

Now you can calculate the greatest common divisor k_2 of f_3 and f_{11} and insert it in the following formula:

$f_{31} = ((f_3 \times f_{22} \times f_{11}) / (k_2 \times k_2)) < 2^{63}$

At this point you have to check whether f_{31} is less than 2^{63}. If not, check whether you can reduce modulo length. You can also make changes in the parameters in the f_1 and f_2 formulas, as long as you ensure that the requirements for f_{11} and f_{22} are still met.

Table 4-3 Parameter description

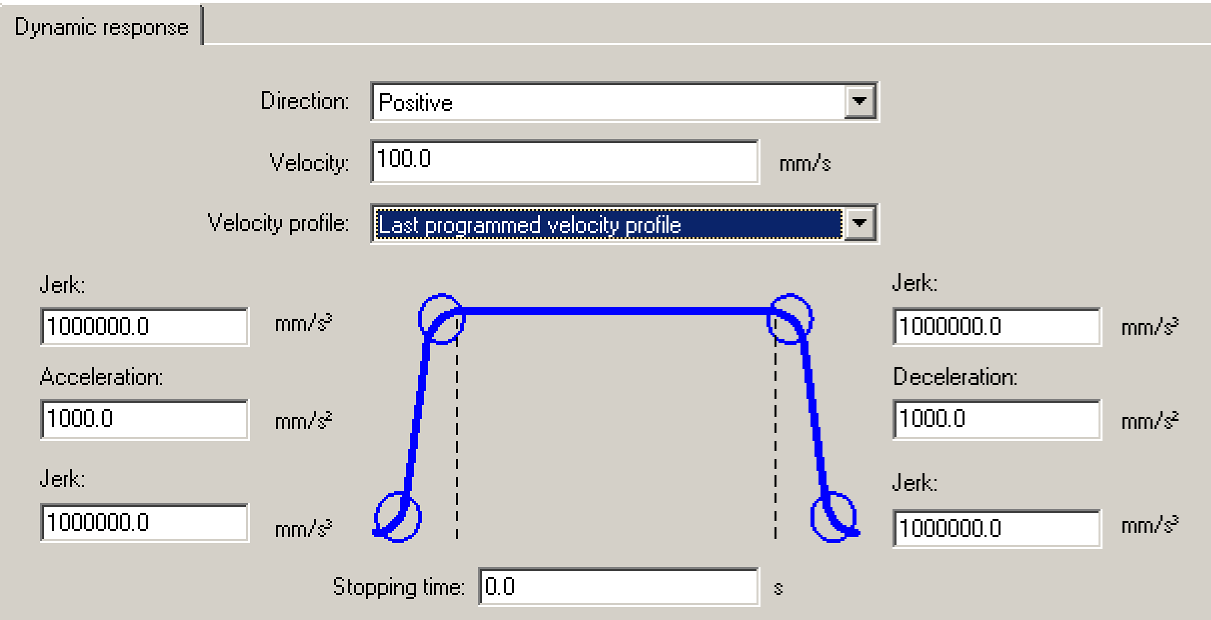
Parameters	Comment	Configuration parameter at the axis
Measuring gear numerator	Numerator of the measuring gear ratio	AdaptDrive.numFactor AdaptExtern.numFactor AdaptLoad.numFactor
Measuring gear denominator	Denominator of the measuring gear ratio	AdaptDrive.denFactor AdaptExtern.denFactor AdaptLoad.denFactor
Load gear numerator	Load gear ratio numerator	Gear.numFactor
Load gear denominator	Load gear ratio denominator	Gear.denFactor
Internal resolution	Internal increments / position unit	Defined in the configuration using the axis wizard.
Actual value coefficient	= X for DP drive	X for absolute value encoders: IncEncoder.incResolution MultiplierCyclic X for absolute value encoders: AbsEncoder.absResolution. MultiplierCyclic

A calculation tool is available in a contribution (FAQ) to this topic on the Internet at http://www.siemens.com/automation/service&support.

4.5.6 Default

4.5.6.1 Default - "Dynamics" tab

The default values for axis dynamics can be set in the "Dynamics" tab of the **Axis > Default** dialog box. The default values are activated if a negative dynamic value is set at a technology function.



Default values supported:

- Direction
- Velocity
- Acceleration
- Deceleration
- Jerk
- Velocity profile

Stop time

The time set at "Stop time" is in effect when a moving axis is disabled, and if *Stopmode* = *2* is set at "MC_Power".

Velocity profile

The velocity profile defines the response of the axis during startup and when braking, and to velocity changes.

The technology functions do not use the default value. If you want to change the velocity profile, use the *Jerk* input parameter.

You can choose between the following profiles:

- Trapezoidal

  The trapezoidal profile is used for linear acceleration in positive and negative direction of the motion (jerk = 0).

- Constant

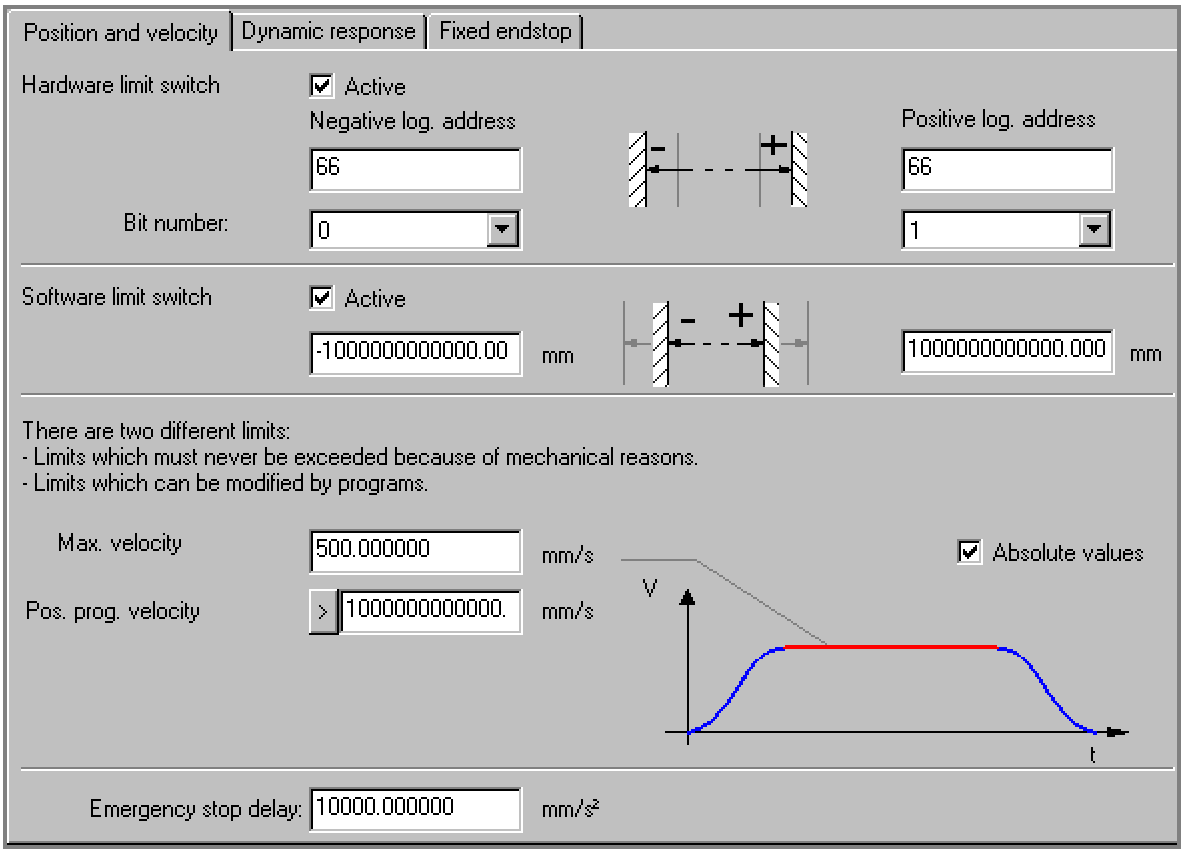
  The profile shows a constant acceleration, the jerk profile is controllable (jerk <> 0).

4.5.7 Limits

4.5.7.1 Limits - "Position and velocity" tab

In the "Position and velocity" tab of the **Axis > Limits** tab you can

- Configure and enable monitoring of the hardware limit switches (Page 147)
- Configure and enable monitoring of the software limit switches (Page 149)
- Set velocity limits
- Set emergency-off deceleration



Emergency off deceleration

The value set for emergency off deceleration is activated when a moving axis is disabled and *Stopmode* = 0 is set at the "MC_Power" parameter.

4.5.7.2 Hardware limit switches

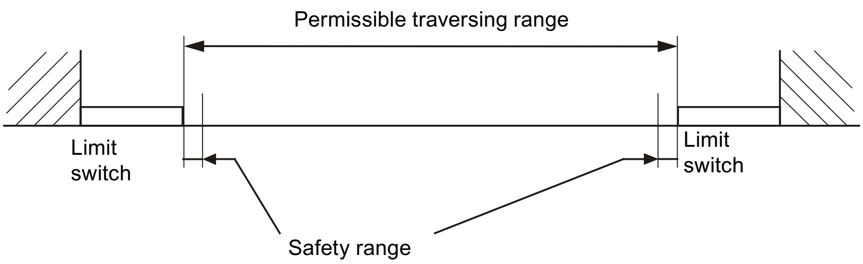
You enable hardware limit switch monitoring in the **Axis > Limits** dialog box, "Position and Velocity (Page 146)" tab. Hardware limit switch monitoring is used to limit the operating range of an axis, or to protect the machine.

Wiring

The HW limit switches can be wired to the four integrated digital inputs of the Technology CPU, or to I/O modules such ET 200 or SINAMICS S120 with TM15/TM17 which are operated on DP(DRIVE).

Traversing range

The permitted traversing range is monitored at the digital inputs of the hardware limit switches.



NOTICE
The hardware limit switch must be implemented as an NC contact. The hardware limit switches must remain active after the axis has passed the permissible traversing range until the mechanical end position is reached.

Retraction

An axis triggering a HW limit switch is stopped with error messages *8013* and *804B*. It is retracted from the HW limit switch (release motion) as described below:

- **Manual retraction**

  The axis is returned manually to the permissible traversing range. The error **at the technology DB** can only be acknowledged after it has been returned to this range.

- **Retraction with drive**

  The error at the technology DB of the axis is acknowledged, but the error message and the *LimitSwitchActive* bit remain active. The axis can now be returned to the permitted traversing range. A reverse motion command once again triggers an axis error. The error messages and the *LimitSwitchActive* status can be acknowledged after the axis has moved out of the range of the limit switch.

The current position of the axis is saved upon reaching the HW limit switch is saved. Only after this position, plus a safety range, has been passed is the axis considered to have left the limit switch.

NOTICE
The controller may **not** be switched off after the axis has passed the HW limit switch in order to avoid a conflict between polarity monitoring of the hardware limit switches and the overrun monitoring of the hardware limit switches in direction of the valid range. In this case, the axis is moved into the valid range without hardware limit switch monitoring, and is then re-enabled. When the controller is **switched on**, the axis must be positioned within the valid traversing range.

Internal states are lost and the configuration is reloaded when the axis passes the hardware limit switch. Reloading without loss of the approach information is only possible within the valid range.

Exception: Deactivation of position limit monitoring after a polarity reversal error

Safety range

The safety range of the HW limit switches is calculated based on the configured resolution of the system of units of the axis.

Safety range = 1000 / (increments / position)

Example: A linear axis is assigned the position unit "mm" and a resolution (increments / position) of "1000/unit" in the "Configure units" dialog box, meaning that the axis position is calculated to an accuracy of 0.001 mm. The safety range in this example is a multiple of 1000 of the accuracy: 1 mm.

4.5.7.3 Software limit switch

You can configure the software limits switches and enable the monitoring function in the **Axis > limits** dialog box, "Position and Velocity (Page 146)" tab. If the software limit switches are activated, the traversing distance of the axis is limited with the software limit switches.

Software limit switches should lie within the range of the HW limit switches with reference to the traversing range in order to limit the working range of an axis, for example.

Monitoring software the software limit switches at start of motion:

- **Check box activated**
  If the target position of a position-controlled motion command lies beyond the software limit switch, the warning *0026* is already displayed in the technology DB of the axis when the motion is started. In S7T Config the interrupt *40105* is signaled.
  The axis travels up to the position of the software limit switch and the error *8014* is reported in the technology DB.
- **Check box not activated**
  If the target position of a position-controlled motion command lies beyond the software limit switch, the axis traverses until the pPosition of the software limit switch is reached. Error *8014* is reported in the technology DB (Warning *0026* is not output).

Behavior for travel to software limit switch:

- **For position-controlled traversing:**
  The axis traverses until the software limit switch is reached. After the software limit switch has been reached, the respectively active position controlled or speed-controlled mode remains.
- **In all operating modes:**
  The axis traverses until the software limit switch is reached. After the software limit switch has been reached, an active position controlled mode remains. In speed-controlled mode the axis changes to position-controlled mode.

Negative position / positive end position:

Enter the positions for the negative and the positive end positions of the software limit switches in these input fields.

Tolerance window for retraction:

Enter a suitable value in this input field in order to prevent renewed triggering of the software limit switch error during retraction.

Note

The response of software limit switches is determined by the axis configuration in terms of the homing function. The limit switches of a configuration requiring a homed axis for the execution of absolute motion commands ("Homing required: Yes" in the **Axis > homing dialog box**) are not monitored if the axis is not homed. By contrast, if the execution of absolute motion commands is allowed when the axis is not homed ("Homing required: No" in the **Axis > Homing dialog box**) the software limit switches are monitored, regardless of the homing state of the axis.

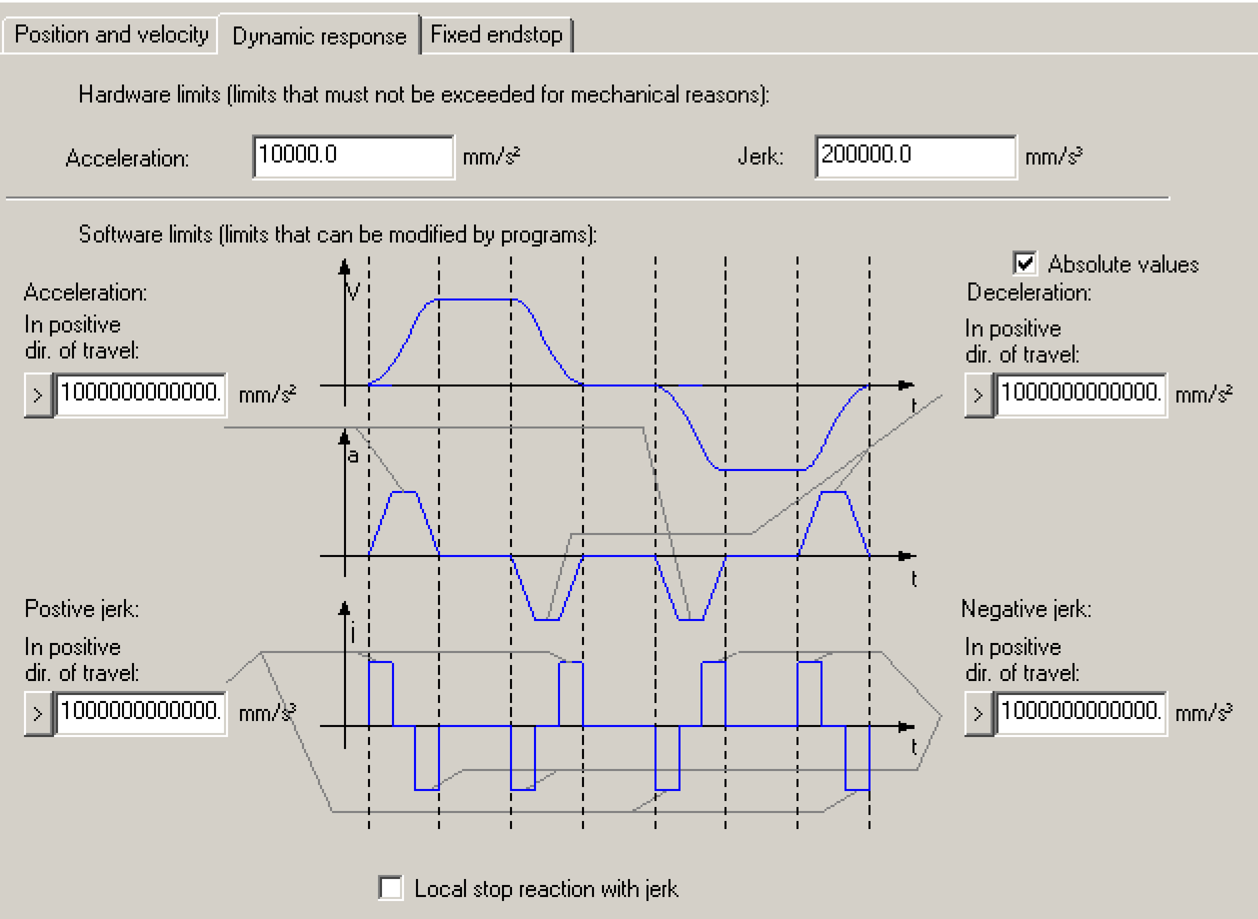
4.5.7.4 Limits - "Dynamic response" tab

You can adjust the HW limits (mechanical) and SW limits in the "Dynamic response" tab of the **Axis > Limits** dialog box.

The set deceleration limit is activated when a moving axis is disabled and *FastStop* = *1* is set at the "MC_Power" parameter.

The **status-dependent** acceleration model is enabled (default) by setting the "Absolute values" check box.

The **direction-dependent** acceleration model is enabled by resetting the "Absolute values" check box. Additional parameters are displayed in this case.



Status-dependent acceleration

- **Acceleration**

  Axis acceleration, independent of the direction of movement

- **Deceleration**

  Axis deceleration, independent of the direction of movement

Direction-dependent acceleration

- **Acceleration**

  Acceleration in the positive motion direction and deceleration in the negative motion direction

- **Deceleration**

  Acceleration in the negative motion direction and deceleration in the positive motion direction

Parameter settings with dynamic direction vector are useful, for example, for suspended axes.

Local stop reaction with jerk

Set the "Local stop reactions with jerk" check box to execute a stop reaction triggered by an alarm reaction at the axis, with jerk limiting and rounding.

Emergency-stop deceleration

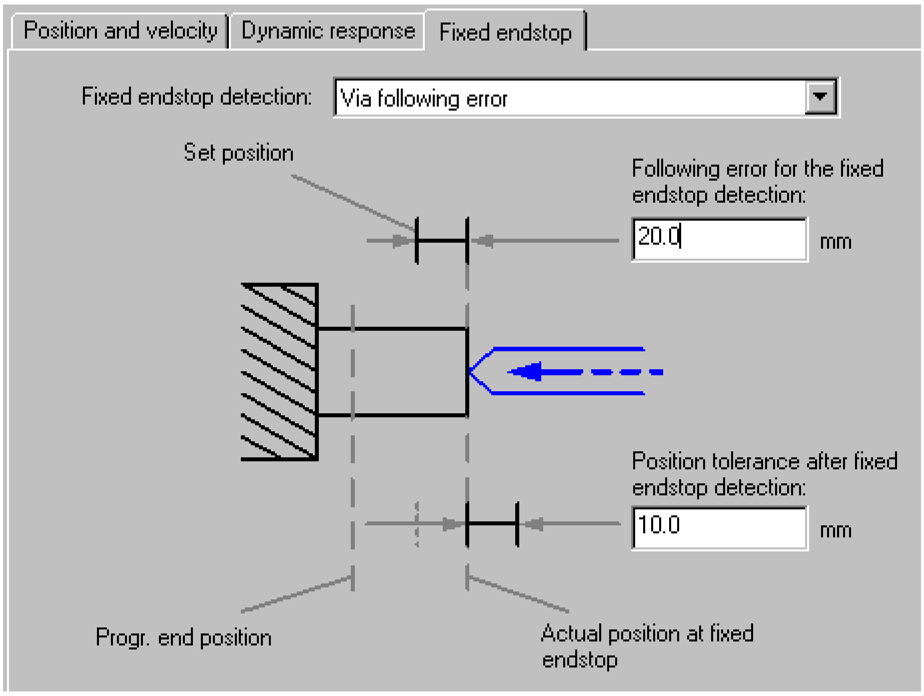
The set emergency-stop deceleration comes into effect when a moving axis is disabled and *Stopmode = 0* is set at "MC_Power."

4.5.7.5 Limits - "Fixed end stop" tab

The "Fixed end stop" tab of the **Axis > Limits** can be used to enable fixed end stop detection and to set the corresponding detection mode:

- Use following error
- Using the force/torque

When the selected condition is met, the "fixed end stop" status is reached. Following error monitoring is disabled if "Move to fixed end stop" is enabled.



Use following error

Note

When the axis moves onto the fixed end stop, and fixed end stop detection with "use following error" is set, the entry in "Position tolerance after fixed end stop detection" should be significantly less than that of "Following error for fixed end stop detection".

Use torque value

The fixed end stop detection function "use torque value" requires that the digital drive assigned to the axis supports torque limiting, and that a corresponding message frame, for example, *message frame 102* or *message frame 105*.

Move to fixed end stop

The "MC_MoveToEndPos function activates the "Move to end stop" function and sets the clamping torque after the end stop is reached. This operation is also known as "clamping."

The motion is stopped when the axis reaches the fixed end stop and the control remains active. The setpoint value at the position controller input is held constant. New motion commands in direction of the clamping position are canceled; new motion control commands in retraction direction are executed to reduce torque. The position setpoint of the axis is used as start position of new motion control commands in retraction direction.

The position setpoint of the axis results from one of the following equations, depending on the fixed end stop detection function:

- "Use following error"
  Position at fixed end stop + following error
- "Use torque value"
  Position at fixed end stop + clamping tolerance

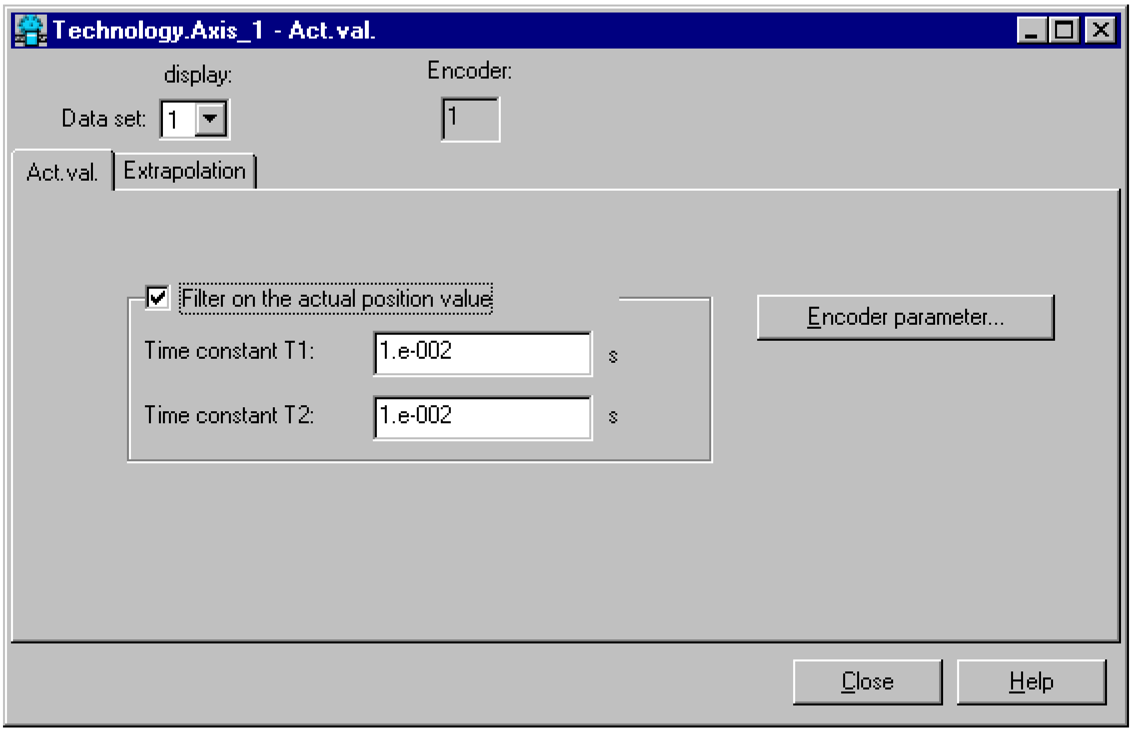
Conditions for the "Fixed end stop detection" function

- The "Move to fixed end stop" function is reset when the axis moves out of the clamping tolerance window.
- A new command can also be output to toggle the direction of torque when clamping is active.
- Non-stepped torque transitions and torque retention over a defined time period can be implemented in the user program, as can definitions of torque profiles.
- Moving to the fixed end stop (clamping) can be disabled by setting a reverse positioning command.
- A reversal command MC_MoveToEndPos is not permitted and is ignored.
- A mechanical break of the end stop is monitored by means of the actual value of the axis (clamping tolerance window monitoring).
- The torque limit of the drive is set in [N/m] at the *Torque* parameter of the MC_MoveToEndPos technology function.
- If the command is active and the fixed end stop is not detected, the system reacts as with active torque limiting.

4.5.8 Actual value

4.5.8.1 Actual value - "Actual value" tab

Activate actual value filtering in the dialog **Axis / External encoder > Actual value** on the "Actual value" tab and set the appropriate time constants.



Filter on the actual position value

Activate the check box if you would like to activate the filtering of the actual position value.

Time constant T1

Here, you set time constant T1 of the PT2 position filter in the actual value system.

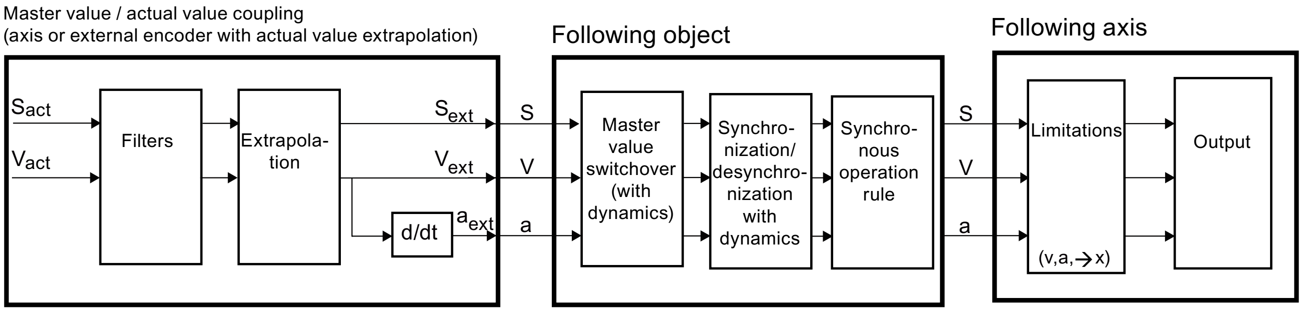
Time constant T2

Here, you set time constant T2 of the PT2 position filter in the actual value system.

"Encoder parameter" button

Opens a dialog box in which the encoder data are displayed.

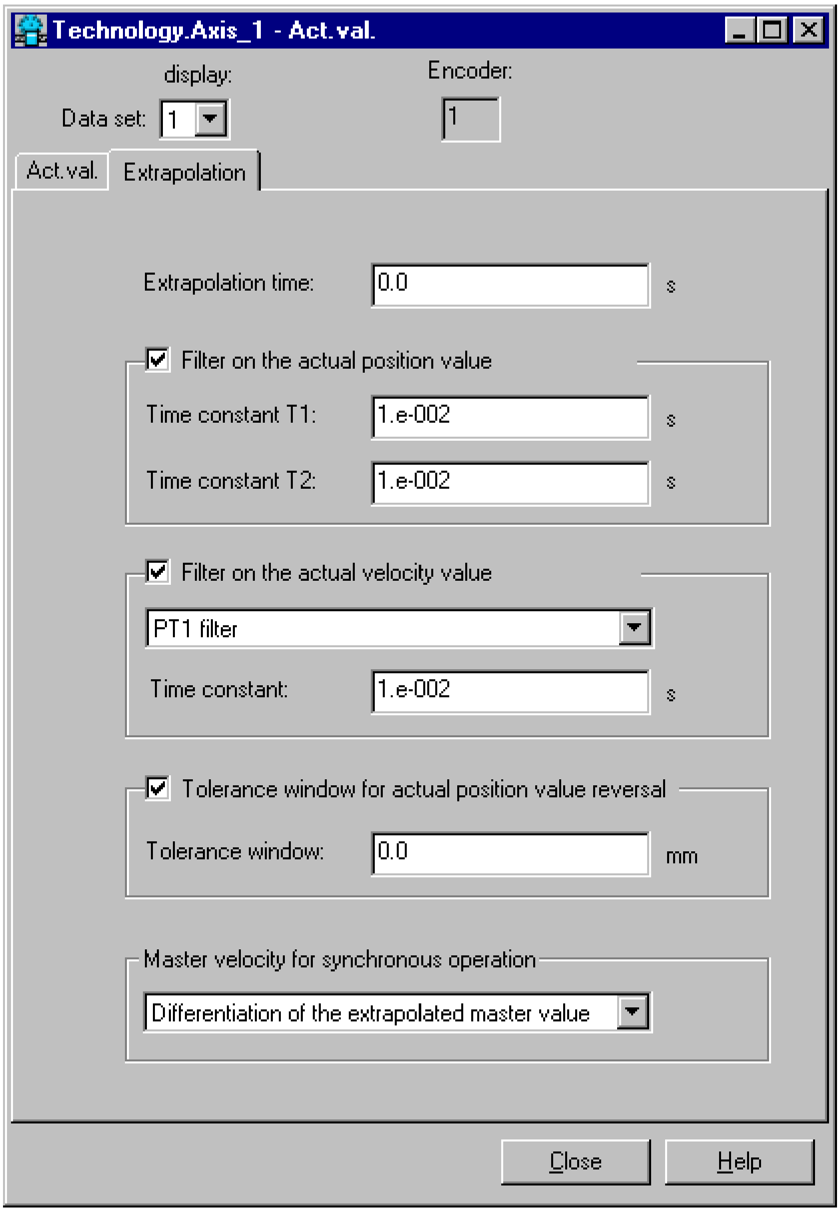
4.5.8.2 Actual value - "Extrapolation" tab



If there is a synchronous operation interconnection within a control, the synchronous operation takes into account the position, velocity, and acceleration of the master value position.

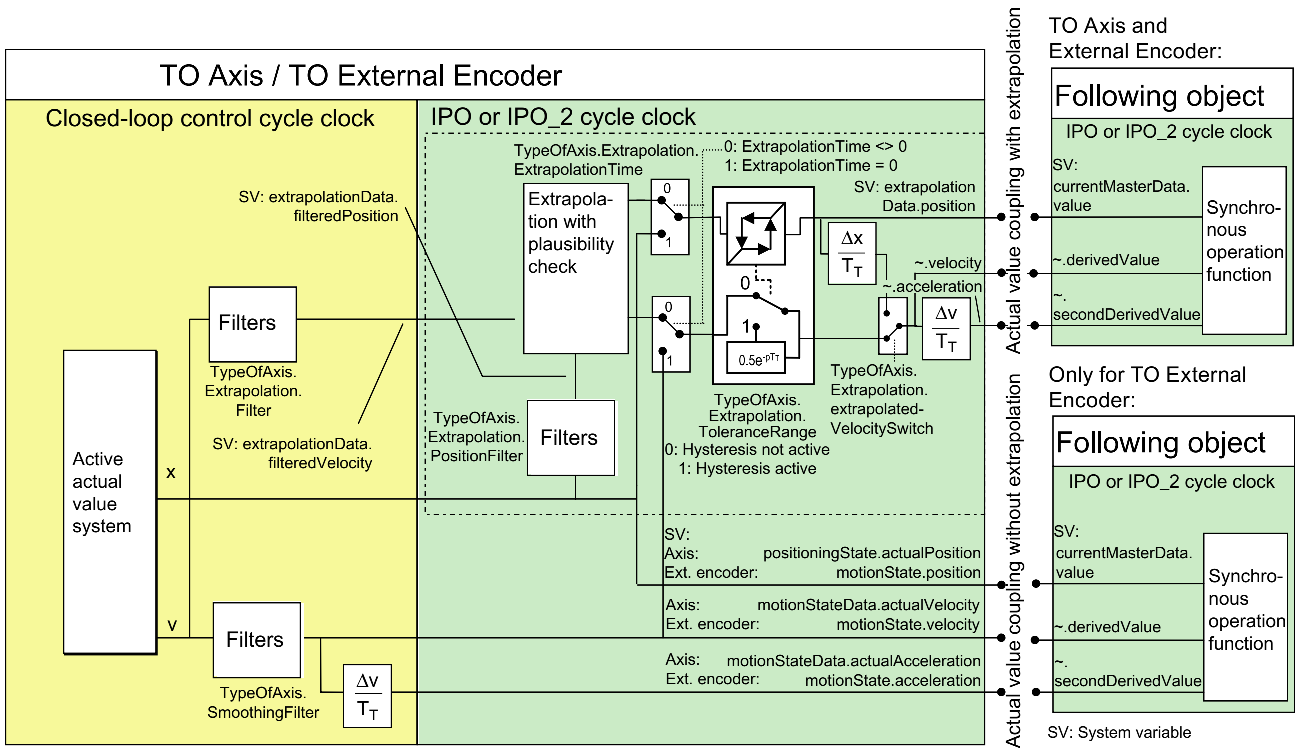
If an actual encoder value is used as the master value, it is useful to extrapolate the measured actual value for the synchronous operation in order to compensate for dead times. Dead times result within the system when measuring actual values, e.g. due to the bus communication and the system processing times.

The extrapolation is set in S7T Config on the leading axis or on the external encoder in the dialog **Axis / External encoder > Actual value** in the "Extrapolation" tab.



The parameters of this dialog can also be read from or written to the user program via technology parameters.

Actual value coupling with extrapolation (axis and external encoder)



Extrapolation time

(Parameter *1110;* configuration data *TypeOfAxis.Extrapolation.ExtrapolationTime*)

Here, you set the time for the extrapolation. No extrapolation if the value entered is *0.0*.

Filter on the actual position value

(Parameter *1130* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.enable*)

Activate the check box if you would like to extrapolate the actual position value.

Time constant T1

(Parameter *1131* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T1*)

Here, you set time constant T1 of the PT2 filter for the extrapolation in the actual value system.

Time constant T2

(Parameter *1132* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T2*)

Here, you set time constant T2 of the PT2 position filter in the actual value system.

The filter acts on the actual position for the extrapolation. The velocity for the extrapolation is taken over from the actual values of the axis or external encoder before application of the smoothing filter (*TypeOfAxis.smoothingFilter*).

Filter on the actual velocity value

(Parameter *1112* configuration data *TypeOfAxis.Extrapolation.Filter.enable*)

Activate the check box if you would like to extrapolate the actual velocity value.

Here, you select the filter for the extrapolation velocity in the drop-down list.

(Parameter *1111* configuration data *TypeOfAxis.Extrapolation.Filter.Mode*)

Time constant

(Parameter *1113* configuration data *TypeOfAxis.Extrapolation.Filter.timeConstant*)

Here, you enter the time constant for the filter.

The position is extrapolated based on the filtered or averaged velocity value. Averaging is via the "time constant".

Tolerance window for actual position value inversion

(Parameter *1114* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.enable*)

Here, you can activate the tolerance window for actual position value inversion

Tolerance window

(Parameter *1115* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.Value*)

Enter the size of the tolerance window here.

If the master value is superimposed with high-frequency noise signals that the synchronous operation cannot follow, this can cause the dynamic response limits to be exceeded or the master value to briefly change directions during synchronization.

In this case, define a tolerance window to prevent the dynamic response limits from being exceeded on the following axis or to prevent direction changes during synchronization.

Master velocity for synchronous operation

(Parameter *1116* configuration data *TypeOfAxis Extrapolation.extrapolatedVelocitySwitch*)

In the drop-down list, select whether the velocity is to be applied for the extrapolation or if the extrapolated master position value is to be differentiated.

Checking the extrapolated and filtered values

The extrapolated and filtered values can be checked in the following system variables:

- *extrapolationdata.position*
- *extrapolationdata.velocity*
- *extrapolationdata.filteredposition*
- *extrapolationdata.filteredvelocity*
- *extrapolationdata.acceleration*

Support of encoders with NIST evaluation

With encoders with NIST evaluation, the speed determined by the encoder and the resulting velocity can be accepted by the encoder. A calculation of the speed and velocity by the integrated technology is not necessary in this case. Two methods of transmission are available:

- **Transmission in the PROFIdrive message frame**

  Setting via the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_PROFIDRIVE_NIST_*

- **Transmission in the I/O area**

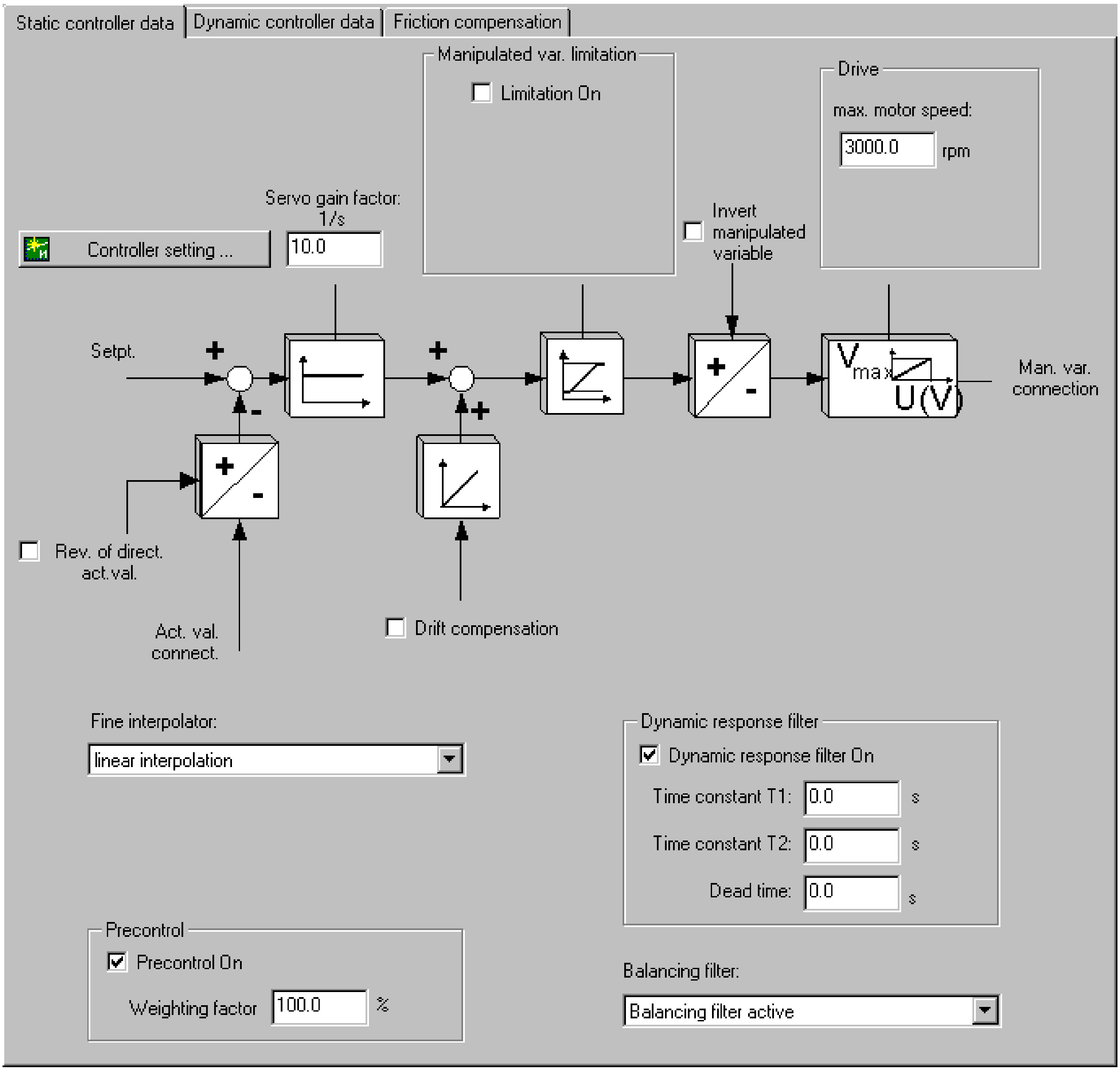
  Setting via the configuration data element

  *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_DIRECT_NIST*In this case, 4000H corresponds to 100%. The address is set in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.logAdress*, and the reference value in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.referenceValue*.

4.5.9 Control

4.5.9.1 Control - "Static controller data" tab

The "Static controller data" tab of the **Axis > Control** dialog can be used to configure the position control of the axes.



Manipulated variable limiting

Manipulated variable limiting represents an absolute high and low limits of the control range. This limitation is applied prior to inversion.

Note

When the Dynamic Servo Control (position controller in the drive) function is active, backlash locking (limiting of manipulated variable for the drive) is not effective. Therefore, when DSC is active, the backstop must be generated in the drive.

Drive

Use this input box to enter the maximum speed of the drive.

"Controller setting" button

This button can be used to implement a simple controller optimization without precontrol and balancing filters with SINAMICS drives.

Follow the instructions in the "Optimizing the position controller - overview (Page 853)" chapter if you want to achieve an implemented position control system.

Controller parameters

For further details on the individual controller parameters and their optimal setting, please refer to the chapter "Optimizing the position controller - overview (Page 853)".

Fine Interpolator

The fine interpolator function is used is to generate interim setpoints when the interpolator and controller have a different duty factor. Optional interpolation mode settings at the fine interpolator:

- No interpolation
- Linear interpolation (constant position for positioning axis)
- Interpolation with constant acceleration (constant acceleration at positioning axis)
- Interpolation with constant velocity (constant velocity at positioning axis)

When set for positioning axes, the position setpoint is interpolated.

When set as speed-controlled axis, the velocity setpoint is interpolated.

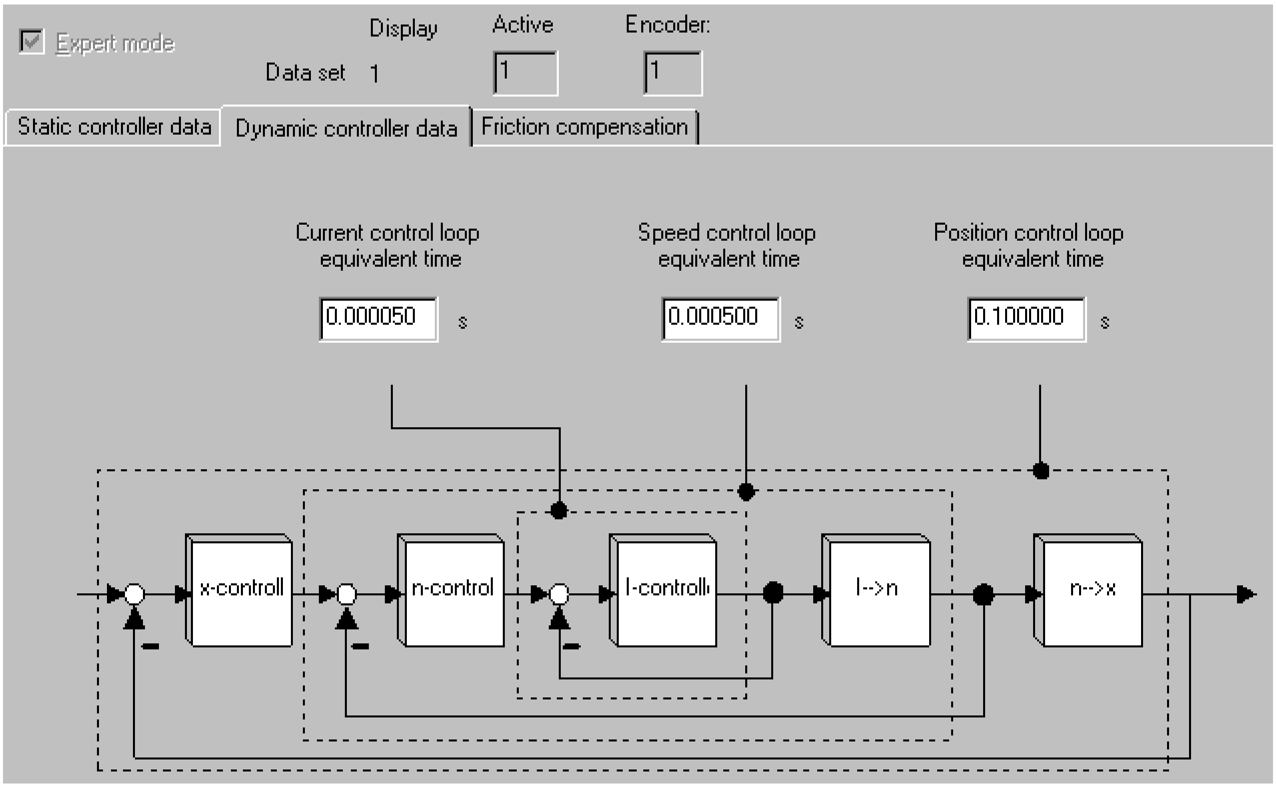
Dynamic filter, precontrol, balancing filter

For further details on the settings please refer to the chapter "Optimizing the position controller - overview (Page 853)".

4.5.9.2 Control - "Dynamic controller data" tab

The "Dynamic controller data" tab of the **Axis > Control** can be used to adjust the dynamic response of the axes, for example, to match their response in synchronous operation. The "Dynamic controller data" tab is available when you set expert mode in the "Static controller data" tab.

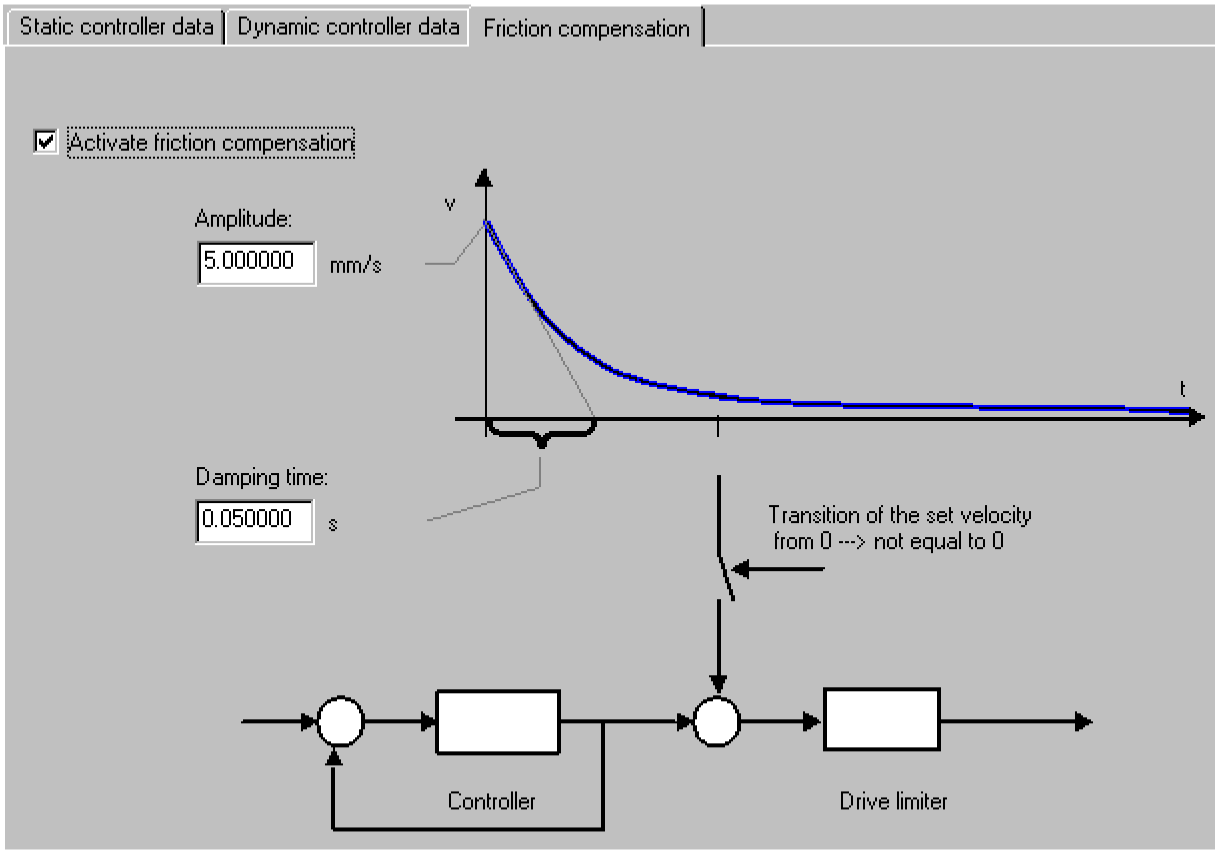
The setpoint branch of the control loop contains a configurable dynamic filter which you can use to adjust the dynamic response of the axes.



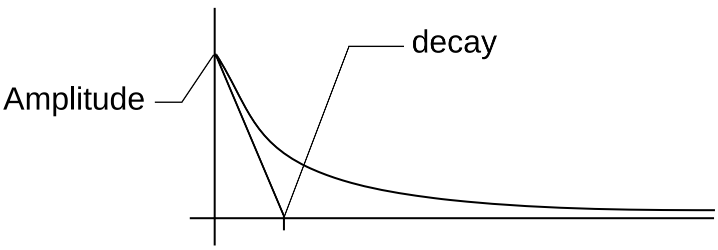
- **Equivalent time current control loop**
  The equivalent time current control loop is not used in this version.
- **Equivalent time speed control loop**
  The "Equivalent time speed control loop" parameter can be used to set time constant vTc (velocity Time constant) of the balancing filter.
- **Equivalent time position control loop**
  The equivalent time of the position control loop is required to toggle from speed-controlled mode to position-controlled mode and to start the emergency stop ramp.

4.5.9.3 Control - "Friction compensation" tab

The "Friction compensation" tab of the **Axis > Control** tab is used to enable friction compensation. The "Friction compensation" tab is available when expert mode is set on the "Static controller data" tab.



The system provides a simple method compensating for the forces of static friction. During startup from a standstill, a DT1 element adds a static friction compensation signal to the manipulated variable.



The friction compensation is added relative to the velocity setpoint. It is only active when motion commands are executed.

The standstill identification for static friction compensation can be set separately, as is the case for the amplitude and the decay response. The amplitude and decay response are set in the configuration.

4.5.10 Homing

4.5.10.1 Introduction

Position-controlled axes equipped with incremental motor measuring systems must be referenced to the position of the mechanical system of the machine each time power is switched on. The axis is synchronized during homing, based on the activation of a certain position value at a defined position of the axis mechanism.

Axes can be homed in:

- Active mode (reference point approach)
- Passive mode (flying referencing)
- Direct mode (with position setpoint)

Detailed settings depend on the measuring systems available for measuring the reference point and on the motion an axis may perform for homing.

4.5.10.2 Homing - "Active homing" tab

Active homing

Active homing can be configured in the "Active homing" tab of the **Axis > Homing** dialog box in S7T Config.

Three homing modes are available for "Active homing":

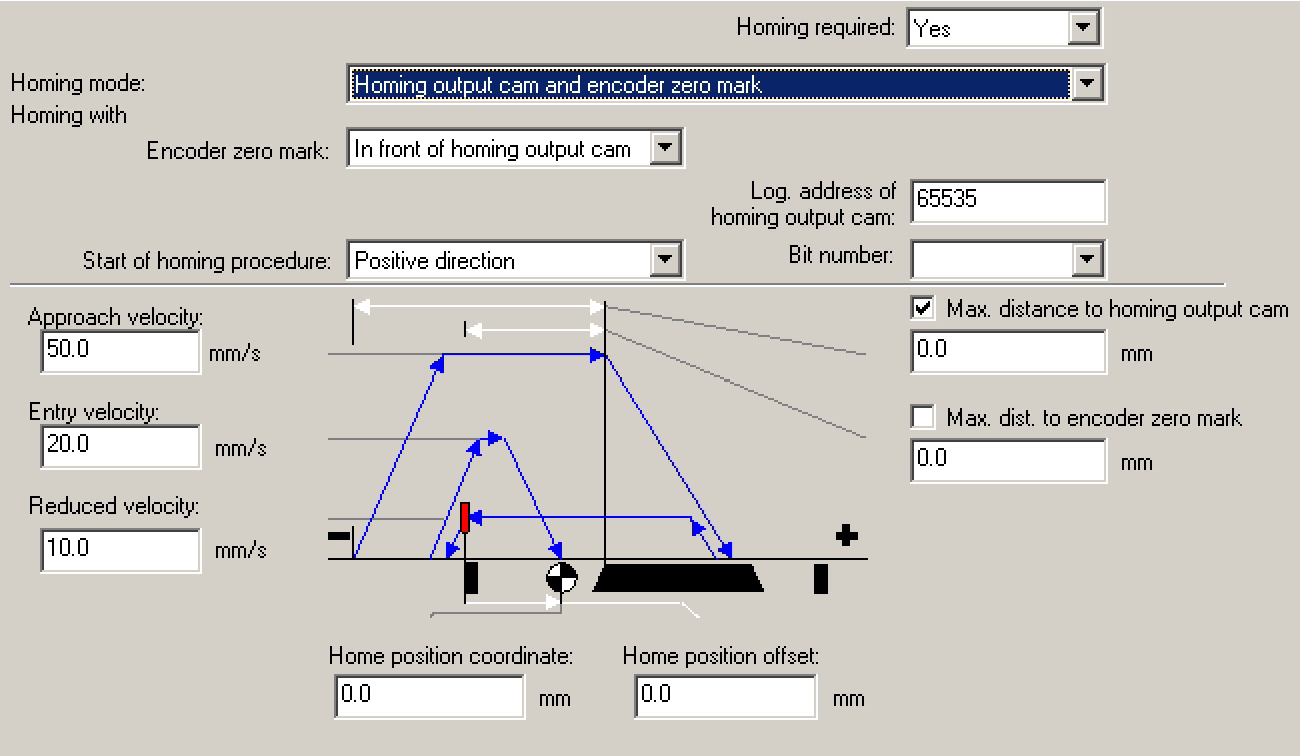
- Reference cam and encoder zero mark
- Encoder zero mark only
- External zero mark only

Start of the homing function:

MC_Home	*Mode = 0*	Determination of the reference point based on the axis configuration
	Mode = 1 *Position = x*	Determination of the reference point based on the axis configuration The reference point is assigned the value of the *Position* input parameter.

After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone* = *TRUE*.

Active homing mode with "reference cam and encoder zero mark"



After homing is started with the "MC_Home" technology function, the axis moves to the reference cam and then approaches the referencing encoder zero mark, according to the axis configuration. The lower section of the diagram shows the resultant motion sequence.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case "Homing output cam and encoder zero mark"
Encoder zero mark	Before reference cam	The axis is homed to the encoder zero mark which lies before the reference cam with reference to the direction of reference point approach.
	After reference cam	The axis is homed to the encoder zero mark which lies after the reference cam with reference to the direction of the reference point approach.
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Logical address of reference cam	[byte address]	Logical byte address of the reference cam You can connect the reference cam to the integrated inputs of the Technology CPU, or to the IO on DP(DRIVE).
Bit number	[Number of the bit]	Bit address of the signal used for the reference cam
Approach velocity	-	Velocity at which the axis approaches the reference cam
Entry velocity	-	Velocity at which the axis approaches the (shifted) reference point after detection of the encoder zero mark
Shutdown velocity	-	Velocity at which the axis approaches the encoder zero mark after detection of the reference cam.
Reference point coordinate (Reference cam and encoder zero mark, External zero mark only, Encoder zero mark only)	-	Here, you enter the actual position value of the home position (relative to the coordinate system of the axis)
Homing position offset	-	The "Home position offset" shifts the reference point by a configured distance. The axis travels along this configured distance at a velocity defined in "Entry velocity", starting at the synchronization position with encoder zero mark. Modulo axes always take the shortest distance.
Maximum distance to homing output cam	deactivated	The distance to reference cam detection is not monitored
	Activated	Monitoring of the distance between the start of reference point approach and detection of the reference cam. If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance an axis travels between the reference cam and the encoder zero mark. If the difference in this distance exceeds the configured distance, the corresponding axis technology DB indicates error *801D*. The reference point approach is canceled.

Sequence of the reference point approach

- Phase 1: Reference cam approach

  The axis starts its reference point approach at the configured approach velocity and towards the direction set in "Start of reference point approach". The reference cam approach ends after the reference cam (Phase 1) is reached.

  You can also monitor the distance an axis travels between the start of reference point approach and detection of the reference cam by setting the "Max. distance to homing output cam" check box. The reference point approach is canceled with error if the reference cam is not detected between the start and end of the configured distance.

- Phase 2: Synchronization with encoder zero mark

  After having reached the reference cam, the axis accelerates / decelerates to shutdown velocity, and approaches the encoder zero mark. The encoder zero mark is derived from the combined settings of "Encoder zero mark" (after or before the reference cam) and "Start of reference point approach" (positive or negative direction).

  After the reference cam is detected, the PLC synchronizes the axis to the first encoder zero mark detected in accordance with the configuration. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

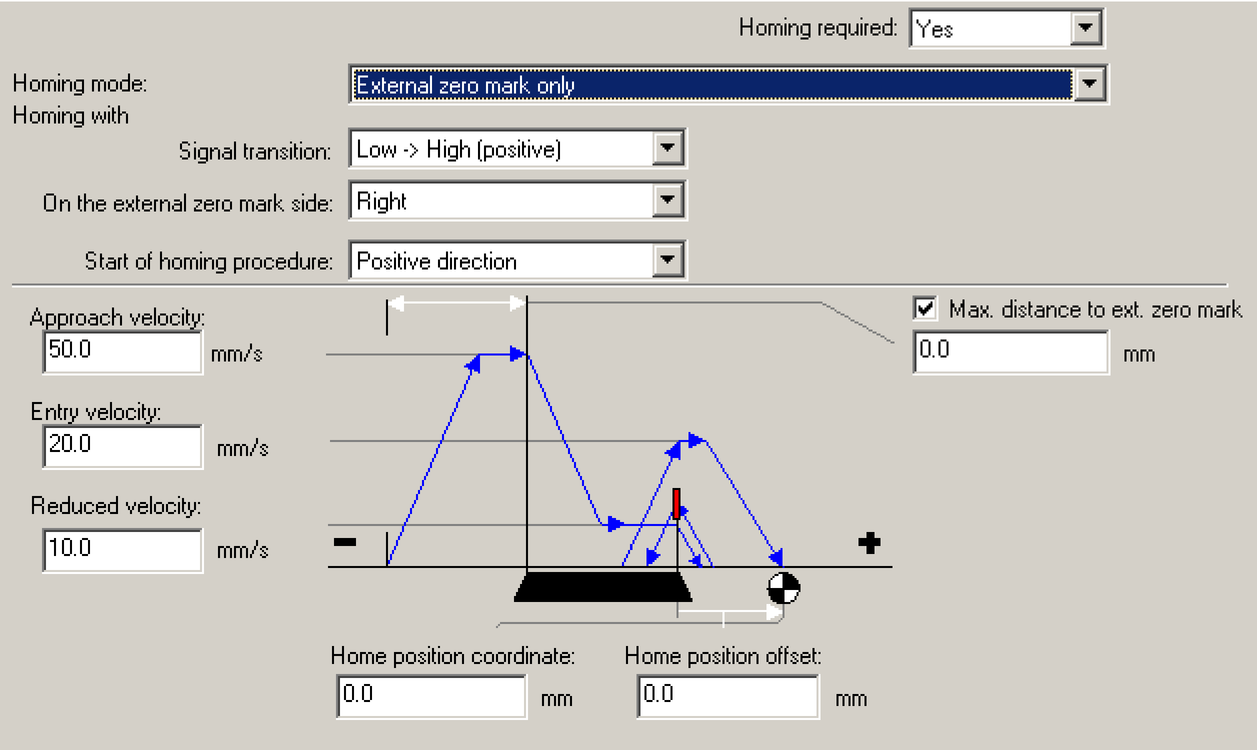
  You can also monitor the distance an axis travels between the reference cam and the encoder zero mark by setting the "Max. distance to encoder zero mark" check box. The reference point approach is canceled with error if the encoder zero mark is not found within the specified distance after the reference cam is detected.

- Phase 3: Reference point approach

  After the encoder zero mark is detected, the axis accelerates / decelerates to approach velocity to approach the reference point.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the encoder zero mark is detected, provided that the reference point lies within the deceleration ramp.

Active homing mode with "External zero mark only"



If homing is started by calling the "MC_Home" technology function the axis approaches the referencing external zero mark in accordance with the configuration. The lower section of the diagram shows the resultant motion sequence.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "External zero mark only"
Signal transition	Low -> High (positive)	The motion is referenced to the positive edge of the external zero mark (setting according to edge evaluation in the drive component).
	High -> Low (negative)	The motion is referenced to the negative edge of the external zero mark (setting according to edge evaluation in the drive component).
On the side of the external zero mark	left	The signal transition is measured viewed from the left side of the external zero mark, in the selected direction of reference point approach.
	right	The signal transition is measured viewed from the right side of the external zero mark, in the selected direction of reference point approach.

Parameters	Value	Explanation
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Approach velocity	-	Velocity at which the axis approaches the external zero mark
Entry velocity	-	Velocity at which the axis approaches the (shifted) homing position after detection of the external zero mark
Shutdown velocity	-	Velocity at which the axis approaches the reference point coordinate after detection of the external zero mark
Homing position offset	-	The "Home position offset" shifts the reference point by a configured distance. The axis moves by a configured distance at the "Homing velocity", after its synchronization at the external zero mark edge. Modulo axes always take the shortest distance.
Max. distance to external zero mark	deactivated	The distance to external zero mark detection is not monitored.
	Activated	Monitoring of the distance between the start of the reference point approach and detection of the external zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.

Note

For homing with "external zero mark", connect the external zero mark as digital measuring input to the drive component.

In order to execute a reference point approach in "external zero mark" homing mode, select "Signal transition" and "on the side of the external zero mark" values which correspond with the drive's configuration or functionality. Instead of being used to configure the measuring function in the drive, the "Signal transition" and "on external zero mark side" parameters merely reflect their functionality in order to control the axis motion according to the configuration.

For information on the configuration of external zero mark detection, refer to the relevant product information or to the drive manuals.

Sequence of the reference point approach

- Phase 1: Synchronization to external zero mark

  The axis starts its reference point approach at the configured approach velocity and towards the direction set in "Start of reference point approach". Synchronization ends at the external zero mark (phase 1) when the configured signal transition (configured at the drive component) is detected at external zero mark. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

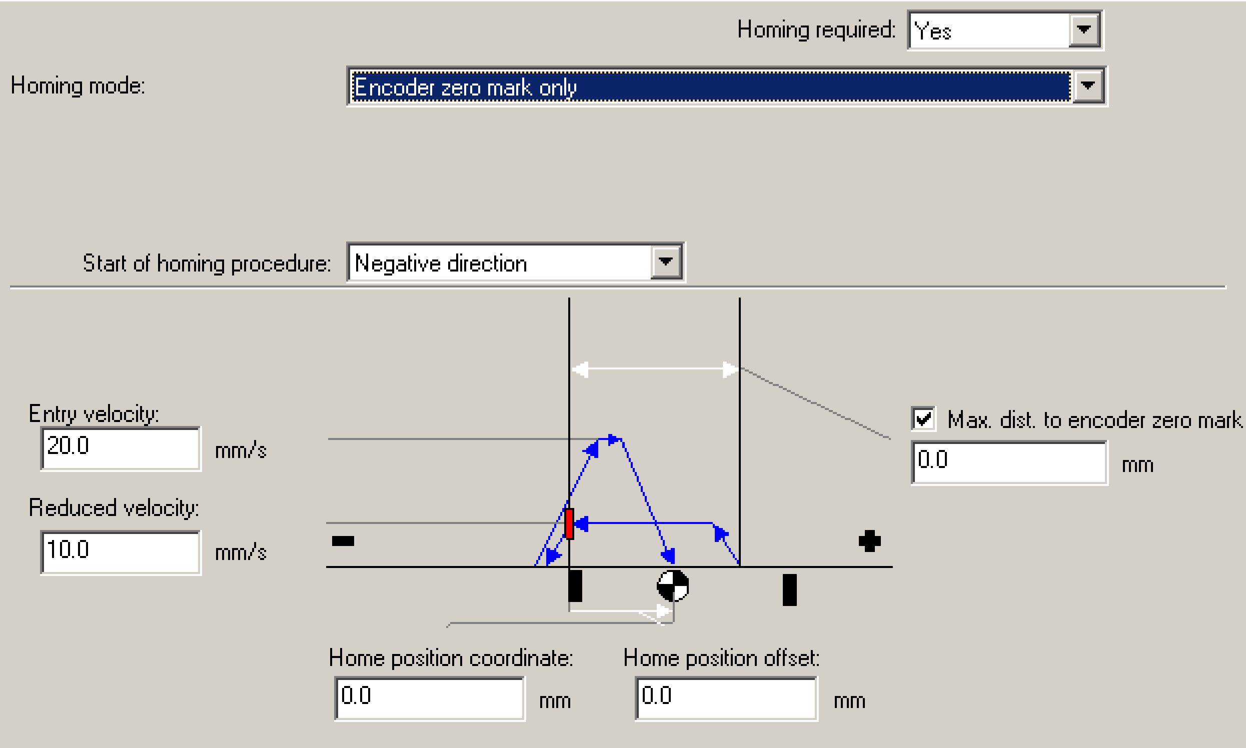
  You can also monitor the distance an axis travels between the start of reference point approach and detection of the signal edge by setting the "Max. distance to external zero mark" check box. Homing is canceled with error if the edge is not detected between the start of reference point approach and the end of the configured distance.

- Phase 2: Reference point approach

  After the configured signal edge is detected, the axis accelerates / decelerates to approach the reference point coordinate at shutdown velocity.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the external zero mark is detected, provided that the reference point lies within the deceleration ramp.

Active homing mode with "Encoder zero mark only"



Homing to encoder zero mark is used, for example, at axes for which the encoder can only set one encoder zero mark in the entire traversing range of the axis. The homing command initiates axis approach to the encoder zero mark. After the encoder zero mark is detected, the axis approaches the shifted reference point at homing velocity. The axis position now has the value defined in the reference point coordinate. The graphic in the figure shows the resultant motion sequence. Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Encoder zero mark only"
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Entry velocity	-	Velocity at which the axis approaches the (shifted) reference point after detection of the encoder zero mark
Shutdown velocity	-	Velocity at which the axis starts the reference point approach and approaches the encoder zero mark
Homing position offset	-	The homing position offset function shifts the homing position by a configured distance, meaning that the axis moves along a configured distance behind the encoder zero mark detection position at a "Homing velocity". Modulo axes always take the shortest distance.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance between the start of the reference point approach and detection of the encoder zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.

Sequence of the reference point approach

- Phase 1: Synchronization to encoder zero mark

  The axis starts its reference point approach towards the direction set in "Start of reference point approach" at the configured shutdown velocity. Synchronization with encoder zero mark (phase 1) ends with the detection of the encoder zero mark. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

  You can also monitor the distance an axis travels between the start of reference point approach and encoder zero mark detection by setting the "Max. distance to zero mark" check box. Homing is canceled with error if the encoder zero mark is not detected between the start of reference point approach and the end of the configured distance.

- Phase 2: Reference point approach

  After the encoder zero mark is detected, the axis accelerates / decelerates to approach the reference point coordinate at entry velocity.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the encoder zero mark is detected, provided that the reference point lies within the deceleration ramp.

4.5.10.3 Homing - "Passive homing" tab

Passive homing can be configured in the "Passive homing" tab of the **Axis > Homing** dialog box of S7T Config. The difference compared to active homing is that the required homing motion is not initiated by a homing command.

Three homing modes are available for "Passive homing":

- Reference cam and encoder zero mark
- External zero mark only
- Encoder zero mark only

Start of the homing function:

MC_Home	*Mode = 2* *Position = x*	The current position is assigned the value of the *Position* input parameter at the reference point.

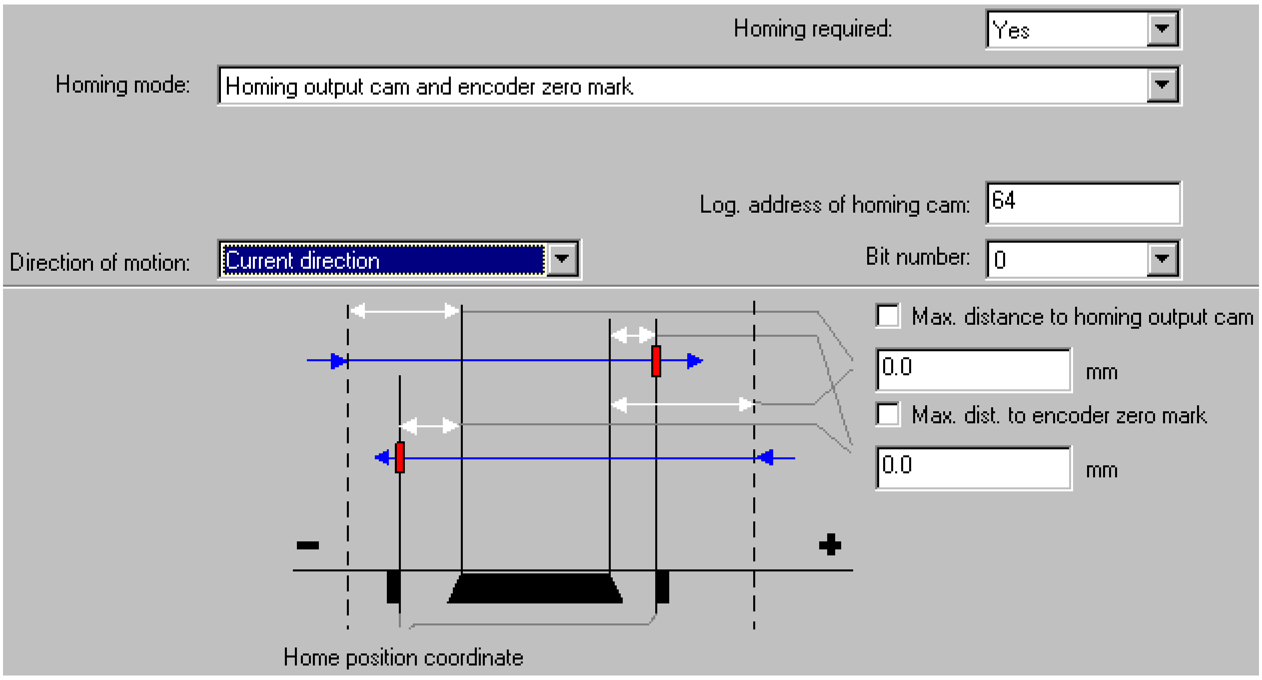
After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone= TRUE*.

Note

Note that not all drive components support all homing mode or measured signal evaluation functions. For details, refer to the documentation of the drive components used.

Triggering of encoder zero marks or reference cam monitoring during passive homing operations is indicated by a corresponding error message at the technology DB. The current axis motion is terminated in this case.

Passive homing mode with "Reference cam and encoder zero mark"



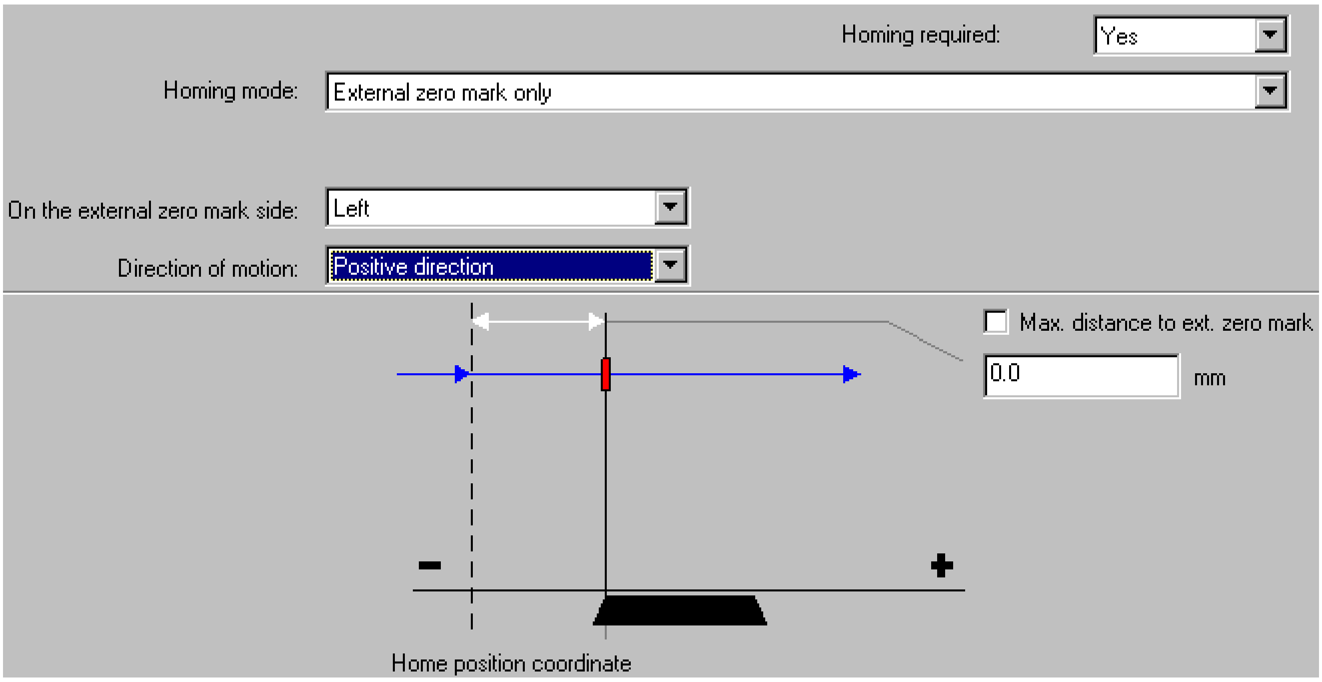
After the axis has passed the reference cam, the next encoder zero mark triggers axis homing. The axis position is set in the reference point to the value defined at the *Position* input parameter of the "MC_Home" technology function.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Reference cam and encoder zero mark"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.
Logical address of reference cam	[byte address]	Logical byte address of the reference cam You can connect the reference cam to the integrated inputs of the Technology CPU, or to the IO on DP(DRIVE).
Bit number	[Number of the bit]	Bit address of the signal used for the reference cam

Parameters	Value	Explanation
Maximum distance to homing output cam	deactivated	The distance to reference cam detection is not monitored
	Activated	Monitoring of the distance between the start of the homing function and detection of the reference cam. If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance an axis travels between the reference cam and the encoder zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.

Passive homing mode with "External zero mark only"



Axis homing starts with the detection of the external zero mark. The axis is set to the value of the *Position* input parameter of the "MC_Home" technology function.

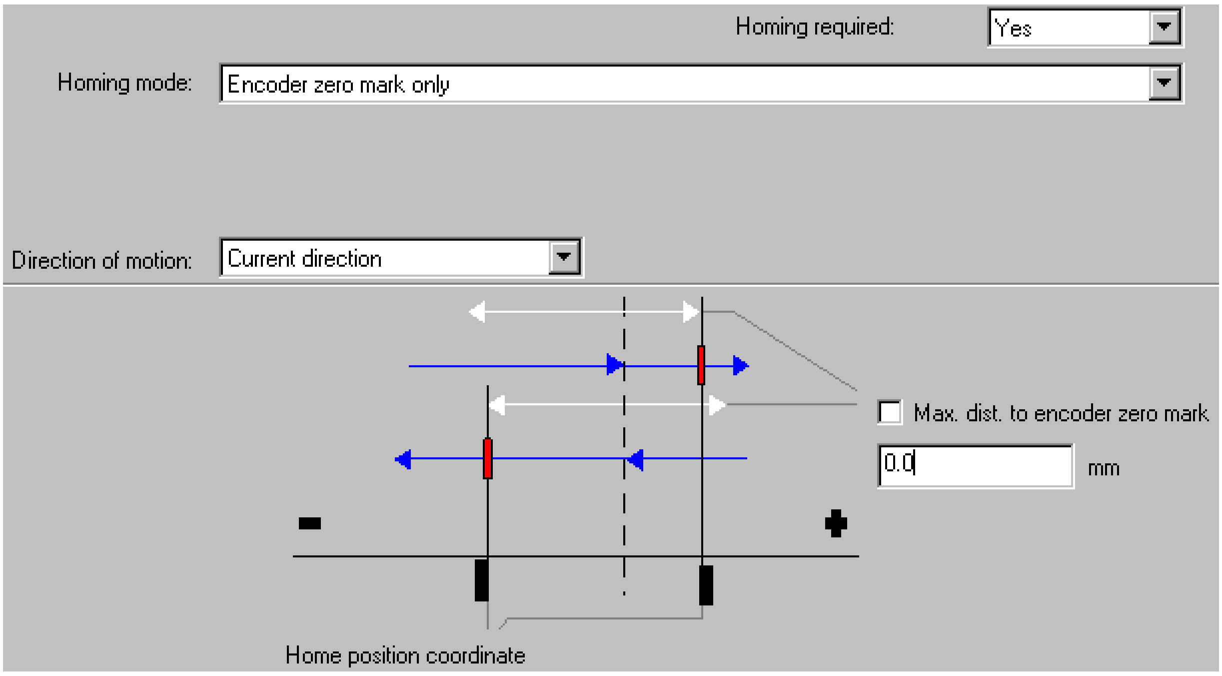
Meaning of the individual parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "External zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the external zero mark.
	Negative direction	The axis is only homed with negative approach to the external zero mark.
	Current direction	The axis is homed when it reaches the next external zero mark.
On the side of the external zero mark	left	Direction of movement: Positive direction Axis homing is triggered at the positive edge. Direction of movement: Negative direction Axis homing is triggered at the negative edge.
	right	Direction of movement: Positive direction Axis homing is triggered at the negative edge. Direction of movement: Negative direction Axis homing is triggered at the positive edge.
Max. distance to external zero mark	deactivated	Distance to go to the external zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the external zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Note

- For homing with "external zero mark", connect the external zero mark as digital measuring input to the drive component.
- In order to be able to execute the reference point approach in "external zero mark" homing mode as required, you should always set values at "Direction of movement" and "on the side of the external zero mark" which are compatible with drive configuration or functionality. The "on the side of the external zero mark" and "Direction of movement" parameters are not used to configure the measuring function in the drive and only reflect their functionality.
- For information on the configuration of external zero mark detection, refer to the relevant product information or to the drive manuals.

Passive homing mode with "Encoder zero mark only"



Homing mode with "Zero mark only" is used for axes, for example, which are equipped with an encoder that outputs only one zero mark signal within the entire traversing range of the axis

Axis homing starts with the detection of the encoder zero mark. The position value of the axis is set to the value of the *Position* input parameter of the "MC_Home" technology function after the encoder zero mark is detected.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Encoder zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the encoder zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Passive homing mode with "Default"

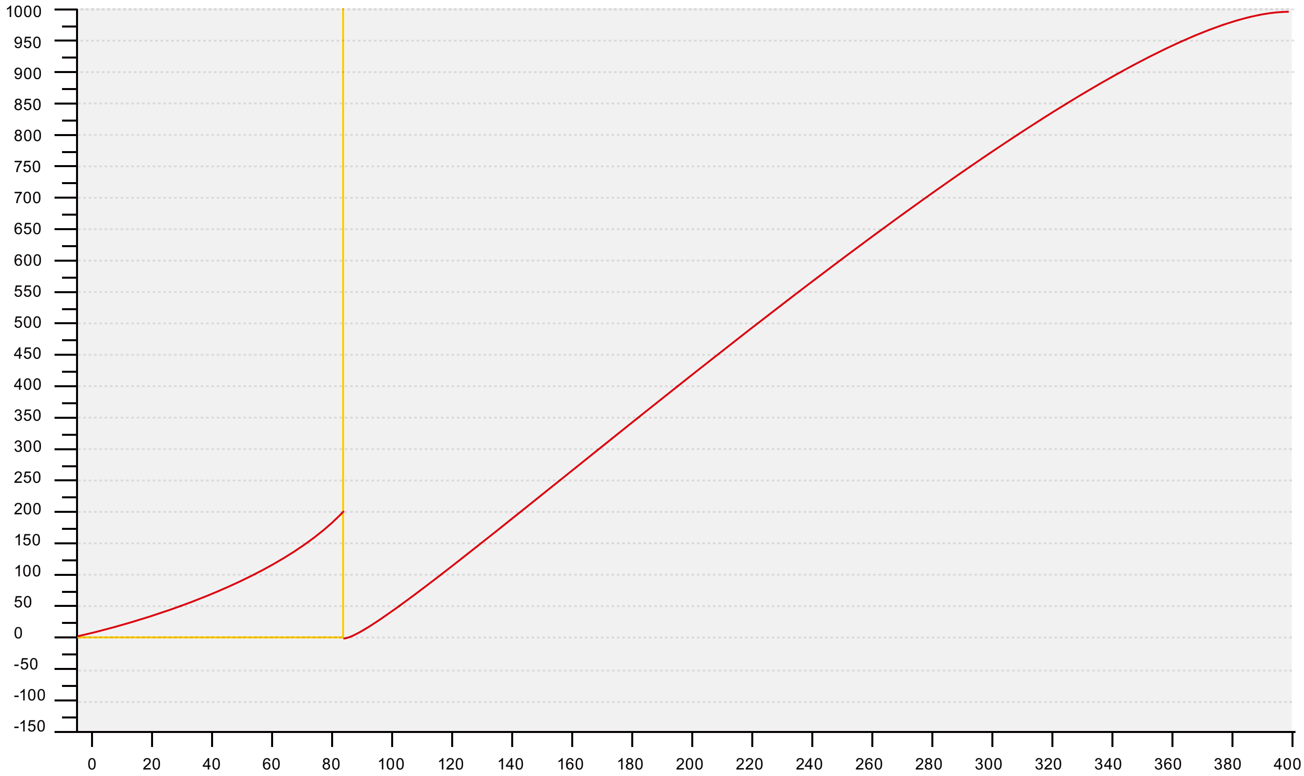
When a new axis is created in S7T Config, the *Default* homing mode is preset.

Homing mode with *encoder zero mark only* is used if the configured encoder provides a zero mark. Homing mode external zero mark only is used if the encoder does not provide a zero mark.

4.5.10.4 Positioning behavior with passive homing

The following examples show the response of the end position with passive homing with the motion commands "MC_MoveRelative" and "MC_MoveAbsolute":

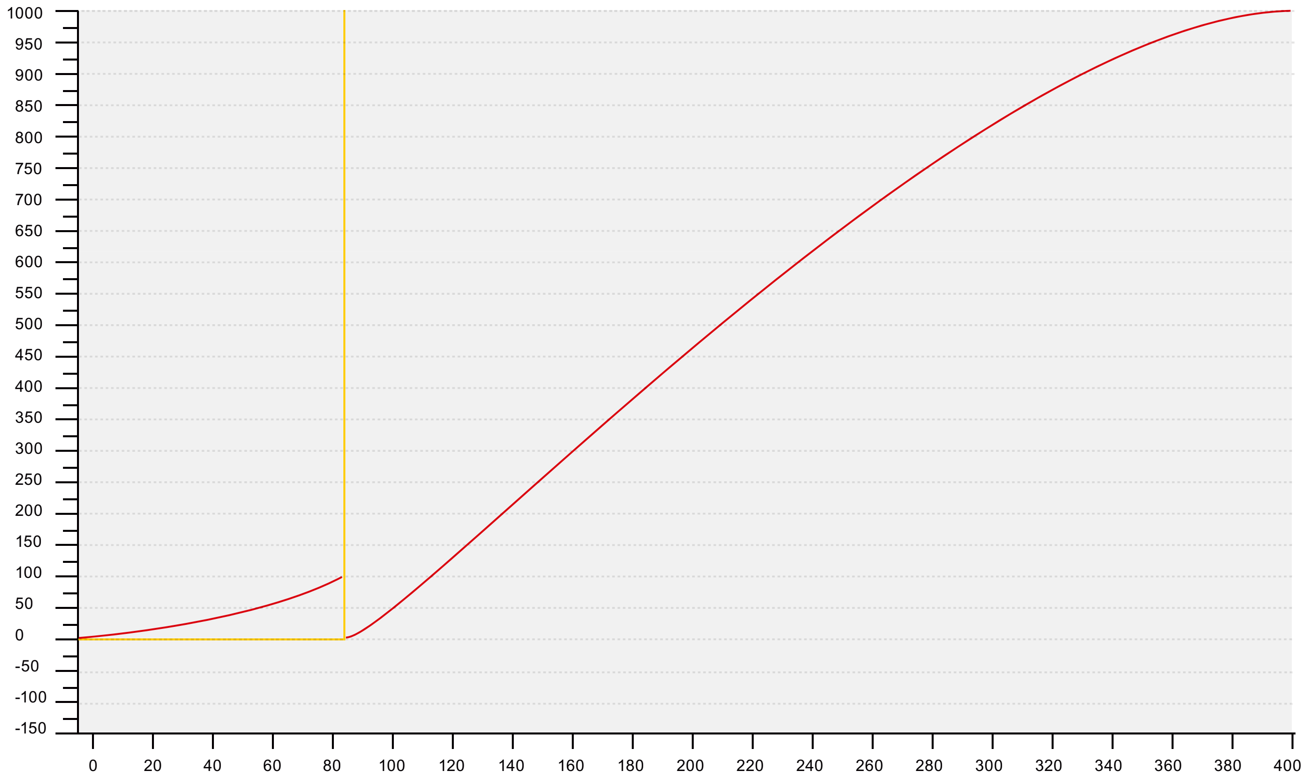
Positioning behavior with passive homing with "MC_MoveRelative"



An MC_MoveRelative command *1000* is started for passive homing.

1. The MC_MoveRelative command and passive homing start at the same time.
2. When the yellow marked homing position *200* is reached, the axis is homed and the position of the axis is set to position *0*.
3. The axis moves by the remaining relative distance to position *800 (1000-200)*.

Positioning behavior with passive homing with "MC_MoveAbsolute"



An MC_MoveAbsolute to position *1000* command is started for passive homing.

1. The MC_MoveAbsolute command and passive homing start at the same time.
2. When the yellow marked homing position *200* is reached, the axis is homed and the position of the axis is set to position *0*.
3. The axis moves to position *1000* in accordance with the new homing point.

4.5.10.5 Direct homing

The actual axis position is set to the value defined at the MC_Home technology function. No reference point shift settings are active. The function does not execute a motion. The axis is homed when the command is executed.

Start of the homing function

MC_Home	*Mode = 3* *Position = x*	Direct homing: The current position is assigned the value of the *Position* input parameter.

After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone= TRUE* .

4.5.10.6 Position correction

A correction value is deducted from the actual position value of the axis. By contrast to other homing modes, the axis maintains its homed state (homed / not homed) in this case.

The position correction function can also be used to manipulate the setpoints of the various coordinate systems (base coordinate system, superimposing coordinate system). This is of significance for superimposing camming in order to be able to generate a reference within a cam disk.

Start of the homing function:

MC_Home	*Mode = 4* *Position = x*	Actual value correction: Position value = (current position) - (parameter position).
	Mode = 6 *Position = x*	Setpoint correction in the base coordinate system: Position value = (current position) - (parameter position).
	Mode = 7 *Position = x*	Setpoint correction in the superimposing coordinate system: Position value = (current position) - (parameter position).

Position correction does not influence the *Statusword.HomingDone* status in the axis technology DB.

4.5.10.7 Motions with non-homed axes

You determine whether absolute positioning is to be available for a non-homed axis in the **Axis > Homing** dialog box.

Optional settings for "Homing required":

- No: Relative and absolute motions are possible. The Software limit switches (Page 149) are monitored.
- Yes: Relative motion only. The software limit switches are not monitored as long as the axis is not homed.

4.5.11 Monitoring functions

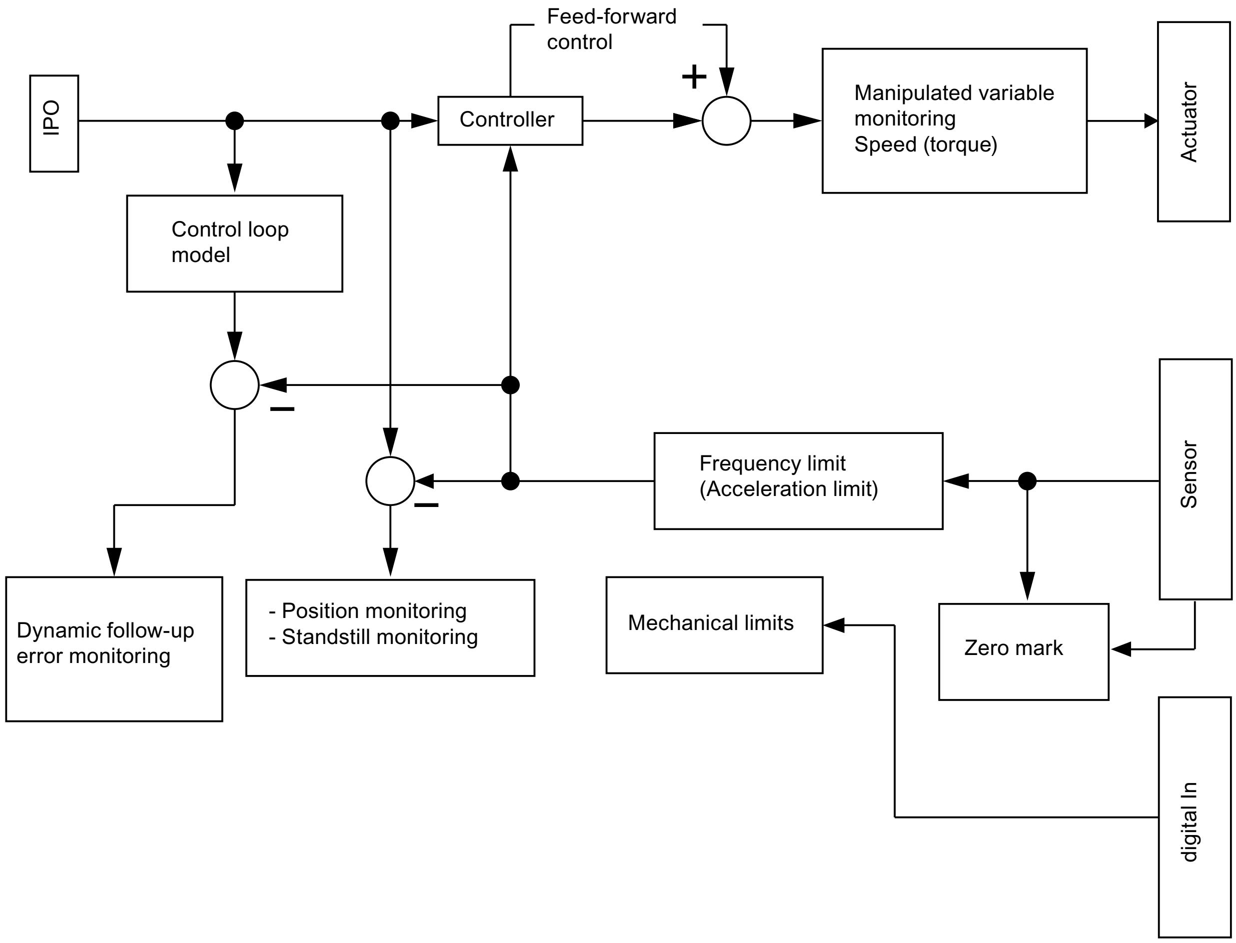
4.5.11.1 Monitoring functions - Overview

Axis monitoring functions you can configure in S7T Config:

Monitoring functions	Speed-controlled axis	Positioning axis	Synchronization axis
Velocity error monitoring (Page 187)	X	-	-
Positioning monitoring (Page 184)	-	X	X
Following error monitoring (Page 185)	-	X	X
Standstill signal (Page 186)	X	X	X
Software limit switch (Page 288)	-	X	X
Hardware limit switch (Page 286)	X	X	X
Synchronization monitoring (Page 321)	-	-	X
Manipulated variable monitoring (Page 187) (always active)	X	X	X
Encoder limit frequency monitoring (Page 919)	X (only with encoder)	X	X

Positioning axis monitoring

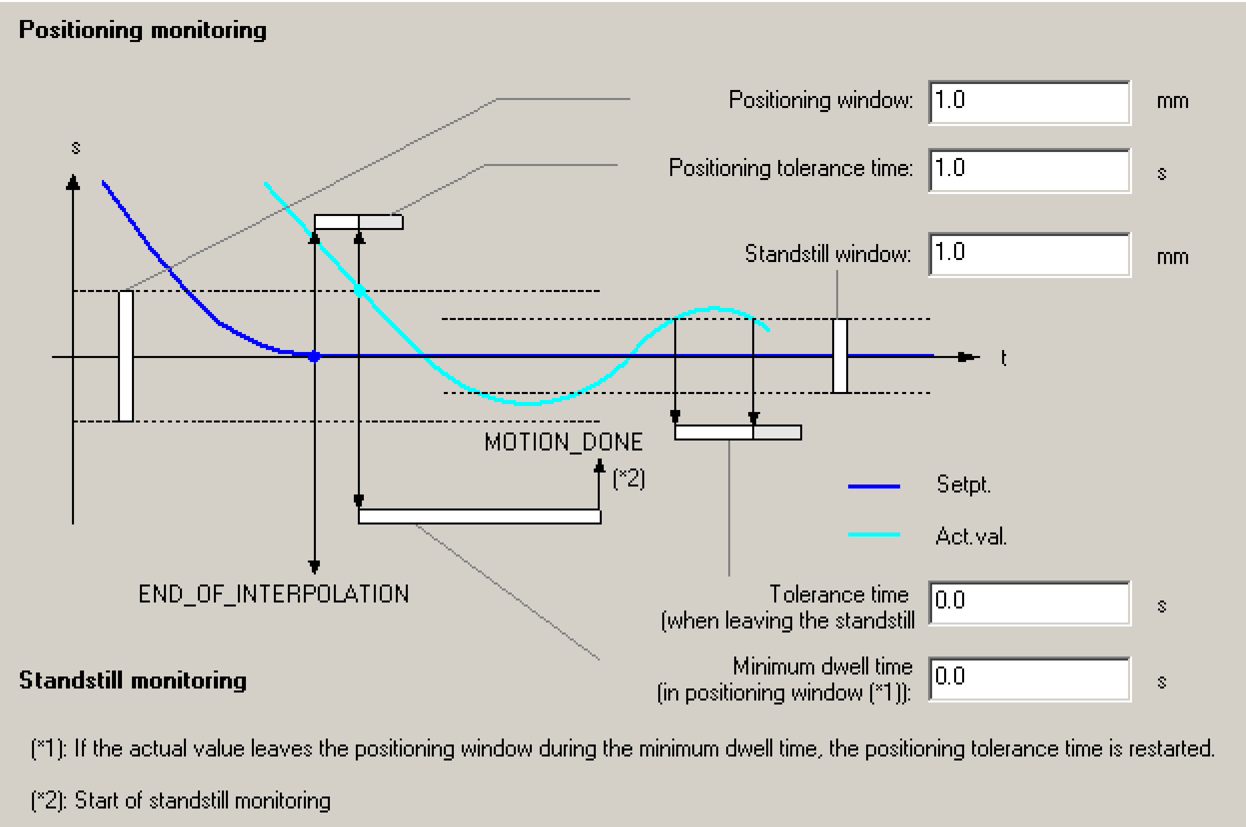
The figure below shows an example of positioning axis monitoring:



Errors are reported in the *ErrorStatus* parameter of the technology DB.

4.5.11.2 Monitoring functions - Positioning and standstill monitoring

Set the limit values for monitoring the positioning of position-controlled and synchronization axes in the "Positioning and standstill monitoring" tab of the **Axis > Monitoring functions** dialog box.



The "Positioning and standstill monitoring" tab shows whether the **Positioning** or **Synchronous operation** technology is set for the axis.

Positioning monitoring

At the end of a positioning motion, the function monitors the axis entry to the target position. For this purpose, a positioning window and a time tolerance within which the end position must be reached are specified. Monitoring is activated at the end of setpoint interpolation.

You can also set a minimum dwell time in the positioning window to expire before the positive feedback of the positioning command is activated. This time can be used for oscillating processes and control loops, for which the tolerance window should be less than the overshoot amplitude.

A positive feedback for the positioning command is output when the actual value reaches the positioning window.

The axis technology DB returns error *8019* if the positioning monitoring function is triggered.

Standstill (zero-speed) monitoring

Standstill monitoring is enabled when the position setpoint of a positioning command equals the value of the target position, and the delay of standstill monitoring activation has expired.

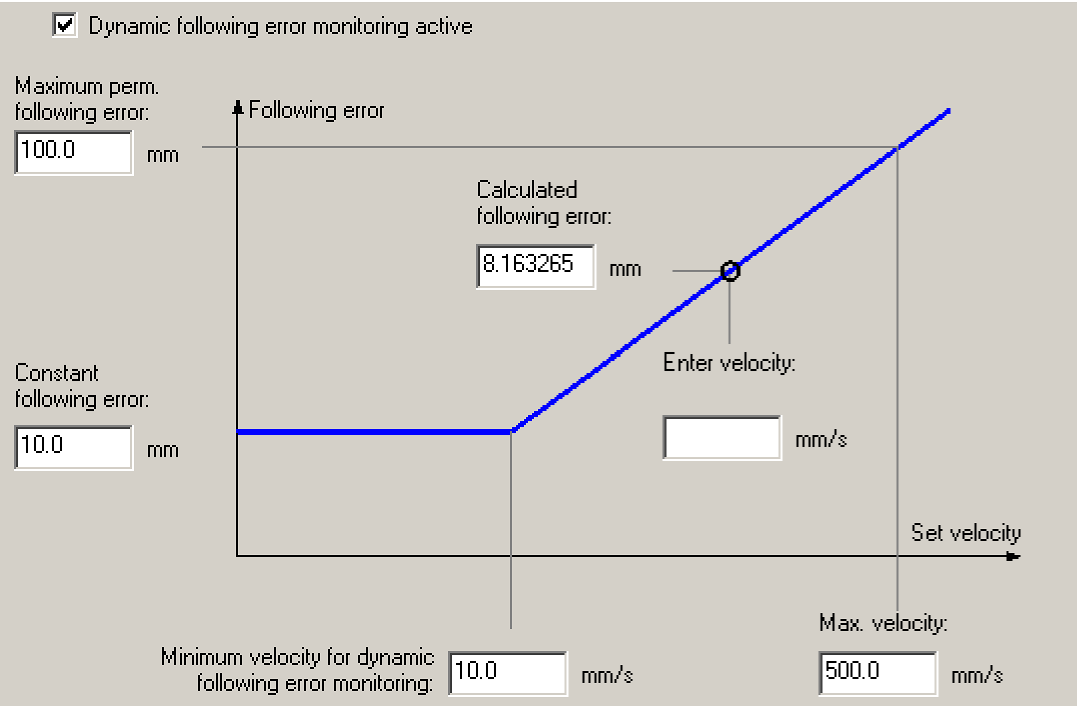
Standstill monitoring is triggered if the axis moves out of the configured standstill tolerance window for any time longer than the configured period. The axis technology DB returns error *8018* if standstill monitoring is triggered.

4.5.11.3 Monitoring functions - "Following error monitoring" tab

Dynamic following error monitoring can be enabled in the "Following error monitoring" tab of the **Axis > Monitoring functions** dialog box.

The following error monitoring on the position-controlled axis is performed using the calculated following error. The axis technology DB returns error *8016* if the offset between the actual position value and the position setpoint exceeds the programmed following error limit. The permitted following error depends on the velocity setpoint of the axis.

At velocities less than the configured minimum the permitted following error is constant and is programmed at the "constant following error" parameter. Above this limit, there is a linear increase of the following error to a maximum value which is defined by the parameter "maximum permitted following error" at maximum velocity. The permissible maximum following error is reached at maximum velocity.



If you specify a value in the "Enter velocity" input box, the corresponding function value is displayed in the "Calculated following error" box. This allows you to control how large the following error may be at the entered speed.

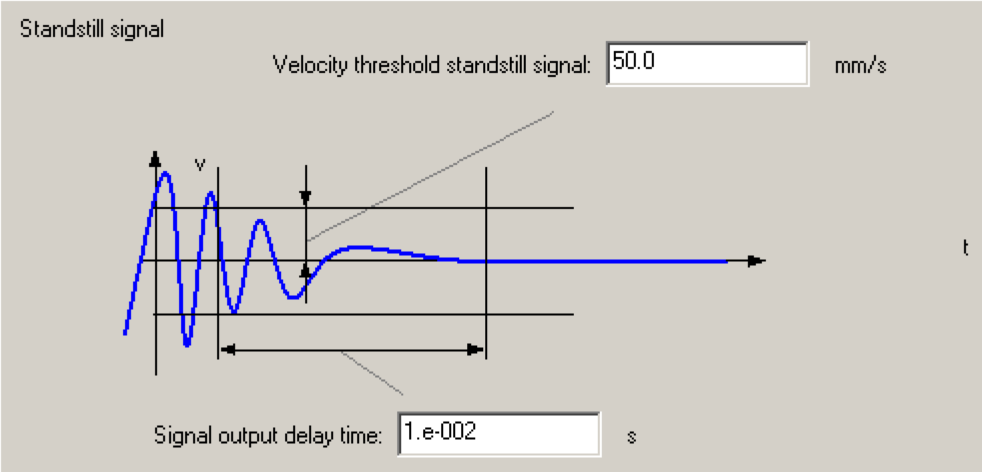
The "Following error monitoring" tab is shown for position-controlled real axes.

4.5.11.4 Monitoring functions - "Standstill signal" tab

The velocity threshold and the "Signal output delay" can be set in the the "Standstill signal" tab of the **Axis > Monitoring functions** dialog box.

The standstill signal sets the *standstill* bit in the status word of the technology DB if the current velocity is below the configured velocity threshold at least for the duration of the set delay time.

At speed-controlled and positioning axes the override is activated in speed-controlled mode. At the positioning axis, the override is activated when the positioning window is reached.

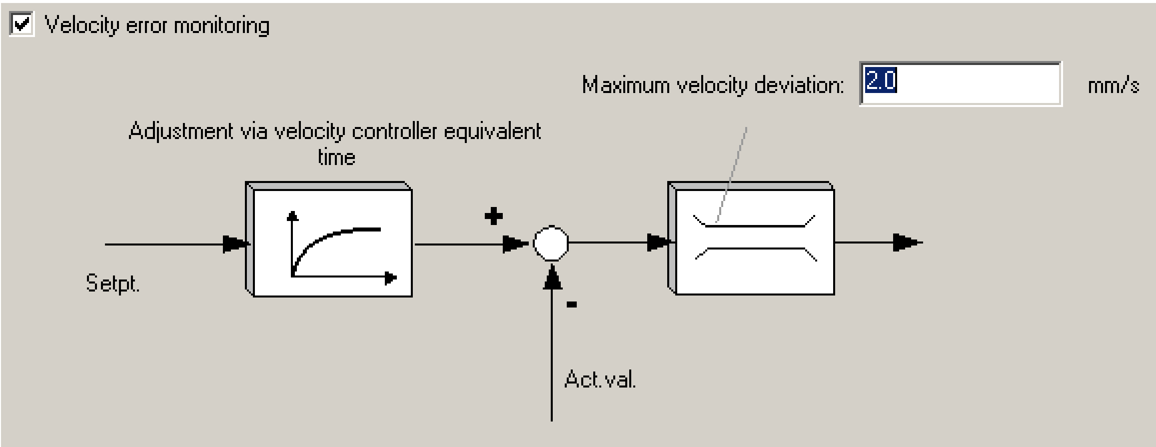


4.5.11.5 Monitoring functions - "Velocity monitoring" tab

Enable velocity error monitoring in the "Velocity error monitoring" tab of the **Axis > Monitoring functions** dialog box.

Velocity error monitoring is only relevant to these axes:

- Speed-controlled axis with encoder
- Positioning axes operating in speed-controlled mode ("MC_MoveVelocity" - input parameter *PositionControl = FALSE*)



Velocity error monitoring

Enable velocity error monitoring in this dialog.

Maximum velocity error:

This input box is only visible if velocity error monitoring is enabled.

Enter the maximum velocity error in this dialog.

4.5.11.6 Manipulated variable monitoring

The maximum values of the manipulated variables are limited for monitoring configured speed limits. The technology DB returns a warning if the values of manipulated variables exceed a configurable maximum.

The maximum possible acceleration and maximum torque are monitored by monitoring the gradient of the manipulated variable.

You define the maximum values of the axis in the **Axis > Limits** dialog box, "Position and velocity" tab.

4.6 Configuring hydraulic axes

4.6.1 Configuring hydraulic axes - inserting an axis

Requirements in HW Config

- A Technology CPU was configured
- An analog control output (analog output module, or analog drive interface IM 174/ADI4) is configured
- An encoder is configured (analog encoder, incremental or absolute encoder)
- The configuration was saved in HW Config

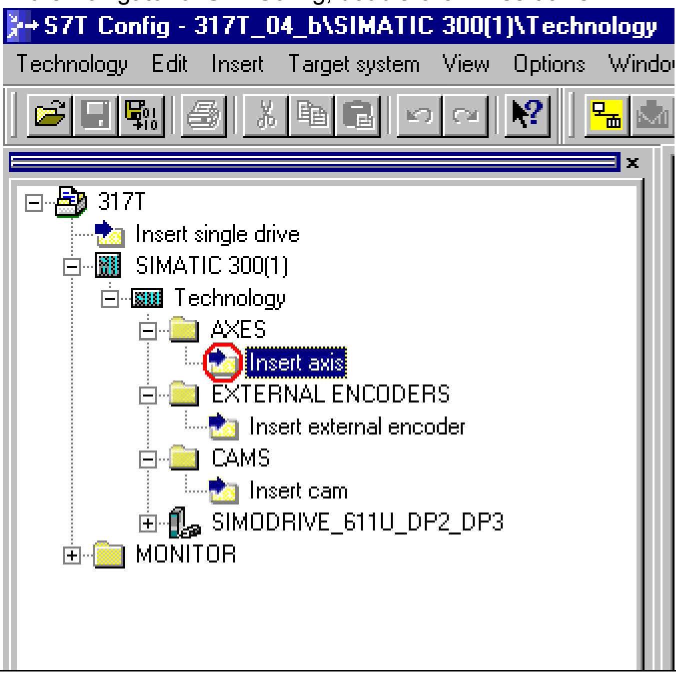
Note

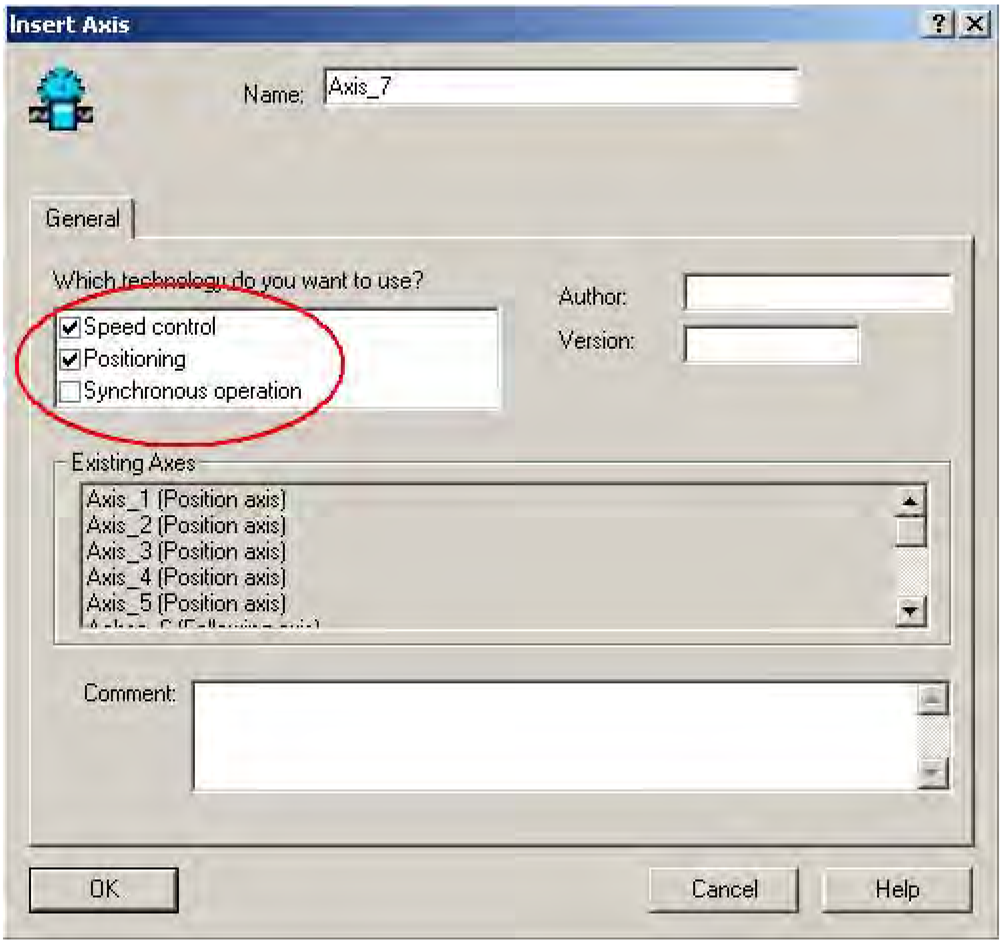
Set "Standard telegram 3" in HW Config when using analog drive interface IM 174/ADI4.

Call "DP Slave properties" in HW Config by double-clicking the IM 174/ADI4 object. Select the "Standard telegram 3" from the "Default" drop-down list in the "Configuration" tab.

Inserting a new axis in S7T Config

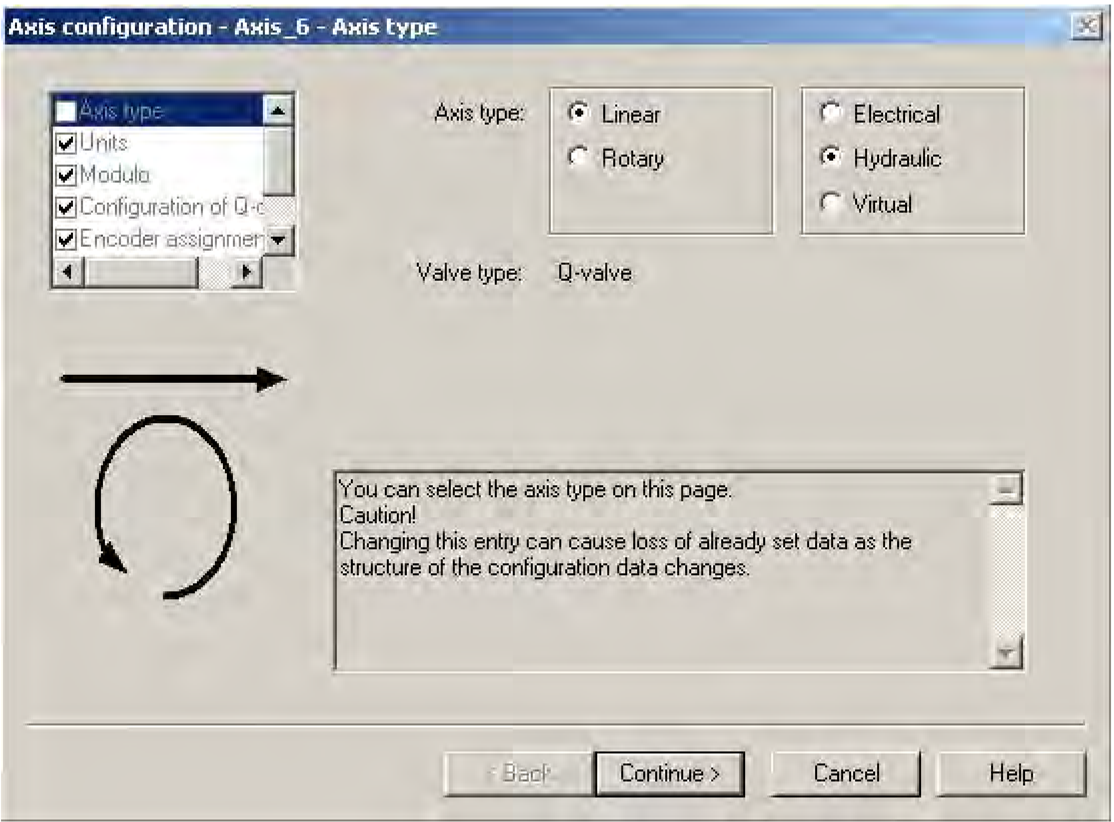
The next steps show the basic procedure of inserting and configuring an axis, based on the example of a hydraulic positioning axis. A hydraulic axis can only be inserted as real axis.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1.</td><td>In the Navigator of S7T Config, double-click "Insert axis".

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

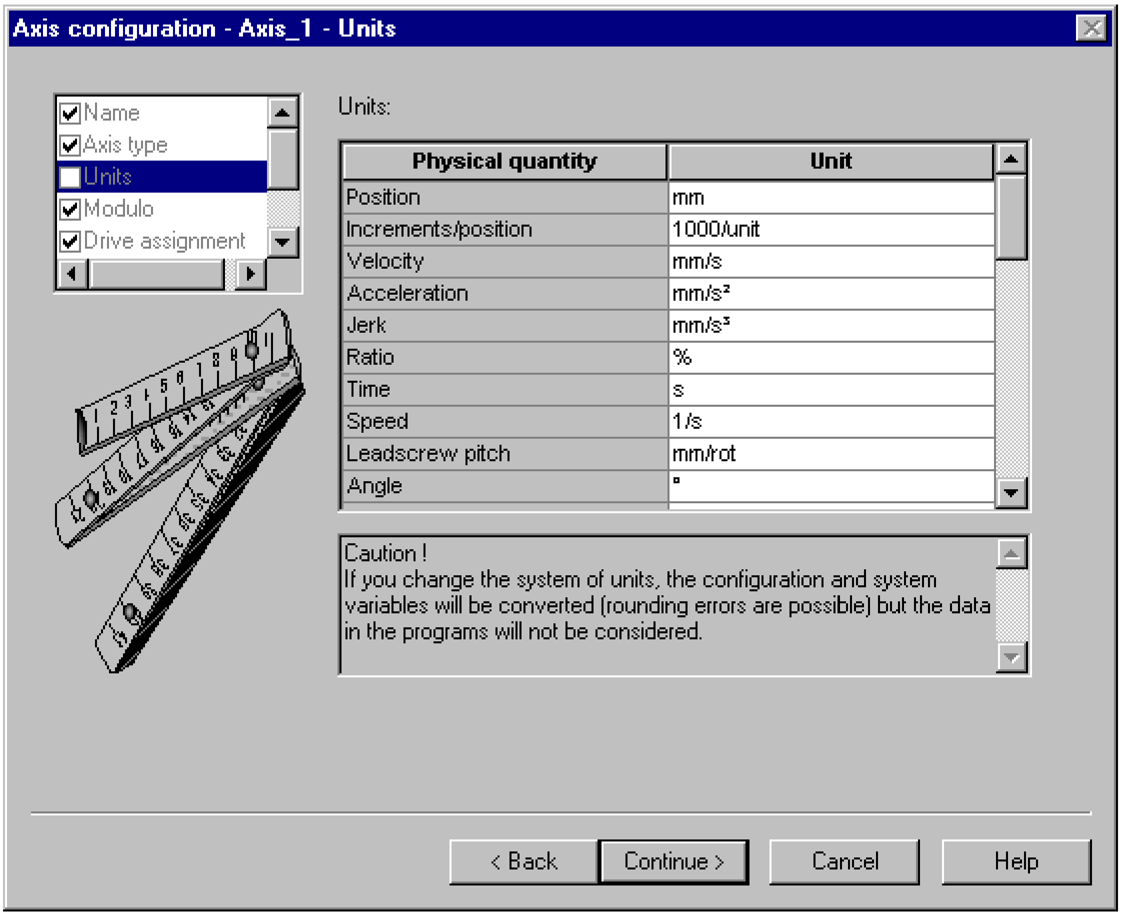
Step	Description
2.	Select the "Speed control" and "Positioning" technologies in the "General" tab.  You can edit the name, enter an author, define a version name, and enter a comment. "Technology Objects Management" applies the name of the technology object as symbolic name to the technology DB of the object in STEP 7.
3.	Click "OK" - this opens the axis wizard.

Note

You define the axis technology (speed, positioning, synchronous operation) once when you configure a new axis. In order to change the axis technology at a later time, you must delete and recreate the axis.

4.	Select a "linear," or "rotary" as well as "hydraulic" axis type in the next dialog box. The valve type is defined as a "Q valve".  Click "Next".

5. Select the physical units for axis configuration and for the user program from the list in the next dialog box.

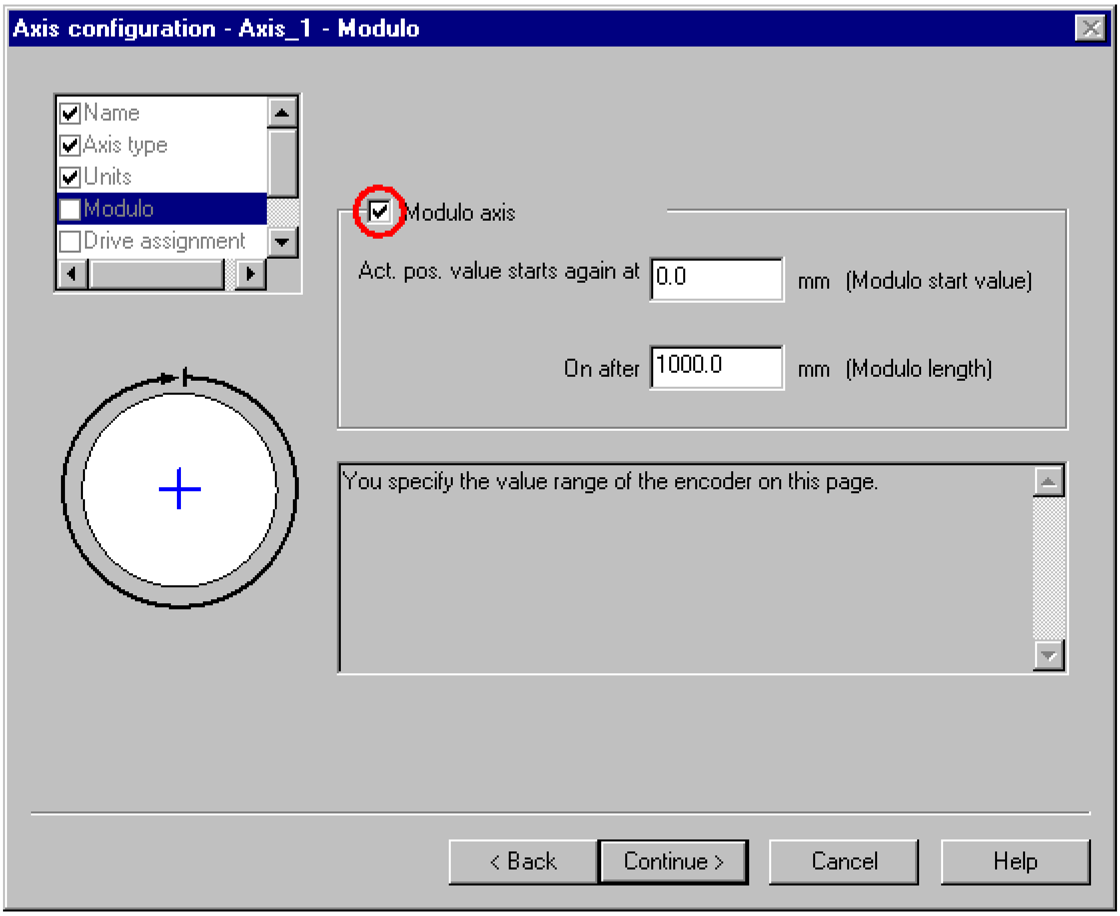


Note:
The physical unit settings only apply to this axis. The list may also show physical units which are irrelevant to this axis due to the axis configuration.

Example: The value "1000.0" with positioning unit in [mm] set at input parameter *Position* of the MC_MoveAbsolute" technology function is equivalent to a position value of 1000.0 mm.

Caution:
Numerical values set in the user programs (for motion commands, for example) are **not** converted into the new units if you change the unit settings. The values always refer to the configured unit!

Click "Next".

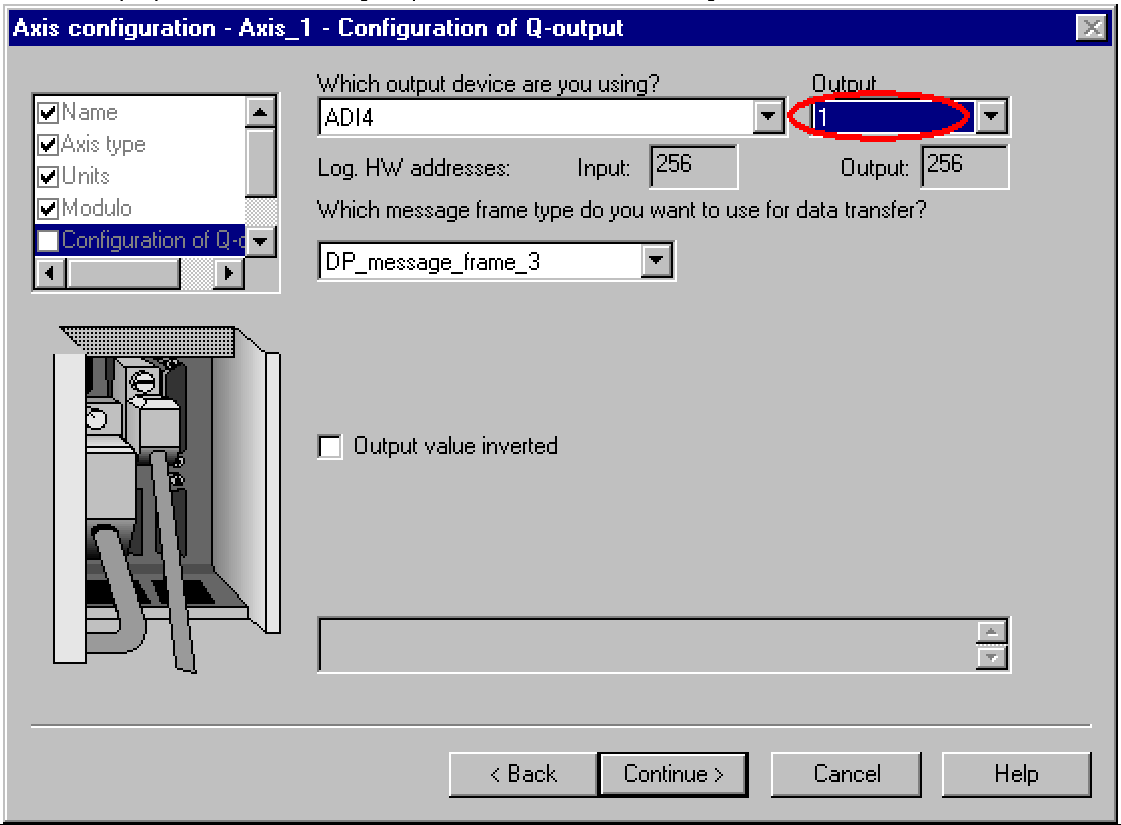
6.	Set the "Modulo axis" check box, and then type in the required modulo length and start value if you are going to operate the axis in modulo mode.  Click "Next". For further information about the procedure, refer to the chapters listed below: • "Configuring hydraulic axes - Q output via IM 174/ADI4" (Page 193) • "Configuring hydraulic axes - Q output via analog output module" (Page 199)

4.6.2 Configuring hydraulic axes - Q output via IM 174/ADI4

Configuring the Q output for analog drive interface IM 174/ADI4

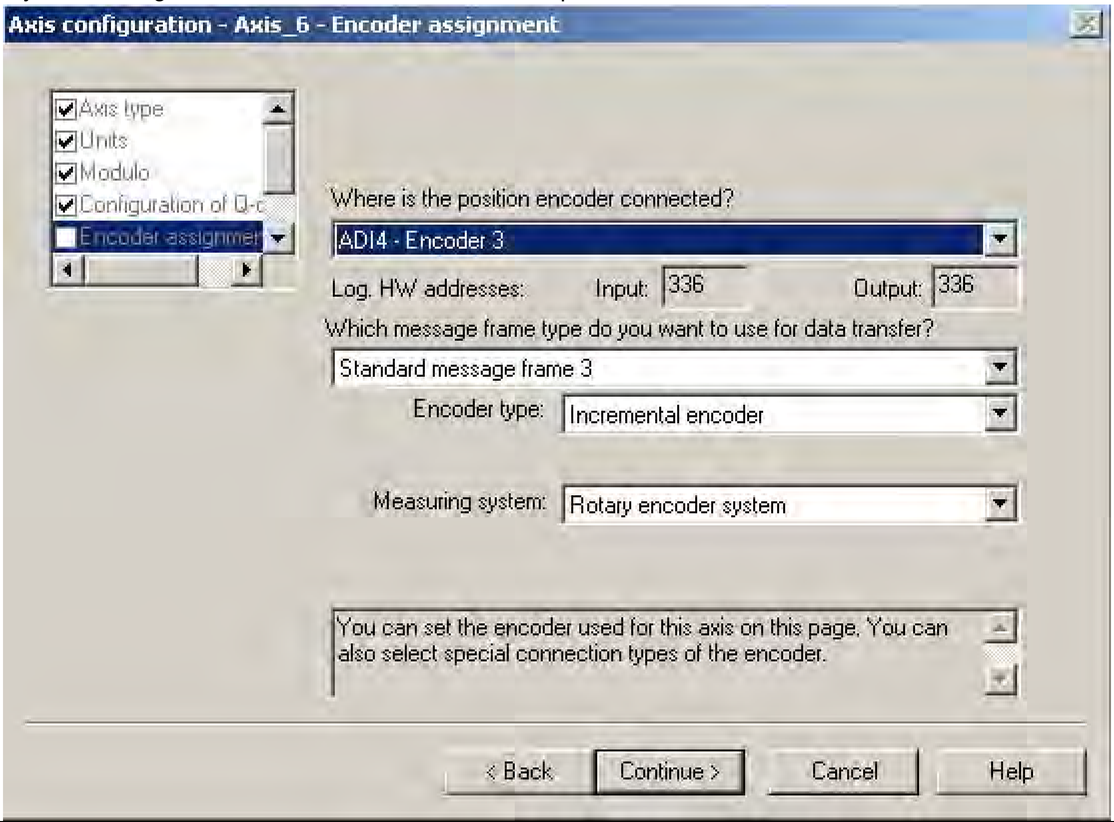
The description below is based on the chapter "Configuring hydraulic axes - Inserting an axis".

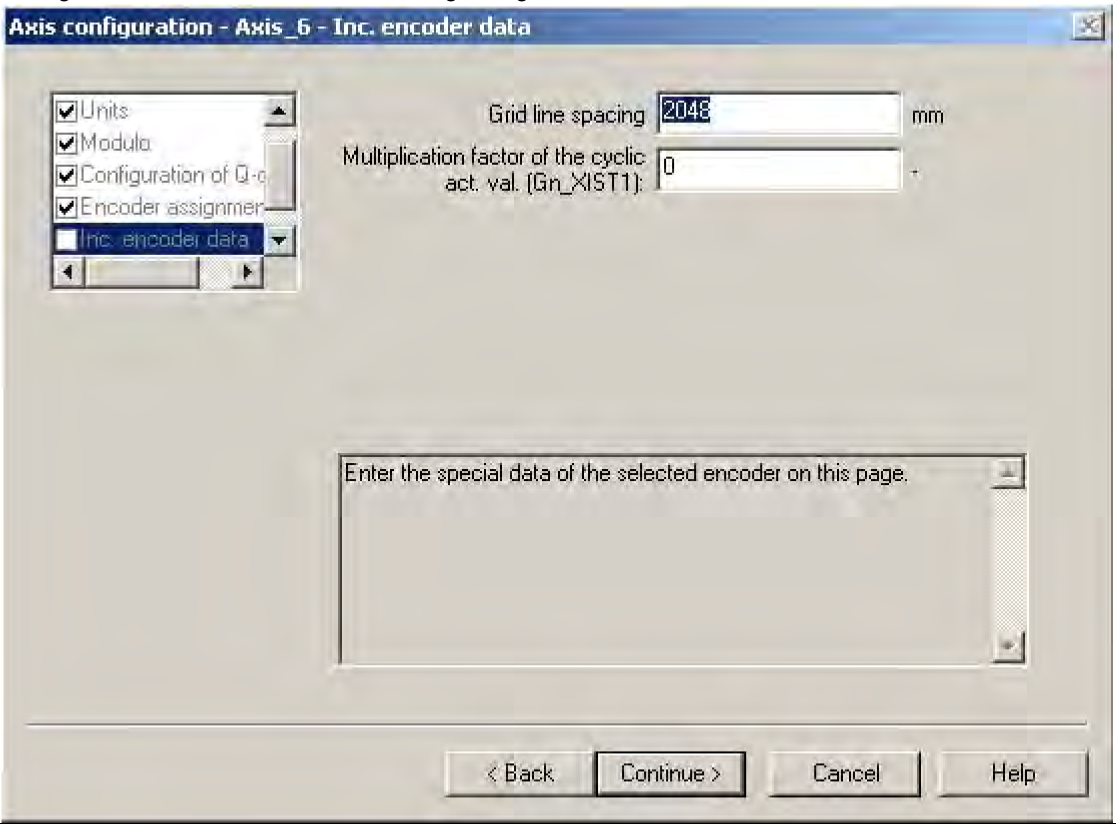
1. Define the properties of the analog output module in the next dialog box.

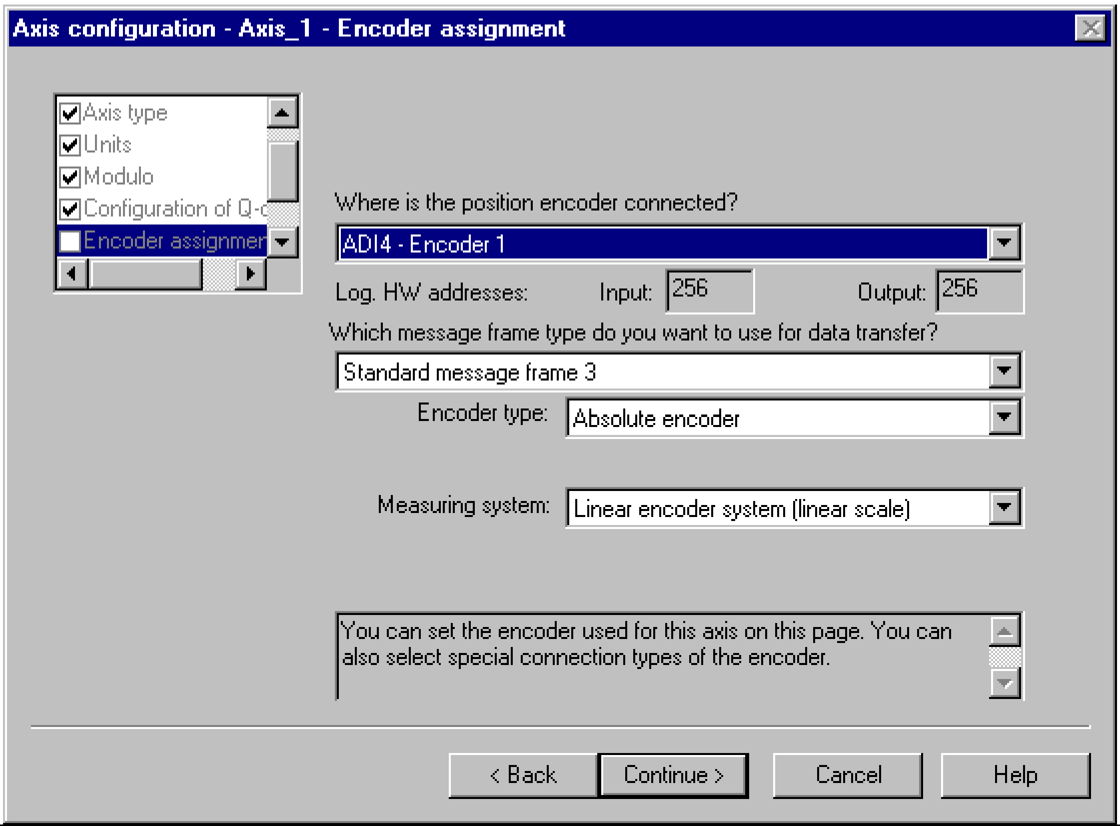


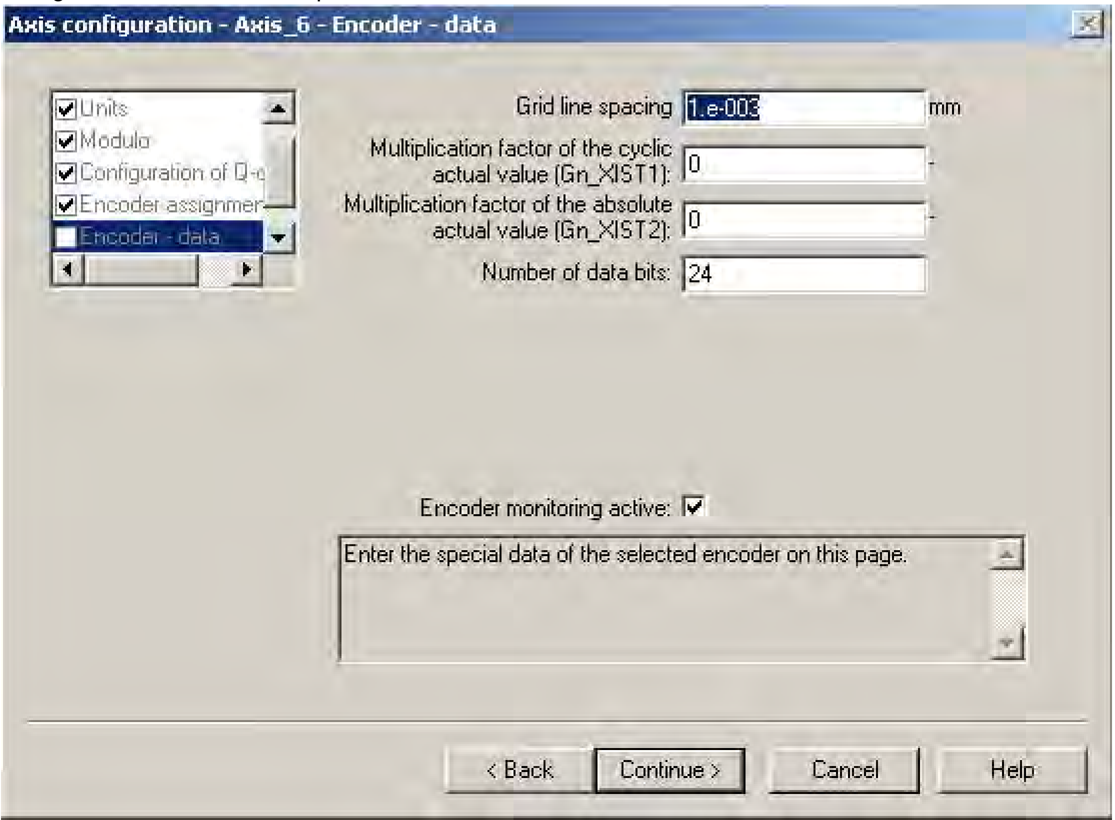
Which output device are you using?	Select the analog drive interface IM 174/ADI4.
Output	Select the analog output of the drive interface.
log. HW addresses: Input, output	The logical addresses can not be edited, as these are defined by the selection of the analog output.
Which message frame type do you want to use for the data transfer?	Only "Standard telegram 3" (here, "DP_Tel3") is permitted as message frame type. The message frame must match the selection in HW Config.
Output of the inverted value	Set this check box to set invert control output to the hydraulic drive.

Click "Next" when you completed your selection.

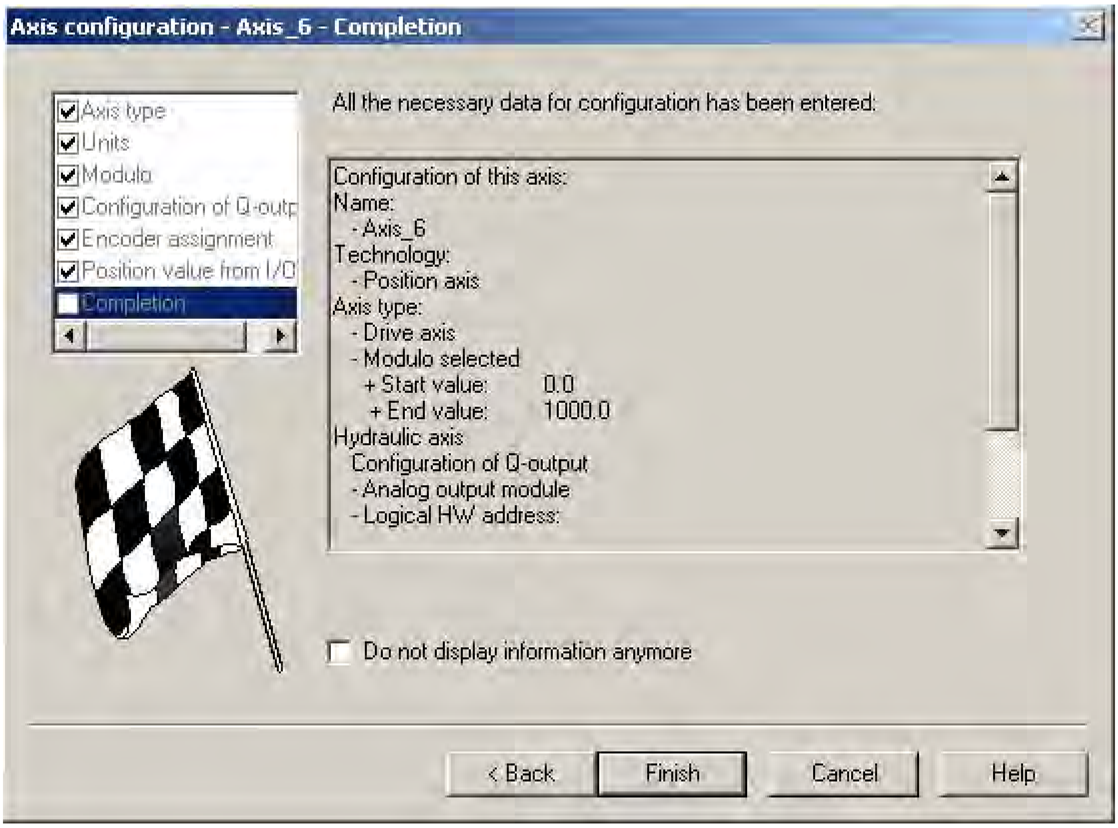
<table>
<tr><td rowspan="7">2a.</td><td colspan="2">Assign an incremental sensor to the hydraulic axis in the subsequent dialog box. The encoder must support "rectangular TTL" mode.
For information on using analog absolute encoders, refer to the chapter Configuring hydraulic axes - Q output via analog output module (Page 199).
If you are using an absolute encoder, continue at step 2b.

</td></tr>
<tr><td>Where is the position encoder connected?</td><td>Select the encoder interface of drive interface module IM 174/ADI4.</td></tr>
<tr><td>log. HW addresses: Input/output:</td><td>The logical addresses can not be edited, as these are defined by the encoders selected.</td></tr>
<tr><td>Which message frame type do you want to use for the data transfer?</td><td>Only "Standard telegram 3" (here, "DP_Tel3") is permitted as message frame type.</td></tr>
<tr><td>Encoder type:</td><td>Select "Incremental encoder" from the drop-down list.</td></tr>
<tr><td>Measuring system:</td><td>Select "Linear encoder system" or "Rotary encoder system", depending on the encoder used.</td></tr>
<tr><td colspan="2">Click "Next" after you completed your settings.</td></tr>
</table>

<table>
<tr><td rowspan="4">3a.</td><td colspan="2">Configure the encoder data in the following dialog box.

</td></tr>
<tr><td>Encoder resolution:</td><td>For information on the encoder resolution, refer to the encoder documentation.</td></tr>
<tr><td>Multiplication factor of the cyclic actual value</td><td>Calibrate the multiplication factor of the cyclic actual value based on the values used in the configuration of the drive component.</td></tr>
<tr><td colspan="2">Click "Next" after you completed your settings.
You have successfully completed the axis configuration.</td></tr>
</table>

<table>
<tr><td rowspan="7">2b.</td><td colspan="2">Assign an absolute encoder to the hydraulic axis in the subsequent dialog box. The encoder must support "SSI" mode.

</td></tr>
<tr><td>Where is the position encoder connected?</td><td>Select the encoder interface of drive interface module IM 174/ADI4.</td></tr>
<tr><td>log. HW addresses: Input/output:</td><td>The logical addresses can not be edited, as these are defined by the encoders selected.</td></tr>
<tr><td>Which message frame type do you want to use for the data transfer?</td><td>Only "Standard telegram 3" (here, "DP_Tel3") is permitted as message frame type.</td></tr>
<tr><td>Encoder type:</td><td>Select "Absolute encoder" or "Absolute encoder cyclic absolute" from the drop-down list</td></tr>
<tr><td>Measuring system:</td><td>Select "Linear encoder system" or "Rotary encoder system", depending on the encoder used.</td></tr>
<tr><td colspan="2">Click "Next" after you completed your settings.</td></tr>
</table>

<table>
<tr><td rowspan="8">3b.</td><td colspan="2">In the next dialog box, define the grid division of the used encoder as well as the multiplication factor of the cyclic actual value. Calibrate the multiplication factor of the cyclic actual value based on the values used in the configuration of the drive component.

</td></tr>
<tr><td>Grid spacing</td><td>The grid spacing describes the distance per encoder pulse.</td></tr>
<tr><td>Multiplication factor of the cyclic actual value</td><td>Encoders connected to a drive can provide pulses with a higher resolution than the actual encoder resolution (fine resolution).
The multiplication factor of the cyclic actual value is set to 0 by default. The system automatically uses multiplication factor 2048 (2^{11}).
Example of a deviating value:
The absolute encoder connected to the drive returns 4096 (2^{12}) pulses per revolution (encoder resolution). In this case, enter multiplier 12 for the cyclic actual value.</td></tr>
<tr><td>Multiplication factor of the absolute actual value</td><td>The encoder actual value transferred at power on (startup) is multiplied by the multiplication factor of the absolute actual value.
The multiplication factor 0 is set by default for the absolute actual value. The system automatically uses multiplication factor 512 (2^9.</td></tr>
<tr><td colspan="2">The multiplication factors also have to be set at the drive!
• At SINAMICS S120: Parameters p0418 and p0419 (for encoder 1);
(effective value can be read in r0979.3 and r0979.4)
• At SIMODRIVE 611U: Parameters 1042 and 1043 (for encoder 1)
• At Masterdrives MC: Parameter 171</td></tr>
<tr><td>Number of the data bits:</td><td>Total number of pulses
Example:
4096 pulses = 2^{12}
Exponent 12 = 12 data bits</td></tr>
<tr><td colspan="2">Click "Next" after you completed your settings.</td></tr>
</table>

<table>
<tr><td>End</td><td>You have successfully completed the axis configuration.



Exit the axis configuration dialog by clicking "Finish".</td></tr>
</table>

Note

Conditions of enabling the hydraulic axis with position control using technology function "MC_Power":

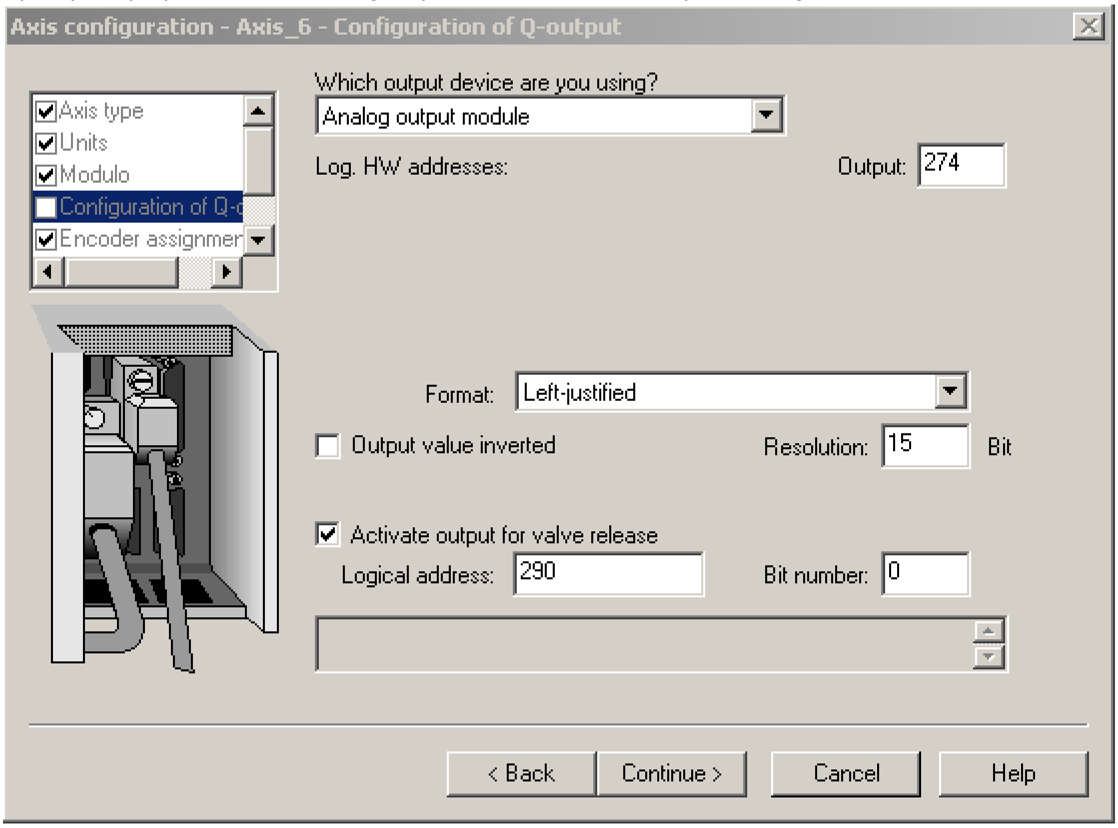
- The hydraulic axis must be assigned a cam disk as valve profile in S7T Config.
- The valve profile must be activated by the "MC_SetCharacteristic" technology function.

4.6.3 Configuring hydraulic axes - Q output via analog output module

Configuring the Q output for an analog output module of ET 200M or ET 200S

The description below is based on the chapter "Configuring hydraulic axes - Inserting an axis (Page 188)".

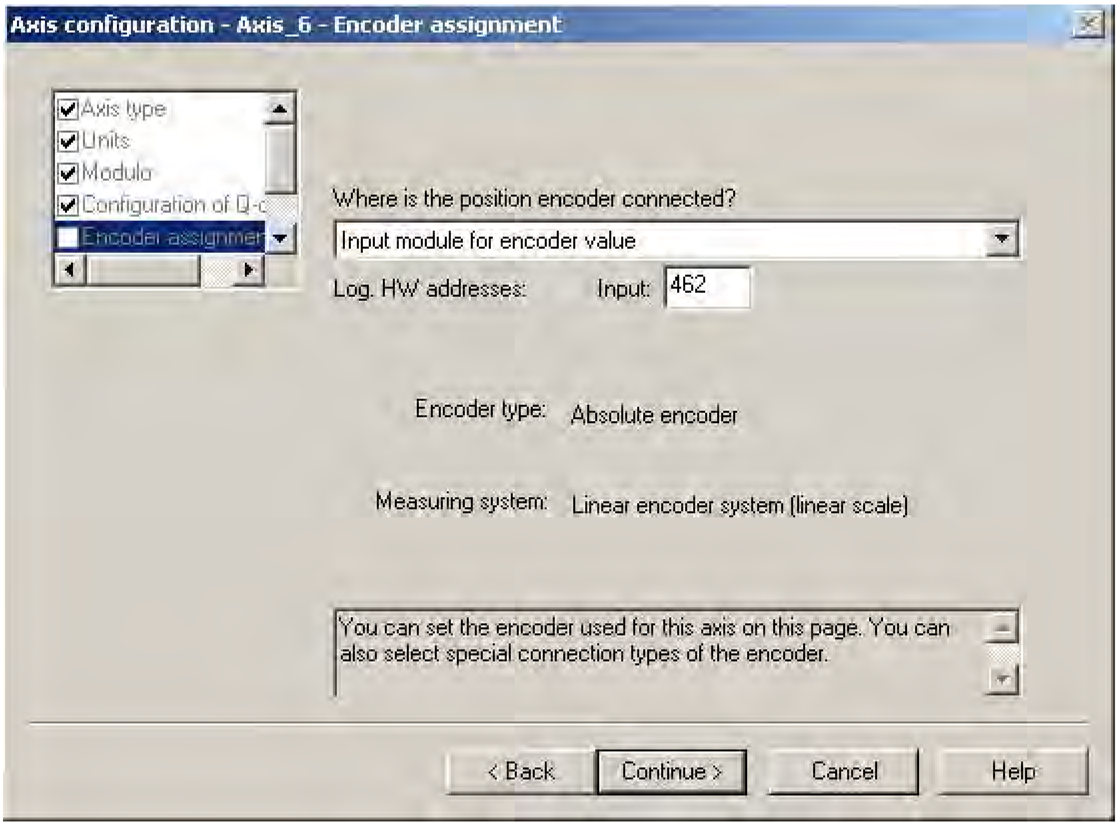
1. Specify the properties of the analog output module in the subsequent dialog box.



Which output device are you using?	Select "analog output module" from the drop-down list.
log. HW addresses: Output:	Enter the HW address of the analog output module of ET 200M or ET 200S.
Format:	Select the "left aligned" format for the analog output modules of ET 200M and ET 200S.
Output of the inverted value:	Set this check box if the analog value is reciprocal to the flow.
Resolution:	For information on the resolution, refer to the hardware catalog of "HW Config" (unsigned resolution).
Activate output for valve enable:	Set this check box to use the enable status of technology function "MC_Power" as control signal.
Logical address:	If the "Valve enable output" check box is activated, you can set the logical address of the enable output signal here.
Bit number:	Enter the bit number for the enable output signal.

Click "Next" after you completed your settings.

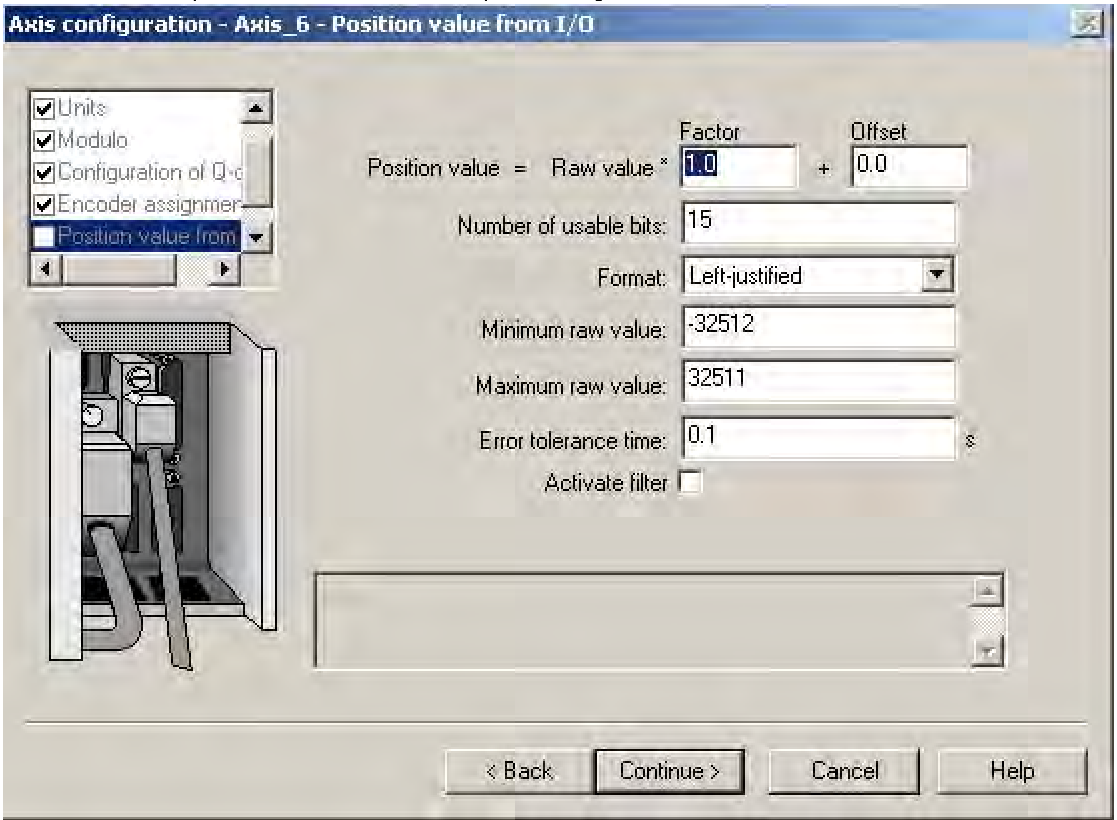
2. Parameterize an analog absolute encoder for the hydraulic axis in the subsequent dialog boxes. For information on using incremental encoders (rectangular TTL) and absolute encoders (SSI), refer to the chapter "Configuring hydraulic axes - Q output via IM 174/ADI4 (Page 193)".



Where is the position encoder connected?	Select "Encoder value input module" from the drop-down list.
log. HW addresses: Input:	Enter the HW address of the analog input module of ET 200M or ET 200S.

Click "Next" after you completed your settings.

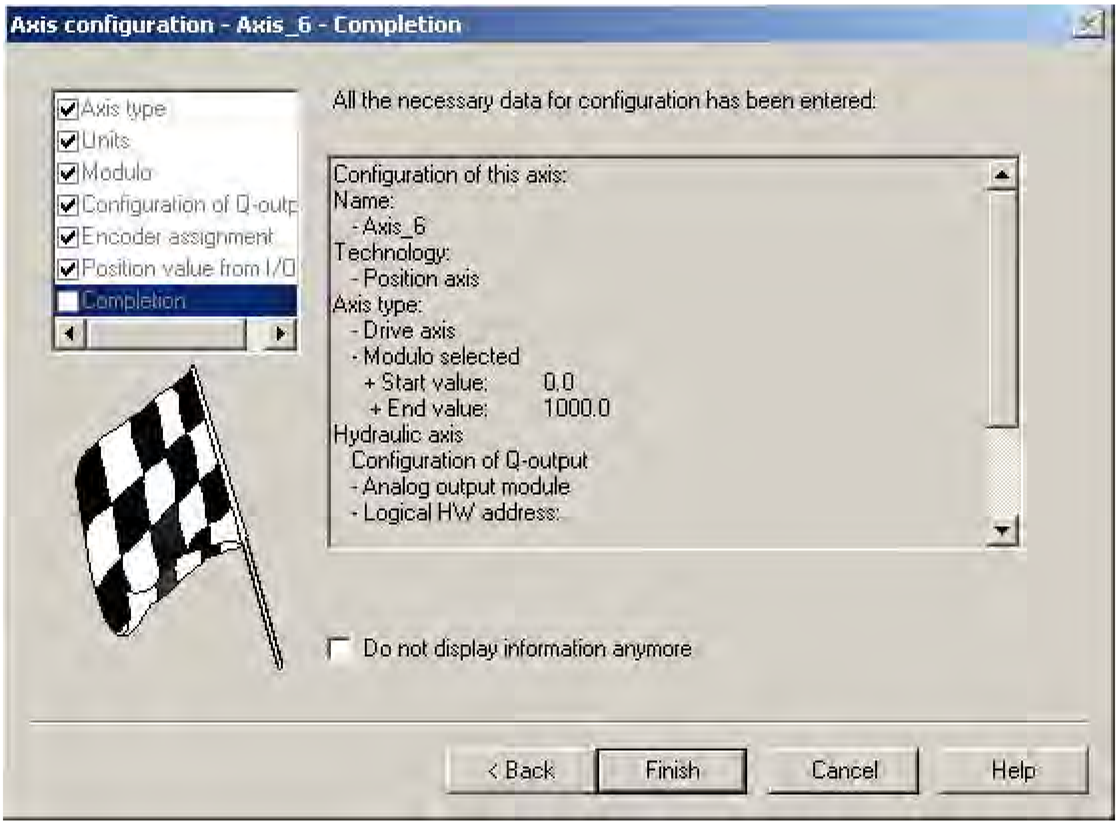
3. Parameterize the position value in the subsequent dialog box.



Factor: / Offset:	Set the *Factor* to define the gradient of the position value, and set the *Offset* to define the shift.
Number of utilizable bits:	Enter the number of utilizable data bits of the analog module.
Format:	Select "left aligned" for the analog modules of ET 200M and ET 200S.
minimum raw value:	Enter the minimum raw value of the analog module.
maximum raw value:	Enter the maximum raw value of the analog module.
Error tolerance time:	Enter an error tolerance time.
Activate filter:	Set the check box to apply a PT1 filter to the analog value of the position.
Time constant PT1 filter	You can enter a time constant for the PT1 filter if the "Activate filter" check box was set.

Click "Next" after you completed your settings.

<table>
<tr>
<td>4.</td>
<td>You have successfully completed the axis configuration.



Exit the axis configuration dialog by clicking "Finish".</td>
</tr>
</table>

Note

Conditions of enabling the hydraulic axis with position control using technology function "MC_Power":

- The hydraulic axis must be assigned a cam disk as valve profile in S7T Config.
- The valve profile must be activated by the "MC_SetCharacteristic" technology function.

4.6.4 Determining and adding a valve profile

The non-linearity between the valve's manipulated variable (-10 V to +10 V, for example) and the velocity of the hydraulic axis is mapped using the valve profile. The "cam disk" technology object is used as valve profile. The valve profile is formed by pairs of values which map the valve position relative to the velocity reached.

Determining a valve profile using the Symbol Browser and the Trace tool

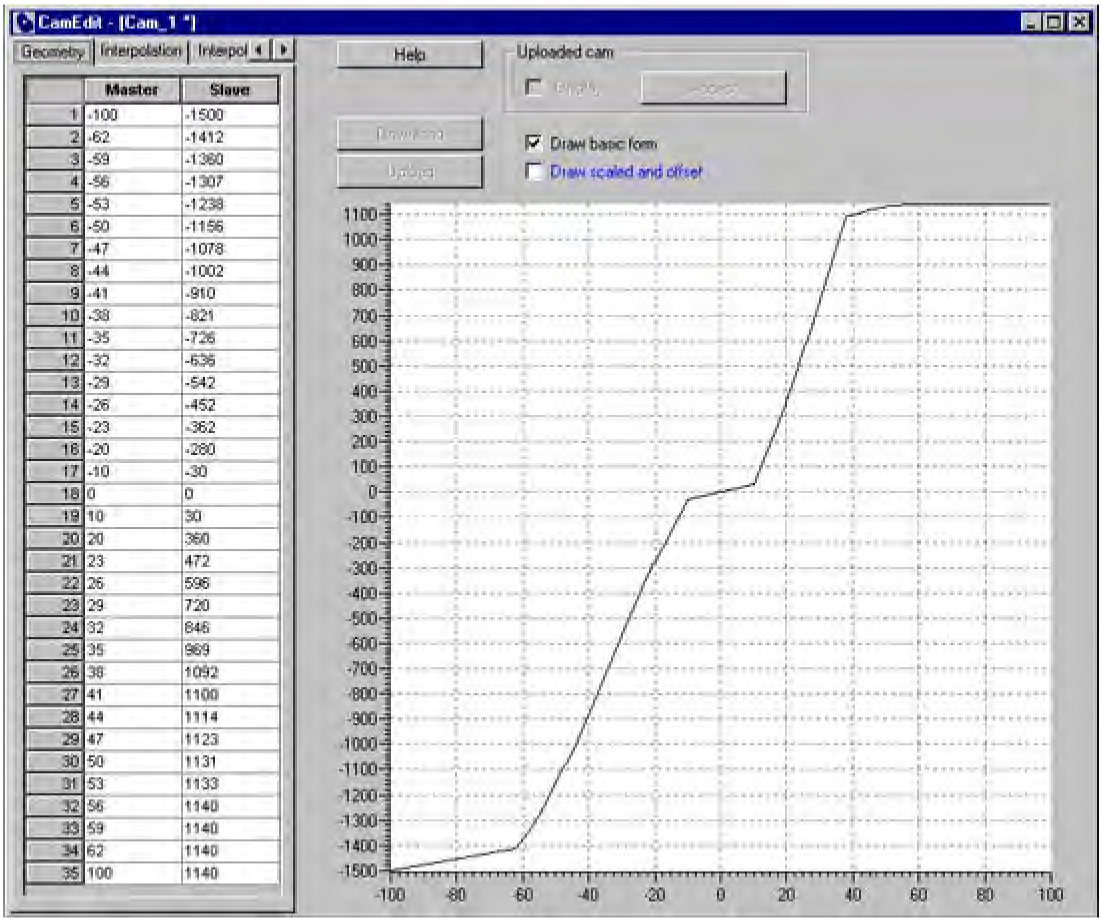
1.	Verify that the hydraulic axis can be moved (hydraulic pressure present, valve ready, etc.).
2.	Verify that the hydraulic axis was not assigned a valve profile ("Profiles" setting in the Navigator of S7T Config).
3.	Enable the hydraulic axis by calling the "MC_Power" technology function in *Mode = 2*.
4.	Program the following variables in S7T Config for the recording in Trace: • *actordata.qoutputvalue* - Manipulated variable as percentage relative to the Q output • *sensordata.sensordata[n].velocity* - Velocity of the hydraulic axis
5.	Set system variable *servosettings.additionalqoutputvalueswitch* to *YES* in the symbol browser of S7T Config.
6.	Start Trace.
7.	Enter a small percentage at system variable *servosettings.additionalqoutputvalue* in the symbol browser to move the axis. Enter value *0* to stop the axis.
8.	Enter values between 0 % and 100 % in succession in order to obtain a sufficient number of interpolation points for the valve profile.
9.	In S7T Config, transfer the recorded values to the cam disk of the valve profile using CamEdit. Transfer the manipulated variable to the "Master" column and the velocity value to the "Slave" column.

Determining a valve profile using a STEP 7 program and Trace

1.	Verify that the hydraulic axis can be moved (hydraulic pressure present, valve ready, etc.).
2.	Verify that the hydraulic axis was not assigned a valve profile ("Profiles" setting in the Navigator of S7T Config).
3.	Enable the hydraulic axis by calling the "MC_Power" technology function in *Mode = 0*.
4.	Program the following variables in S7T Config for the recording in Trace: • *actordata.qoutputvalue* - Manipulated variable as percentage relative to the Q output • *sensordata.sensordata[n].velocity* - Velocity of the hydraulic axis
5.	Start Trace.
6.	Move the axis at different velocities using the "MC_MoveVelocity" technology function with *PositionControl = FALSE*. Caution: The velocity displayed in *sensordata.sensordata[n].velocity* does not necessarily coincide with the default velocity set at "MC_MoveVelocity" as you have not yet activated a valve profile. You can stop the axis at a defined position in order to prevent the drive from moving to a fixed end stop. Use technology function "MC_Stop" to set this position.
7.	Set several velocity values up to maximum in succession in order to obtain a sufficient number of interpolation points for the valve profile.
8.	In S7T Config, transfer the recorded values to the cam disk of the valve profile using CamEdit. Transfer the manipulated variable to the "Master" column and the velocity value to the "Slave" column.

Instead of using CamEdit to create the cam disk you can also use the "MC_CamClear", "MC_CamSectorAdd" and "MC_CamInterpolate" technology functions.

Example - Profile of a proportional directional valve

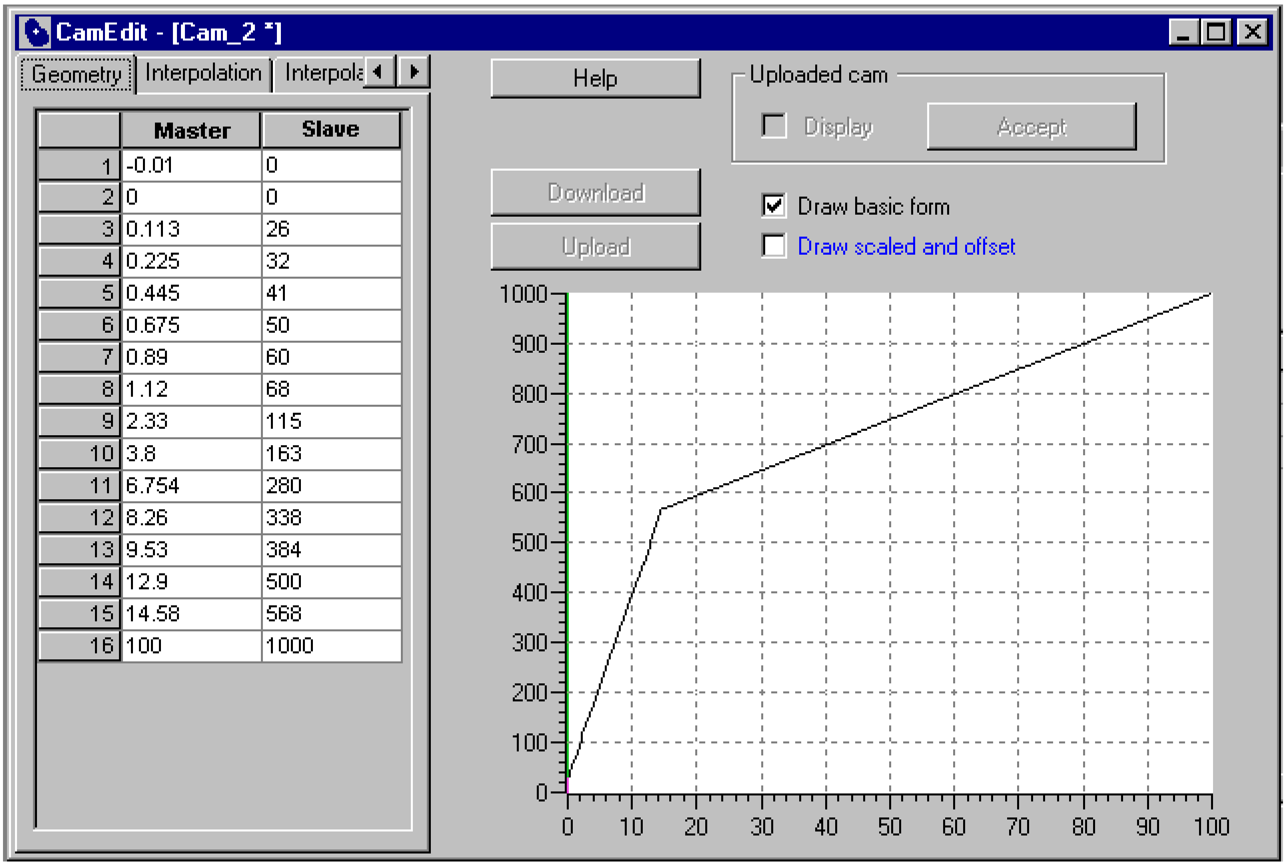


Enter the valve control output as a "%" unit in the "Master" column. The "Slave" column contains the corresponding velocity values of the hydraulic axis.

Example - Valve profiles for a variable speed pump with switching directional valve

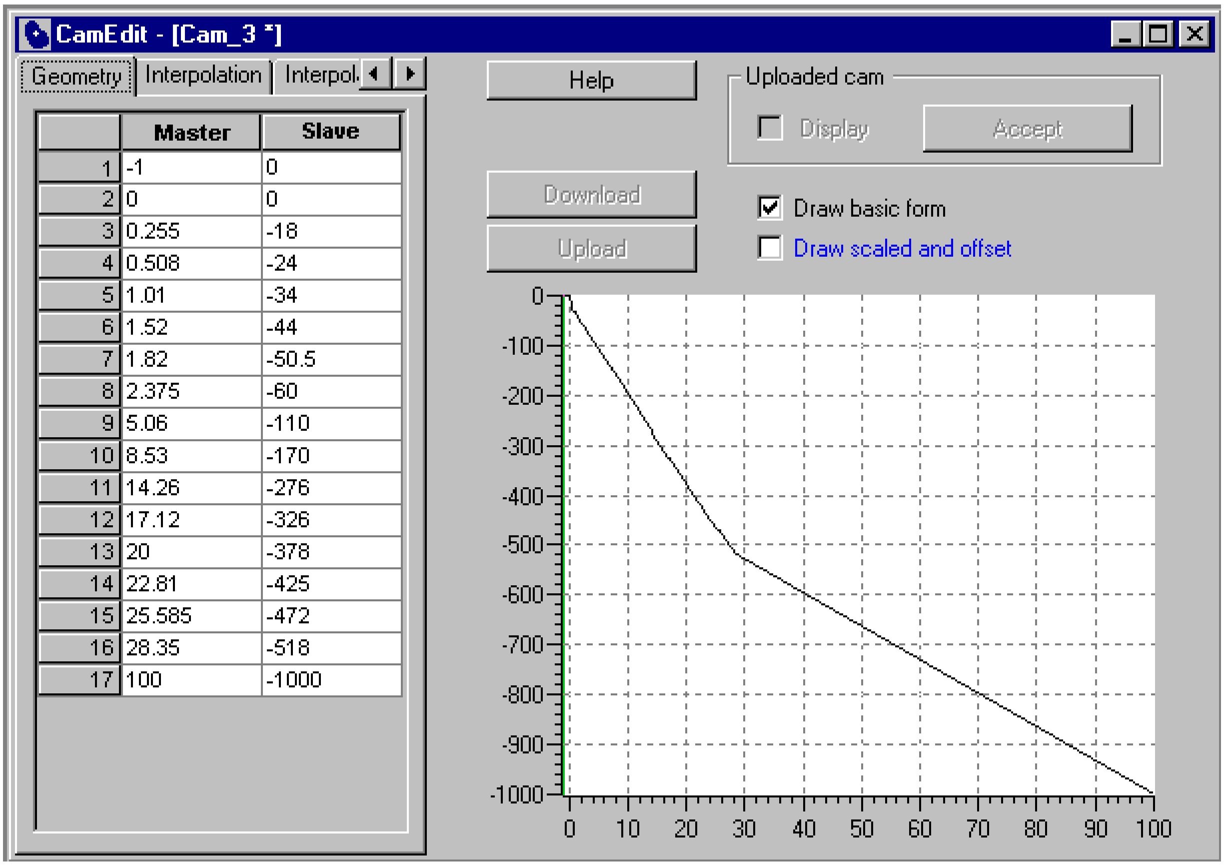
In a system containing a variable speed pump with switching directional valve, negative motion directions are achieved by controlling the switching directional valve. Use separate valve profiles for the positive and negative direction of movement.

Example of a valve profile for positive direction of movement:



Enter the valve control output as a "%" unit in the "Master" column. The "Slave" column contains the corresponding velocity values of the hydraulic axis. Any negative control values which may briefly develop due to control functions are limited to *0* in row 1.

Example of a valve profile for negative direction of movement:



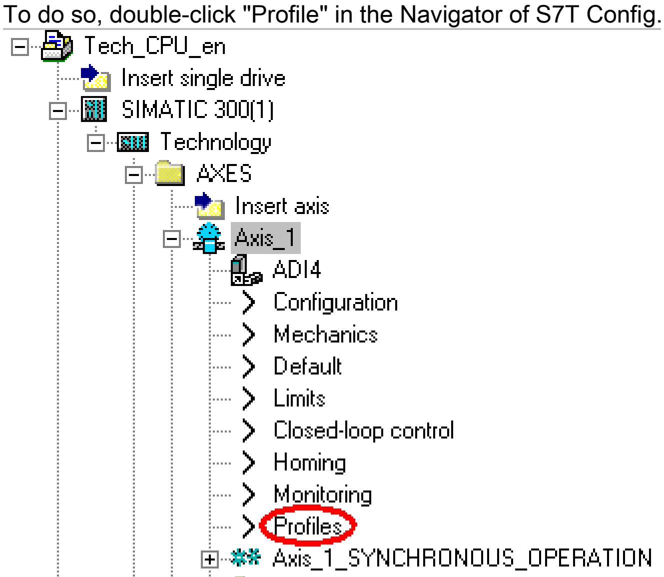
The structure of the valve profile for negative direction of movement corresponds to that of the positive direction of movement. The first value pair also limits negative manipulated variables to *0*.

Activate the relevant valve profile in the user program by calling the "MC_SetCharacteristic" technology function, depending on the direction of movement.

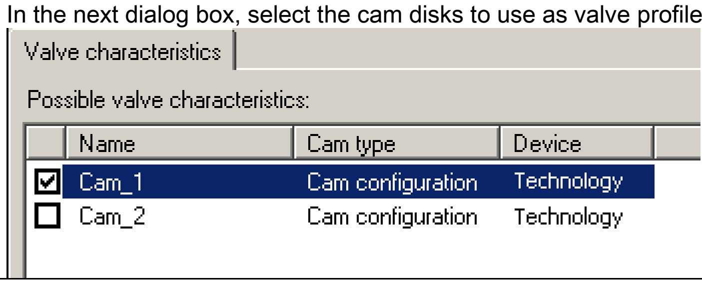
4.6.5 Assigning the valve profile

Assign the corresponding cam disk to the hydraulic axis as possible valve profile in order to let you activate a valve profile by calling the "MC_SetCharacteristic" technology function.

<table>
<tr><td>1.</td><td>To do so, double-click "Profile" in the Navigator of S7T Config.



</td></tr>
<tr><td>2.</td><td>In the next dialog box, select the cam disks to use as valve profile for this hydraulic axis.



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

4.6.6 Adding a data record for data record changeover

Introduction

The "MC_ChangeDataset" technology function can be used to perform a data record changeover at an axis. You can create data records for speed-controlled axes, positioning axes and synchronization axes. Virtual axes always have only one data record.

The data record changeover function can be used, for example, to toggle between the motor encoder and the machine encoder, or to edit controller parameters.

Note

When editing data record parameters, note that certain parameters must be identical in the data records:

- Parameters changing the structure (different controller types such as PV or PID controllers, for example)
- Important activating parameters (following monitoring on/off, DSC on/off, for example)

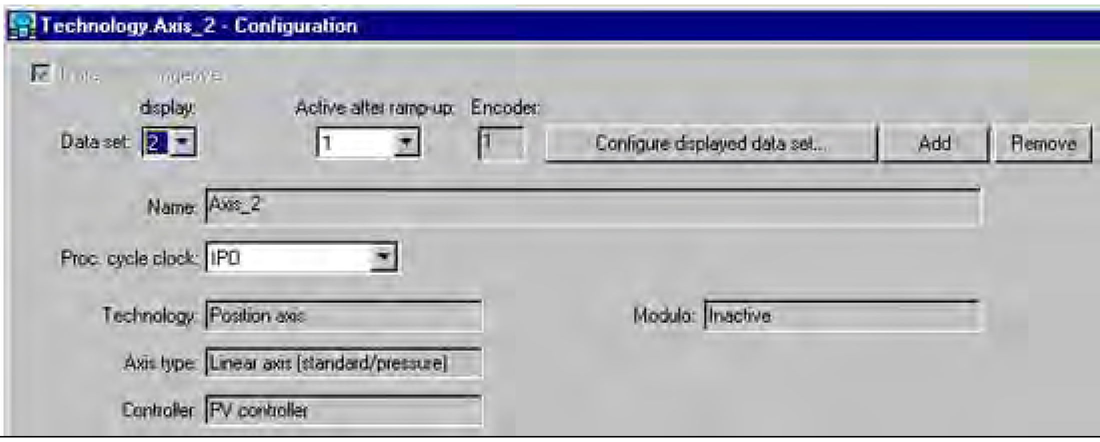
The consistency check reports any illegal technology parameter settings when you configure data records in S7T Config.

Prerequisite

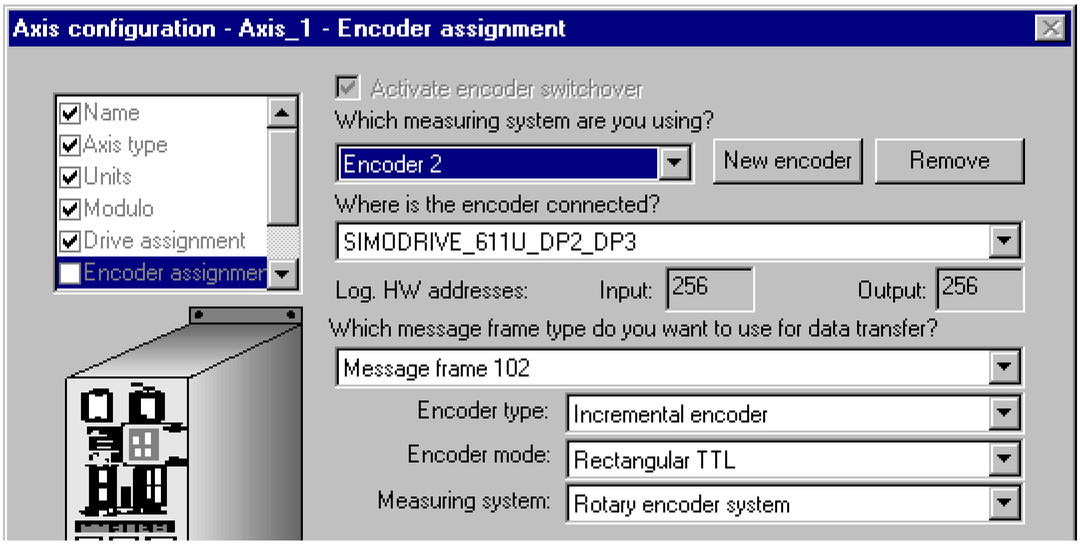
- The axis is configured.

Adding a new data record

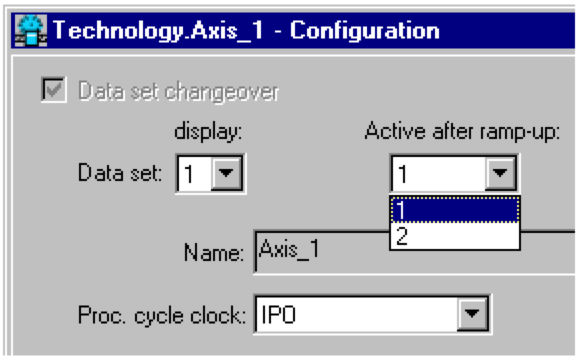
To configure a new data record:

Step	Description
1.	In the Navigator of S7T Config, open the configuration dialog box of the axis to which you want to add a data record.
2.	Set the "data record changeover" check box. The check box is hidden if the axis already contains several data records.
3.	Click "Add". Result: A new data record is added to the axis. 
4.	Click "Configure active data set." Result: The Axis Wizard appears.
5.	Use the axis wizard to configure the data record. Note: The technology used is set once when you create the axis. You can not use a second data record to change the axis technology.
6.	You may also perform an encoder changeover using the data record changeover function. Do so by setting the "enable encoder changeover" check box on the "Axis configuration - Encoder assignment" dialog box. Ignore this step and the next otherwise. Click "new encoder" to create the second encoder. Result: A drop-down list opens from which you can select an encoder for this data record.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>7.</td><td>
Select the connection and message frame type for the second encoder. In the final step, enter the encoder information ("Encoder type", "Encoder mode" and "Measuring system").
For the encoder configuration select settings from the documentation of the encoder or configuration settings of the drive components to which the encoder is connected.



Click "Next". Continue by editing the steps in the wizard to complete the configuration.
</td></tr>
<tr><td>8.</td><td>
In the axis configuration dialog box, select the data record to be loaded after CPU startup ("active after startup").



Result: The "Encoder" table shows an overview of the configured encoders of the axis.
</td></tr>
</table>

Deleting the data record of an axis

You start to delete a data record by selecting it from the axis configuration dialog box. Next, click "Delete". The data record is deleted.

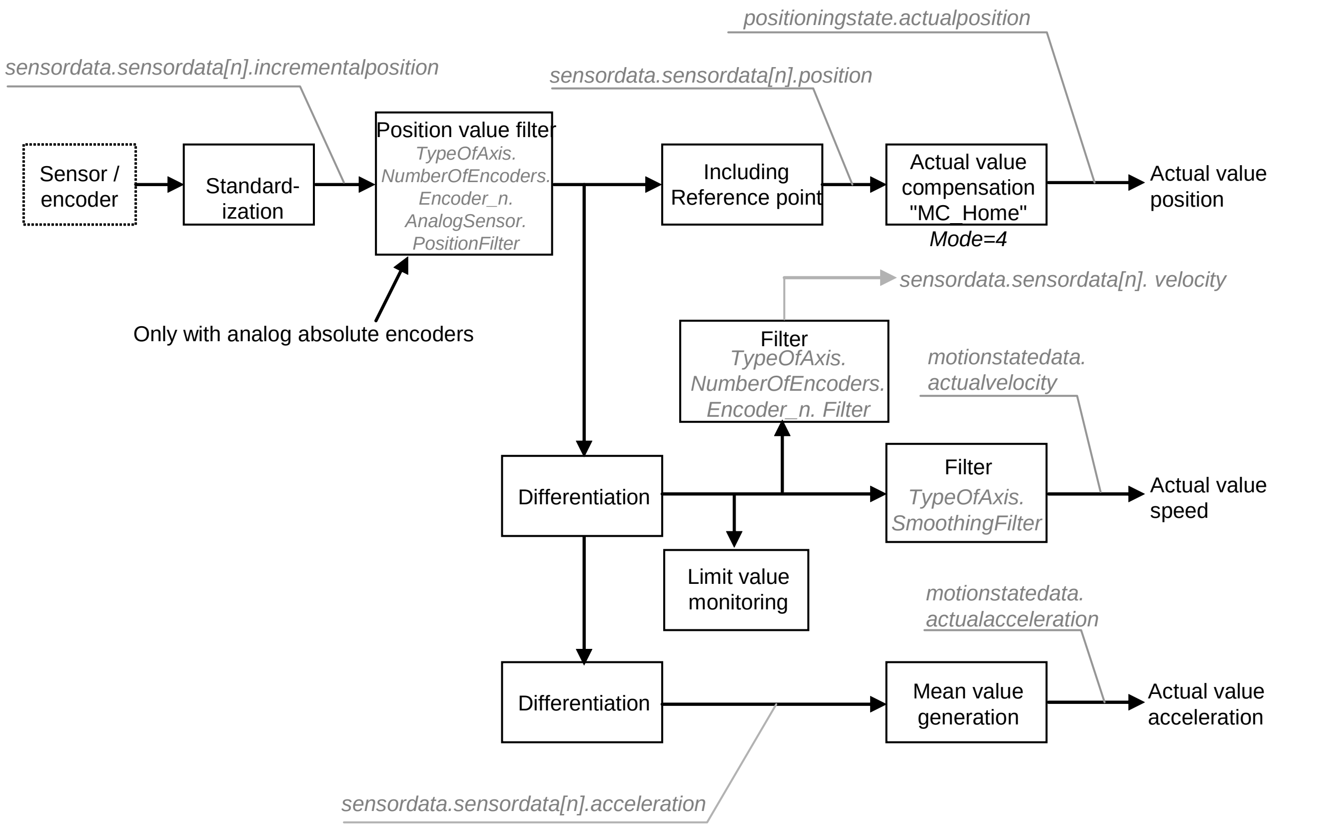
Selecting a data record

Select the data record to be edited from the "Display data record" drop-down list. This selection is available in all dialog boxes of the axis.

Select the data record which is used by default after startup from the "Data record active after startup".

4.6.7 Actual value logging

The diagram below shows the preparation of the actual value of an axis or external encoder in the Technology CPU.



Data are logged within the position control cycle. These data can be used to calculate further variables such as velocity and acceleration.

The system variables *sensordata.sensordata[n].incrementalposition* and *sensordata.sensordata[n].position* are calculated in the position control cycle clock. All other system variables are calculated in the corresponding execution cycle clock of the axis or external encoder. For external encoders, the system variables *sensordata.sensordata[n]....* are called *sensordata... .*

You use the *TypeOfAxis.NumberOfEncoders.Encoder_n.AnalogSensor.PositionFilter* configuration data element to activate or deactivate the actual position value. The filter is only available for analog absolute encoders:

Configuration data element	Significance	
TypeOfAxis.NumberOfEncoders.Encoder_n. AnalogSensor.PositionFilter.enable	*YES*	Actual position value activated
	NO	Actual position value deactivated
TypeOfAxis.NumberOfEncoders.Encoder_n. AnalogSensor.PositionFilter.timeConstant	Time constant for PT1 smoothing.	

The "sensor" velocity filter can be activated/deactivated by means of the *TypeOfAxis.NumberOfEncoders.Encoder_n.Filter* configuration data element:

Configuration data element	Significance	
TypeOfAxis.NumberOfEncoders.Encoder_n. Filter.enable	*YES*	Actual value filter activated
	NO	Actual value filter deactivated
TypeOfAxis.NumberOfEncoders.Encoder_n. Filter.timeConstant	Time constant for PT1 smoothing.	

(For external encoders, the configuration data element is called *TypeOfAxis.Encoder_1.Filter*... .

The actual velocity value filter can be activated/deactivated by means of the *TypeOfAxis.SmoothingFilter* configuration data element:

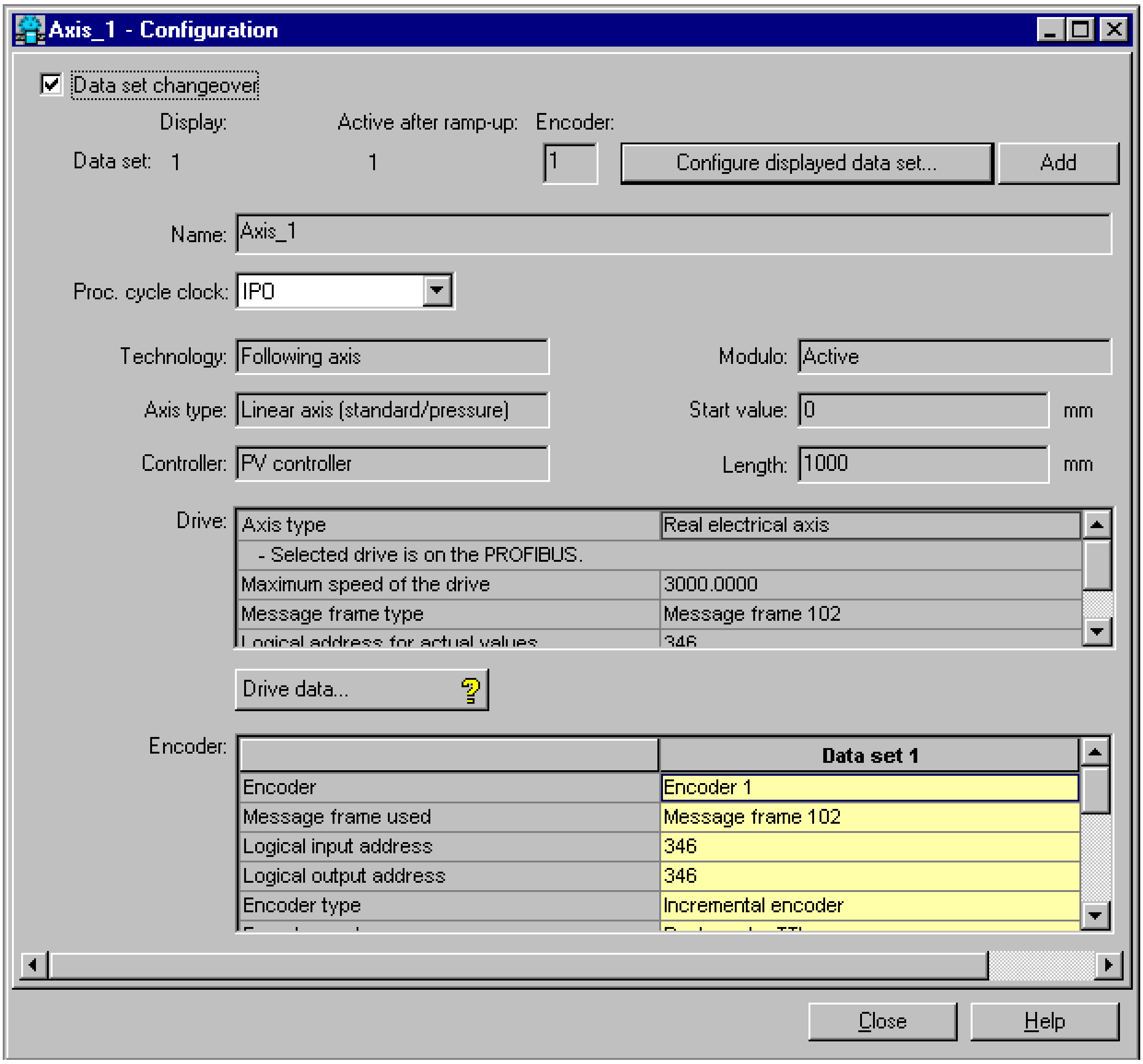
Configuration data element	Significance	
TypeOfAxis.SmoothingFilter.enable	*YES*	Actual value filter activated
	NO	Actual value filter deactivated
TypeOfAxis.SmoothingFilter.mode	You can set the filter calculation method at this configuration parameter:	
	DEFAULT_MODE	Averaging as a function of the ratio: Execution cycle of the axis / external encoder to the position control cycle Example: A *3 : 1* ratio is set between the execution cycle and the position control cycle. In this case, the mean value is formed in three position controller cycles.
	AVERAGING	Averaging using a time constant
	PT	PT1 smoothing using a time constant
TypeOfAxis.SmoothingFilter.timeConstant	Time constant for PT1 smoothing or "*AVERAGING*".	

4.6.8 Configuration

4.6.8.1 Configuration - Axis

The **Axis > Configuration** dialog box shows the axis and drive settings

You can edit these settings in a dialog box by clicking "Configure active data set".



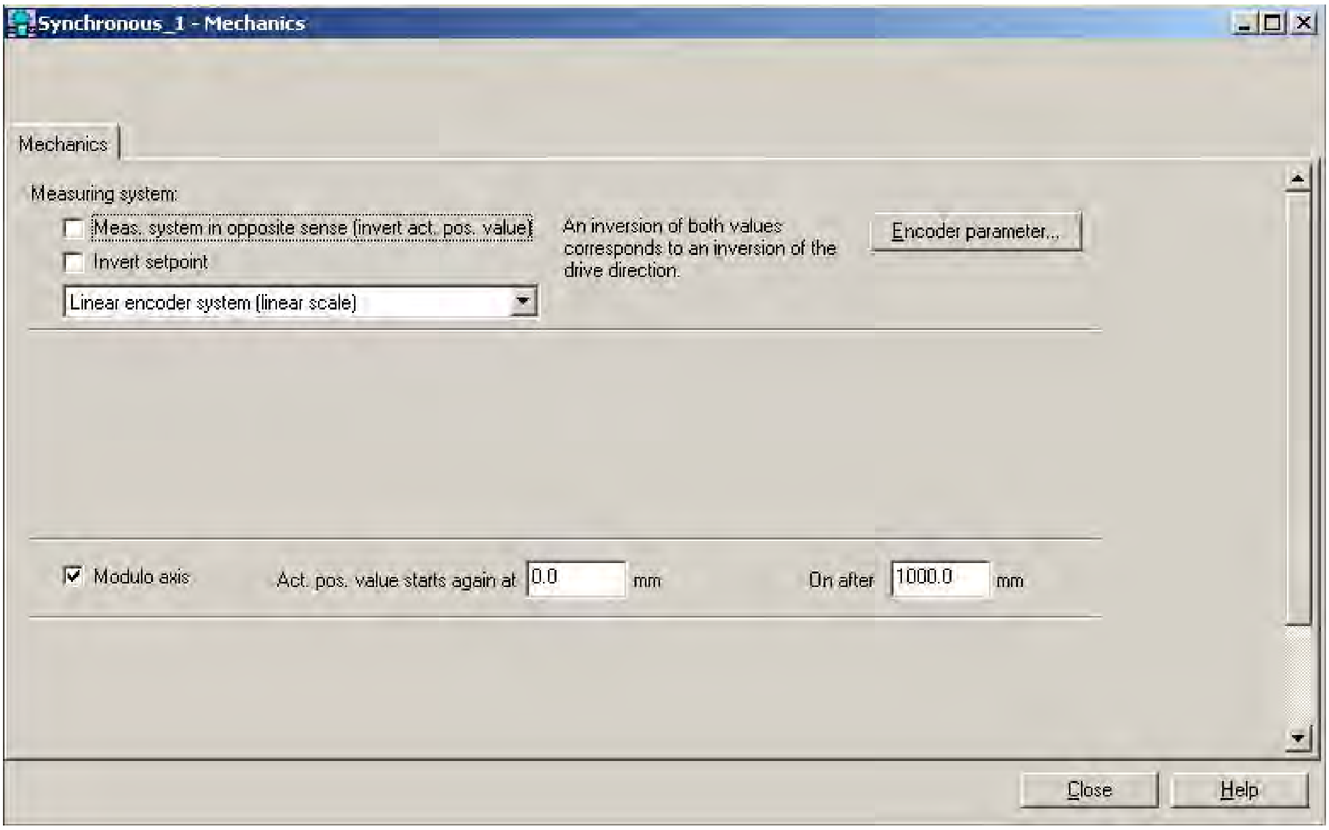
This dialog box also contains functions for managing the data records for a data record changeover. If the technology object supports multiple data records, you can:

- Create new data records
- Delete data records
- Define which data record to load for the technology object during CPU startup, and
- Reconfigure the selected data record.

4.6.9 Mechanical system

4.6.9.1 Mechanics - Hydraulic axis

You configure the mechanical parameters of the axis and encoder in the **Axis > Mechanics** dialog box.



Settings for the axis:

- Drive direction settings
- Leadscrew pitch of a linear axis
- Modulo settings
- Backlash compensation settings

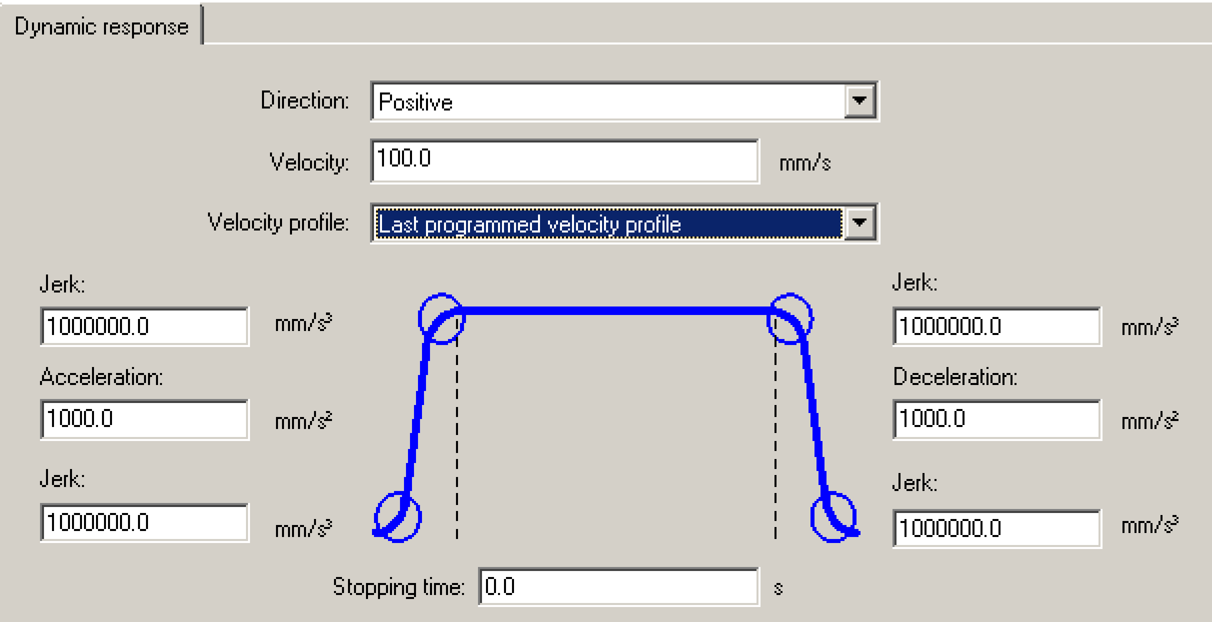
Settings for the encoder:

- Backlash compensation
- Encoder mounting type

4.6.10 Default

4.6.10.1 Default - "Dynamics" tab

The default values for axis dynamics can be set in the "Dynamics" tab of the **Axis > Default** dialog box. The default values are activated if a negative dynamic value is set at a technology function.



Default values supported:

- Direction
- Velocity
- Acceleration
- Deceleration
- Jerk
- Velocity profile

Stop time

The time set at "Stop time" is in effect when a moving axis is disabled, and if *Stopmode* = *2* is set at "MC_Power".

Velocity profile

The velocity profile defines the response of the axis during startup and when braking, and to velocity changes.

The technology functions do not use the default value. If you want to change the velocity profile, use the *Jerk* input parameter.

You can choose between the following profiles:

- Trapezoidal

  The trapezoidal profile is used for linear acceleration in positive and negative direction of the motion (jerk = 0).
- Constant

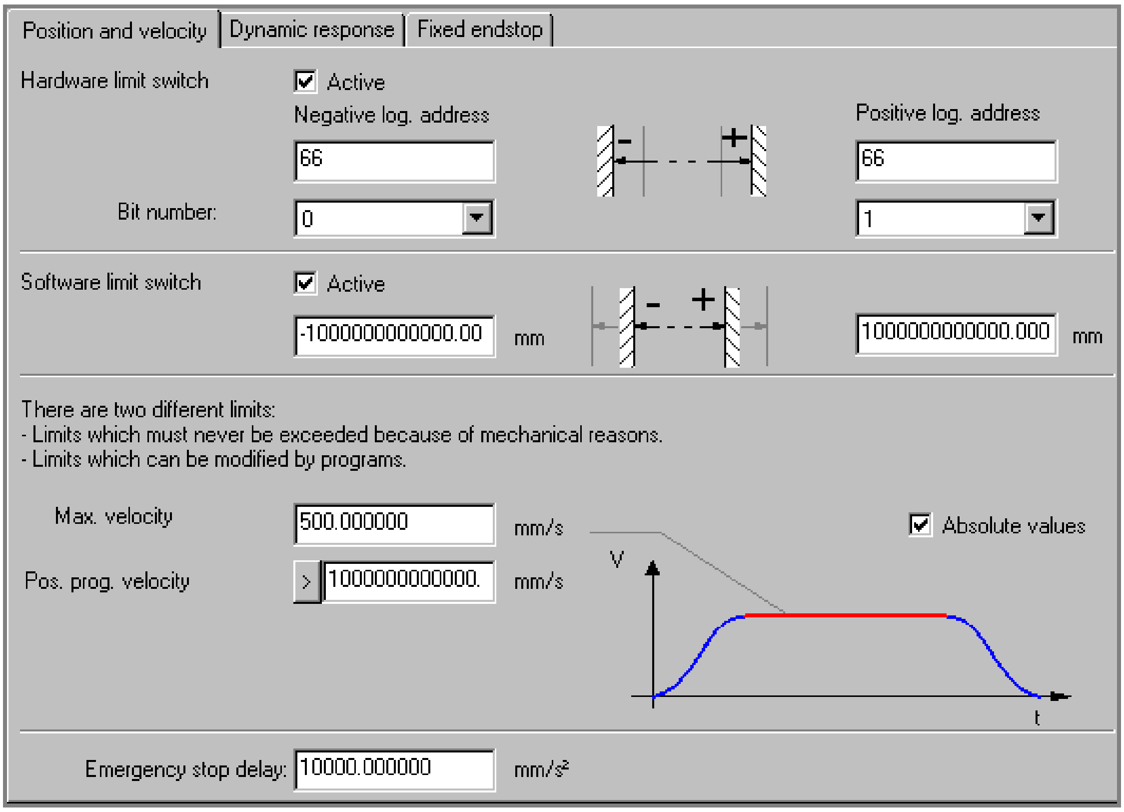
  The profile shows a constant acceleration, the jerk profile is controllable (jerk <> 0).

4.6.11 Limits

4.6.11.1 Limits - "Position and velocity" tab

In the "Position and velocity" tab of the **Axis > Limits** tab you can

- Configure and enable monitoring of the hardware limit switches (Page 147)
- Configure and enable monitoring of the software limit switches (Page 219)
- Set velocity limits
- Set emergency-off deceleration



Emergency off deceleration

The value set for emergency off deceleration is activated when a moving axis is disabled and *Stopmode* = 0 is set at the "MC_Power" parameter.

4.6.11.2 Hardware limit switches

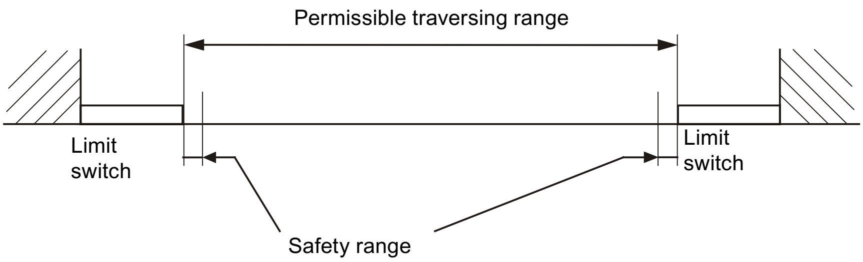
You enable hardware limit switch monitoring in the **Axis > Limits** dialog box, "Position and Velocity (Page 146)" tab. Hardware limit switch monitoring is used to limit the operating range of an axis, or to protect the machine.

Wiring

The HW limit switches can be wired to the four integrated digital inputs of the Technology CPU, or to I/O modules such ET 200 or SINAMICS S120 with TM15/TM17 which are operated on DP(DRIVE).

Traversing range

The permitted traversing range is monitored at the digital inputs of the hardware limit switches.



NOTICE
The hardware limit switch must be implemented as an NC contact. The hardware limit switches must remain active after the axis has passed the permissible traversing range until the mechanical end position is reached.

Retraction

An axis triggering a HW limit switch is stopped with error messages *8013* and *804B*. It is retracted from the HW limit switch (release motion) as described below:

- Manual retraction

  The axis is returned manually to the permissible traversing range. The error at the technology DB can only be acknowledged after it has been returned to this range.

- Retraction with drive

  The error at the technology DB of the axis is acknowledged, but the error message and the *LimitSwitchActive* bit remain active. The axis can now be returned to the permitted traversing range. A reverse motion command once again triggers an axis error. The error messages and the *LimitSwitchActive* status can be acknowledged after the axis has moved out of the range of the limit switch.

The current position of the axis is saved upon reaching the HW limit switch is saved. Only after this position, plus a safety range, has been passed is the axis considered to have left the limit switch.

NOTICE
The controller may not be switched off after the axis has passed the HW limit switch in order to avoid a conflict between polarity monitoring of the hardware limit switches and the overrun monitoring of the hardware limit switches in direction of the valid range. In this case, the axis is moved into the valid range without hardware limit switch monitoring, and is then re-enabled. When the controller is switched on, the axis must be positioned within the valid traversing range.

Internal states are lost and the configuration is reloaded when the axis passes the hardware limit switch. Reloading without loss of the approach information is only possible within the valid range.

Exception: Deactivation of position limit monitoring after a polarity reversal error

Safety range

The safety range of the HW limit switches is calculated based on the configured resolution of the system of units of the axis.

Safety range = 1000 / (increments / position)

Example: A linear axis is assigned the position unit "mm" and a resolution (increments / position) of "1000/unit" in the "Configure units" dialog box, meaning that the axis position is calculated to an accuracy of 0.001 mm. The safety range in this example is a multiple of 1000 of the accuracy: 1 mm.

4.6.11.3 Software limit switch

You can configure the software limits switches and enable the monitoring function in the **Axis > limits** dialog box, "Position and Velocity (Page 146)" tab. If the software limit switches are activated, the traversing distance of the axis is limited with the software limit switches.

Software limit switches should lie within the range of the HW limit switches with reference to the traversing range in order to limit the working range of an axis, for example.

Monitoring software the software limit switches at start of motion:

- **Check box activated**
  If the target position of a position-controlled motion command lies beyond the software limit switch, the warning *0026* is already displayed in the technology DB of the axis when the motion is started. In S7T Config the interrupt *40105* is signaled.
  The axis travels up to the position of the software limit switch and the error *8014* is reported in the technology DB.
- **Check box not activated**
  If the target position of a position-controlled motion command lies beyond the software limit switch, the axis traverses until the pPosition of the software limit switch is reached. Error *8014* is reported in the technology DB (Warning *0026* is not output).

Behavior for travel to software limit switch:

- **For position-controlled traversing:**
  The axis traverses until the software limit switch is reached. After the software limit switch has been reached, the respectively active position controlled or speed-controlled mode remains.
- **In all operating modes:**
  The axis traverses until the software limit switch is reached. After the software limit switch has been reached, an active position controlled mode remains. In speed-controlled mode the axis changes to position-controlled mode.

Negative position / positive end position:

Enter the positions for the negative and the positive end positions of the software limit switches in these input fields.

Tolerance window for retraction:

Enter a suitable value in this input field in order to prevent renewed triggering of the software limit switch error during retraction.

Note

The response of software limit switches is determined by the axis configuration in terms of the homing function. The limit switches of a configuration requiring a homed axis for the execution of absolute motion commands ("Homing required: Yes" in the **Axis > homing dialog box**) are not monitored if the axis is not homed. By contrast, if the execution of absolute motion commands is allowed when the axis is not homed ("Homing required: No" in the **Axis > Homing dialog box**) the software limit switches are monitored, regardless of the homing state of the axis.

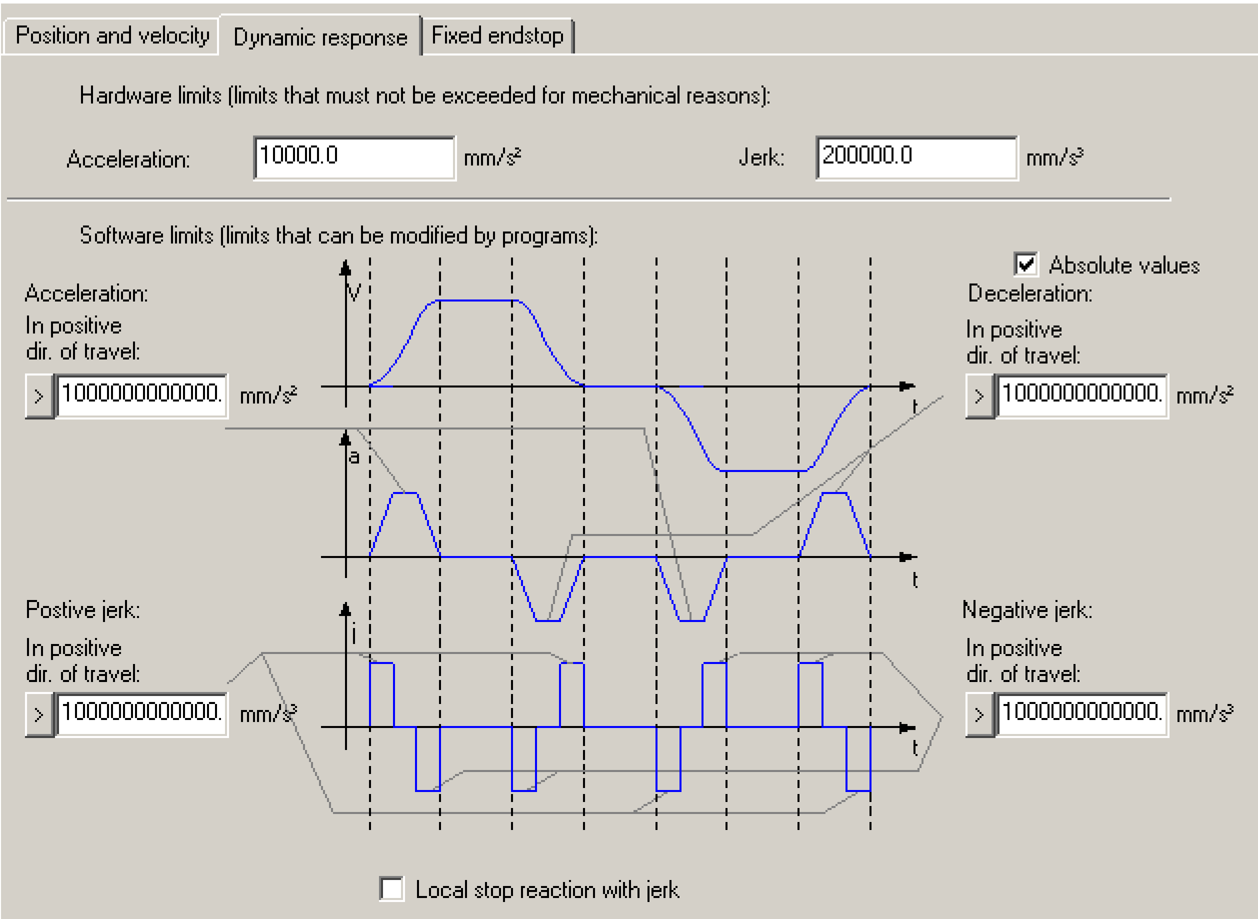
4.6.11.4 Limits - "Dynamic response" tab

You can adjust the HW limits (mechanical) and SW limits in the "Dynamic response" tab of the **Axis > Limits** dialog box.

The set deceleration limit is activated when a moving axis is disabled and *FastStop = 1* is set at the "MC_Power" parameter.

The **status-dependent** acceleration model is enabled (default) by setting the "Absolute values" check box.

The **direction-dependent** acceleration model is enabled by resetting the "Absolute values" check box. Additional parameters are displayed in this case.



Status-dependent acceleration

- **Acceleration**

  Axis acceleration, independent of the direction of movement

- **Deceleration**

  Axis deceleration, independent of the direction of movement

Direction-dependent acceleration

- **Acceleration**

  Acceleration in the positive motion direction and deceleration in the negative motion direction

- **Deceleration**

  Acceleration in the negative motion direction and deceleration in the positive motion direction

Parameter settings with dynamic direction vector are useful, for example, for suspended axes.

Local stop reaction with jerk

Set the "Local stop reactions with jerk" check box to execute a stop reaction triggered by an alarm reaction at the axis, with jerk limiting and rounding.

Emergency-stop deceleration

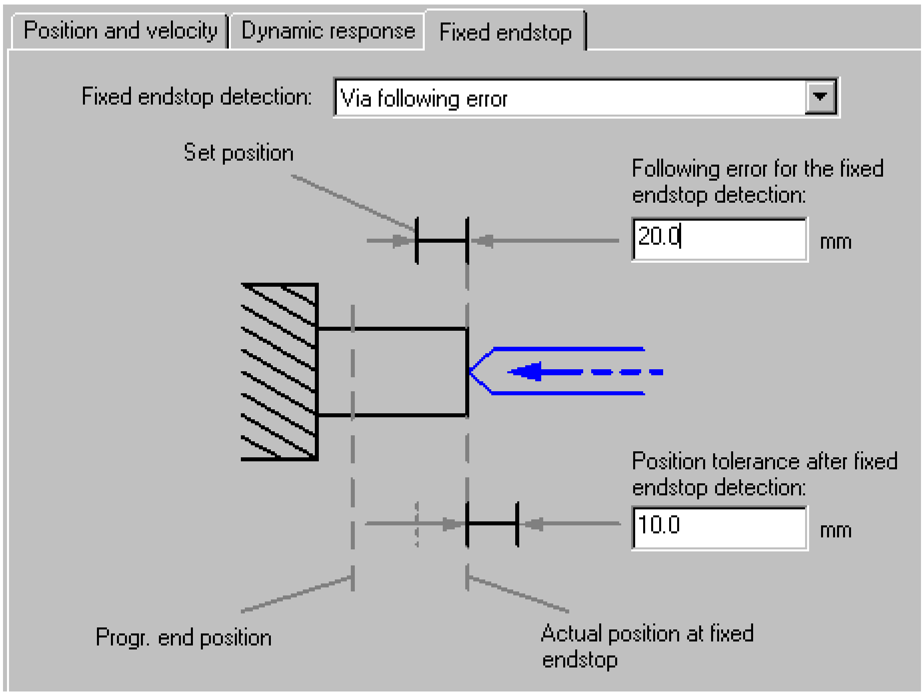
The set emergency-stop deceleration comes into effect when a moving axis is disabled and *Stopmode = 0* is set at "MC_Power."

4.6.11.5 Limits - "Fixed end stop" tab

The "Fixed end stop" tab of the **Axis > Limits** can be used to enable fixed end stop detection and to set the corresponding detection mode:

- Use following error
- Using the force/torque

When the selected condition is met, the "fixed end stop" status is reached. Following error monitoring is disabled if "Move to fixed end stop" is enabled.



Use following error

Note

When the axis moves onto the fixed end stop, and fixed end stop detection with "use following error" is set, the entry in "Position tolerance after fixed end stop detection" should be significantly less than that of "Following error for fixed end stop detection".

Use torque value

The fixed end stop detection function "use torque value" requires that the digital drive assigned to the axis supports torque limiting, and that a corresponding message frame, for example, *message frame 102* or *message frame 105*.

Move to fixed end stop

The "MC_MoveToEndPos function activates the "Move to end stop" function and sets the clamping torque after the end stop is reached. This operation is also known as "clamping."

The motion is stopped when the axis reaches the fixed end stop and the control remains active. The setpoint value at the position controller input is held constant. New motion commands in direction of the clamping position are canceled; new motion control commands in retraction direction are executed to reduce torque. The position setpoint of the axis is used as start position of new motion control commands in retraction direction.

The position setpoint of the axis results from one of the following equations, depending on the fixed end stop detection function:

- "Use following error"
  Position at fixed end stop + following error
- "Use torque value"
  Position at fixed end stop + clamping tolerance

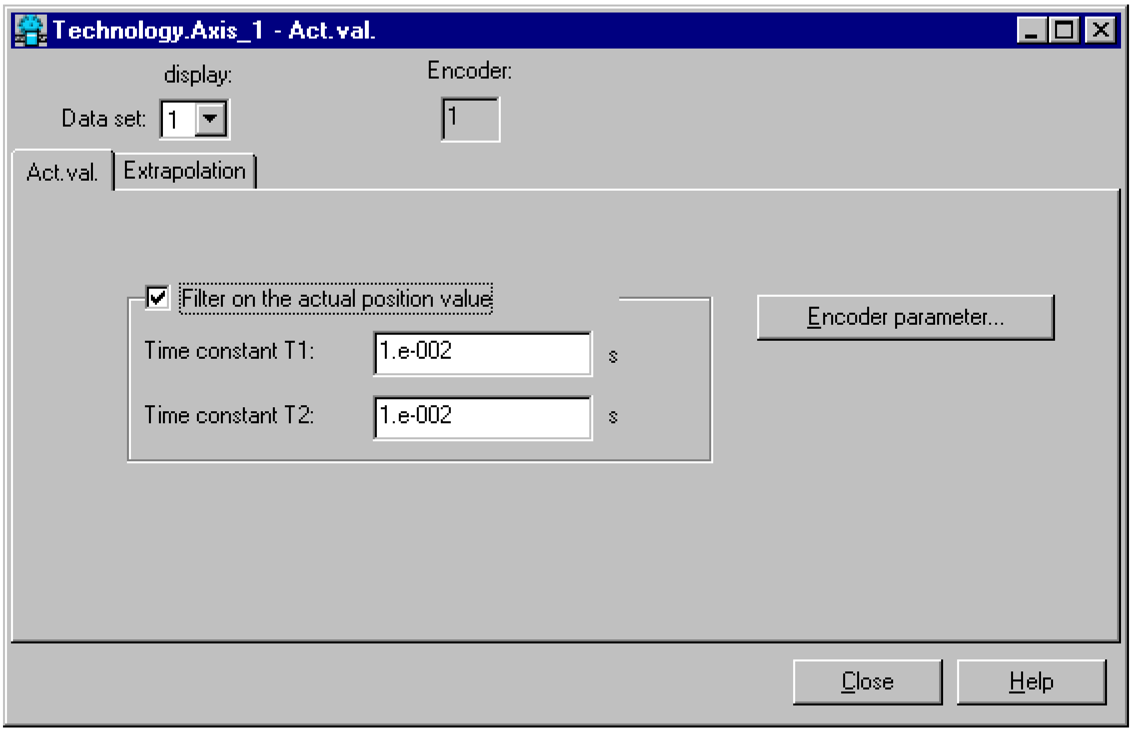
Conditions for the "Fixed end stop detection" function

- The "Move to fixed end stop" function is reset when the axis moves out of the clamping tolerance window.
- A new command can also be output to toggle the direction of torque when clamping is active.
- Non-stepped torque transitions and torque retention over a defined time period can be implemented in the user program, as can definitions of torque profiles.
- Moving to the fixed end stop (clamping) can be disabled by setting a reverse positioning command.
- A reversal command MC_MoveToEndPos is not permitted and is ignored.
- A mechanical break of the end stop is monitored by means of the actual value of the axis (clamping tolerance window monitoring).
- The torque limit of the drive is set in [N/m] at the *Torque* parameter of the MC_MoveToEndPos technology function.
- If the command is active and the fixed end stop is not detected, the system reacts as with active torque limiting.

4.6.12 actual value

4.6.12.1 Actual value - "Actual value" tab

Activate actual value filtering in the dialog **Axis / External encoder > Actual value** on the "Actual value" tab and set the appropriate time constants.



Filter on the actual position value

Activate the check box if you would like to activate the filtering of the actual position value.

Time constant T1

Here, you set time constant T1 of the PT2 position filter in the actual value system.

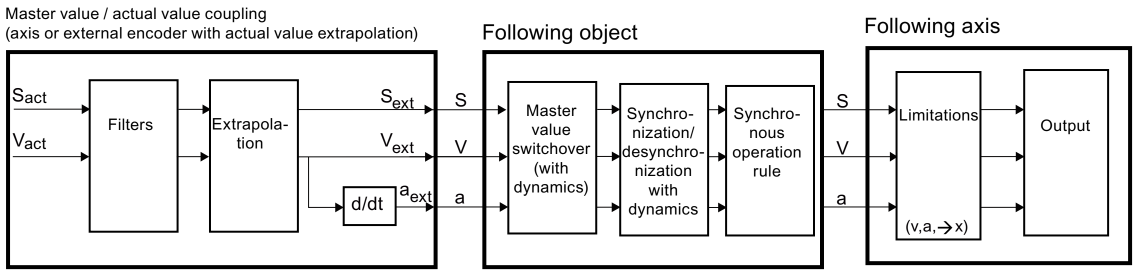
Time constant T2

Here, you set time constant T2 of the PT2 position filter in the actual value system.

"Encoder parameter" button

Opens a dialog box in which the encoder data are displayed.

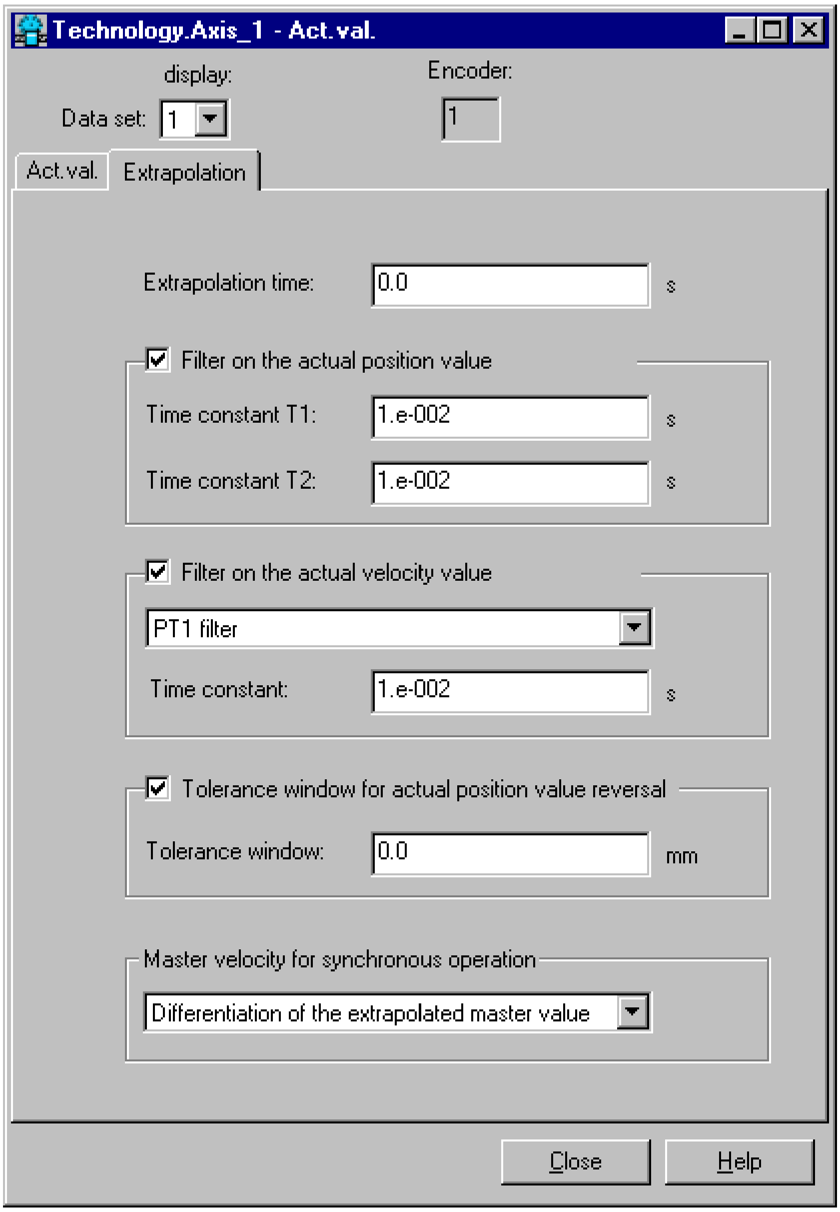
4.6.12.2 Actual value - "Extrapolation" tab



If there is a synchronous operation interconnection within a control, the synchronous operation takes into account the position, velocity, and acceleration of the master value position.

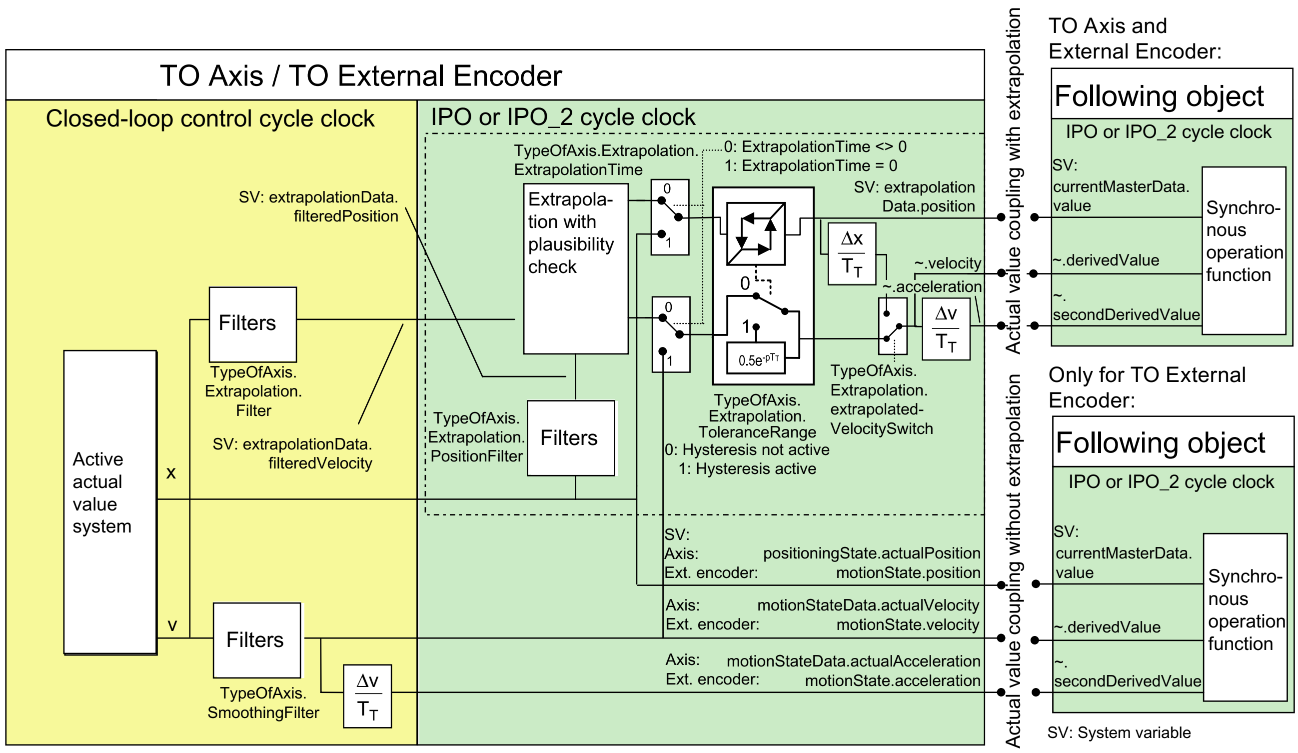
If an actual encoder value is used as the master value, it is useful to extrapolate the measured actual value for the synchronous operation in order to compensate for dead times. Dead times result within the system when measuring actual values, e.g. due to the bus communication and the system processing times.

The extrapolation is set in S7T Config on the leading axis or on the external encoder in the dialog **Axis / External encoder > Actual value** in the "Extrapolation" tab.



The parameters of this dialog can also be read from or written to the user program via technology parameters.

Actual value coupling with extrapolation (axis and external encoder)



Extrapolation time

(Parameter *1110;* configuration data *TypeOfAxis.Extrapolation.ExtrapolationTime*)

Here, you set the time for the extrapolation. No extrapolation if the value entered is *0.0*.

Filter on the actual position value

(Parameter *1130* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.enable*)

Activate the check box if you would like to extrapolate the actual position value.

Time constant T1

(Parameter *1131* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T1*)

Here, you set time constant T1 of the PT2 filter for the extrapolation in the actual value system.

Time constant T2

(Parameter *1132* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T2*)

Here, you set time constant T2 of the PT2 position filter in the actual value system.

The filter acts on the actual position for the extrapolation. The velocity for the extrapolation is taken over from the actual values of the axis or external encoder before application of the smoothing filter (*TypeOfAxis.smoothingFilter*).

Filter on the actual velocity value

(Parameter *1112* configuration data *TypeOfAxis.Extrapolation.Filter.enable*)

Activate the check box if you would like to extrapolate the actual velocity value.

Here, you select the filter for the extrapolation velocity in the drop-down list.

(Parameter *1111* configuration data *TypeOfAxis.Extrapolation.Filter.Mode*)

Time constant

(Parameter *1113* configuration data *TypeOfAxis.Extrapolation.Filter.timeConstant*)

Here, you enter the time constant for the filter.

The position is extrapolated based on the filtered or averaged velocity value. Averaging is via the "time constant".

Tolerance window for actual position value inversion

(Parameter *1114* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.enable*)

Here, you can activate the tolerance window for actual position value inversion

Tolerance window

(Parameter *1115* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.Value*)

Enter the size of the tolerance window here.

If the master value is superimposed with high-frequency noise signals that the synchronous operation cannot follow, this can cause the dynamic response limits to be exceeded or the master value to briefly change directions during synchronization.

In this case, define a tolerance window to prevent the dynamic response limits from being exceeded on the following axis or to prevent direction changes during synchronization.

Master velocity for synchronous operation

(Parameter *1116* configuration data *TypeOfAxis Extrapolation.extrapolatedVelocitySwitch*)

In the drop-down list, select whether the velocity is to be applied for the extrapolation or if the extrapolated master position value is to be differentiated.

Checking the extrapolated and filtered values

The extrapolated and filtered values can be checked in the following system variables:

- *extrapolationdata.position*
- *extrapolationdata.velocity*
- *extrapolationdata.filteredposition*
- *extrapolationdata.filteredvelocity*
- *extrapolationdata.acceleration*

Support of encoders with NIST evaluation

With encoders with NIST evaluation, the speed determined by the encoder and the resulting velocity can be accepted by the encoder. A calculation of the speed and velocity by the integrated technology is not necessary in this case. Two methods of transmission are available:

- **Transmission in the PROFIdrive message frame**

  Setting via the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_PROFIDRIVE_NIST_*

- **Transmission in the I/O area**

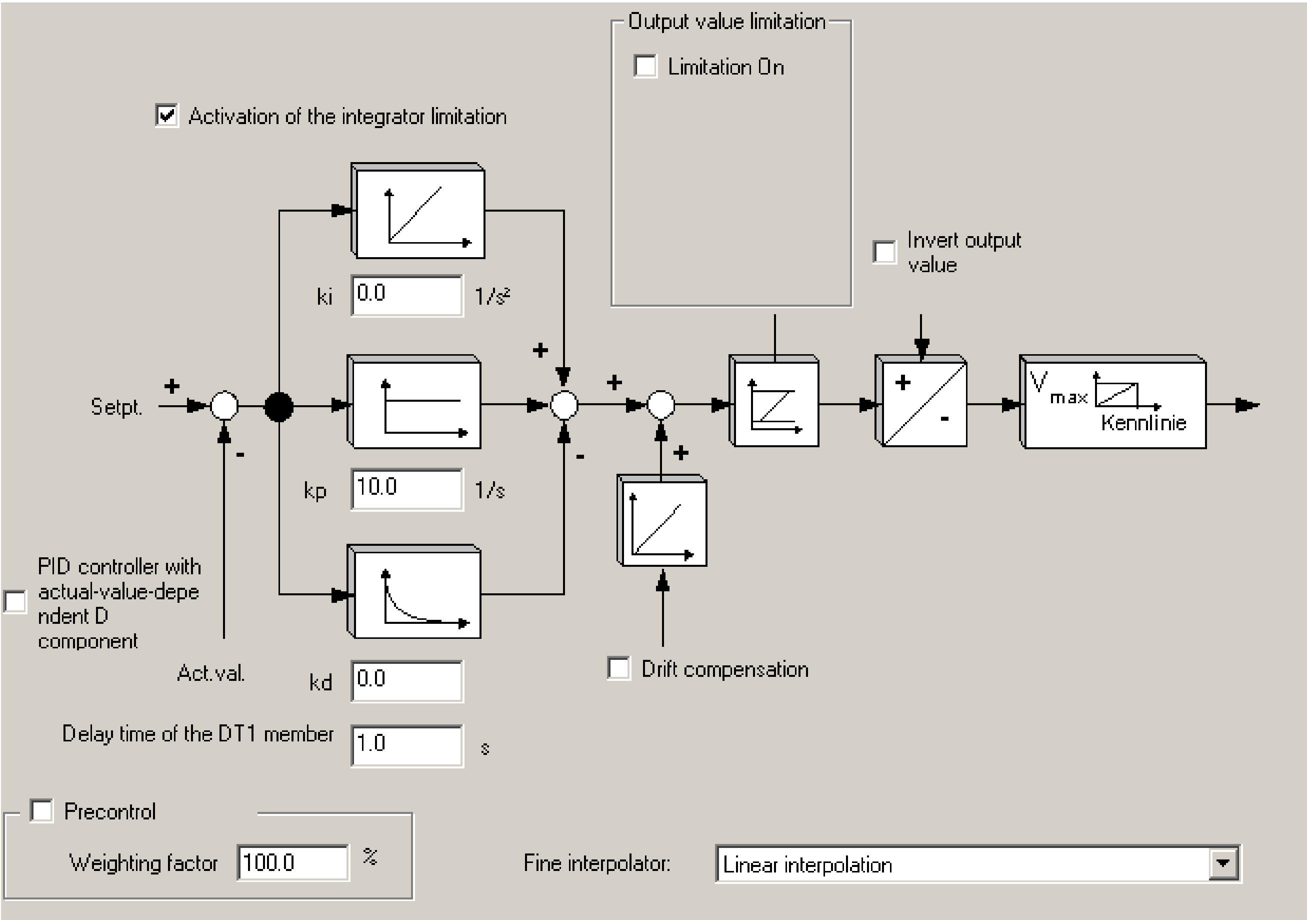
  Setting via the configuration data element

  *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_DIRECT_NIST*In this case, 4000H corresponds to 100%. The address is set in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.logAdress*, and the reference value in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.referenceValue*.

4.6.13 Control

4.6.13.1 Control - "Static controller data" tab

Set the position control of the axes in the "Static controller data" tab of the **Axis > Control** dialog box.



Manipulated variable limiting

Manipulated variable limiting sets the absolute hi and lo limits of the control range. This limitation is applied prior to inversion.

Fine Interpolator

The fine interpolator function is used is to generate interim setpoints when the interpolator and controller have a different duty factor. Optional interpolation mode settings at the fine interpolator:

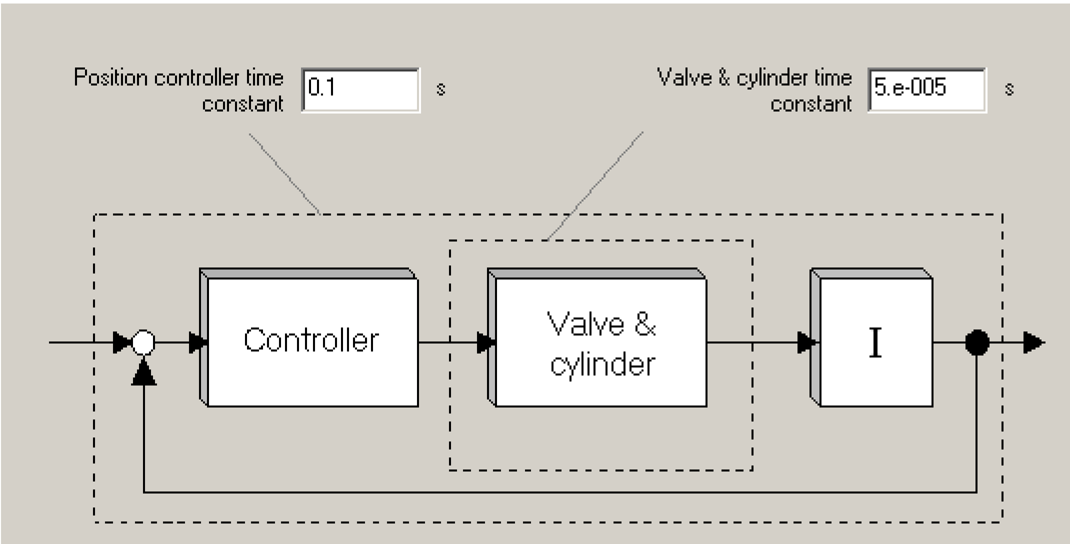
- No interpolation
- Linear interpolation (constant position for positioning axis)
- Interpolation with constant acceleration (constant acceleration at positioning axis)
- Interpolation with constant velocity (constant velocity at positioning axis)

When set for positioning axes, the position setpoint is interpolated.

When set as speed-controlled axis, the velocity setpoint is interpolated.

4.6.13.2 Control - "Dynamic controller data" tab

Set the dynamic controller data of the axis in the "Dynamic controller data" tab of the **Axis > Control** dialog box.



Position controller time constant

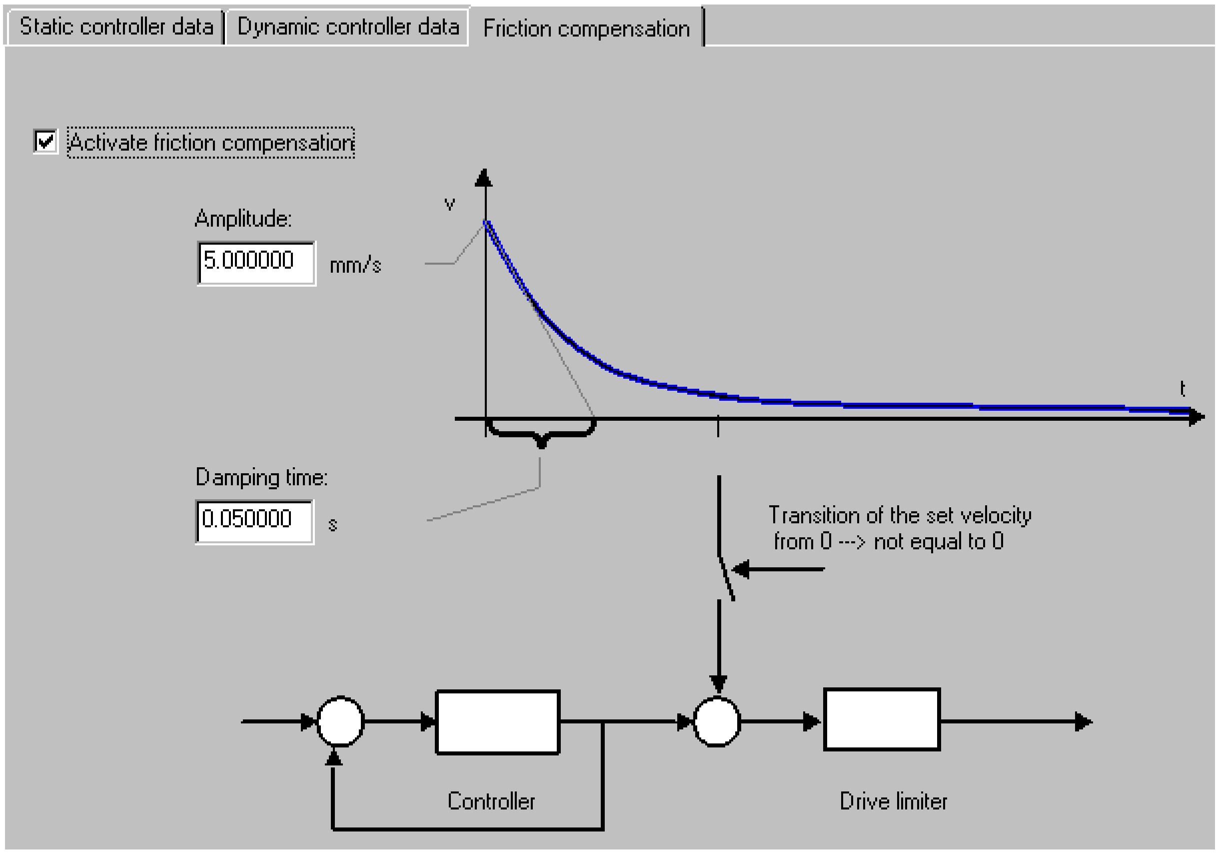
Enter the equivalent time constant for the position controller of the axis.

Valve & cylinder time constant

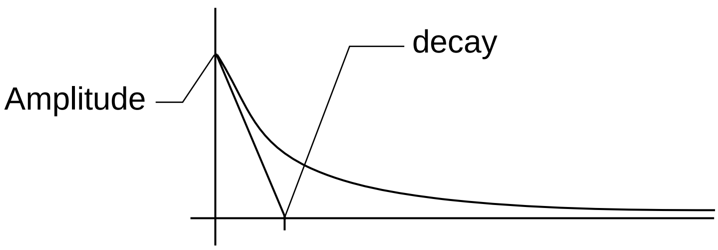
Enter the equivalent time constant for the hydraulic control loop at this parameter.

4.6.13.3 Control - "Friction compensation" tab

The "Friction compensation" tab of the **Axis > Control** tab is used to enable friction compensation. The "Friction compensation" tab is available when expert mode is set on the "Static controller data" tab.



The system provides a simple method compensating for the forces of static friction. During startup from a standstill, a DT1 element adds a static friction compensation signal to the manipulated variable.

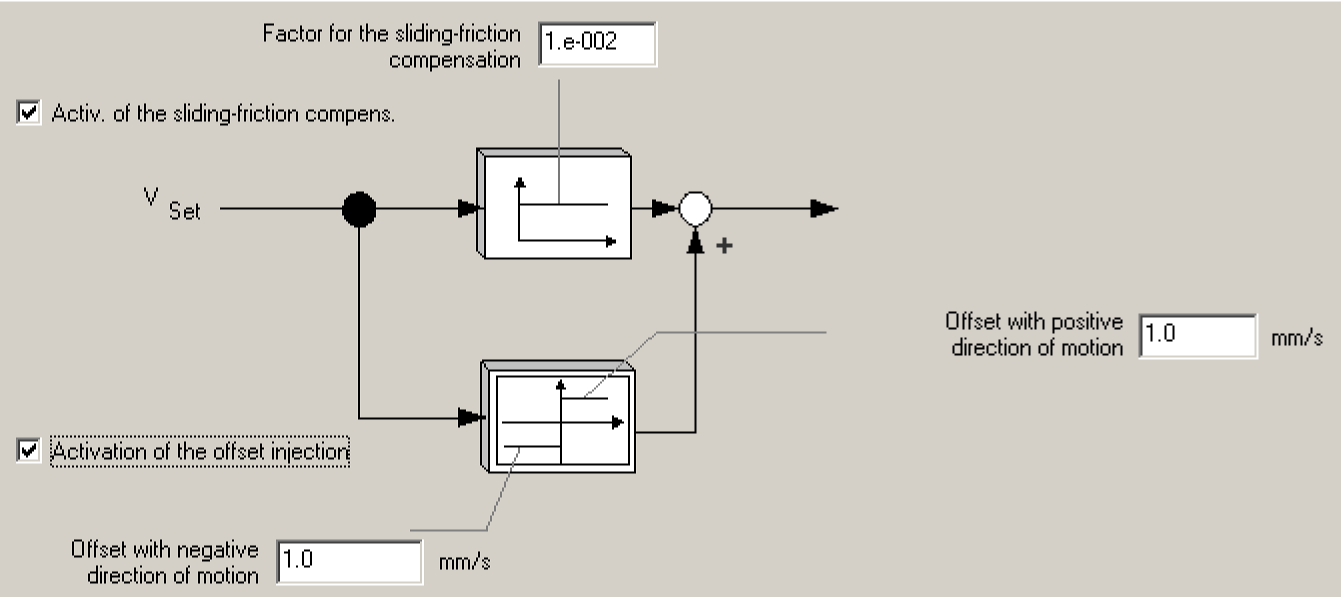


The friction compensation is added relative to the velocity setpoint. It is only active when motion commands are executed.

The standstill identification for static friction compensation can be set separately, as is the case for the amplitude and the decay response. The amplitude and decay response are set in the configuration.

4.6.13.4 Control - "Additional compensation functions" tab

Select the "Additional compensation functions" tab of the **Axis > Control** dialog box to configure a static compensation element (additive friction coefficient) and a compensation element which is proportional to velocity (friction) for the position controller of the hydraulic axis.



- **Friction compensation factor:** (friction compensation activated)
  Enter the friction compensation factor at this parameter.
- **Activate friction compensation:**
  Set this box to activate friction compensation.
- **Offset with positive direction of movement:** (Offset activated)
  Enter the offset for positive direction of movement.
- **Activate offset:**
  Set this box to activate the offset.
- **Offset for negative direction of movement:** (Offset activated)
  Enter the offset for negative direction of movement.

4.6.14 Homing

4.6.14.1 Introduction

Position-controlled axes equipped with incremental motor measuring systems must be referenced to the position of the mechanical system of the machine each time power is switched on. The axis is synchronized during homing, based on the activation of a certain position value at a defined position of the axis mechanism.

Axes can be homed in:

- Active mode (reference point approach)
- Passive mode (flying referencing)
- Direct mode (with position setpoint)

Detailed settings depend on the measuring systems available for measuring the reference point and on the motion an axis may perform for homing.

4.6.14.2 Homing - "Active homing" tab

Active homing

Active homing can be configured in the "Active homing" tab of the **Axis > Homing** dialog box in S7T Config.

Three homing modes are available for "Active homing":

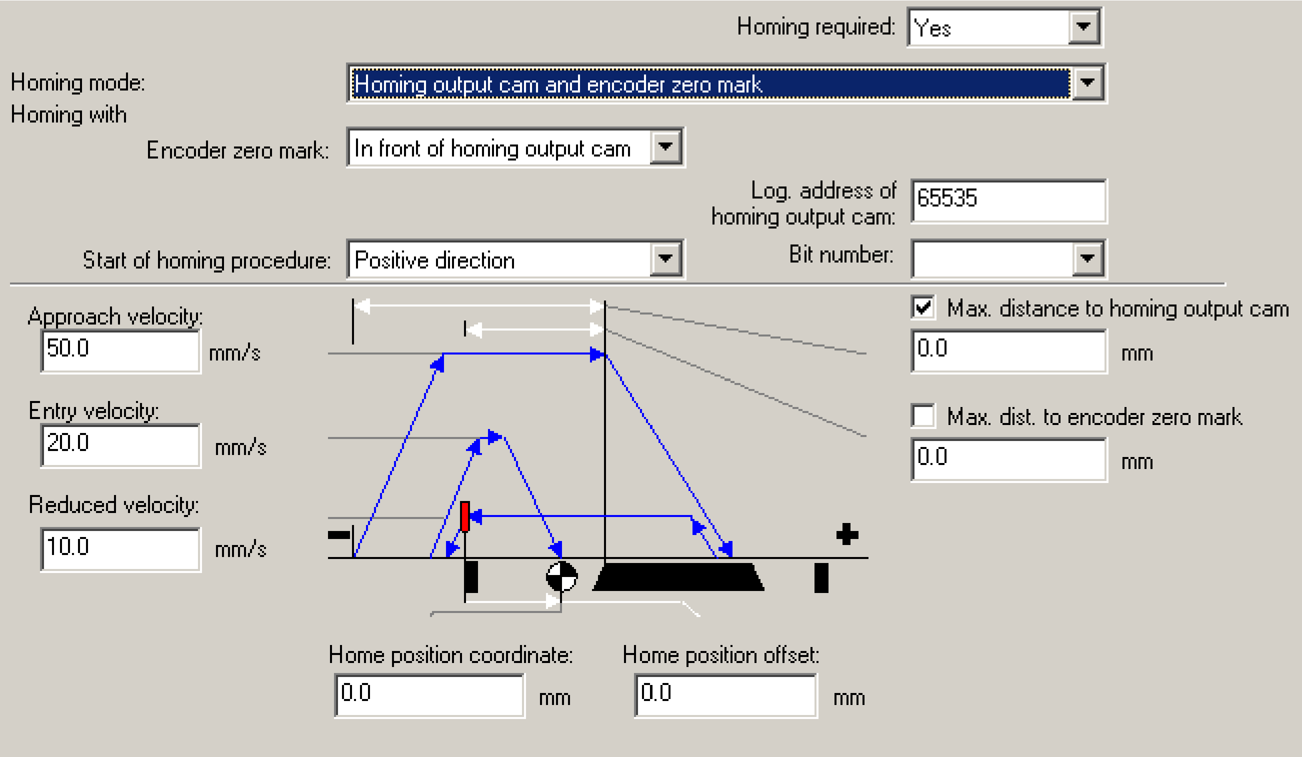
- Reference cam and encoder zero mark
- Encoder zero mark only
- External zero mark only

Start of the homing function:

MC_Home	*Mode = 0*	Determination of the reference point based on the axis configuration
	Mode = 1 *Position = x*	Determination of the reference point based on the axis configuration The reference point is assigned the value of the *Position* input parameter.

After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone* = *TRUE*.

Active homing mode with "reference cam and encoder zero mark"



After homing is started with the "MC_Home" technology function, the axis moves to the reference cam and then approaches the referencing encoder zero mark, according to the axis configuration. The lower section of the diagram shows the resultant motion sequence.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case "Homing output cam and encoder zero mark"
Encoder zero mark	Before reference cam	The axis is homed to the encoder zero mark which lies before the reference cam with reference to the direction of reference point approach.
	After reference cam	The axis is homed to the encoder zero mark which lies after the reference cam with reference to the direction of the reference point approach.
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Logical address of reference cam	[byte address]	Logical byte address of the reference cam You can connect the reference cam to the integrated inputs of the Technology CPU, or to the IO on DP(DRIVE).
Bit number	[Number of the bit]	Bit address of the signal used for the reference cam
Approach velocity	-	Velocity at which the axis approaches the reference cam
Entry velocity	-	Velocity at which the axis approaches the (shifted) reference point after detection of the encoder zero mark
Shutdown velocity	-	Velocity at which the axis approaches the encoder zero mark after detection of the reference cam.
Reference point coordinate (Reference cam and encoder zero mark, External zero mark only, Encoder zero mark only)	-	Here, you enter the actual position value of the home position (relative to the coordinate system of the axis)
Homing position offset	-	The "Home position offset" shifts the reference point by a configured distance. The axis travels along this configured distance at a velocity defined in "Entry velocity", starting at the synchronization position with encoder zero mark. Modulo axes always take the shortest distance.
Maximum distance to homing output cam	deactivated	The distance to reference cam detection is not monitored
	Activated	Monitoring of the distance between the start of reference point approach and detection of the reference cam. If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance an axis travels between the reference cam and the encoder zero mark. If the difference in this distance exceeds the configured distance, the corresponding axis technology DB indicates error *801D*. The reference point approach is canceled.

Sequence of the reference point approach

- Phase 1: Reference cam approach

  The axis starts its reference point approach at the configured approach velocity and towards the direction set in "Start of reference point approach". The reference cam approach ends after the reference cam (Phase 1) is reached.

  You can also monitor the distance an axis travels between the start of reference point approach and detection of the reference cam by setting the "Max. distance to homing output cam" check box. The reference point approach is canceled with error if the reference cam is not detected between the start and end of the configured distance.

- Phase 2: Synchronization with encoder zero mark

  After having reached the reference cam, the axis accelerates / decelerates to shutdown velocity, and approaches the encoder zero mark. The encoder zero mark is derived from the combined settings of "Encoder zero mark" (after or before the reference cam) and "Start of reference point approach" (positive or negative direction).

  After the reference cam is detected, the PLC synchronizes the axis to the first encoder zero mark detected in accordance with the configuration. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

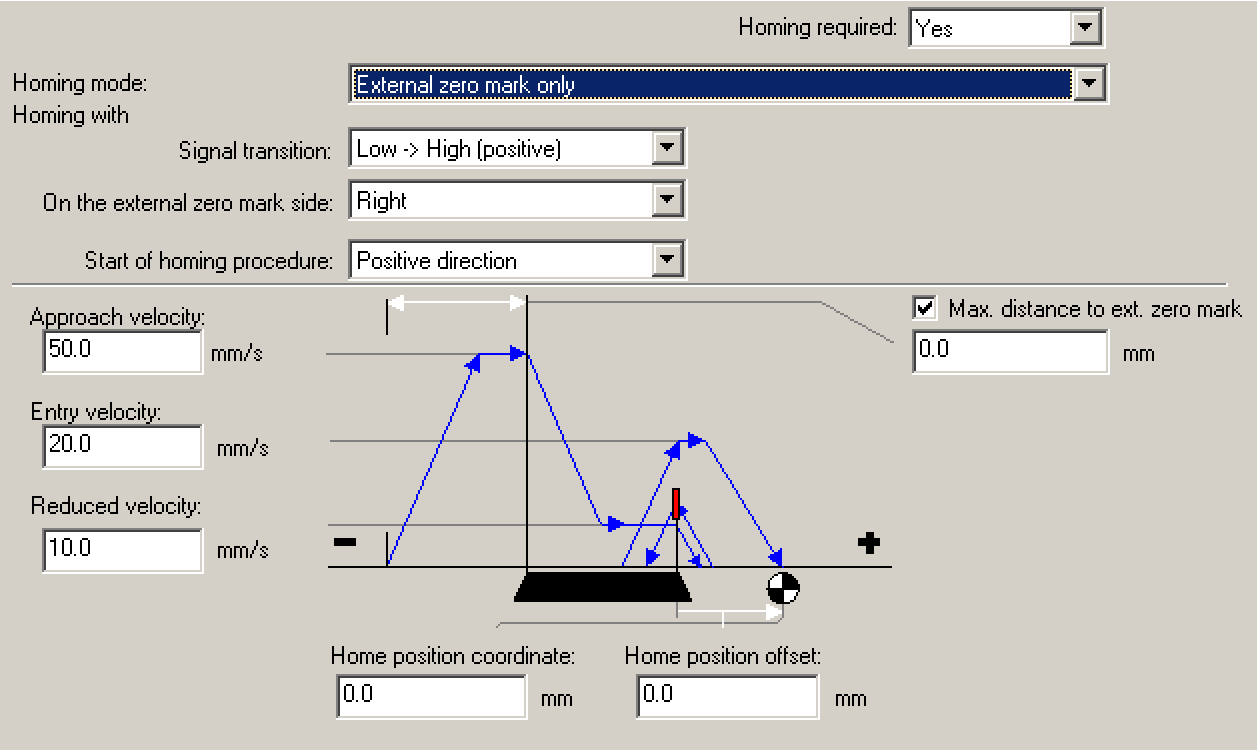
  You can also monitor the distance an axis travels between the reference cam and the encoder zero mark by setting the "Max. distance to encoder zero mark" check box. The reference point approach is canceled with error if the encoder zero mark is not found within the specified distance after the reference cam is detected.

- Phase 3: Reference point approach

  After the encoder zero mark is detected, the axis accelerates / decelerates to approach velocity to approach the reference point.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the encoder zero mark is detected, provided that the reference point lies within the deceleration ramp.

Active homing mode with "External zero mark only"



If homing is started by calling the "MC_Home" technology function the axis approaches the referencing external zero mark in accordance with the configuration. The lower section of the diagram shows the resultant motion sequence.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "External zero mark only"
Signal transition	Low -> High (positive)	The motion is referenced to the positive edge of the external zero mark (setting according to edge evaluation in the drive component).
	High -> Low (negative)	The motion is referenced to the negative edge of the external zero mark (setting according to edge evaluation in the drive component).
On the side of the external zero mark	left	The signal transition is measured viewed from the left side of the external zero mark, in the selected direction of reference point approach.
	right	The signal transition is measured viewed from the right side of the external zero mark, in the selected direction of reference point approach.

Parameters	Value	Explanation
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Approach velocity	-	Velocity at which the axis approaches the external zero mark
Entry velocity	-	Velocity at which the axis approaches the (shifted) homing position after detection of the external zero mark
Shutdown velocity	-	Velocity at which the axis approaches the reference point coordinate after detection of the external zero mark
Homing position offset	-	The "Home position offset" shifts the reference point by a configured distance. The axis moves by a configured distance at the "Homing velocity", after its synchronization at the external zero mark edge. Modulo axes always take the shortest distance.
Max. distance to external zero mark	deactivated	The distance to external zero mark detection is not monitored.
	Activated	Monitoring of the distance between the start of the reference point approach and detection of the external zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.

Note

For homing with "external zero mark", connect the external zero mark as digital measuring input to the drive component.

In order to execute a reference point approach in "external zero mark" homing mode, select "Signal transition" and "on the side of the external zero mark" values which correspond with the drive's configuration or functionality. Instead of being used to configure the measuring function in the drive, the "Signal transition" and "on external zero mark side" parameters merely reflect their functionality in order to control the axis motion according to the configuration.

For information on the configuration of external zero mark detection, refer to the relevant product information or to the drive manuals.

Sequence of the reference point approach

- Phase 1: Synchronization to external zero mark

  The axis starts its reference point approach at the configured approach velocity and towards the direction set in "Start of reference point approach". Synchronization ends at the external zero mark (phase 1) when the configured signal transition (configured at the drive component) is detected at external zero mark. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

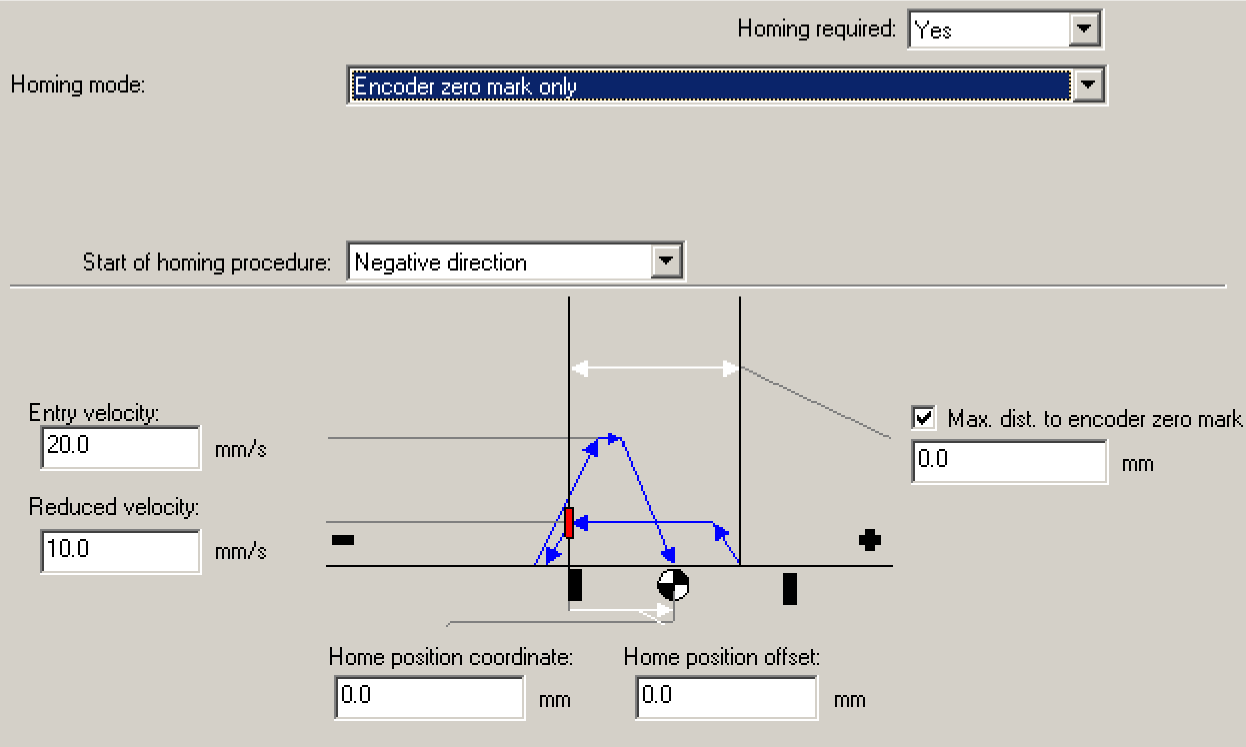
  You can also monitor the distance an axis travels between the start of reference point approach and detection of the signal edge by setting the "Max. distance to external zero mark" check box. Homing is canceled with error if the edge is not detected between the start of reference point approach and the end of the configured distance.

- Phase 2: Reference point approach

  After the configured signal edge is detected, the axis accelerates / decelerates to approach the reference point coordinate at shutdown velocity.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the external zero mark is detected, provided that the reference point lies within the deceleration ramp.

Active homing mode with "Encoder zero mark only"



Homing to encoder zero mark is used, for example, at axes for which the encoder can only set one encoder zero mark in the entire traversing range of the axis. The homing command initiates axis approach to the encoder zero mark. After the encoder zero mark is detected, the axis approaches the shifted reference point at homing velocity. The axis position now has the value defined in the reference point coordinate. The graphic in the figure shows the resultant motion sequence. Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Encoder zero mark only"
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Entry velocity	-	Velocity at which the axis approaches the (shifted) reference point after detection of the encoder zero mark
Shutdown velocity	-	Velocity at which the axis starts the reference point approach and approaches the encoder zero mark
Homing position offset	-	The homing position offset function shifts the homing position by a configured distance, meaning that the axis moves along a configured distance behind the encoder zero mark detection position at a "Homing velocity". Modulo axes always take the shortest distance.

Parameters	Value	Explanation
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance between the start of the reference point approach and detection of the encoder zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.

Sequence of the reference point approach

- Phase 1: Synchronization to encoder zero mark

  The axis starts its reference point approach towards the direction set in "Start of reference point approach" at the configured shutdown velocity. Synchronization with encoder zero mark (phase 1) ends with the detection of the encoder zero mark. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

  You can also monitor the distance an axis travels between the start of reference point approach and encoder zero mark detection by setting the "Max. distance to zero mark" check box. Homing is canceled with error if the encoder zero mark is not detected between the start of reference point approach and the end of the configured distance.

- Phase 2: Reference point approach

  After the encoder zero mark is detected, the axis accelerates / decelerates to approach the reference point coordinate at entry velocity.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the encoder zero mark is detected, provided that the reference point lies within the deceleration ramp.

4.6.14.3 Homing - "Passive homing" tab

Passive homing can be configured in the "Passive homing" tab of the Axis > Homing dialog box of S7T Config. The difference compared to active homing is that the required homing motion is not initiated by a homing command.

Three homing modes are available for "Passive homing":

- Reference cam and encoder zero mark
- External zero mark only
- Encoder zero mark only

Start of the homing function:

MC_Home	*Mode = 2* *Position = x*	The current position is assigned the value of the *Position* input parameter at the reference point.

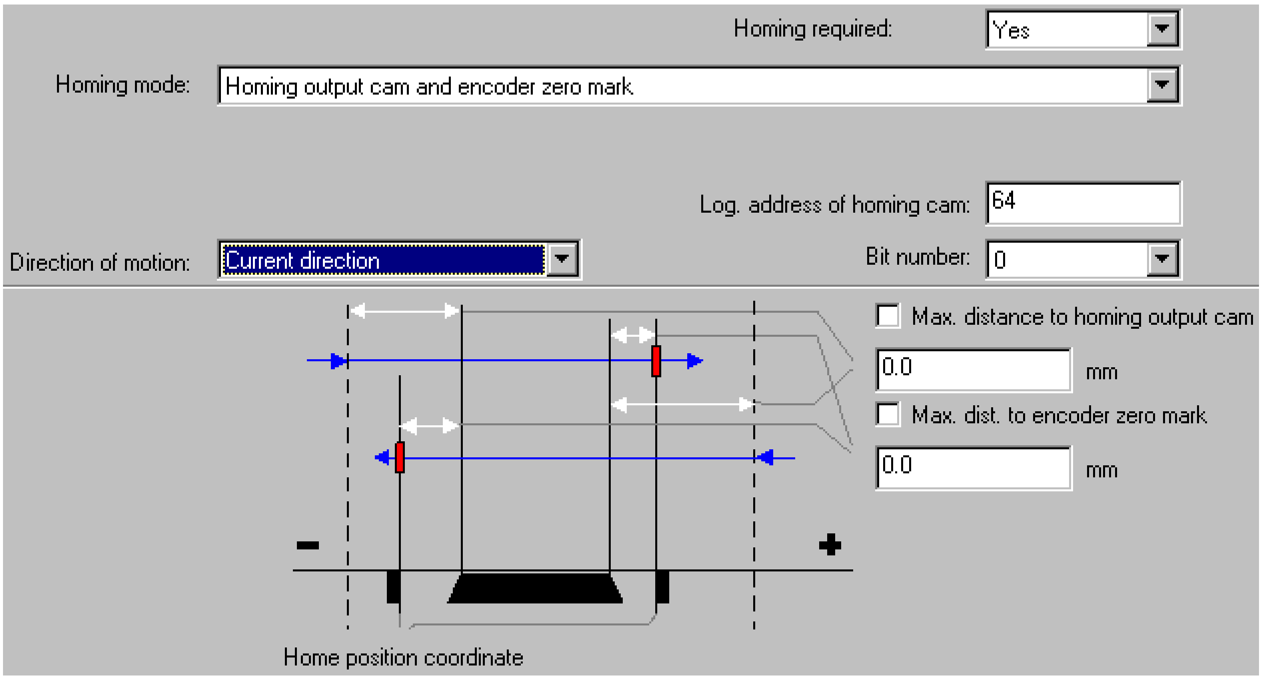
After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone= TRUE*.

Note

Note that not all drive components support all homing mode or measured signal evaluation functions. For details, refer to the documentation of the drive components used.

Triggering of encoder zero marks or reference cam monitoring during passive homing operations is indicated by a corresponding error message at the technology DB. The current axis motion is terminated in this case.

Passive homing mode with "Reference cam and encoder zero mark"

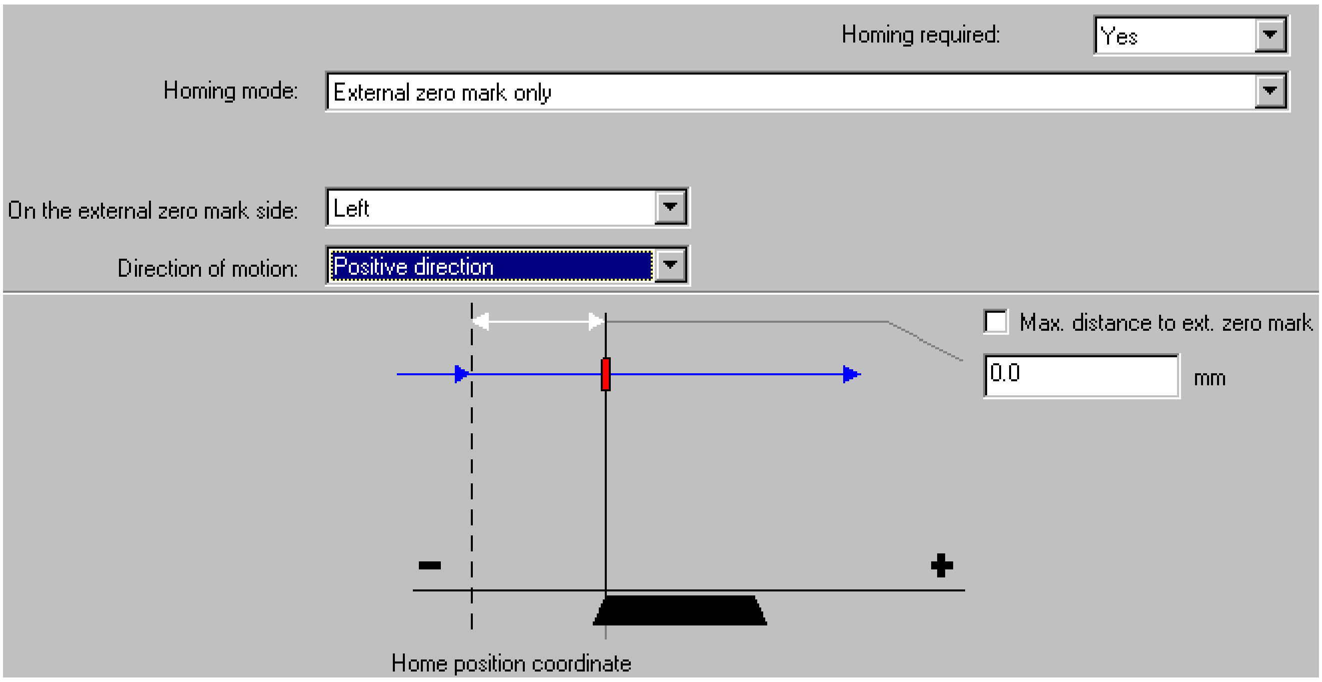


After the axis has passed the reference cam, the next encoder zero mark triggers axis homing. The axis position is set in the reference point to the value defined at the *Position* input parameter of the "MC_Home" technology function.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Reference cam and encoder zero mark"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.
Logical address of reference cam	[byte address]	Logical byte address of the reference cam You can connect the reference cam to the integrated inputs of the Technology CPU, or to the IO on DP(DRIVE).
Bit number	[Number of the bit]	Bit address of the signal used for the reference cam
Maximum distance to homing output cam	deactivated	The distance to reference cam detection is not monitored
	Activated	Monitoring of the distance between the start of the homing function and detection of the reference cam. If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance an axis travels between the reference cam and the encoder zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.

Passive homing mode with "External zero mark only"



Axis homing starts with the detection of the external zero mark. The axis is set to the value of the *Position* input parameter of the "MC_Home" technology function.

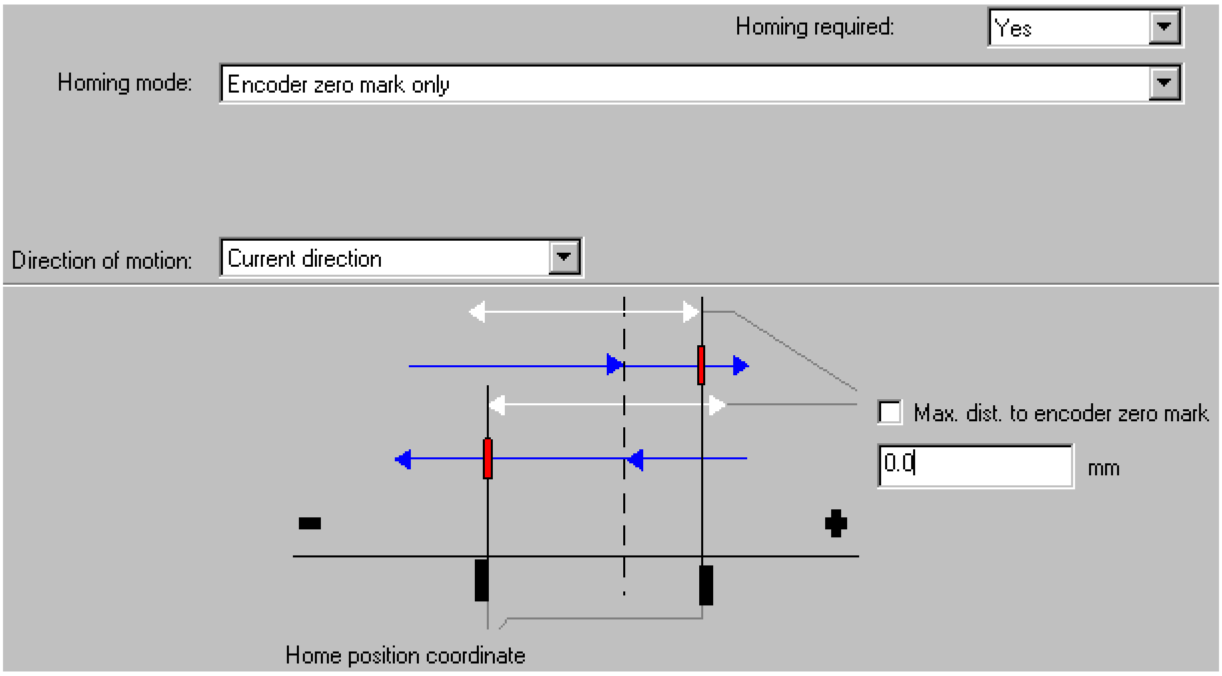
Meaning of the individual parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "External zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the external zero mark.
	Negative direction	The axis is only homed with negative approach to the external zero mark.
	Current direction	The axis is homed when it reaches the next external zero mark.
On the side of the external zero mark	left	Direction of movement: Positive direction Axis homing is triggered at the positive edge. Direction of movement: Negative direction Axis homing is triggered at the negative edge.
	right	Direction of movement: Positive direction Axis homing is triggered at the negative edge. Direction of movement: Negative direction Axis homing is triggered at the positive edge.
Max. distance to external zero mark	deactivated	Distance to go to the external zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the external zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Note

- For homing with "external zero mark", connect the external zero mark as digital measuring input to the drive component.
- In order to be able to execute the reference point approach in "external zero mark" homing mode as required, you should always set values at "Direction of movement" and "on the side of the external zero mark" which are compatible with drive configuration or functionality. The "on the side of the external zero mark" and "Direction of movement" parameters are not used to configure the measuring function in the drive and only reflect their functionality.
- For information on the configuration of external zero mark detection, refer to the relevant product information or to the drive manuals.

Passive homing mode with "Encoder zero mark only"



Homing mode with "Zero mark only" is used for axes, for example, which are equipped with an encoder that outputs only one zero mark signal within the entire traversing range of the axis

Axis homing starts with the detection of the encoder zero mark. The position value of the axis is set to the value of the *Position* input parameter of the "MC_Home" technology function after the encoder zero mark is detected.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Encoder zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the encoder zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Passive homing mode with "Default"

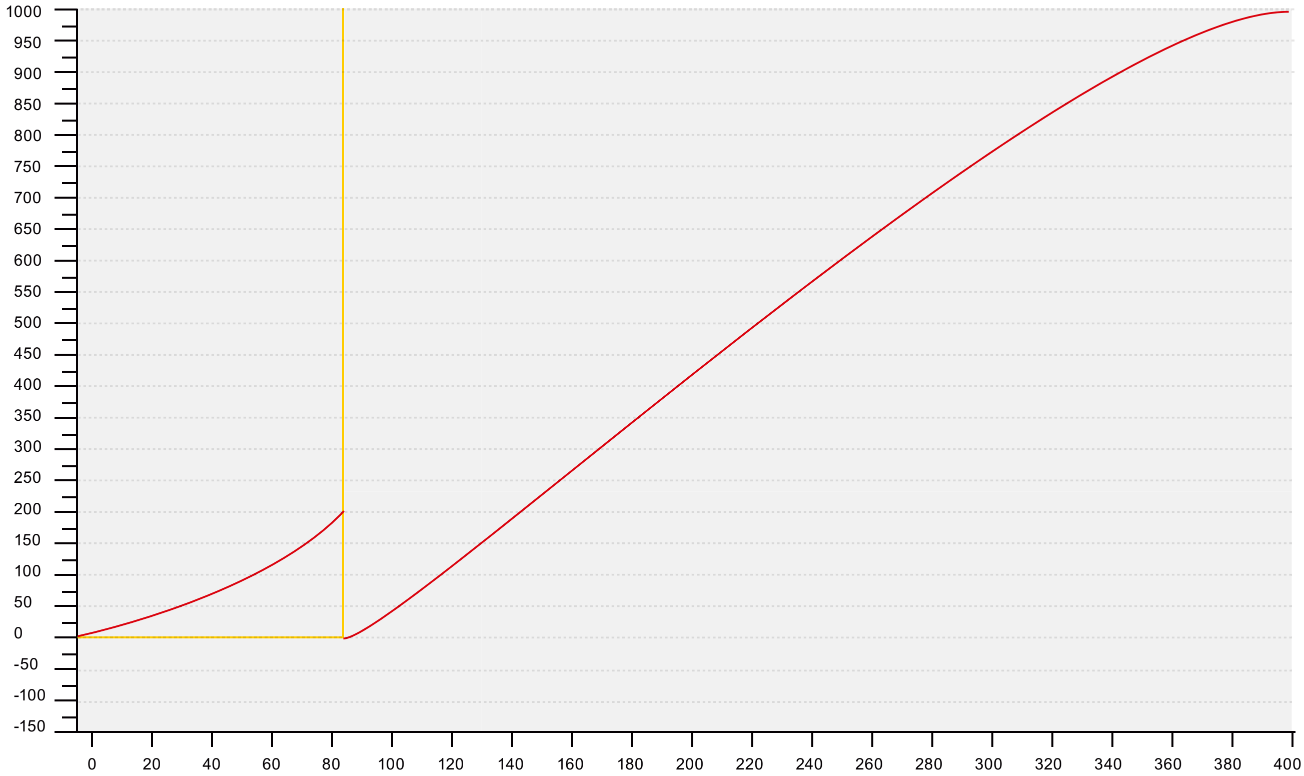
When a new axis is created in S7T Config, the *Default* homing mode is preset.

Homing mode with *encoder zero mark only* is used if the configured encoder provides a zero mark. Homing mode external zero mark only is used if the encoder does not provide a zero mark.

4.6.14.4 Positioning behavior with passive homing

The following examples show the response of the end position with passive homing with the motion commands "MC_MoveRelative" and "MC_MoveAbsolute":

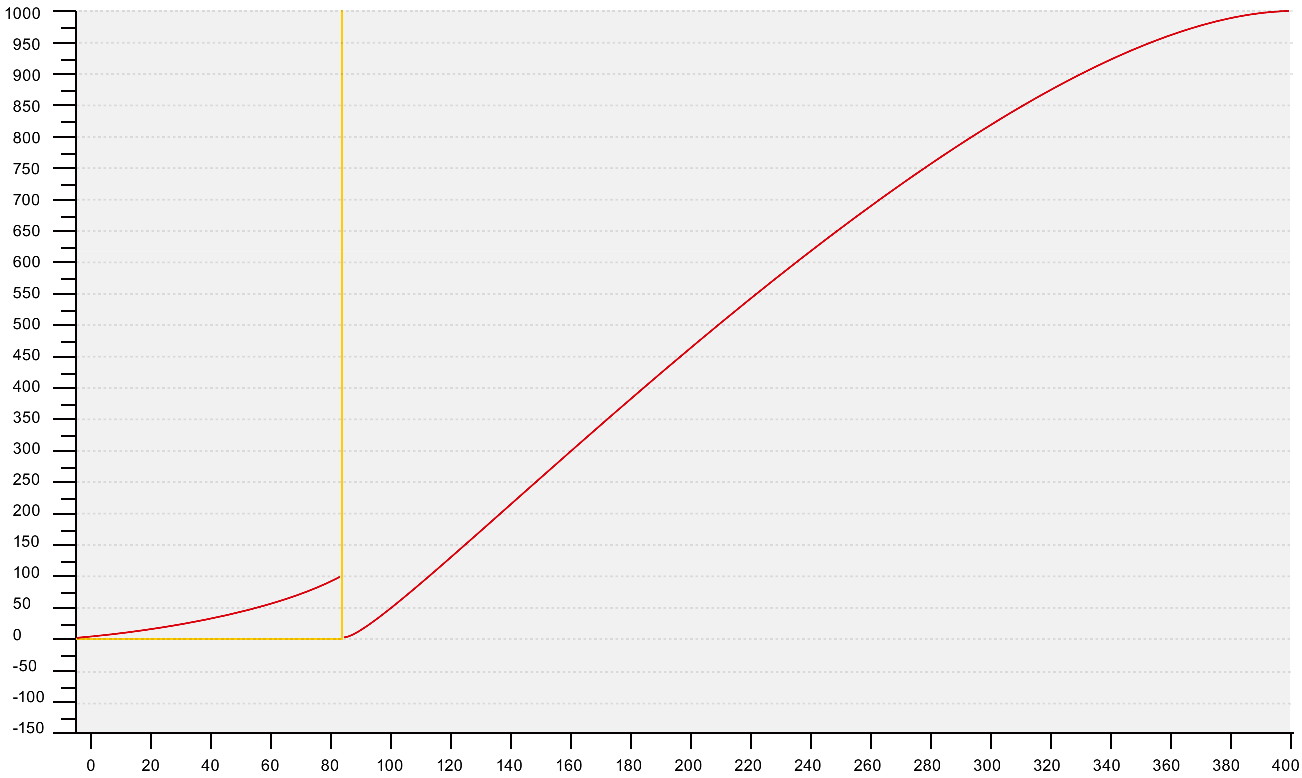
Positioning behavior with passive homing with "MC_MoveRelative"



An MC_MoveRelative command *1000* is started for passive homing.

1. The MC_MoveRelative command and passive homing start at the same time.
2. When the yellow marked homing position *200* is reached, the axis is homed and the position of the axis is set to position *0*.
3. The axis moves by the remaining relative distance to position *800 (1000-200)*.

Positioning behavior with passive homing with "MC_MoveAbsolute"



An MC_MoveAbsolute to position *1000* command is started for passive homing.

1. The MC_MoveAbsolute command and passive homing start at the same time.
2. When the yellow marked homing position *200* is reached, the axis is homed and the position of the axis is set to position *0*.
3. The axis moves to position *1000* in accordance with the new homing point.

4.6.14.5 Direct homing

The actual axis position is set to the value defined at the MC_Home technology function. No reference point shift settings are active. The function does not execute a motion. The axis is homed when the command is executed.

Start of the homing function

MC_Home	*Mode = 3* *Position = x*	Direct homing: The current position is assigned the value of the *Position* input parameter.

After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone= TRUE* .

4.6.14.6 Position correction

A correction value is deducted from the actual position value of the axis. By contrast to other homing modes, the axis maintains its homed state (homed / not homed) in this case.

The position correction function can also be used to manipulate the setpoints of the various coordinate systems (base coordinate system, superimposing coordinate system). This is of significance for superimposing camming in order to be able to generate a reference within a cam disk.

Start of the homing function:

MC_Home	*Mode = 4* *Position = x*	Actual value correction: Position value = (current position) - (parameter position).
	Mode = 6 *Position = x*	Setpoint correction in the base coordinate system: Position value = (current position) - (parameter position).
	Mode = 7 *Position = x*	Setpoint correction in the superimposing coordinate system: Position value = (current position) - (parameter position).

Position correction does not influence the *Statusword.HomingDone* status in the axis technology DB.

4.6.14.7 Motions with non-homed axes

You determine whether absolute positioning is to be available for a non-homed axis in the **Axis > Homing** dialog box.

Optional settings for "Homing required":

- No: Relative and absolute motions are possible. The Software limit switches (Page 149) are monitored.
- Yes: Relative motion only. The software limit switches are not monitored as long as the axis is not homed.

4.6.15 Monitoring functions

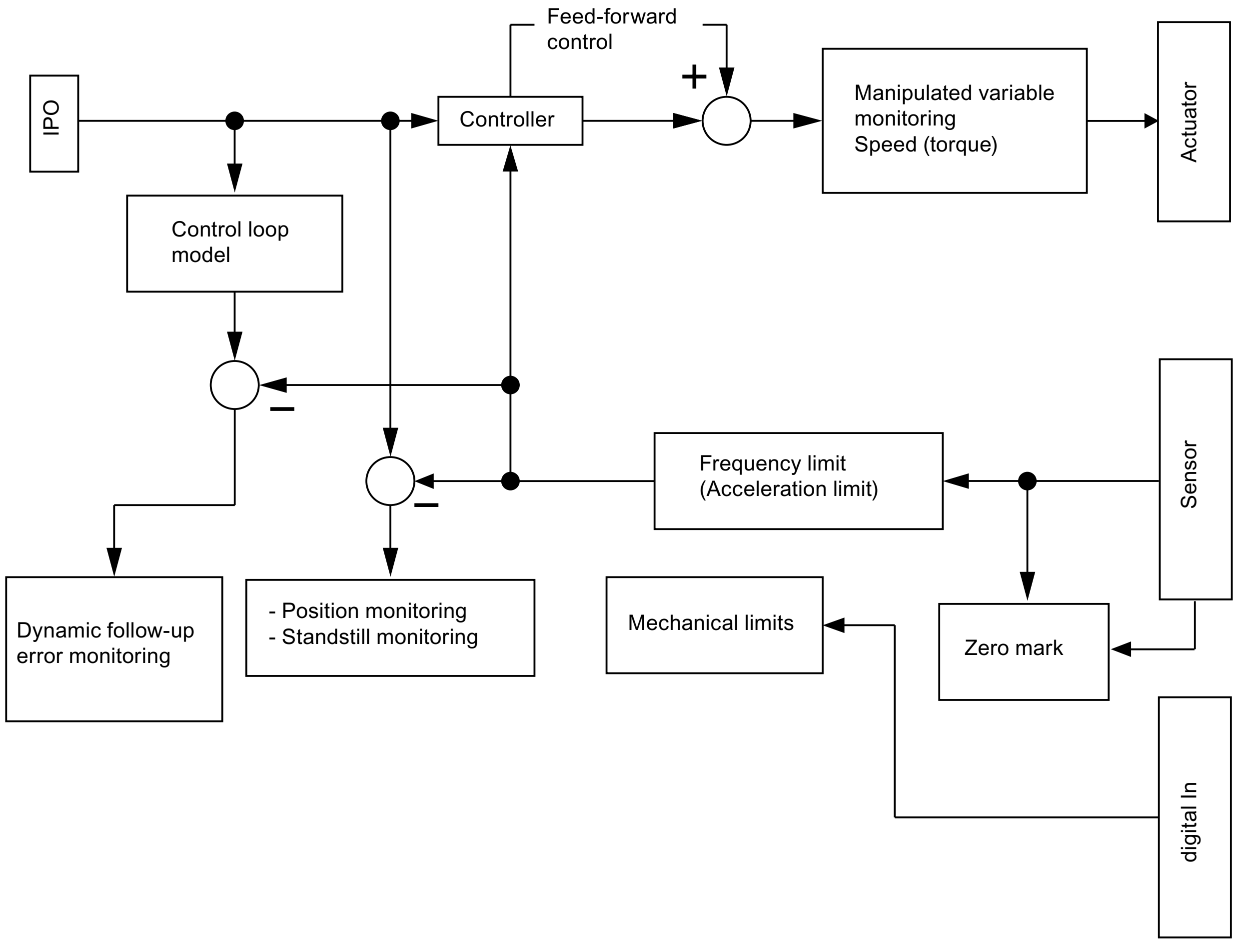
4.6.15.1 Monitoring functions - Overview

Axis monitoring functions you can configure in S7T Config:

Monitoring functions	Speed-controlled axis	Positioning axis	Synchronization axis
Velocity error monitoring (Page 187)	X	-	-
Positioning monitoring (Page 184)	-	X	X
Following error monitoring (Page 185)	-	X	X
Standstill signal (Page 186)	X	X	X
Software limit switch (Page 149)	-	X	X
Hardware limit switch (Page 147)	X	X	X
Synchronization monitoring (Page 321)	-	-	X
Manipulated variable monitoring (Page 187) (always active)	X	X	X
Encoder limit frequency monitoring (Page 919)	X (only with encoder)	X	X

Positioning axis monitoring

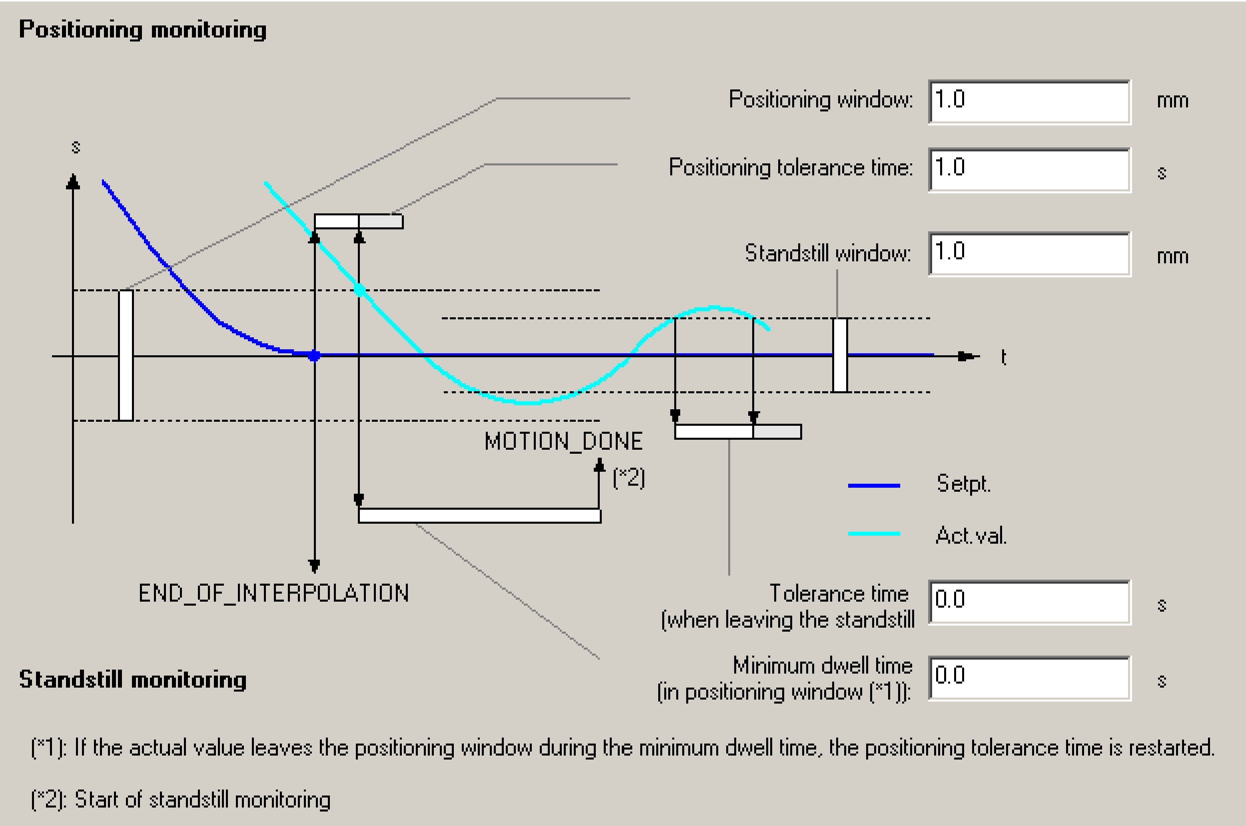
The figure below shows an example of positioning axis monitoring:



Errors are reported in the *ErrorStatus* parameter of the technology DB.

4.6.15.2 Monitoring functions - Positioning and standstill monitoring

Set the limit values for monitoring the positioning of position-controlled and synchronization axes in the "Positioning and standstill monitoring" tab of the **Axis > Monitoring functions** dialog box.



The "Positioning and standstill monitoring" tab shows whether the **Positioning** or **Synchronous operation** technology is set for the axis.

Positioning monitoring

At the end of a positioning motion, the function monitors the axis entry to the target position. For this purpose, a positioning window and a time tolerance within which the end position must be reached are specified. Monitoring is activated at the end of setpoint interpolation.

You can also set a minimum dwell time in the positioning window to expire before the positive feedback of the positioning command is activated. This time can be used for oscillating processes and control loops, for which the tolerance window should be less than the overshoot amplitude.

A positive feedback for the positioning command is output when the actual value reaches the positioning window.

The axis technology DB returns error *8019* if the positioning monitoring function is triggered.

Standstill (zero-speed) monitoring

Standstill monitoring is enabled when the position setpoint of a positioning command equals the value of the target position, and the delay of standstill monitoring activation has expired.

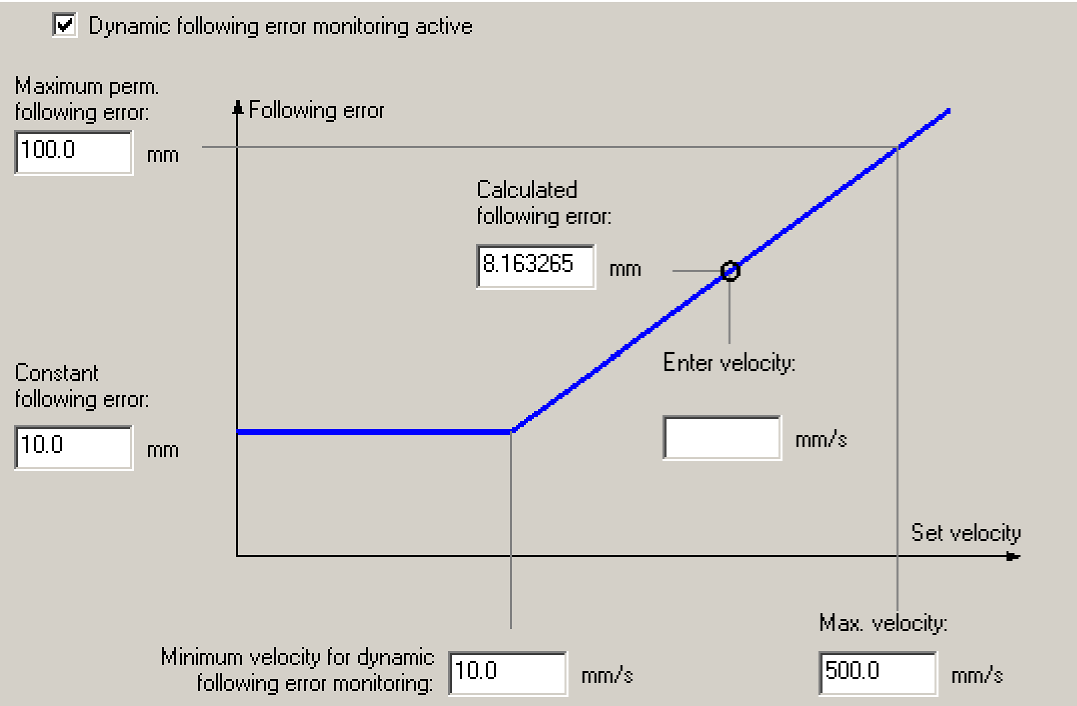
Standstill monitoring is triggered if the axis moves out of the configured standstill tolerance window for any time longer than the configured period. The axis technology DB returns error *8018* if standstill monitoring is triggered.

4.6.15.3 Monitoring functions - "Following error monitoring" tab

Dynamic following error monitoring can be enabled in the "Following error monitoring" tab of the Axis > Monitoring functions dialog box.

The following error monitoring on the position-controlled axis is performed using the calculated following error. The axis technology DB returns error *8016* if the offset between the actual position value and the position setpoint exceeds the programmed following error limit. The permitted following error depends on the velocity setpoint of the axis.

At velocities less than the configured minimum the permitted following error is constant and is programmed at the "constant following error" parameter. Above this limit, there is a linear increase of the following error to a maximum value which is defined by the parameter "maximum permitted following error" at maximum velocity. The permissible maximum following error is reached at maximum velocity.



If you specify a value in the "Enter velocity" input box, the corresponding function value is displayed in the "Calculated following error" box. This allows you to control how large the following error may be at the entered speed.

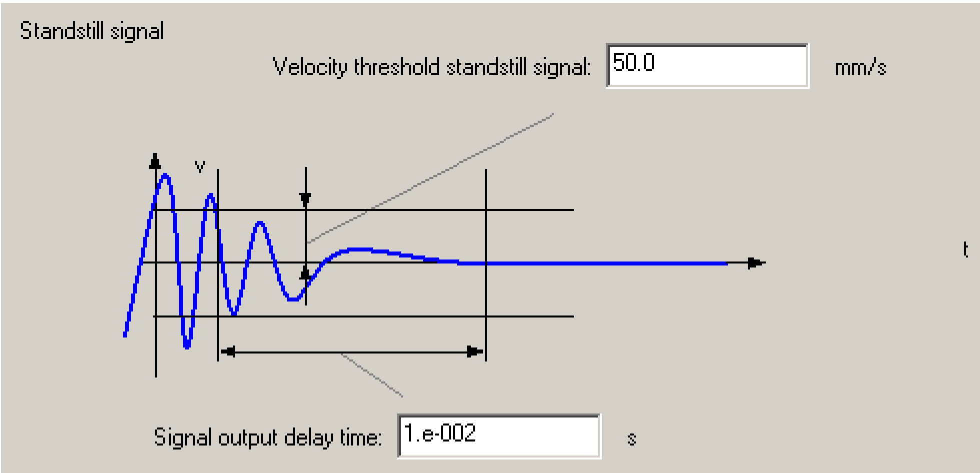
The "Following error monitoring" tab is shown for position-controlled real axes.

4.6.15.4 Monitoring functions - "Standstill signal" tab

The velocity threshold and the "Signal output delay" can be set in the the "Standstill signal" tab of the **Axis > Monitoring functions** dialog box.

The standstill signal sets the *standstill* bit in the status word of the technology DB if the current velocity is below the configured velocity threshold at least for the duration of the set delay time.

At speed-controlled and positioning axes the override is activated in speed-controlled mode. At the positioning axis, the override is activated when the positioning window is reached.

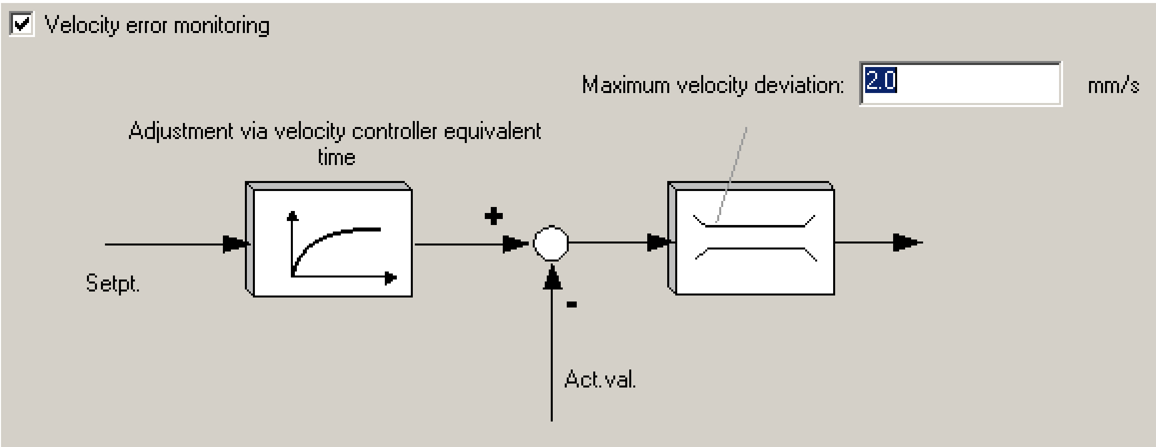


4.6.15.5 Monitoring functions - "Velocity monitoring" tab

Enable velocity error monitoring in the "Velocity error monitoring" tab of the **Axis > Monitoring functions** dialog box.

Velocity error monitoring is only relevant to these axes:

- Speed-controlled axis with encoder
- Positioning axes operating in speed-controlled mode ("MC_MoveVelocity" - input parameter *PositionControl = FALSE*)



Velocity error monitoring

Enable velocity error monitoring in this dialog.

Maximum velocity error:

This input box is only visible if velocity error monitoring is enabled.

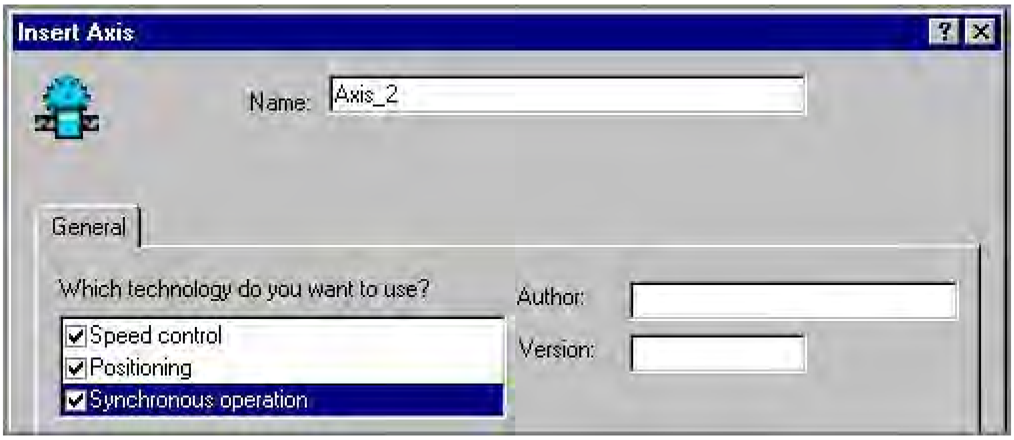
Enter the maximum velocity error in this dialog.

4.7 Configuring synchronization axes

Prerequisite

- The Technology CPU was configured in HW Config and configuration data were compiled.

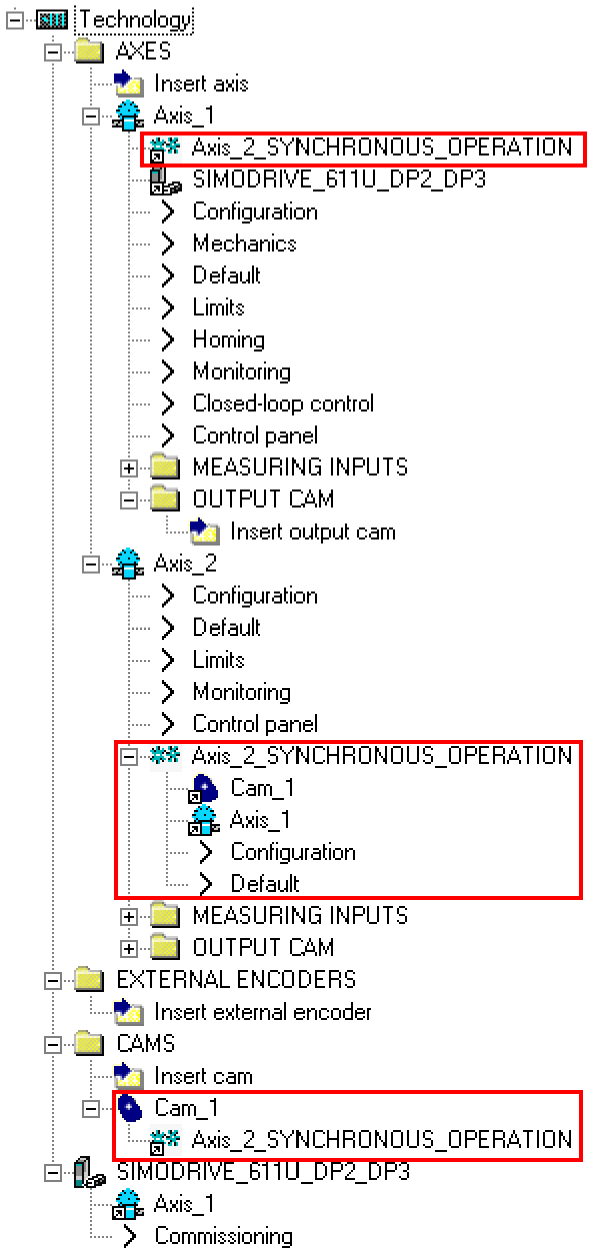
How to insert a synchronization axis in S7T Config

Step	Description
1.	In the Navigator of S7T Config, double-click "Insert axis".
2.	Activate the **synchronous operation** technology when you configure the axis.  This "Synchronization" technology requires the active "Positioning" and "Speed control" technologies. You can not for this reason disable these technologies.
3.	Go through the steps as described in the chapter "Configuring axes (Page 126)".
	Result: The synchronization axis appears in the Navigator of S7T Config, and its corresponding synchronization object will be created automatically. The object is automatically given the name of the axis with **_SYNCHRONOUS_OPERATION** ending.

Representation in the Navigator

The leading axes and cam disks assigned to the synchronization axis are symbolized by means of logic links:

- Below the synchronous operation object
- Below the cams used
- Below the leading axes (axes, external encoders)



Further procedures

For synchronous operation, assign the synchronization axis the corresponding leading axes and / or cam disks.

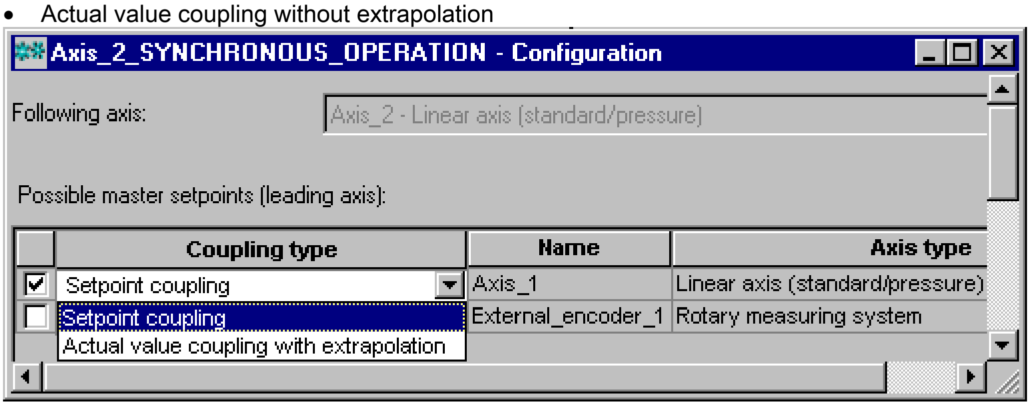
4.7.1 Assigning leading axes and cam disks

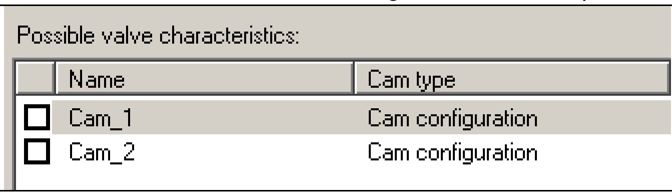
In order to implement gearing you must assign the following axis an available leading axis. In order to implement camming, you must assign the following axis an additionally available cam disk. Error! Bookmark not defined.

Prerequisite

- A following axis with the "Synchronism" technology was inserted in S7T Config.
- A leading axis or cam disk was inserted in S7T Config.

How to assign leading axes and cam disks to a following axis

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1.</td><td>In Navigator of S7T Config, double-click "Configuration" in the synchronization object (child object of the synchronization axis).</td></tr>
<tr><td>2.</td><td>In the next dialog box, assign the following axis one or several available leading axes. To do so, set the check box in the left column, and then select the relevant coupling mode. Possible coupling modes:
Leading axis is a real axis:
• Setpoint coupling
• Actual value coupling with extrapolation
Leading axis is a virtual axis:
• Setpoint coupling
Leading axis is an external encoder:
• Actual value coupling with extrapolation
• Actual value coupling without extrapolation


Which one of the axes is to provide the master setpoints to the following axis is determined in runtime by calling the relevant technology function, for example, "MC_GearIn" or "MC_CamIn".</td></tr>
<tr><td>3.</td><td>From the bottom section of the dialog box, select the optional cam disks for the implementation of camming.

</td></tr>
<tr><td>4.</td><td>Click "Close"</td></tr>
</table>

4.7.2 Configuring superimposing synchronism

A synchronization axis consists of an axis object and of a synchronization object. Both objects are generated when you create the axis. You configure superimposing synchronism by assigning the synchronization axis an additional synchronization object which coordinates the superimposing motions.

A synchronization axis may contain up to two synchronization objects:

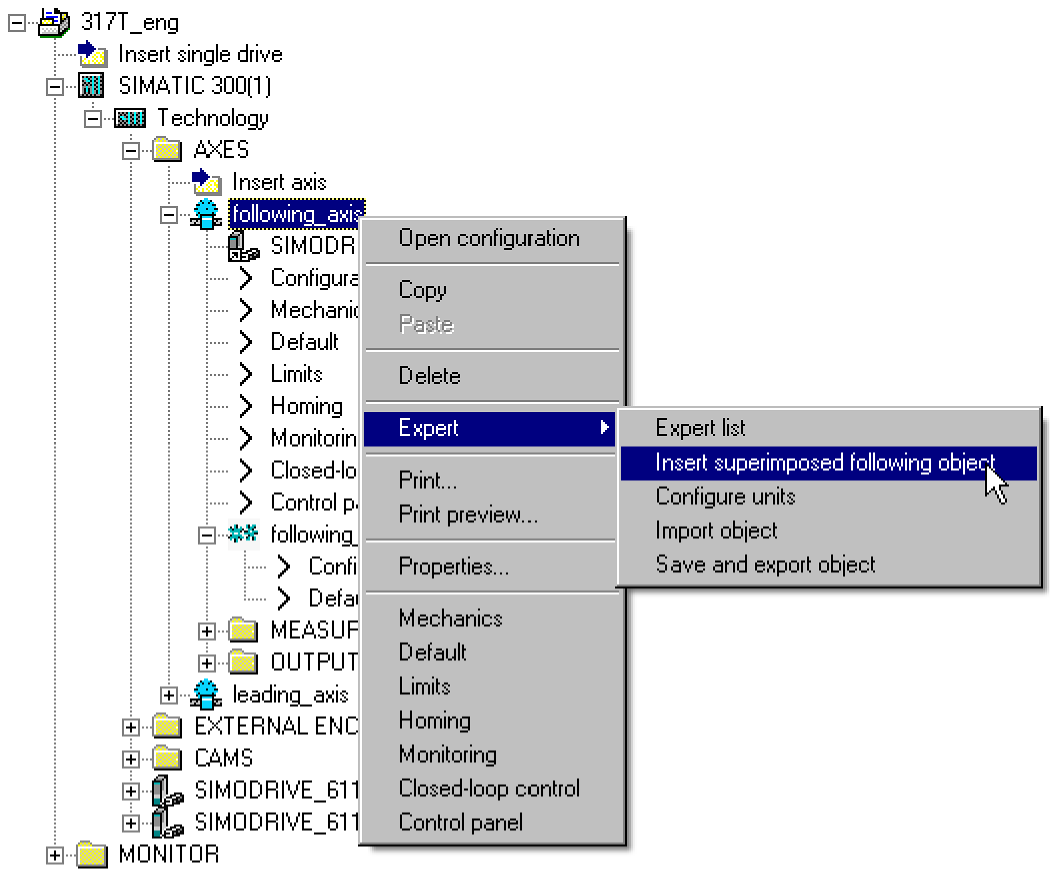
- One synchronization object for base synchronism
- One synchronization object for superimposing synchronism

Prerequisites

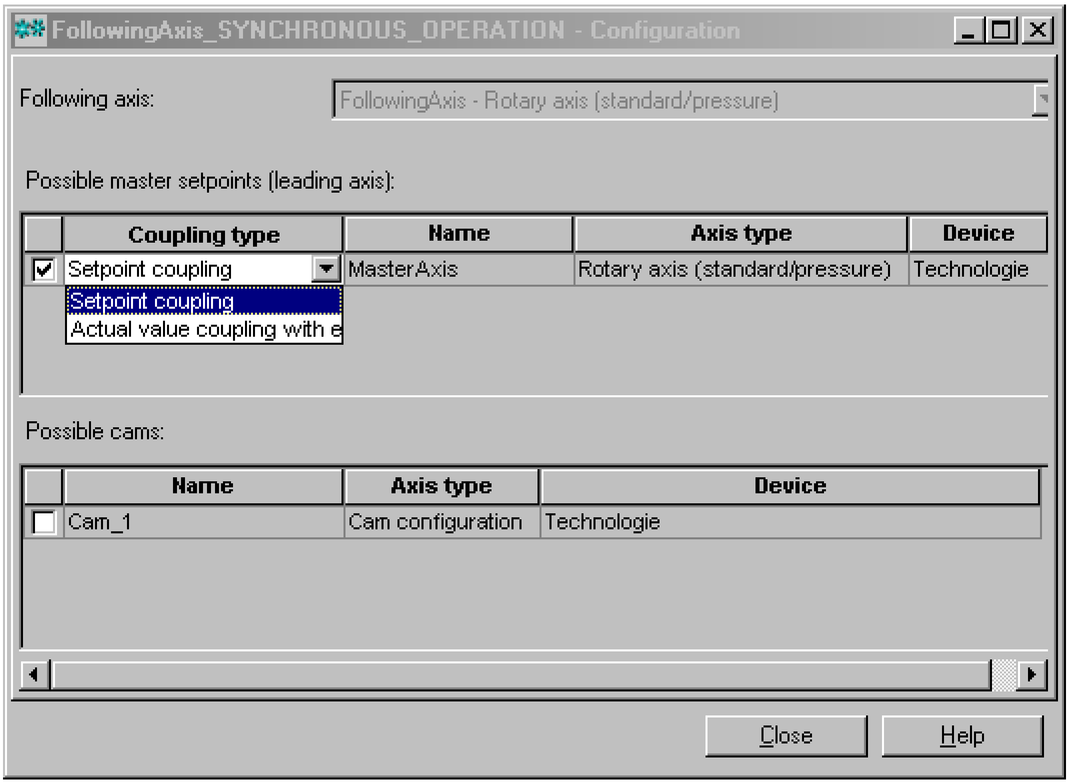
- A synchronization axis was configured in S7T Config.
- A superimposing synchronization object was not configured for the axis

Configuring a superimposing synchronization object

The next steps show you the basic procedures of configuring a superimposing synchronization object.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1.</td><td>In S7T Config Navigator, select the synchronization axis at which you want to configure superimposing synchronization. Select Expert > Superimposed synchronous object from the context menu.


Result: The Navigator shows a second synchronization object under synchronization axis.</td></tr>
<tr><td>2.</td><td>Double-click the Configuration entry of the second synchronization object to open the configuration dialog box.</td></tr>
</table>

Step	Description
3.	Finalize the synchronous group of the superimposing synchronism by assigning it a leading axis (including cam disks as required). Set the corresponding check boxes, and then select the coupling type at the leading axes.  The synchronization-specific parameters of the superimposing synchronization object are configured and assigned values as for basic synchronous operation.

Distinguishing between synchronization objects

Whether the synchronization object at a synchronization axis represents a basic synchronism or superimposing synchronism is defined at the *SyncingMotion.motionImpact*configuration parameter in the expert list of the synchronization object.

SyncingMotion.motionImpact	STANDARD (0)	Basic synchronous operation
	SUPERIMPOSED_MOTION (1)	Superimposing synchronism

4.7.3 Synchronization

Coupling to the master setpoint during synchronization or desynchronization can be defined differently according to the application.

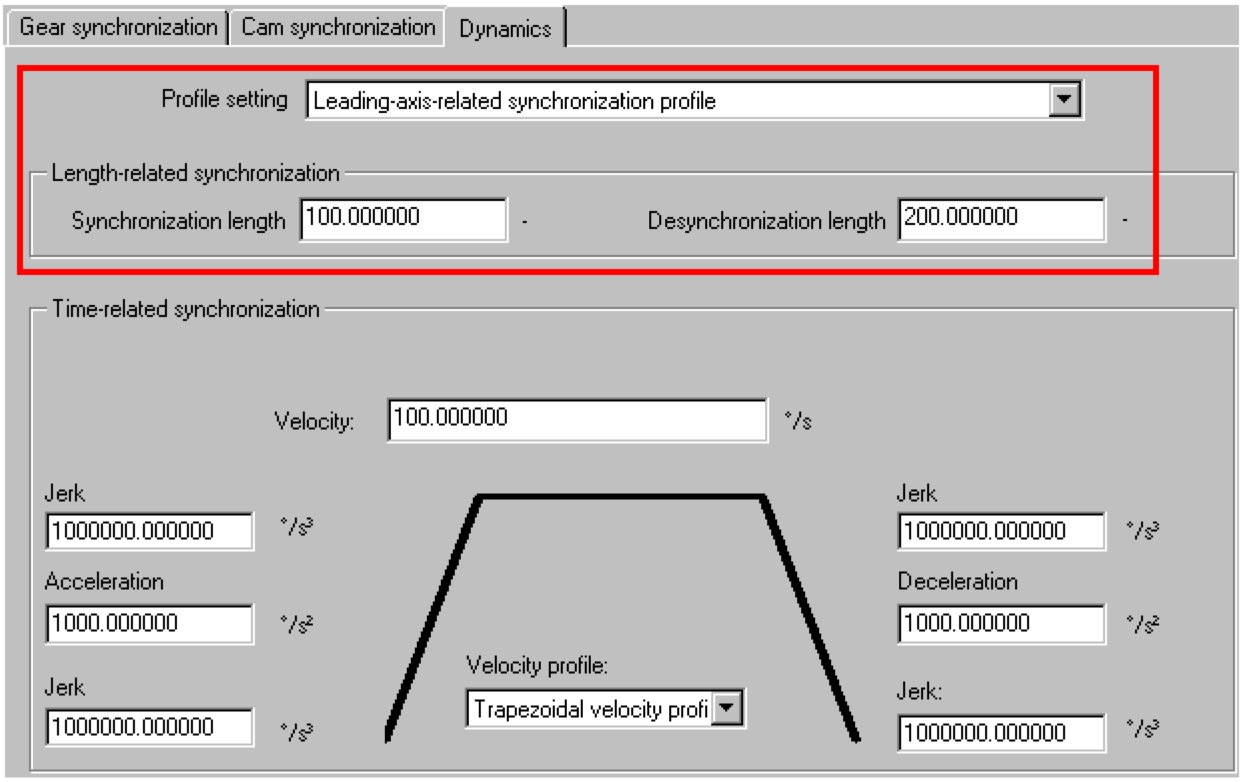
It is determined by:

- The synchronization mode
- The synchronization point
- The position of the synchronization point
- The synchronization criterion
- The direction of synchronization (modulo axes only)

You can set the following synchronization modes in the "Dynamics" tab of the **Synchronous operation > Default** dialog box:

- **Leading axis-related synchronization profile:** Synchronization and desynchronization with master value reference
- **Time-related synchronization profile:** Synchronization and desynchronization on the basis of specified dynamic values

Synchronization based on a leading axis-related synchronization profile



The leading axis-related synchronization profile is defined at parameters "Sync. length" and "Desync. length".

Sync. Length	Synchronization length
Desync. Length	Desynchronization length

For leading axis-related synchronization, a start or end value relative to the master setpoint is defined for synchronization, including a synchronization length ("Sync. length" or "Desync. length").

A synchronization profile is determined, that is, a dynamic transition is calculated independent of the dynamic profile of the master setpoint.

The velocities at the edges of the synchronization operation are constant.

The synchronization length defined in the dialog forms the basis for the definition of the synchronization range with reference to the master setpoint. Synchronization is handled using the "MC_GearIn" (gearing), or "MC_CamIn" (camming), or "MC_GearInSuperImposed" (superimposed gearing) or "MC_CamInSuperImposed" (superimposed camming) technology functions.

Input parameter *Mode* at the technology function determines whether to activate the values set in S7T Config (*Mode = 0*) for the synchronization profile, or whether to activate dynamic defaults at the technology function (*Mode = 1*).

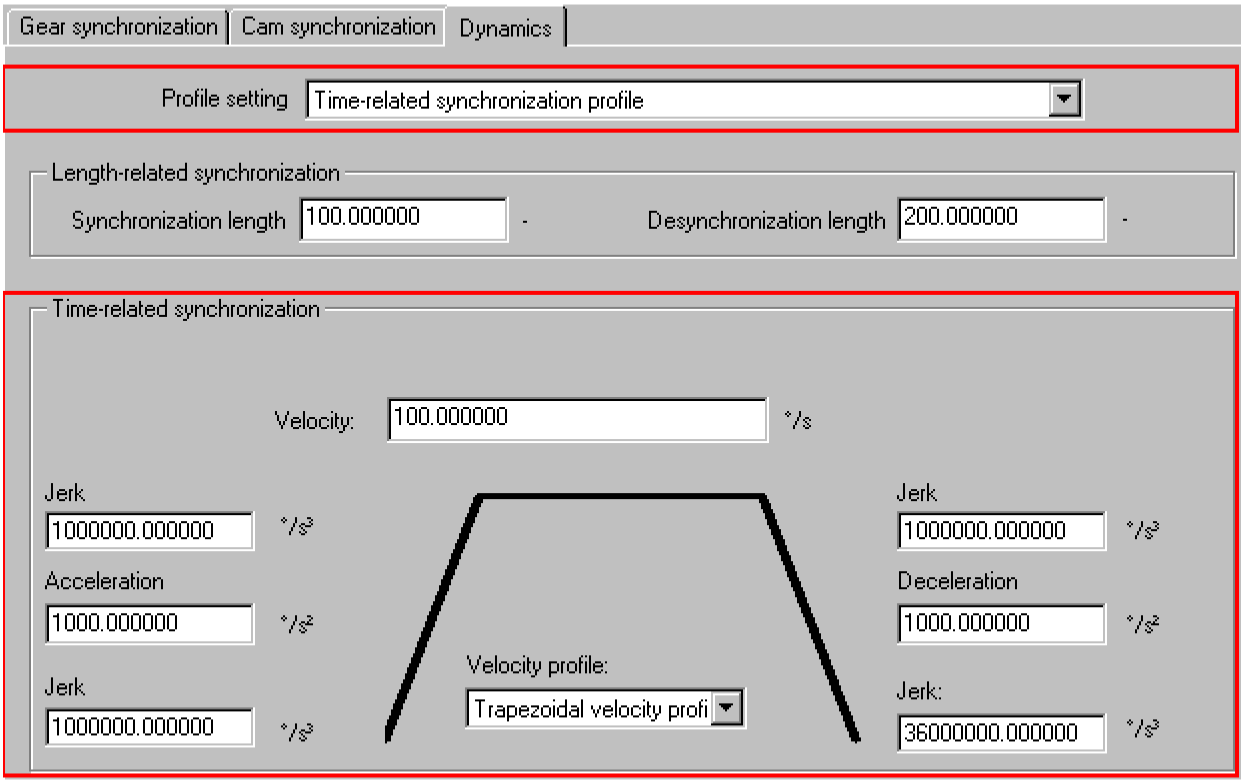
Note

The synchronization profile depends on the velocity of the leading axis and on the set synchronization length. Any dynamic definitions at the synchronization object will be ignored. For this reason, dynamic response is limited only at the slave axis.

Synchronization based on a time-related synchronization profile

Based on dynamic parameters set in S7T Config or at the block input, time-based synchronization is performed after the start of the synchronization function.

Input parameter *Mode* at the technology function determines whether to activate the values set in S7T Config (*Mode = 0*) for the synchronization profile, or whether to activate dynamic defaults at the technology function (*Mode = 1*):



The time-related synchronization profile is defined by the "Jerk", "Acceleration", "Deceleration", "Velocity" and "Velocity profile" parameters.

Jerk	Jerk settings supported for the velocity transitions listed below: • Jerk at the start of acceleration • Jerk at the end of acceleration • Jerk at the start of deceleration • Jerk at the end of deceleration
Acceleration	Acceleration of the axis during synchronization
Deceleration	Deceleration of the axis during synchronization
Velocity	Maximum velocity of the axis during synchronization Whether the velocity can be reached depends on the synchronization conditions
Velocity profile	The following velocity profiles are available: • Trapezoidal velocity profile • Constant velocity profile

Retarded and advanced synchronization

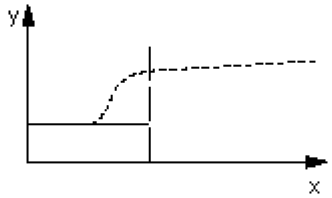
In time-based synchronization mode **advanced synchronization** and **retarded synchronization** can be differentiated.

	Synchronization start point…	Synchronization end point…
Advanced synchronization	Calculated by the system based on default dynamic values and master setpoint characteristic.	Defined implicitly based on the following axis position, or directly
Retarded synchronization	Defined implicitly based on the following axis position, or directly	Derived from the dynamic values and master setpoint characteristic

Note

When synchronizing a synchronization axis with default dynamic values for which the start point of synchronization is defined, the following axis accelerates to its target velocity along a hyperbolic ramp.

Advanced synchronization (synchronize before synchronization position)



For advanced synchronization, the synchronization motion starts.

- At the start point of synchronization, based on default dynamic response parameters and constant master setpoint value, if this synchronization can be performed within an optimal time (dynamic master setpoint)
- Immediately, if an optimal synchronization time can not be calculated and the synchronization position can be reached (for example, with static master setpoint)

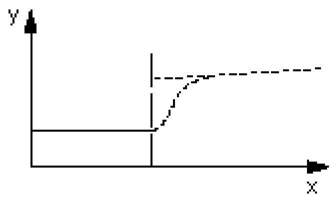
Note

If synchronization cannot be performed optimally a reversal of the motion may be generated.

Any change of the velocity master setpoint is also applied correspondingly to the dynamic values of the synchronization operation. A warning message is output if the configured tolerance is exceeded.

The synchronization is canceled with error if the master setpoint is inverted while synchronization is active.

Retarded synchronization (start symmetrically to synchronization position)



In retarded synchronization mode

- The synchronization motion starts when the start criterion is reached
- Only a trapezoidal velocity profile is supported

  If a velocity profile with constant acceleration is set, a "Dynamic response of motion profiles... cannot be maintained" alarm is generated, and the system automatically toggles to a trapezoidal velocity profile.

Synchronization direction

You can set the synchronization direction in the following dialog boxes:

- **Gear synchronization**
  Axis_SYNCHRONOUS_OPERATION > Default dialog box, "Gear synchronization" tab, "Synchronization direction"
- **Cam synchronization**
  Axis_SYNCHRONOUS_OPERATION > Default dialog box, "Cam synchronization" tab, "Synchronization direction"

The settings described below are valid for synchronization and desynchronization.

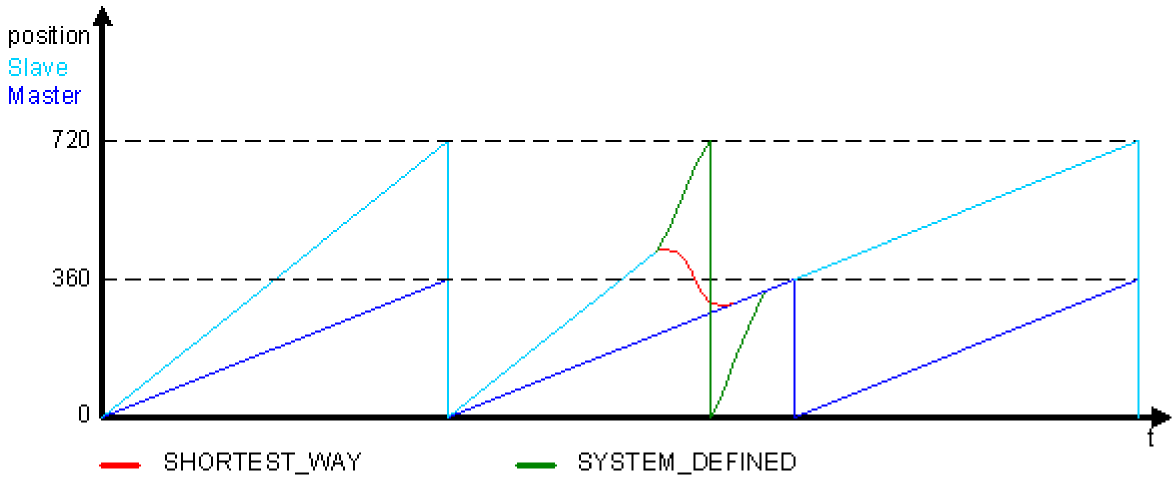
You can also program the settings at the *userdefault.gearingsettings.synchronizingdirection* or *userdefault.cammingsettings.synchronizingdirection* system variables.

- **Compatibility mode** (*SYSTEM_DEFINED*)
  The following axis is synchronized within the shortest possible distance. The direction will be retained if synchronization of the moving following axis is possible in the currently set direction. The direction is reversed if the current direction of motion has to be changed for synchronization. The system calculates the motion sequences required for synchronization.
- **Shortest distance** (*SHORTEST_WAY*)
  The following axis is synchronized within the shortest possible distance.
- **Negative synchronization direction** (*NEGATIVE_DIRECTION*)
  Only synchronization in negative direction is allowed, that is, transitions from negative to positive velocities during synchronization must be avoided. If synchronization fails, S7T Config outputs error message "50007 Error when activating/deactivating synchronous operation".
- **Positive synchronization direction** (*POSITIVE_DIRECTION*)
  Only synchronization in positive direction is allowed, that is, transitions from positive to negative velocities during synchronization must be avoided. If synchronization fails, S7T Config outputs error message "50007 Error when activating/deactivating synchronous operation".

Compatibility mode is always used if *Mode = 1* at "MC_CamIn" or "MC_GearIn".

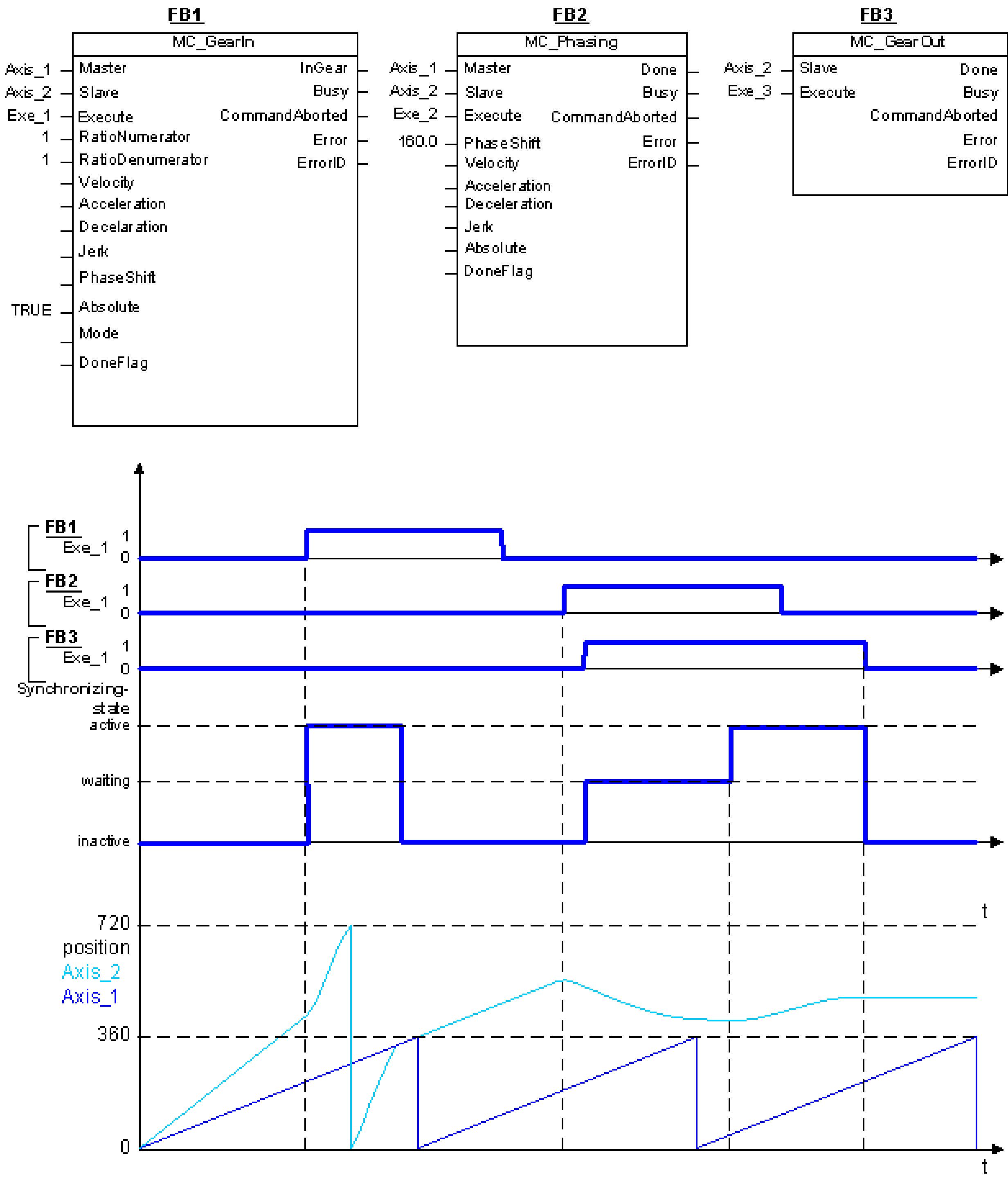
Difference between "SHORTEST_WAY" and "SYSTEM_DEFINED"

Both axes move, for example, with positive velocity. Absolute gearing with a ratio of 1:1 and immediate synchronization are initiated. The green curve shows the course when the current direction of motion of the following axis is retained. The red curve, on the other hand, shows synchronization with a reversal of direction.



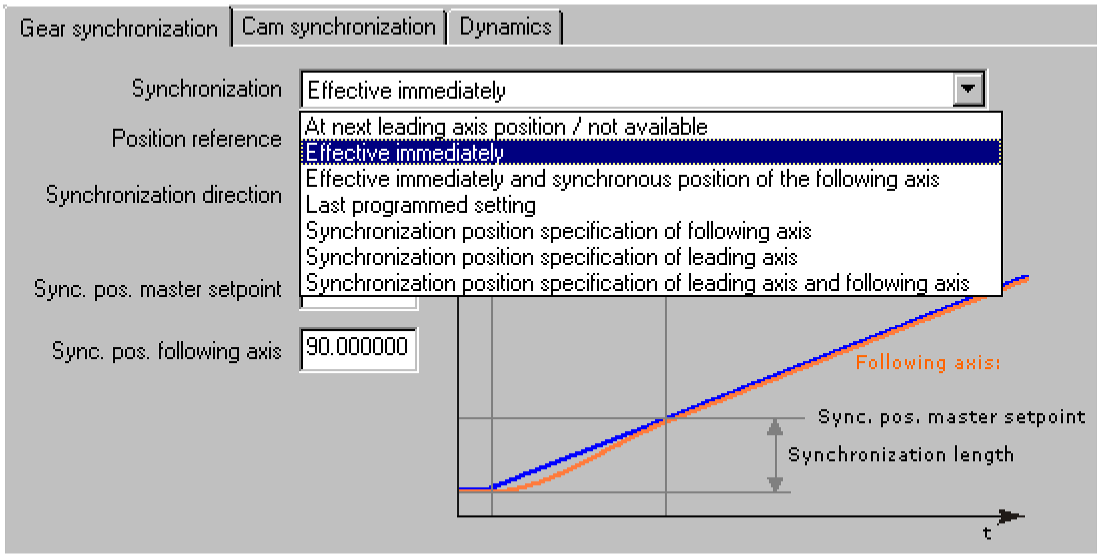
Example of "Positive synchronization direction" (Gearingsettings.synchronizingdirection = POSITIVE_DIRECTION)

The next example shows a default setting which only allows synchronization / desynchronization motions in positive direction of movement. The default settings for synchronization / desynchronization are "Effective immediately". Synchronization starts immediately with *Exe_1*. The slave is now to be moved by a phase shift with *Exe_2*. This motion is carried out in negative direction. If the "immediately" active "MC_CamOut" is started in negative direction with *Exe_3*, the axis waits until it is moving in positive direction before it is desynchronized


FB1
MC_GearIn
Axis_1
Axis_2
Exe_1
1
1
TRUE
Master
Slave
Execute
RatioNumerator
RatioDenumerator
Velocity
Acceleration
Decelaration
Jerk
PhaseShift
Absolute
Mode
DoneFlag
InGear
Busy
CommandAborted
Error
ErrorID
FB2
MC_Phasing
Axis_1
Axis_2
Exe_2
160.0
Master
Slave
Execute
PhaseShift
Velocity
Acceleration
Deceleration
Jerk
Absolute
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
FB3
MC_GearOut
Axis_2
Exe_3
Slave
Execute
Done
Busy
CommandAborted
Error
ErrorID
FB1
Exe_1
FB2
Exe_1
FB3
Exe_1
1
0
Synchronizing-
state
active
waiting
inactive
720
position
Axis_2
Axis_1
360
0
t
t

4.7.4 Synchronization

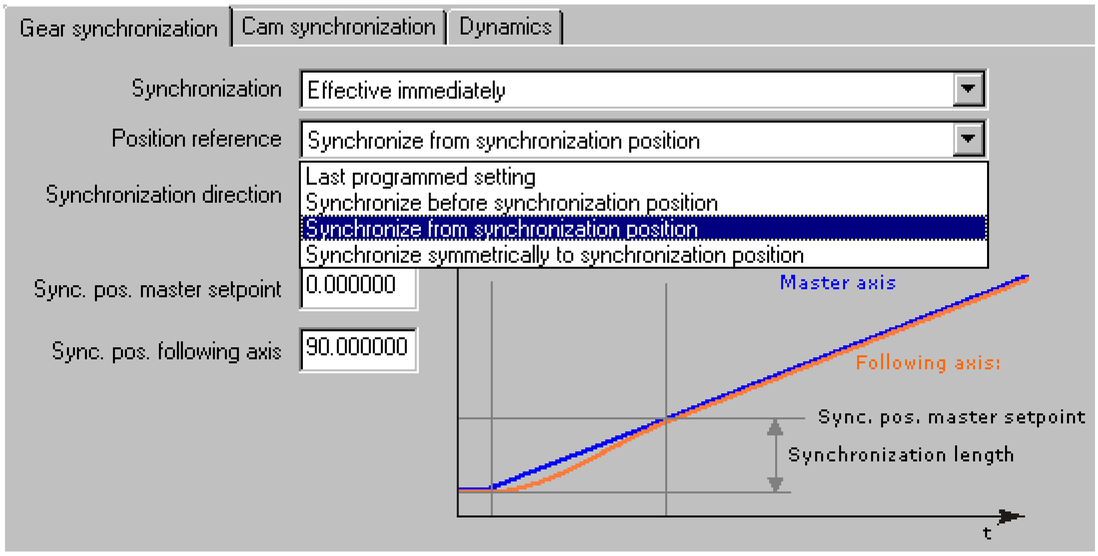
You can set the synchronization conditions in the "Gear synchronization" and "Cam synchronization" tabs of the **Synchronous operation > Default** dialog box.



The synchronization condition is set using the "Synchronization" drop-down list box:

Setting	Meaning
Effective immediately	Synchronization is performed immediately after the start of the function. The settings in "Sync. pos. master setpoint", "Sync. pos. following axis" and "Position reference" are not evaluated.
Default synchronization position of the leading axis	The synchronization criterion depends on the position of the leading axis. The synchronization position is defined in "Sync. pos. master setpoint". The setting in "Sync. pos. following axis" is ignored.
Default synchronization position of the following axis (only with gearing)	The synchronization criterion depends on the position of the following axis. The synchronization position is defined in "Sync. pos. following axis". The setting in "Sync. pos. master setpoint" is ignored.
Transition at the end of the active cam disk (only with camming)	This can only be set with relative leading axis reference. The synchronization criterion is the master setpoint position at the end of the current cam disk cycle. The setting in "Sync. pos. following axis" is ignored.
Default synchronization position of the leading axis and following axis	The synchronization criterion depends on the position of the leading axis. The synchronization position is defined in "Sync. pos. master setpoint". In addition, an offset is generated at the following axis as a result of the setting in "Sync. Pos. following axis", that is, instead of being synchronized with the programmed (for example, via cam disk) following position but, the following axis is synchronized with the "Sync. Pos. following axis" position **plus** the absolute position value of the following axis from the cam disk. **Example:** The following axis is coupled to the master axis by way of cam disk. The cam disk was created so that the following axis is set to position *70* when the leading axis is at position *50*. If parameter "SyncPos following axis" = *80* the following axis is synchronized at position 150 (*70 + 80*) when the leading axis is at position *50*. Coupling by way of the cam disk is still active (scaling remains the same), however, with an offset of *80*.
Effective immediately and synchronous position of the slave axis	Synchronization is performed immediately after the start of the function. The resulting phase shift between the leading and following slave axes results is derived from the position of the following axis when the function was started. This compensates for offset caused by the acceleration ramp of the following axis. The synchronization motion is based on dynamic response defaults. The setting in "Sync. pos. master setpoint" is ignored.
Last programmed setting	Not usable

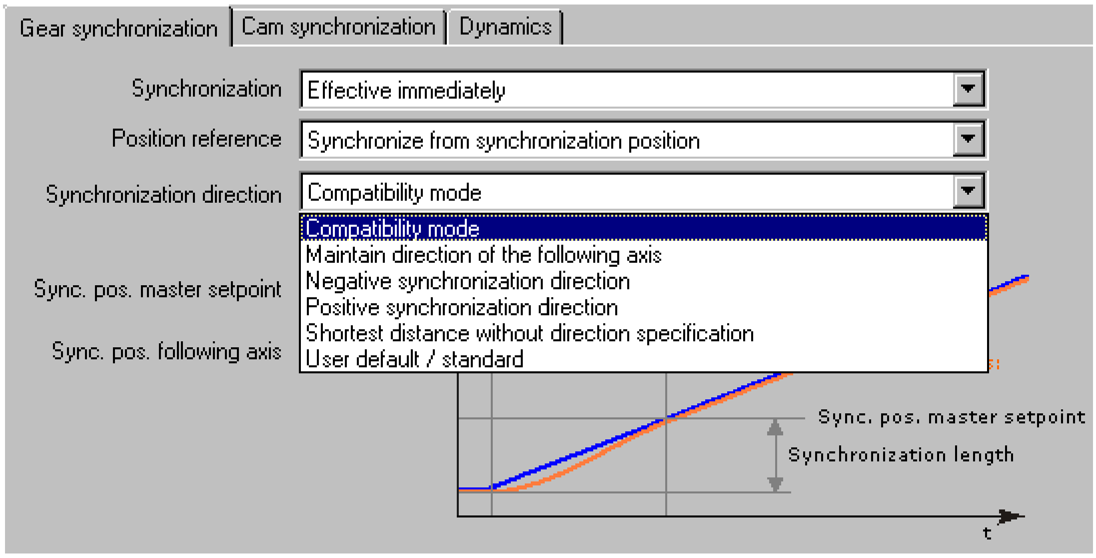
Position reference



The synchronization mode is defined in the "Position reference" drop-down list box:

Setting	Meaning
Synchronize from synchronization position	Synchronization starts at the synchronization position. The synchronization length is derived from the time-related synchronization profile in dynamic data, and for a leading axis-related synchronization profile from the "Synchronization length" parameter. The following axis is synchronized when this length has been passed.
Synchronize before synchronization position	In this synchronization mode, the following axis is in synchronism with the leading axis when it reaches the synchronization position. The start position of the synchronization with time-related synchronization profile is determined by the dynamic data, and with leading axis-related synchronization profile by the "Synchronization length" parameter.
Synchronize symmetrically to synchronization position	In this mode, synchronism is reached at the middle of the synchronization length. The synchronization movement starts before the synchronization position is reached and is stopped after the position has been passed. The synchronization is performed depending on the profile in relation to the synchronization length or dynamic defaults.
Last programmed setting	Not usable

Synchronization direction

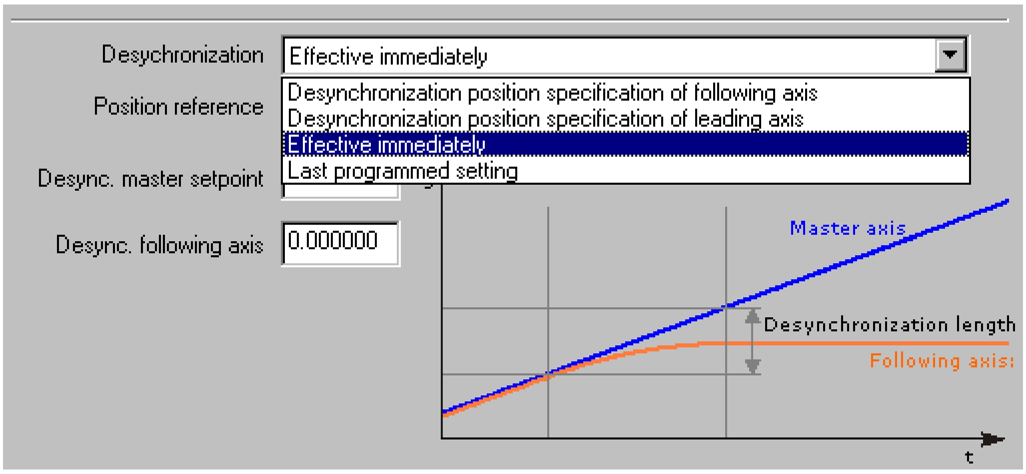


The direction of synchronization is determined by the setting in the "Synchronization Direction" drop-down list box. This setting is only available for modulo axes.

Setting	Meaning
Same direction as master	Synchronization always takes place in the direction of motion of the following axis.
Compatibility mode	**Gearing:** The following axis is synchronized in positive direction **Camming:** The following axis is synchronized within the shortest distance.
Shortest path without direction preset	The slave axis is synchronized over the shortest path.
Negative synchronization direction	The axis is always synchronized in negative direction.
Positive synchronization direction	The axis is always synchronized in positive direction.

4.7.5 Desynchronization

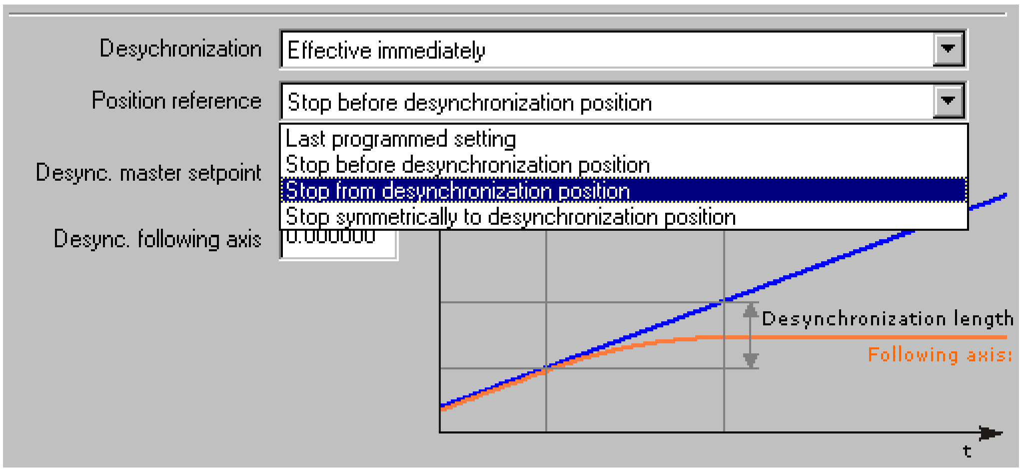
You can set the desynchronization conditions in the "Gear synchronization" and "Cam synchronization" tab of the **Synchronous operation > Default** dialog box.



The desynchronization condition is set in the "Desynchronization" drop-down list box.

Setting	Meaning
Effective immediately	Desynchronization starts immediately after the start of the function. The settings in "DeSync. master setpoint", "Desync. following axis" and "Position reference" are ignored.
Default desynchronization position of the following axis	Desynchronization starts based on the following axis position set in "DeSync following axis". The settings in "DeSync. leading axis" are ignored.
Specified by the desynchronization position of the leading axis	Desynchronization starts based on the leading axis position set in "DeSync master setpoint". The settings in "DeSync. following axis" are ignored.
End of cam disk cycle (only with camming)	The desynchronization is performed at the end of the current cam disk cycle.
Last programmed setting	Not usable

Position reference



The desynchronization mode is set in the "Position reference" drop-down list box:

Setting	Meaning
Stop after desynchronization position	Starts desynchronization at the desynchronization position. The desynchronization length is derived from dynamic data for a time-based synchronization profile, and from the "Desynchronization length" parameter for a leading axis-related synchronization profile. Desynchronization starts after this length has been passed.
Stop symmetrically to desynchronization position	In this mode, desynchronization starts when the axis reaches the middle of the desynchronization length. The desynchronization motion is started in advance of the desynchronization position, and stopped when the position is passed. Desynchronization is performed based on the profile in relation to the desynchronization length or dynamic defaults.
Stop before desynchronization position	In this mode, desynchronization starts when the axis reaches the desynchronization position. The start position of the desynchronization with time-related synchronization profile is determined by the dynamic data, and with leading axis-related synchronization profile by the "Synchronization length" parameter.
Last programmed setting	Not usable

4.7.6 Synchronization status

The section below shows system variables of S7T Config that can be used to view the synchronization status:

System variable at the synchronization object:

- enablecommand
  The system variable indicates whether a "MC_GearIn" or "MC_CamIn" command was started at the synchronization object. The following states are indicated:
  - *ACTIVE* - Command active
  - *NONE* - No command active
  - *WAITING_TO_START* - Command waiting for the synchronization criterion
- disablecommand
  The system variable indicates whether a "MC_GearOut" or "MC_CamOut" command was started at the synchronization object. The following states are indicated:
  - *ACTIVE* - Command active, following axis is being desynchronized
  - *NONE* - No command active
  - *WAITING_TO_START* - Command waiting for the desynchronization criterion
- syncstate
  This system variable returns the synchronization status. The following states are indicated:
  - *NO* - The synchronous operation has not been synchronized yet.
  - *YES* - The synchronous operation is synchronized.
- state
  This system variable indicates the active synchronization mode at the synchronization object. The following states are indicated:
  - *CAMMING* - A curve synchronization is active
  - *INACTIVE* - No synchronous operation active

System variable at the following axis:

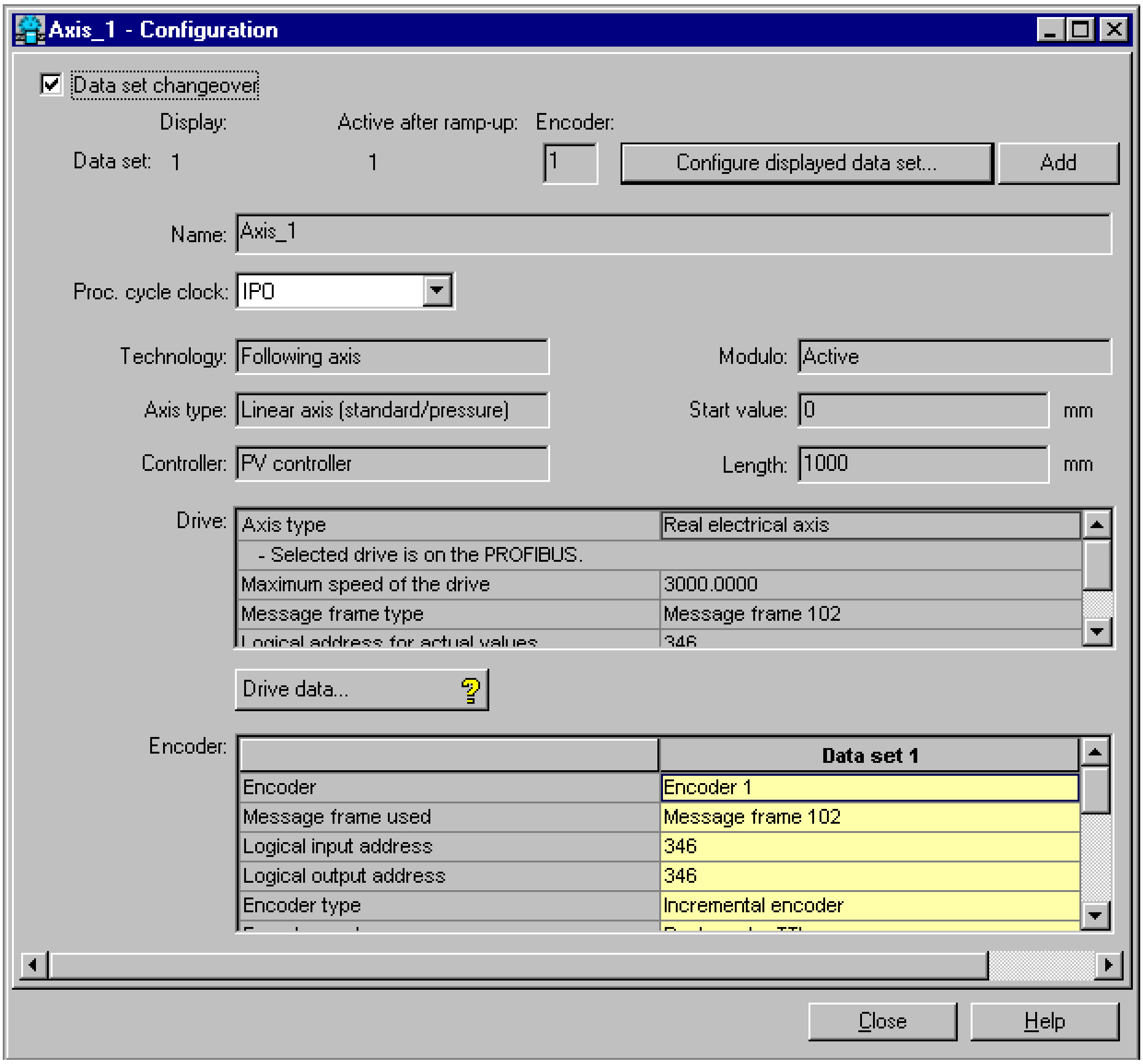
- syncmonitoring.differencecommandvalue
  This system variable indicates the master setpoint difference between the leading axis and the following axis.

4.7.7 Configuration

4.7.7.1 Configuration - Axis

The **Axis > Configuration** dialog box shows the axis and drive settings

You can edit these settings in a dialog box by clicking "Configure active data set".



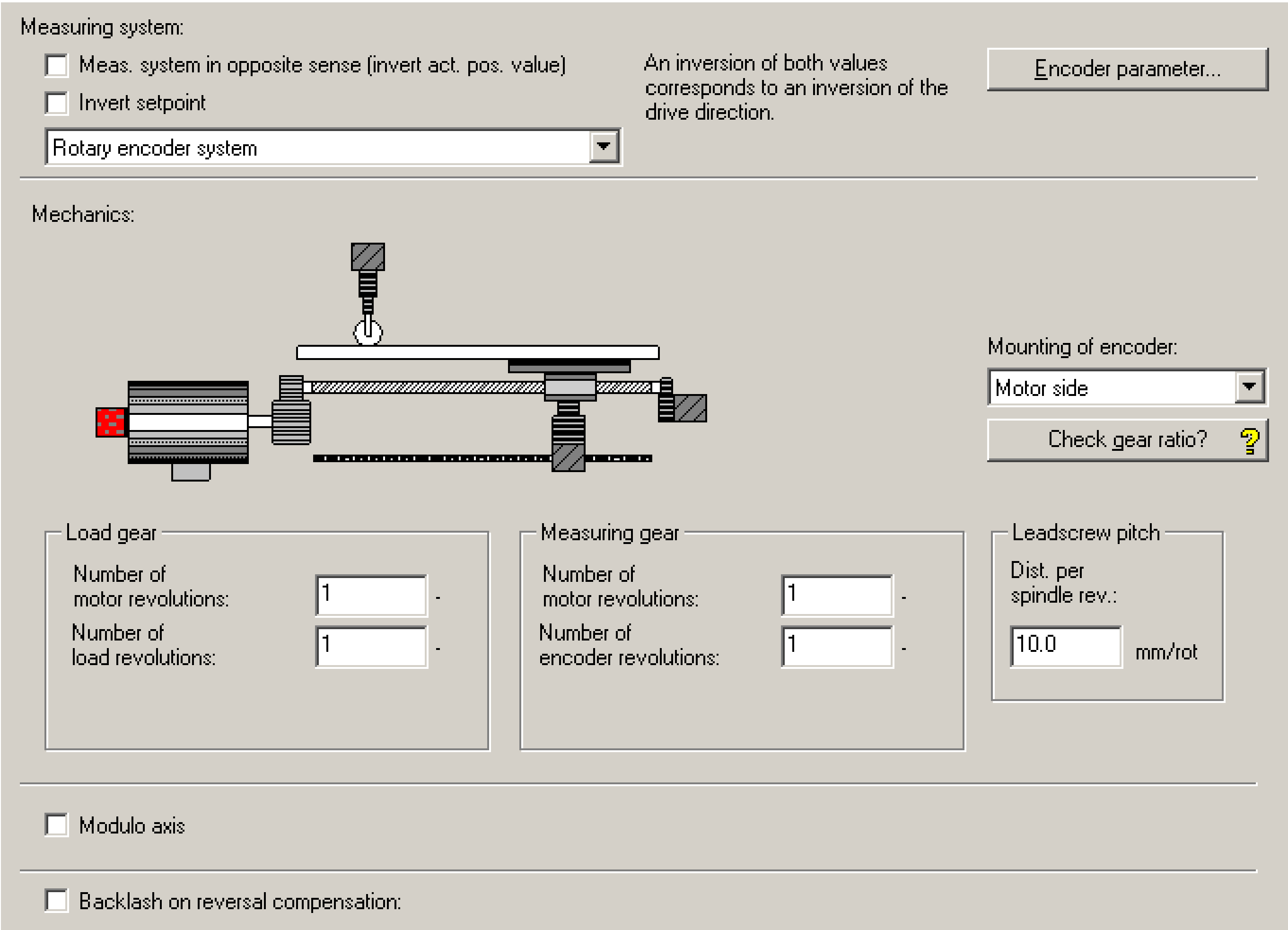
This dialog box also contains functions for managing the data records for a data record changeover. If the technology object supports multiple data records, you can:

- Create new data records
- Delete data records
- Define which data record to load for the technology object during CPU startup, and
- Reconfigure the selected data record.

4.7.8 Mechanical system

4.7.8.1 Mechanics - Electrical axis

The **Axis > Mechanics** dialog box is used to configure the mechanical parameters of the axis and of the encoder.



Settings for the axis:

- Drive direction settings
- Load gear settings
- Leadscrew pitch of a linear axis
- Modulo settings
- Backlash compensation settings

Settings for the encoder:

- Measuring gear settings
- Backlash compensation
- Encoder mounting type

Conditions of determining gear parameters

The configuration may return error messages indicating incompatible gear parameters. The selection of incompatible configuration data may lead to internal overflows in the integrated technology. Formulae used to determine suitable parameters:

f_1 = Numerator measuring gear x 360 x internal resolution x denominator load gear

f_2 = Denominator measuring gear x encoder resolution x actual value factor x numerator load gear

From f_1 and f_2, the greatest common divisor **k** must be determined and used in the following formula:

$f_{11} = (f_1/k) < 2^{32}$

$f_{22} = (f_2/k) < 2^{32}$

The results of f_{11} and f_{22} must be less than 2^{32}. If this is not the case, check whether an appropriate modification of the parameters in the f_1 and f_2 formulas will produce values for f_{11} and f_{22} that do not exceed the maximum permissible value.

Check the following value if f_{11} and f_{22} meets the requirements described above and a configuration error is detected nonetheless:

f_3 = Modulo length x internal resolution

Now you can calculate the greatest common divisor k_2 of f_3 and f_{11} and insert it in the following formula:

$f_{31} = ((f_3 \times f_{22} \times f_{11}) / (k_2 \times k_2)) < 2^{63}$

At this point you have to check whether f_{31} is less than 2^{63}. If not, check whether you can reduce modulo length. You can also make changes in the parameters in the f_1 and f_2 formulas, as long as you ensure that the requirements for f_{11} and f_{22} are still met.

Table 4-4 Parameter description

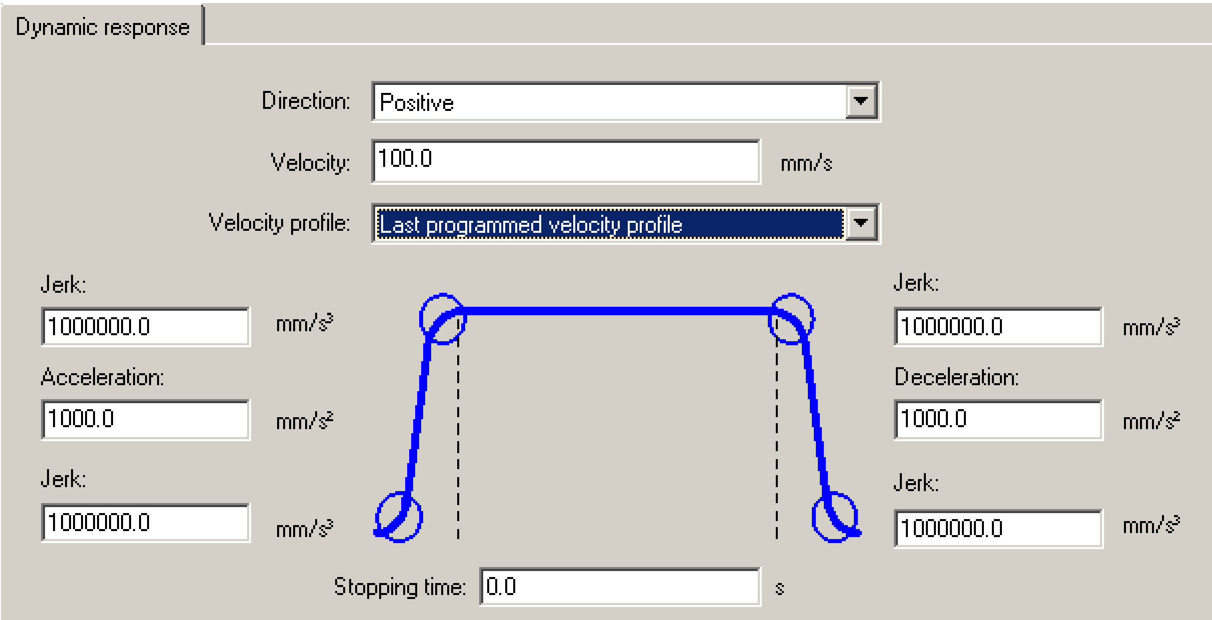
Parameters	Comment	Configuration parameter at the axis
Measuring gear numerator	Numerator of the measuring gear ratio	AdaptDrive.numFactor AdaptExtern.numFactor AdaptLoad.numFactor
Measuring gear denominator	Denominator of the measuring gear ratio	AdaptDrive.denFactor AdaptExtern.denFactor AdaptLoad.denFactor
Load gear numerator	Load gear ratio numerator	Gear.numFactor
Load gear denominator	Load gear ratio denominator	Gear.denFactor
Internal resolution	Internal increments / position unit	Defined in the configuration using the axis wizard.
Actual value coefficient	= X for DP drive	X for absolute value encoders: IncEncoder.incResolution MultiplierCyclic X for absolute value encoders: AbsEncoder.absResolution.MultiplierCyclic

A calculation tool is available in a contribution (FAQ) to this topic on the Internet at http://www.siemens.com/automation/service&support.

4.7.9 Default

4.7.9.1 Default - "Dynamics" tab

The default values for axis dynamics can be set in the "Dynamics" tab of the **Axis > Default** dialog box. The default values are activated if a negative dynamic value is set at a technology function.



Default values supported:

- Direction
- Velocity
- Acceleration
- Deceleration
- Jerk
- Velocity profile

Stop time

The time set at "Stop time" is in effect when a moving axis is disabled, and if *Stopmode* = *2* is set at "MC_Power".

Velocity profile

The velocity profile defines the response of the axis during startup and when braking, and to velocity changes.

The technology functions do not use the default value. If you want to change the velocity profile, use the *Jerk* input parameter.

You can choose between the following profiles:

- Trapezoidal

  The trapezoidal profile is used for linear acceleration in positive and negative direction of the motion (jerk = 0).

- Constant

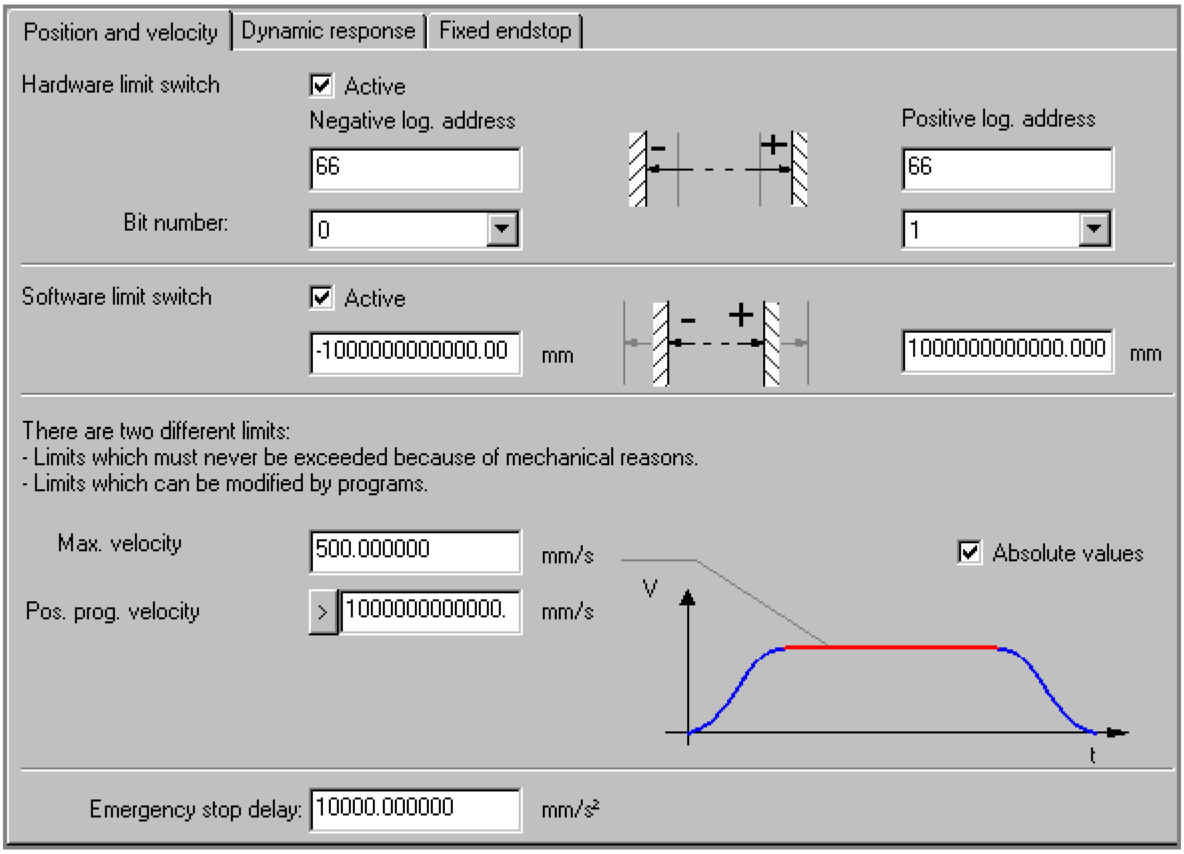
  The profile shows a constant acceleration, the jerk profile is controllable (jerk <> 0).

4.7.10 Limits

4.7.10.1 Limits - "Position and velocity" tab

In the "Position and velocity" tab of the **Axis > Limits** tab you can

- Configure and enable monitoring of the hardware limit switches (Page 147)
- Configure and enable monitoring of the software limit switches (Page 149)
- Set velocity limits
- Set emergency-off deceleration



Emergency off deceleration

The value set for emergency off deceleration is activated when a moving axis is disabled and *Stopmode* = 0 is set at the "MC_Power" parameter.

4.7.10.2 Hardware limit switches

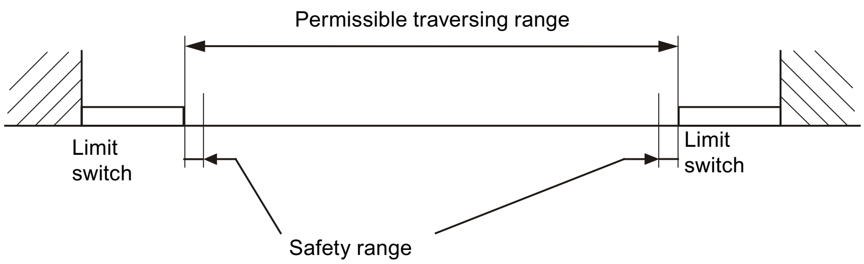
You enable hardware limit switch monitoring in the **Axis > Limits** dialog box, "Position and Velocity (Page 146)" tab. Hardware limit switch monitoring is used to limit the operating range of an axis, or to protect the machine.

Wiring

The HW limit switches can be wired to the four integrated digital inputs of the Technology CPU, or to I/O modules such ET 200 or SINAMICS S120 with TM15/TM17 which are operated on DP(DRIVE).

Traversing range

The permitted traversing range is monitored at the digital inputs of the hardware limit switches.



NOTICE
The hardware limit switch must be implemented as an NC contact. The hardware limit switches must remain active after the axis has passed the permissible traversing range until the mechanical end position is reached.

Retraction

An axis triggering a HW limit switch is stopped with error messages *8013* and *804B*. It is retracted from the HW limit switch (release motion) as described below:

- **Manual retraction**

  The axis is returned manually to the permissible traversing range. The error **at the technology DB** can only be acknowledged after it has been returned to this range.

- **Retraction with drive**

  The error at the technology DB of the axis is acknowledged, but the error message and the *LimitSwitchActive* bit remain active. The axis can now be returned to the permitted traversing range. A reverse motion command once again triggers an axis error. The error messages and the *LimitSwitchActive* status can be acknowledged after the axis has moved out of the range of the limit switch.

The current position of the axis is saved upon reaching the HW limit switch is saved. Only after this position, plus a safety range, has been passed is the axis considered to have left the limit switch.

NOTICE
The controller may **not** be switched off after the axis has passed the HW limit switch in order to avoid a conflict between polarity monitoring of the hardware limit switches and the overrun monitoring of the hardware limit switches in direction of the valid range. In this case, the axis is moved into the valid range without hardware limit switch monitoring, and is then re-enabled. When the controller is **switched on**, the axis must be positioned within the valid traversing range.

Internal states are lost and the configuration is reloaded when the axis passes the hardware limit switch. Reloading without loss of the approach information is only possible within the valid range.

Exception: Deactivation of position limit monitoring after a polarity reversal error

Safety range

The safety range of the HW limit switches is calculated based on the configured resolution of the system of units of the axis.

Safety range = 1000 / (increments / position)

Example: A linear axis is assigned the position unit "mm" and a resolution (increments / position) of "1000/unit" in the "Configure units" dialog box, meaning that the axis position is calculated to an accuracy of 0.001 mm. The safety range in this example is a multiple of 1000 of the accuracy: 1 mm.

4.7.10.3 Software limit switch

You can configure the software limits switches and enable the monitoring function in the **Axis > limits** dialog box, "Position and Velocity (Page 146)" tab. If the software limit switches are activated, the traversing distance of the axis is limited with the software limit switches.

Software limit switches should lie within the range of the HW limit switches with reference to the traversing range in order to limit the working range of an axis, for example.

Monitoring software the software limit switches at start of motion:

- **Check box activated**
  If the target position of a position-controlled motion command lies beyond the software limit switch, the warning *0026* is already displayed in the technology DB of the axis when the motion is started. In S7T Config the interrupt *40105* is signaled.
  The axis travels up to the position of the software limit switch and the error *8014* is reported in the technology DB.
- **Check box not activated**
  If the target position of a position-controlled motion command lies beyond the software limit switch, the axis traverses until the pPosition of the software limit switch is reached. Error *8014* is reported in the technology DB (Warning *0026* is not output).

Behavior for travel to software limit switch:

- **For position-controlled traversing:**
  The axis traverses until the software limit switch is reached. After the software limit switch has been reached, the respectively active position controlled or speed-controlled mode remains.
- **In all operating modes:**
  The axis traverses until the software limit switch is reached. After the software limit switch has been reached, an active position controlled mode remains. In speed-controlled mode the axis changes to position-controlled mode.

Negative position / positive end position:

Enter the positions for the negative and the positive end positions of the software limit switches in these input fields.

Tolerance window for retraction:

Enter a suitable value in this input field in order to prevent renewed triggering of the software limit switch error during retraction.

Note

The response of software limit switches is determined by the axis configuration in terms of the homing function. The limit switches of a configuration requiring a homed axis for the execution of absolute motion commands ("Homing required: Yes" in the **Axis > homing dialog box**) are not monitored if the axis is not homed. By contrast, if the execution of absolute motion commands is allowed when the axis is not homed ("Homing required: No" in the **Axis > Homing dialog box**) the software limit switches are monitored, regardless of the homing state of the axis.

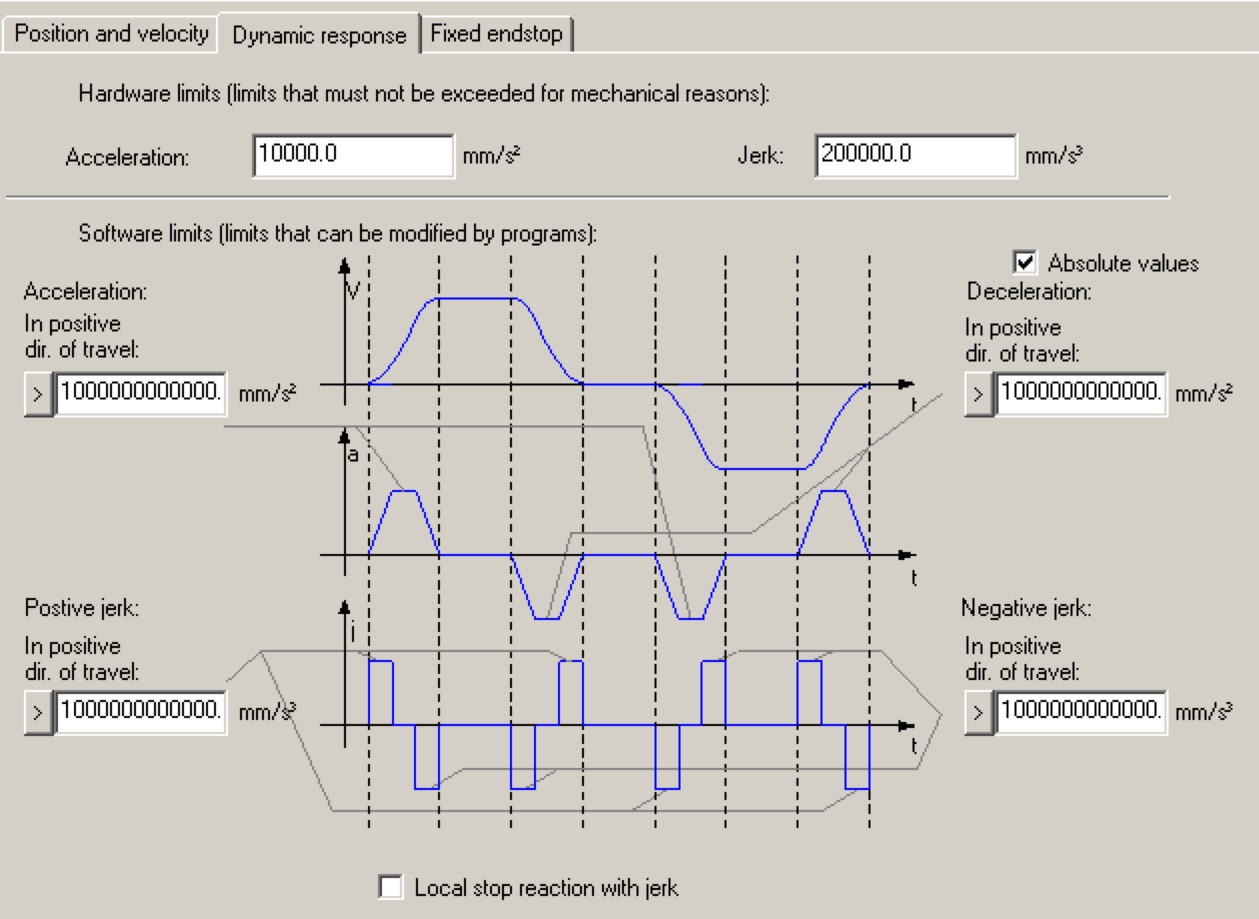
4.7.10.4 Limits - "Dynamic response" tab

You can adjust the HW limits (mechanical) and SW limits in the "Dynamic response" tab of the **Axis > Limits** dialog box.

The set deceleration limit is activated when a moving axis is disabled and *FastStop* = *1* is set at the "MC_Power" parameter.

The **status-dependent** acceleration model is enabled (default) by setting the "Absolute values" check box.

The **direction-dependent** acceleration model is enabled by resetting the "Absolute values" check box. Additional parameters are displayed in this case.



Status-dependent acceleration

- **Acceleration**

  Axis acceleration, independent of the direction of movement

- **Deceleration**

  Axis deceleration, independent of the direction of movement

Direction-dependent acceleration

- Acceleration

  Acceleration in the positive motion direction and deceleration in the negative motion direction
- Deceleration

  Acceleration in the negative motion direction and deceleration in the positive motion direction

Parameter settings with dynamic direction vector are useful, for example, for suspended axes.

Local stop reaction with jerk

Set the "Local stop reactions with jerk" check box to execute a stop reaction triggered by an alarm reaction at the axis, with jerk limiting and rounding.

Emergency-stop deceleration

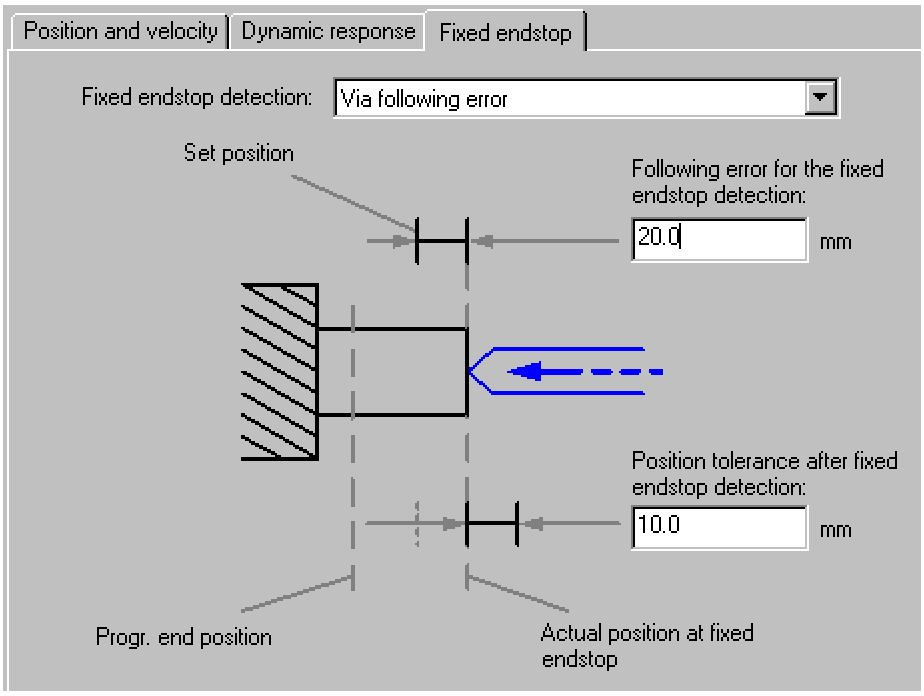
The set emergency-stop deceleration comes into effect when a moving axis is disabled and *Stopmode = 0* is set at "MC_Power."

4.7.10.5 Limits - "Fixed end stop" tab

The "Fixed end stop" tab of the **Axis > Limits** can be used to enable fixed end stop detection and to set the corresponding detection mode:

- Use following error
- Using the force/torque

When the selected condition is met, the "fixed end stop" status is reached. Following error monitoring is disabled if "Move to fixed end stop" is enabled.



Use following error

Note

When the axis moves onto the fixed end stop, and fixed end stop detection with "use following error" is set, the entry in "Position tolerance after fixed end stop detection" should be significantly less than that of "Following error for fixed end stop detection".

Use torque value

The fixed end stop detection function "use torque value" requires that the digital drive assigned to the axis supports torque limiting, and that a corresponding message frame, for example, *message frame 102* or *message frame 105*.

Move to fixed end stop

The "MC_MoveToEndPos function activates the "Move to end stop" function and sets the clamping torque after the end stop is reached. This operation is also known as "clamping."

The motion is stopped when the axis reaches the fixed end stop and the control remains active. The setpoint value at the position controller input is held constant. New motion commands in direction of the clamping position are canceled; new motion control commands in retraction direction are executed to reduce torque. The position setpoint of the axis is used as start position of new motion control commands in retraction direction.

The position setpoint of the axis results from one of the following equations, depending on the fixed end stop detection function:

- "Use following error"
  Position at fixed end stop + following error
- "Use torque value"
  Position at fixed end stop + clamping tolerance

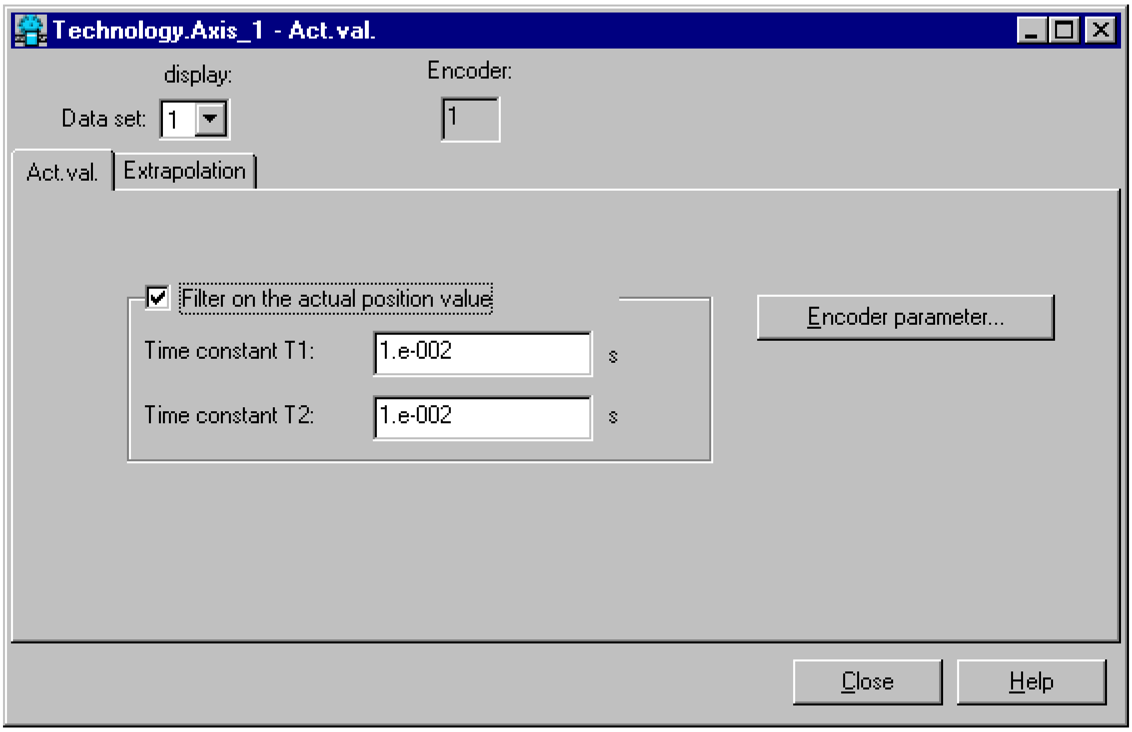
Conditions for the "Fixed end stop detection" function

- The "Move to fixed end stop" function is reset when the axis moves out of the clamping tolerance window.
- A new command can also be output to toggle the direction of torque when clamping is active.
- Non-stepped torque transitions and torque retention over a defined time period can be implemented in the user program, as can definitions of torque profiles.
- Moving to the fixed end stop (clamping) can be disabled by setting a reverse positioning command.
- A reversal command MC_MoveToEndPos is not permitted and is ignored.
- A mechanical break of the end stop is monitored by means of the actual value of the axis (clamping tolerance window monitoring).
- The torque limit of the drive is set in [N/m] at the *Torque* parameter of the MC_MoveToEndPos technology function.
- If the command is active and the fixed end stop is not detected, the system reacts as with active torque limiting.

4.7.11 actual value

4.7.11.1 Actual value - "Actual value" tab

Activate actual value filtering in the dialog **Axis / External encoder > Actual value** on the "Actual value" tab and set the appropriate time constants.



Filter on the actual position value

Activate the check box if you would like to activate the filtering of the actual position value.

Time constant T1

Here, you set time constant T1 of the PT2 position filter in the actual value system.

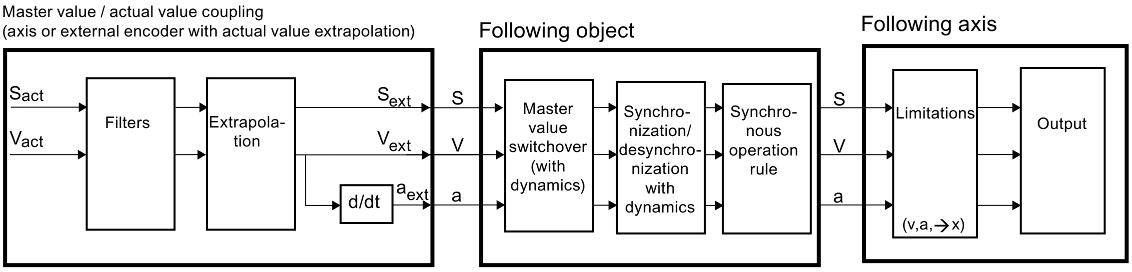
Time constant T2

Here, you set time constant T2 of the PT2 position filter in the actual value system.

"Encoder parameter" button

Opens a dialog box in which the encoder data are displayed.

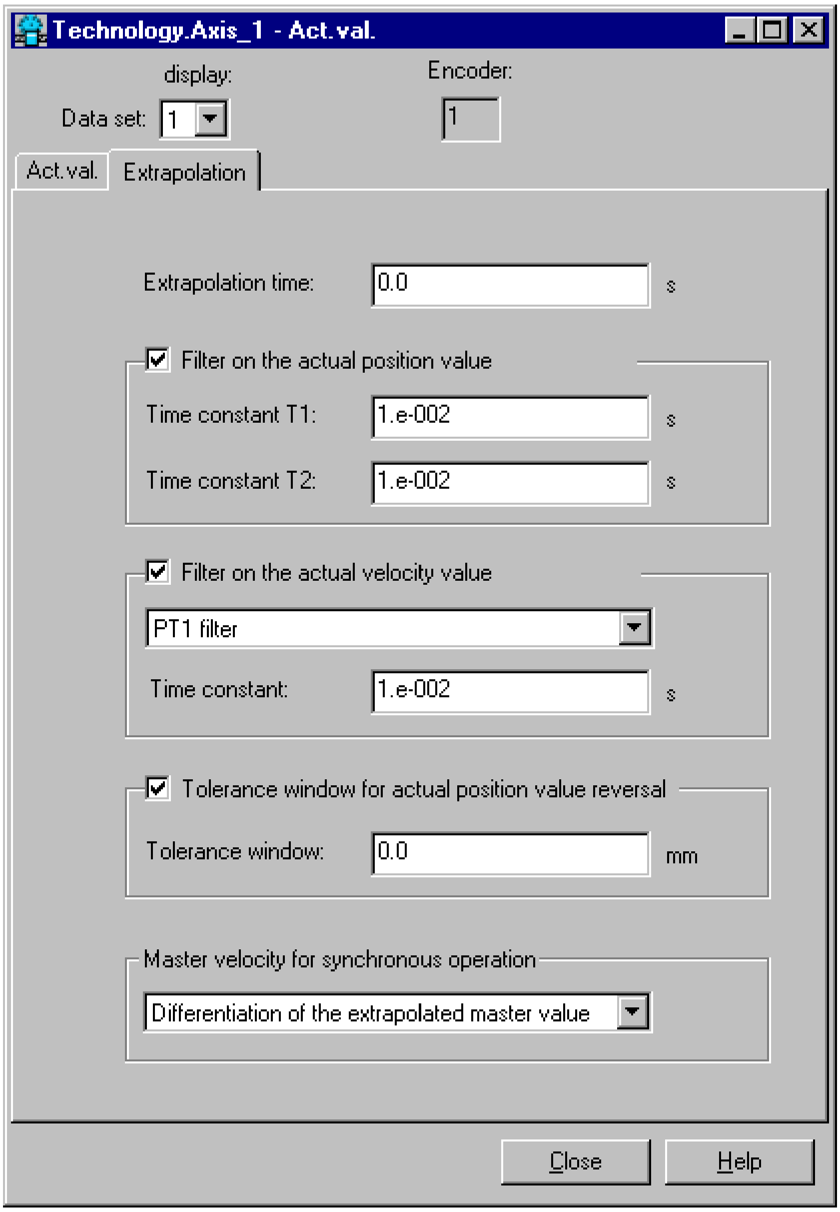
4.7.11.2 Actual value - "Extrapolation" tab



If there is a synchronous operation interconnection within a control, the synchronous operation takes into account the position, velocity, and acceleration of the master value position.

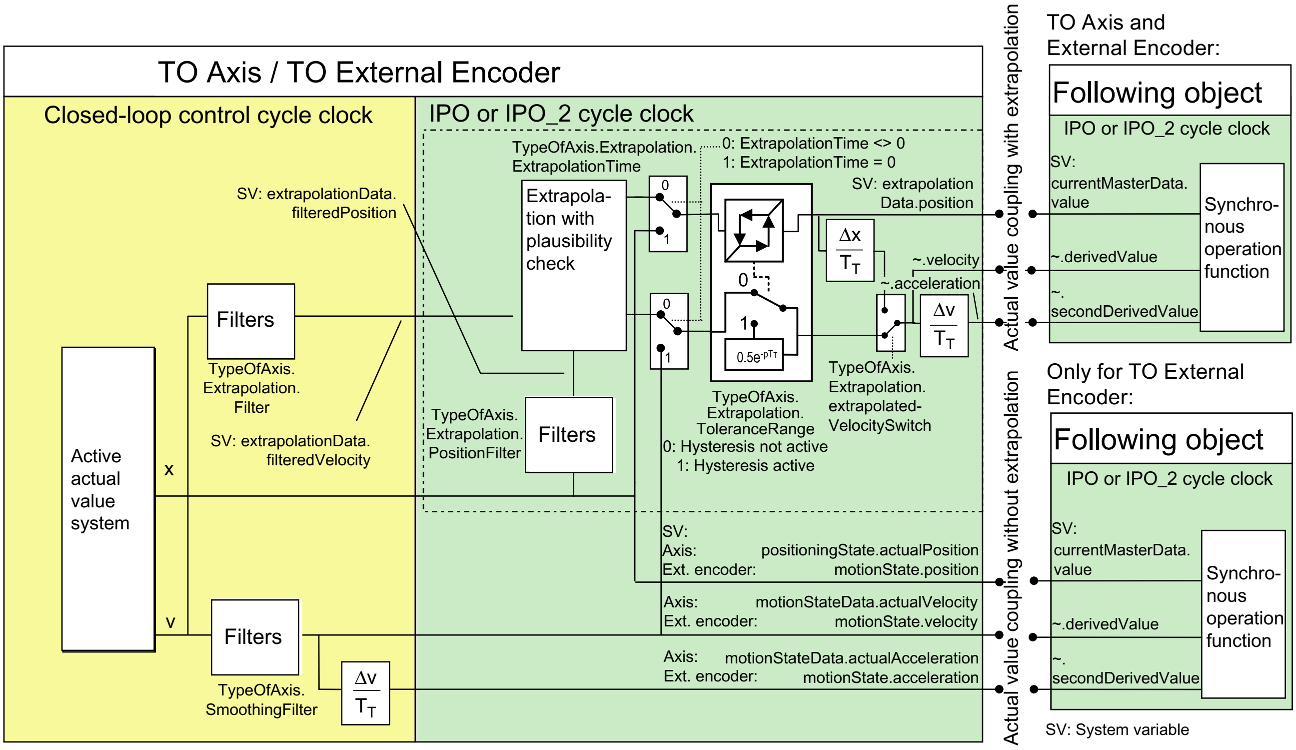
If an actual encoder value is used as the master value, it is useful to extrapolate the measured actual value for the synchronous operation in order to compensate for dead times. Dead times result within the system when measuring actual values, e.g. due to the bus communication and the system processing times.

The extrapolation is set in S7T Config on the leading axis or on the external encoder in the dialog **Axis / External encoder > Actual value** in the "Extrapolation" tab.



The parameters of this dialog can also be read from or written to the user program via technology parameters.

Actual value coupling with extrapolation (axis and external encoder)



Extrapolation time

(Parameter *1110;* configuration data *TypeOfAxis.Extrapolation.ExtrapolationTime*)

Here, you set the time for the extrapolation. No extrapolation if the value entered is *0.0*.

Filter on the actual position value

(Parameter *1130* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.enable*)

Activate the check box if you would like to extrapolate the actual position value.

Time constant T1

(Parameter *1131* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T1*)

Here, you set time constant T1 of the PT2 filter for the extrapolation in the actual value system.

Time constant T2

(Parameter *1132* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T2*)

Here, you set time constant T2 of the PT2 position filter in the actual value system.

The filter acts on the actual position for the extrapolation. The velocity for the extrapolation is taken over from the actual values of the axis or external encoder before application of the smoothing filter (*TypeOfAxis.smoothingFilter*).

Filter on the actual velocity value

(Parameter *1112* configuration data *TypeOfAxis.Extrapolation.Filter.enable*)

Activate the check box if you would like to extrapolate the actual velocity value.

Here, you select the filter for the extrapolation velocity in the drop-down list.

(Parameter *1111* configuration data *TypeOfAxis.Extrapolation.Filter.Mode*)

Time constant

(Parameter *1113* configuration data *TypeOfAxis.Extrapolation.Filter.timeConstant*)

Here, you enter the time constant for the filter.

The position is extrapolated based on the filtered or averaged velocity value. Averaging is via the "time constant".

Tolerance window for actual position value inversion

(Parameter *1114* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.enable*)

Here, you can activate the tolerance window for actual position value inversion

Tolerance window

(Parameter *1115* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.Value*)

Enter the size of the tolerance window here.

If the master value is superimposed with high-frequency noise signals that the synchronous operation cannot follow, this can cause the dynamic response limits to be exceeded or the master value to briefly change directions during synchronization.

In this case, define a tolerance window to prevent the dynamic response limits from being exceeded on the following axis or to prevent direction changes during synchronization.

Master velocity for synchronous operation

(Parameter *1116* configuration data *TypeOfAxis Extrapolation.extrapolatedVelocitySwitch*)

In the drop-down list, select whether the velocity is to be applied for the extrapolation or if the extrapolated master position value is to be differentiated.

Checking the extrapolated and filtered values

The extrapolated and filtered values can be checked in the following system variables:

- *extrapolationdata.position*
- *extrapolationdata.velocity*
- *extrapolationdata.filteredposition*
- *extrapolationdata.filteredvelocity*
- *extrapolationdata.acceleration*

Support of encoders with NIST evaluation

With encoders with NIST evaluation, the speed determined by the encoder and the resulting velocity can be accepted by the encoder. A calculation of the speed and velocity by the integrated technology is not necessary in this case. Two methods of transmission are available:

- **Transmission in the PROFIdrive message frame**

  Setting via the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_PROFIDRIVE_NIST_*

- **Transmission in the I/O area**

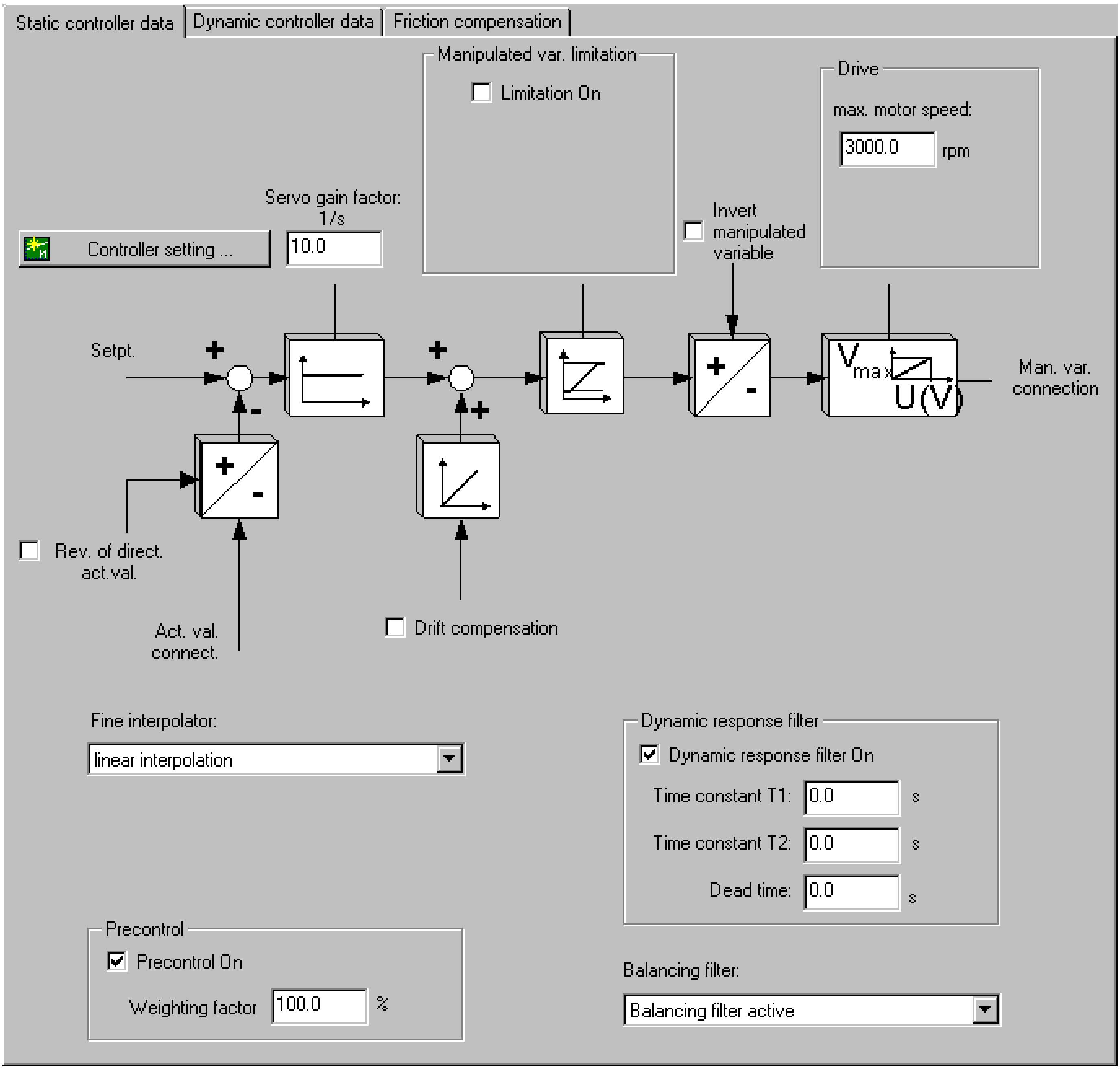
  Setting via the configuration data element

  *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_DIRECT_NIST*In this case, 4000H corresponds to 100%. The address is set in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.logAdress*, and the reference value in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.referenceValue*.

4.7.12 Control

4.7.12.1 Control - "Static controller data" tab

The "Static controller data" tab of the **Axis** > **Control** dialog can be used to configure the position control of the axes.



Manipulated variable limiting

Manipulated variable limiting represents an absolute high and low limits of the control range. This limitation is applied prior to inversion.

Note

When the Dynamic Servo Control (position controller in the drive) function is active, backlash locking (limiting of manipulated variable for the drive) is not effective. Therefore, when DSC is active, the backstop must be generated in the drive.

Drive

Use this input box to enter the maximum speed of the drive.

"Controller setting" button

This button can be used to implement a simple controller optimization without precontrol and balancing filters with SINAMICS drives.

Follow the instructions in the "Optimizing the position controller - overview (Page 853)" chapter if you want to achieve an implemented position control system.

Controller parameters

For further details on the individual controller parameters and their optimal setting, please refer to the chapter "Optimizing the position controller - overview (Page 853)".

Fine Interpolator

The fine interpolator function is used is to generate interim setpoints when the interpolator and controller have a different duty factor. Optional interpolation mode settings at the fine interpolator:

- No interpolation
- Linear interpolation (constant position for positioning axis)
- Interpolation with constant acceleration (constant acceleration at positioning axis)
- Interpolation with constant velocity (constant velocity at positioning axis)

When set for positioning axes, the position setpoint is interpolated.

When set as speed-controlled axis, the velocity setpoint is interpolated.

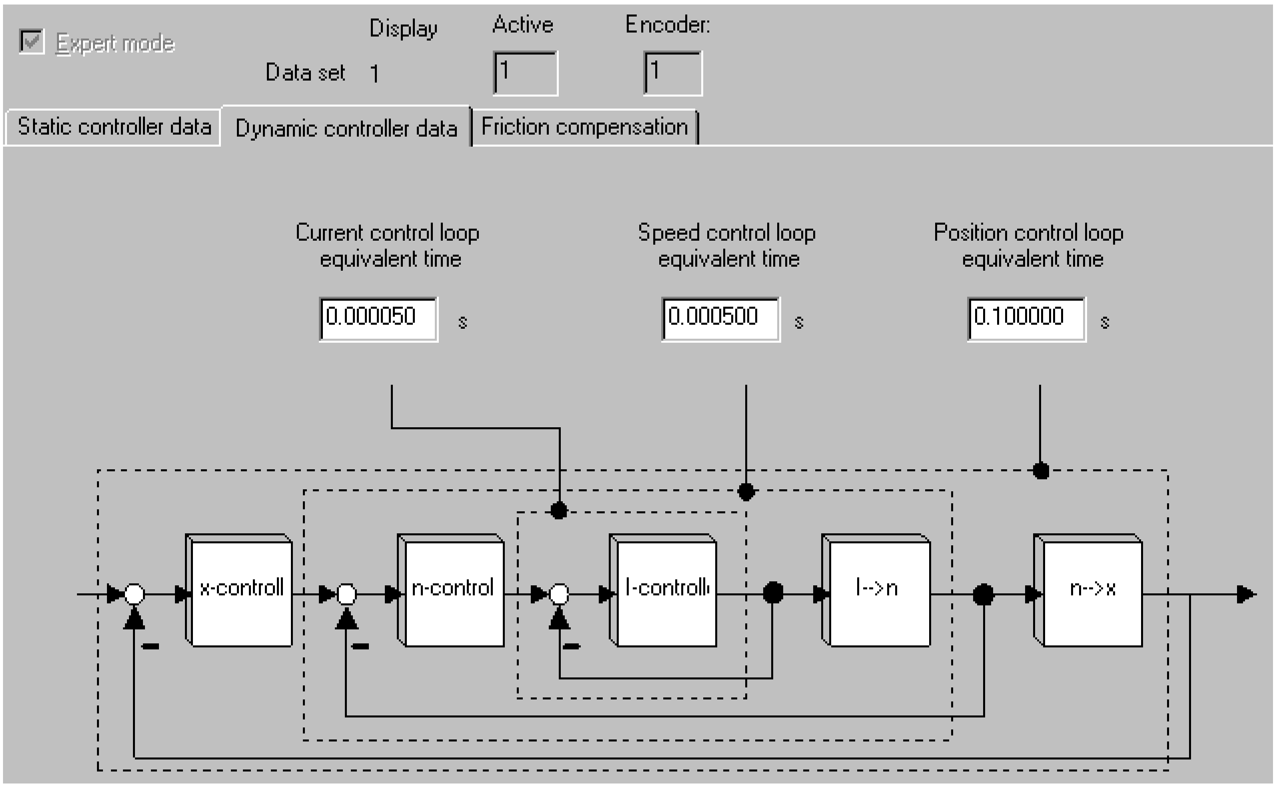
Dynamic filter, precontrol, balancing filter

For further details on the settings please refer to the chapter "Optimizing the position controller - overview (Page 853)".

4.7.12.2 Control - "Dynamic controller data" tab

The "Dynamic controller data" tab of the **Axis > Control** can be used to adjust the dynamic response of the axes, for example, to match their response in synchronous operation. The "Dynamic controller data" tab is available when you set expert mode in the "Static controller data" tab.

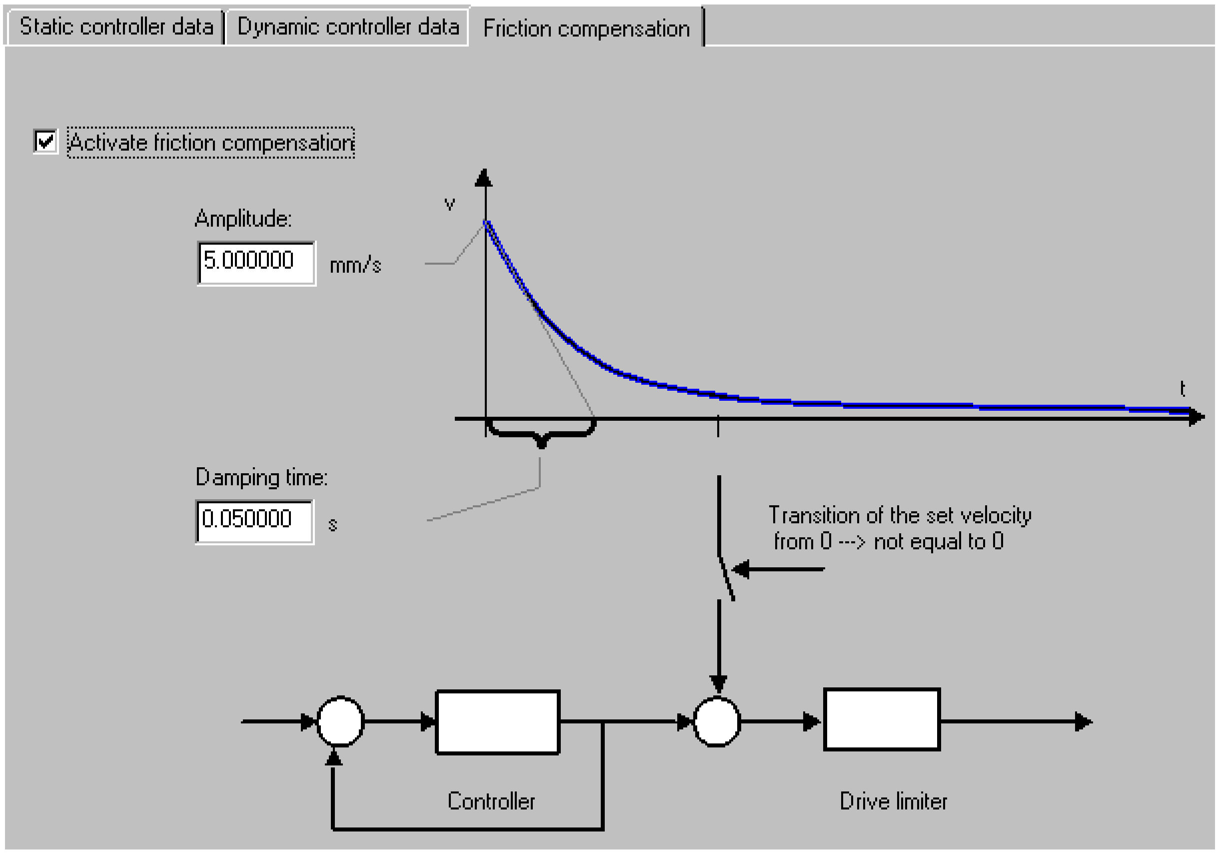
The setpoint branch of the control loop contains a configurable dynamic filter which you can use to adjust the dynamic response of the axes.



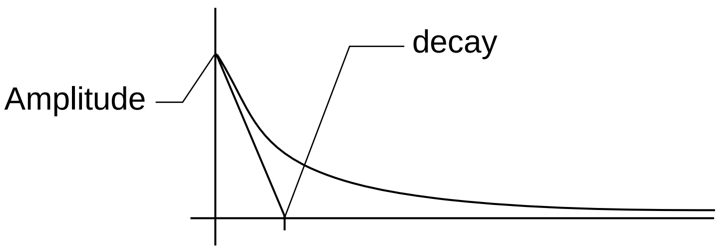
- **Equivalent time current control loop**
  The equivalent time current control loop is not used in this version.
- **Equivalent time speed control loop**
  The "Equivalent time speed control loop" parameter can be used to set time constant vTc (velocity Time constant) of the balancing filter.
- **Equivalent time position control loop**
  The equivalent time of the position control loop is required to toggle from speed-controlled mode to position-controlled mode and to start the emergency stop ramp.

4.7.12.3 Control - "Friction compensation" tab

The "Friction compensation" tab of the **Axis > Control** tab is used to enable friction compensation. The "Friction compensation" tab is available when expert mode is set on the "Static controller data" tab.



The system provides a simple method compensating for the forces of static friction. During startup from a standstill, a DT1 element adds a static friction compensation signal to the manipulated variable.



The friction compensation is added relative to the velocity setpoint. It is only active when motion commands are executed.

The standstill identification for static friction compensation can be set separately, as is the case for the amplitude and the decay response. The amplitude and decay response are set in the configuration.

4.7.13 Homing

4.7.13.1 Introduction

Position-controlled axes equipped with incremental motor measuring systems must be referenced to the position of the mechanical system of the machine each time power is switched on. The axis is synchronized during homing, based on the activation of a certain position value at a defined position of the axis mechanism.

Axes can be homed in:

- Active mode (reference point approach)
- Passive mode (flying referencing)
- Direct mode (with position setpoint)

Detailed settings depend on the measuring systems available for measuring the reference point and on the motion an axis may perform for homing.

4.7.13.2 Homing - "Active homing" tab

Active homing

Active homing can be configured in the "Active homing" tab of the **Axis > Homing** dialog box in S7T Config.

Three homing modes are available for "Active homing":

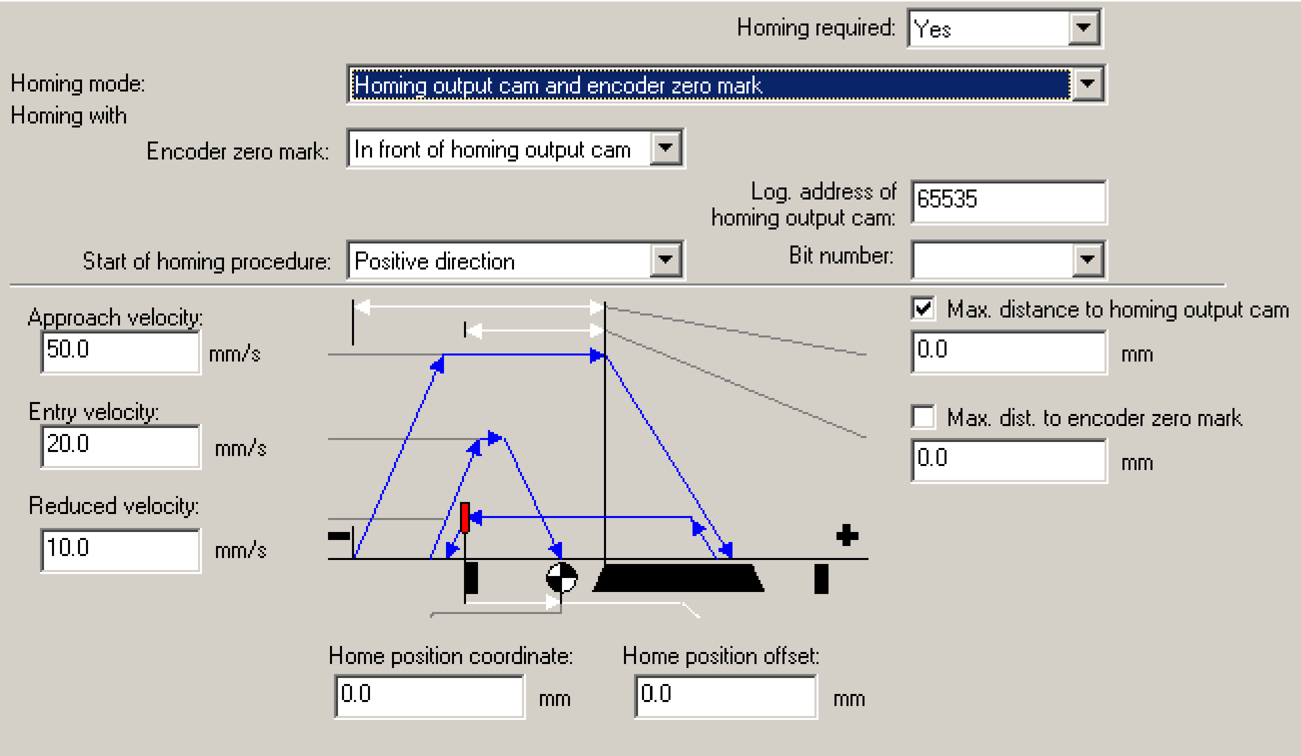
- Reference cam and encoder zero mark
- Encoder zero mark only
- External zero mark only

Start of the homing function:

MC_Home	*Mode = 0*	Determination of the reference point based on the axis configuration
	Mode = 1 *Position = x*	Determination of the reference point based on the axis configuration The reference point is assigned the value of the *Position* input parameter.

After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone* = *TRUE*.

Active homing mode with "reference cam and encoder zero mark"



After homing is started with the "MC_Home" technology function, the axis moves to the reference cam and then approaches the referencing encoder zero mark, according to the axis configuration. The lower section of the diagram shows the resultant motion sequence.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case "Homing output cam and encoder zero mark"
Encoder zero mark	Before reference cam	The axis is homed to the encoder zero mark which lies before the reference cam with reference to the direction of reference point approach.
	After reference cam	The axis is homed to the encoder zero mark which lies after the reference cam with reference to the direction of the reference point approach.
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Logical address of reference cam	[byte address]	Logical byte address of the reference cam You can connect the reference cam to the integrated inputs of the Technology CPU, or to the IO on DP(DRIVE).
Bit number	[Number of the bit]	Bit address of the signal used for the reference cam
Approach velocity	-	Velocity at which the axis approaches the reference cam
Entry velocity	-	Velocity at which the axis approaches the (shifted) reference point after detection of the encoder zero mark
Shutdown velocity	-	Velocity at which the axis approaches the encoder zero mark after detection of the reference cam.
Reference point coordinate (Reference cam and encoder zero mark, External zero mark only, Encoder zero mark only)	-	Here, you enter the actual position value of the home position (relative to the coordinate system of the axis)
Homing position offset	-	The "Home position offset" shifts the reference point by a configured distance. The axis travels along this configured distance at a velocity defined in "Entry velocity", starting at the synchronization position with encoder zero mark. Modulo axes always take the shortest distance.
Maximum distance to homing output cam	deactivated	The distance to reference cam detection is not monitored
	Activated	Monitoring of the distance between the start of reference point approach and detection of the reference cam. If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance an axis travels between the reference cam and the encoder zero mark. If the difference in this distance exceeds the configured distance, the corresponding axis technology DB indicates error *801D*. The reference point approach is canceled.

Sequence of the reference point approach

- Phase 1: Reference cam approach

  The axis starts its reference point approach at the configured approach velocity and towards the direction set in "Start of reference point approach". The reference cam approach ends after the reference cam (Phase 1) is reached.

  You can also monitor the distance an axis travels between the start of reference point approach and detection of the reference cam by setting the "Max. distance to homing output cam" check box. The reference point approach is canceled with error if the reference cam is not detected between the start and end of the configured distance.

- Phase 2: Synchronization with encoder zero mark

  After having reached the reference cam, the axis accelerates / decelerates to shutdown velocity, and approaches the encoder zero mark. The encoder zero mark is derived from the combined settings of "Encoder zero mark" (after or before the reference cam) and "Start of reference point approach" (positive or negative direction).

  After the reference cam is detected, the PLC synchronizes the axis to the first encoder zero mark detected in accordance with the configuration. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

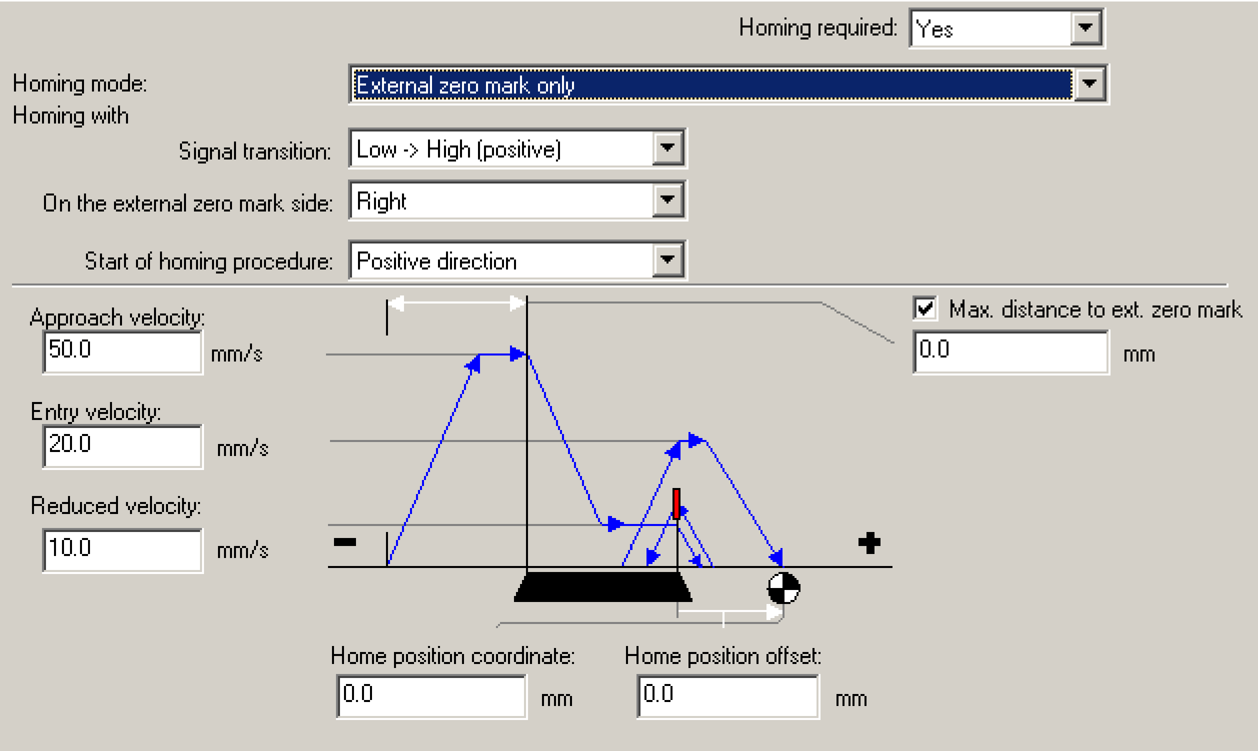
  You can also monitor the distance an axis travels between the reference cam and the encoder zero mark by setting the "Max. distance to encoder zero mark" check box. The reference point approach is canceled with error if the encoder zero mark is not found within the specified distance after the reference cam is detected.

- Phase 3: Reference point approach

  After the encoder zero mark is detected, the axis accelerates / decelerates to approach velocity to approach the reference point.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the encoder zero mark is detected, provided that the reference point lies within the deceleration ramp.

Active homing mode with "External zero mark only"



If homing is started by calling the "MC_Home" technology function the axis approaches the referencing external zero mark in accordance with the configuration. The lower section of the diagram shows the resultant motion sequence.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "External zero mark only"
Signal transition	Low -> High (positive)	The motion is referenced to the positive edge of the external zero mark (setting according to edge evaluation in the drive component).
	High -> Low (negative)	The motion is referenced to the negative edge of the external zero mark (setting according to edge evaluation in the drive component).
On the side of the external zero mark	left	The signal transition is measured viewed from the left side of the external zero mark, in the selected direction of reference point approach.
	right	The signal transition is measured viewed from the right side of the external zero mark, in the selected direction of reference point approach.
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Approach velocity	-	Velocity at which the axis approaches the external zero mark
Entry velocity	-	Velocity at which the axis approaches the (shifted) homing position after detection of the external zero mark
Shutdown velocity	-	Velocity at which the axis approaches the reference point coordinate after detection of the external zero mark
Homing position offset	-	The "Home position offset" shifts the reference point by a configured distance. The axis moves by a configured distance at the "Homing velocity", after its synchronization at the external zero mark edge. Modulo axes always take the shortest distance.
Max. distance to external zero mark	deactivated	The distance to external zero mark detection is not monitored.
	Activated	Monitoring of the distance between the start of the reference point approach and detection of the external zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.

Note

For homing with "external zero mark", connect the external zero mark as digital measuring input to the drive component.

In order to execute a reference point approach in "external zero mark" homing mode, select "Signal transition" and "on the side of the external zero mark" values which correspond with the drive's configuration or functionality. Instead of being used to configure the measuring function in the drive, the "Signal transition" and "on external zero mark side" parameters merely reflect their functionality in order to control the axis motion according to the configuration.

For information on the configuration of external zero mark detection, refer to the relevant product information or to the drive manuals.

Sequence of the reference point approach

- Phase 1: Synchronization to external zero mark

  The axis starts its reference point approach at the configured approach velocity and towards the direction set in "Start of reference point approach". Synchronization ends at the external zero mark (phase 1) when the configured signal transition (configured at the drive component) is detected at external zero mark. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

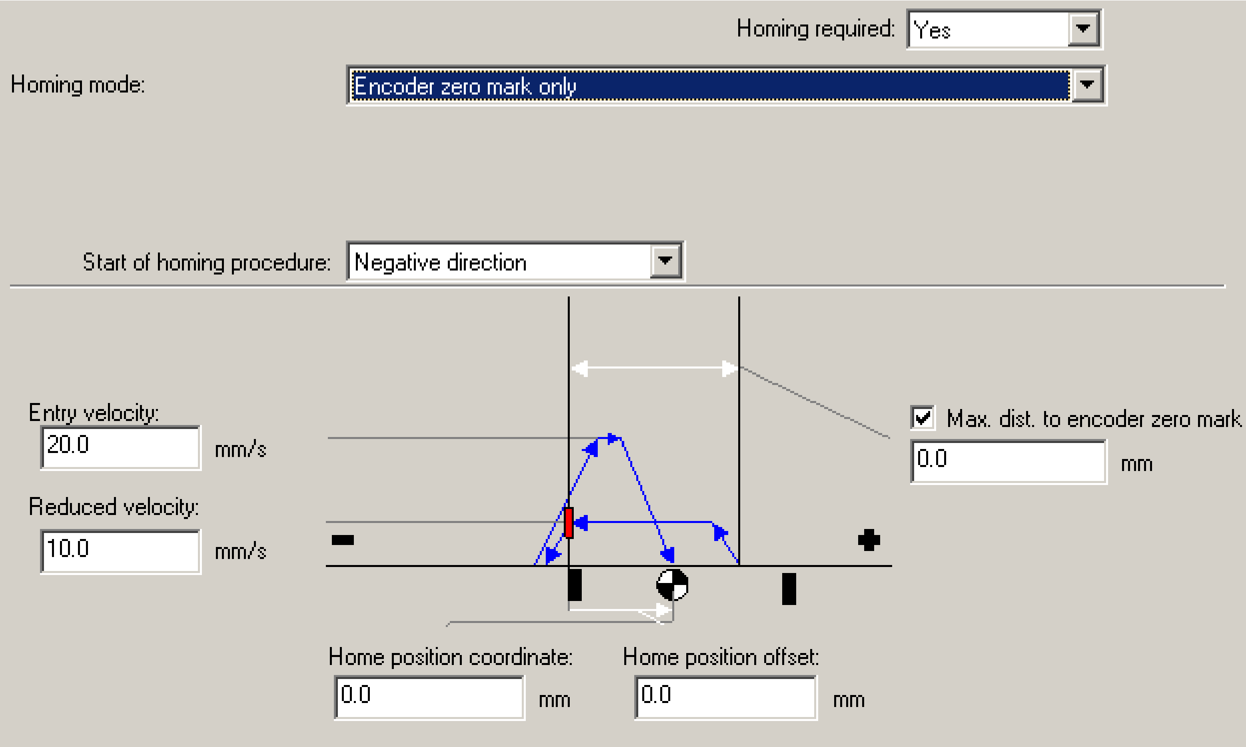
  You can also monitor the distance an axis travels between the start of reference point approach and detection of the signal edge by setting the "Max. distance to external zero mark" check box. Homing is canceled with error if the edge is not detected between the start of reference point approach and the end of the configured distance.

- Phase 2: Reference point approach

  After the configured signal edge is detected, the axis accelerates / decelerates to approach the reference point coordinate at shutdown velocity.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the external zero mark is detected, provided that the reference point lies within the deceleration ramp.

Active homing mode with "Encoder zero mark only"



Homing to encoder zero mark is used, for example, at axes for which the encoder can only set one encoder zero mark in the entire traversing range of the axis. The homing command initiates axis approach to the encoder zero mark. After the encoder zero mark is detected, the axis approaches the shifted reference point at homing velocity. The axis position now has the value defined in the reference point coordinate. The graphic in the figure shows the resultant motion sequence. Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Encoder zero mark only"
Start reference point approach	Positive direction	Reference point approach in positive direction.
	Negative direction	Reference point approach in negative direction.
Entry velocity	-	Velocity at which the axis approaches the (shifted) reference point after detection of the encoder zero mark
Shutdown velocity	-	Velocity at which the axis starts the reference point approach and approaches the encoder zero mark
Homing position offset	-	The homing position offset function shifts the homing position by a configured distance, meaning that the axis moves along a configured distance behind the encoder zero mark detection position at a "Homing velocity". Modulo axes always take the shortest distance.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance between the start of the reference point approach and detection of the encoder zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The reference point approach is canceled.

Sequence of the reference point approach

- Phase 1: Synchronization to encoder zero mark

  The axis starts its reference point approach towards the direction set in "Start of reference point approach" at the configured shutdown velocity. Synchronization with encoder zero mark (phase 1) ends with the detection of the encoder zero mark. The axis position is set to the default value minus the reference point shift defined in "Reference point coordinate" (*Mode = 0*) or at input parameter *Position* (*Mode = 1*).

  You can also monitor the distance an axis travels between the start of reference point approach and encoder zero mark detection by setting the "Max. distance to zero mark" check box. Homing is canceled with error if the encoder zero mark is not detected between the start of reference point approach and the end of the configured distance.

- Phase 2: Reference point approach

  After the encoder zero mark is detected, the axis accelerates / decelerates to approach the reference point coordinate at entry velocity.

  If a reference point shift was configured, the axis approaches this reference point by the corresponding distance starting at the synchronization position. The direction of motion is determined by the sign of the reference point shift and by the length of the deceleration ramp after the encoder zero mark is detected, provided that the reference point lies within the deceleration ramp.

4.7.13.3 Homing - "Passive homing" tab

Passive homing can be configured in the "Passive homing" tab of the Axis > Homing dialog box of S7T Config. The difference compared to active homing is that the required homing motion is not initiated by a homing command.

Three homing modes are available for "Passive homing":

- Reference cam and encoder zero mark
- External zero mark only
- Encoder zero mark only

Start of the homing function:

MC_Home	*Mode = 2* *Position = x*	The current position is assigned the value of the *Position* input parameter at the reference point.

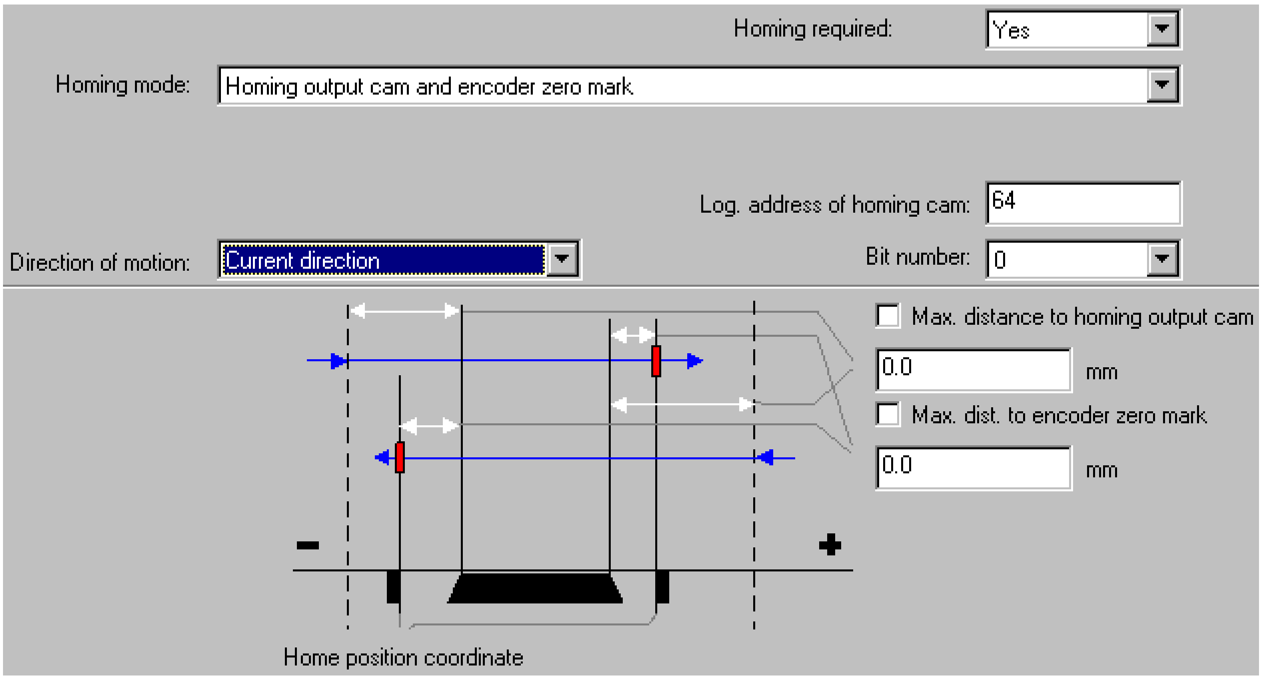
After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone= TRUE*.

Note

Note that not all drive components support all homing mode or measured signal evaluation functions. For details, refer to the documentation of the drive components used.

Triggering of encoder zero marks or reference cam monitoring during passive homing operations is indicated by a corresponding error message at the technology DB. The current axis motion is terminated in this case.

Passive homing mode with "Reference cam and encoder zero mark"

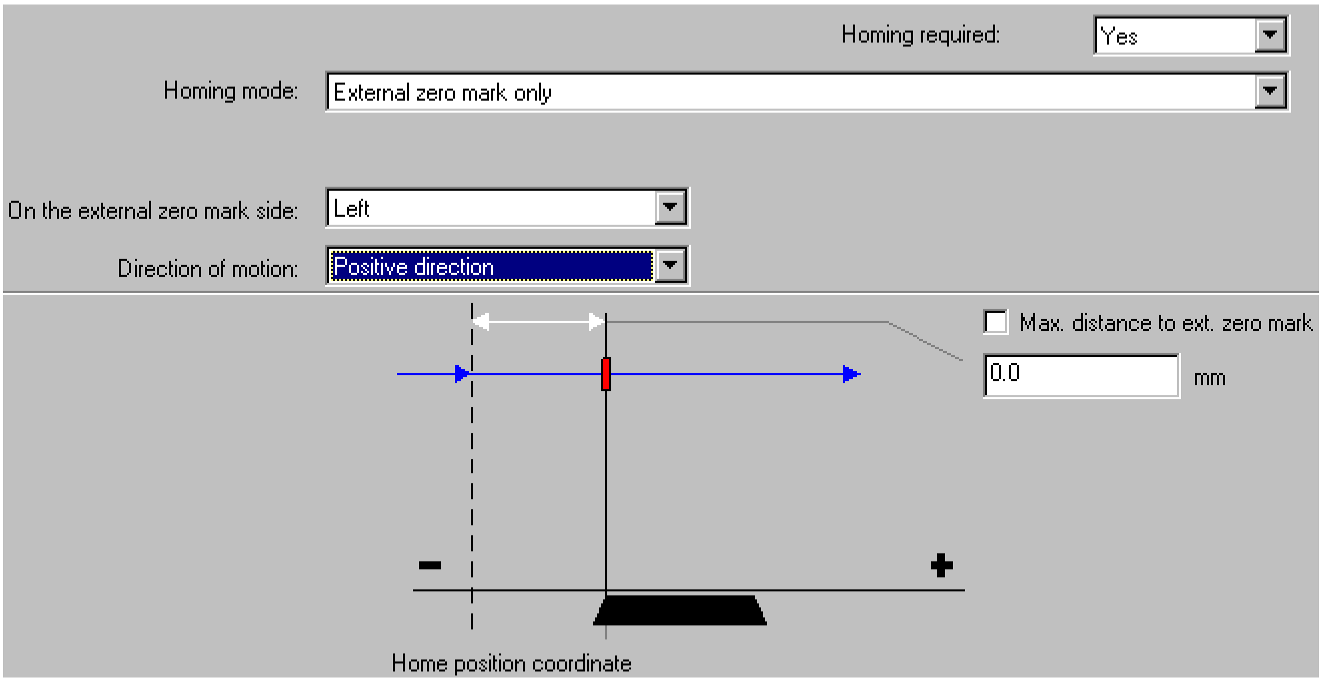


After the axis has passed the reference cam, the next encoder zero mark triggers axis homing. The axis position is set in the reference point to the value defined at the *Position* input parameter of the "MC_Home" technology function.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Reference cam and encoder zero mark"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.
Logical address of reference cam	[byte address]	Logical byte address of the reference cam You can connect the reference cam to the integrated inputs of the Technology CPU, or to the IO on DP(DRIVE).
Bit number	[Number of the bit]	Bit address of the signal used for the reference cam
Maximum distance to homing output cam	deactivated	The distance to reference cam detection is not monitored
	Activated	Monitoring of the distance between the start of the homing function and detection of the reference cam. If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance an axis travels between the reference cam and the encoder zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.

Passive homing mode with "External zero mark only"



Axis homing starts with the detection of the external zero mark. The axis is set to the value of the *Position* input parameter of the "MC_Home" technology function.

Meaning of the individual parameters:

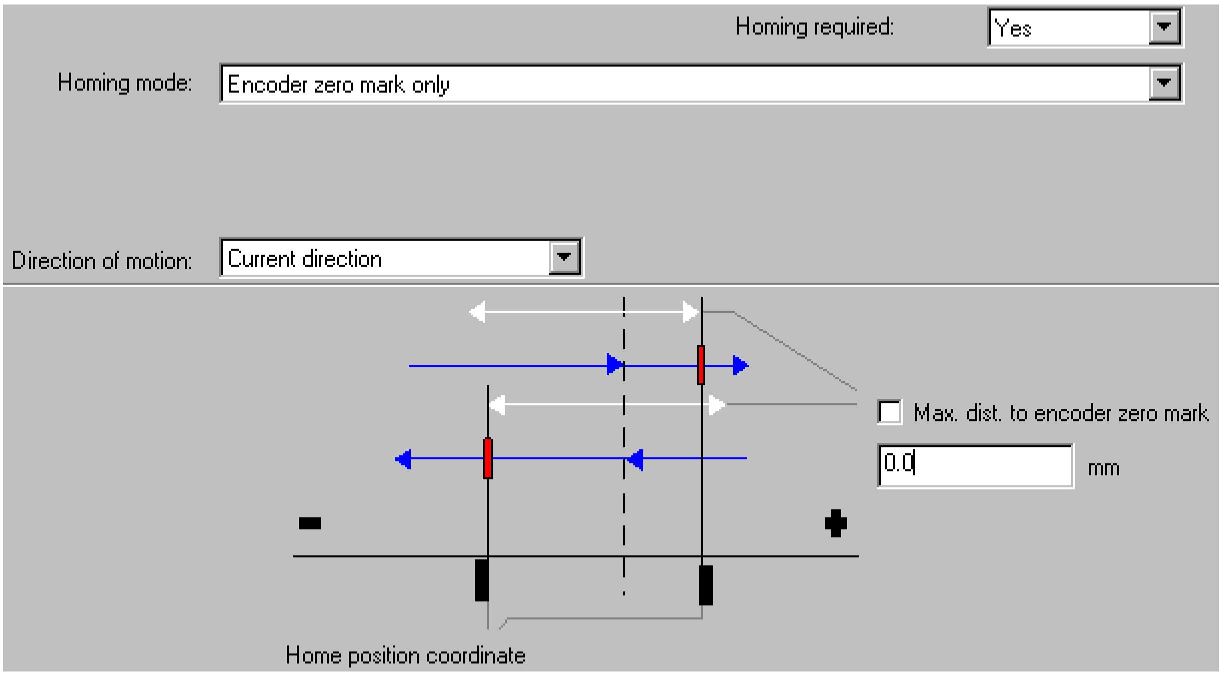
Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "External zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the external zero mark.
	Negative direction	The axis is only homed with negative approach to the external zero mark.
	Current direction	The axis is homed when it reaches the next external zero mark.
On the side of the external zero mark	left	Direction of movement: Positive direction Axis homing is triggered at the positive edge. Direction of movement: Negative direction Axis homing is triggered at the negative edge.
	right	Direction of movement: Positive direction Axis homing is triggered at the negative edge. Direction of movement: Negative direction Axis homing is triggered at the positive edge.

Parameters	Value	Explanation
Max. distance to external zero mark	deactivated	Distance to go to the external zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the external zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Note

- For homing with "external zero mark", connect the external zero mark as digital measuring input to the drive component.
- In order to be able to execute the reference point approach in "external zero mark" homing mode as required, you should always set values at "Direction of movement" and "on the side of the external zero mark" which are compatible with drive configuration or functionality. The "on the side of the external zero mark" and "Direction of movement" parameters are not used to configure the measuring function in the drive and only reflect their functionality.
- For information on the configuration of external zero mark detection, refer to the relevant product information or to the drive manuals.

Passive homing mode with "Encoder zero mark only"



Homing mode with "Zero mark only" is used for axes, for example, which are equipped with an encoder that outputs only one zero mark signal within the entire traversing range of the axis

Axis homing starts with the detection of the encoder zero mark. The position value of the axis is set to the value of the *Position* input parameter of the "MC_Home" technology function after the encoder zero mark is detected.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Encoder zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.

Parameters	Value	Explanation
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the encoder zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Passive homing mode with "Default"

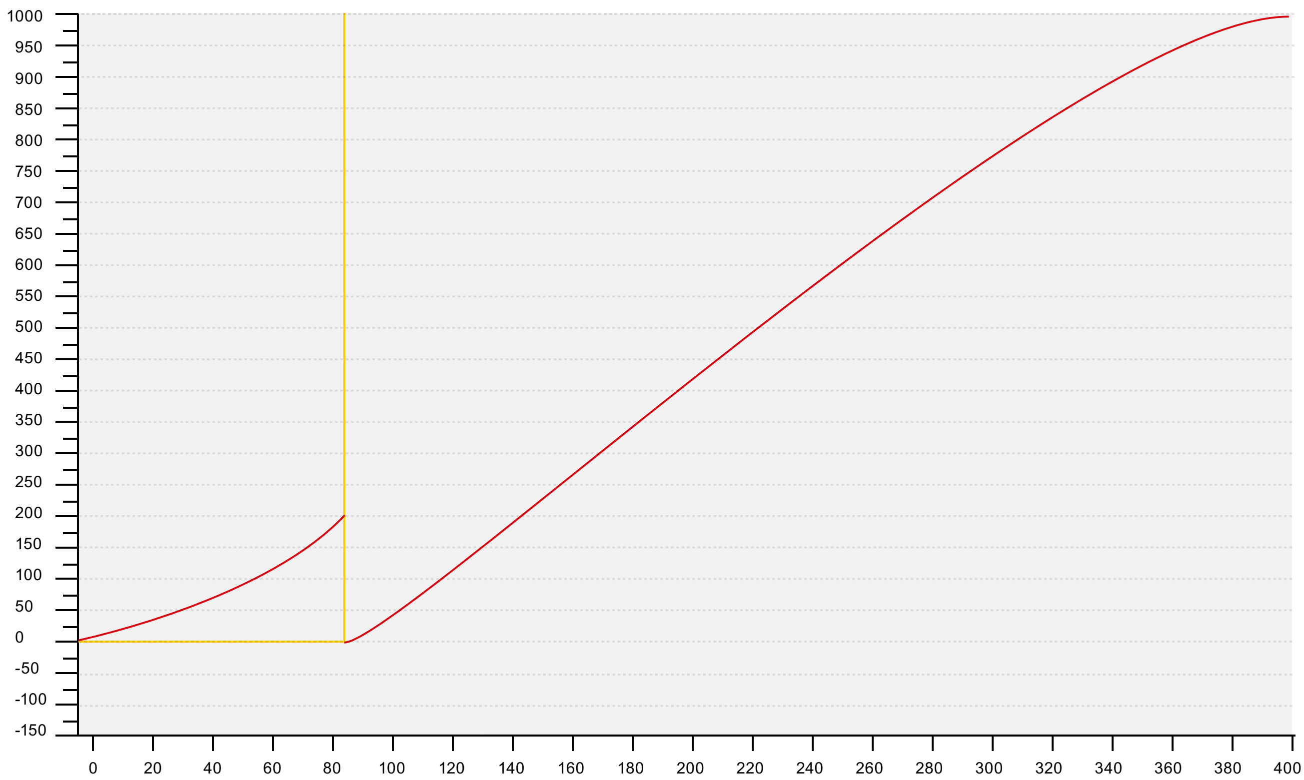
When a new axis is created in S7T Config, the *Default* homing mode is preset.

Homing mode with *encoder zero mark only* is used if the configured encoder provides a zero mark. Homing mode external zero mark only is used if the encoder does not provide a zero mark.

4.7.13.4 Positioning behavior with passive homing

The following examples show the response of the end position with passive homing with the motion commands "MC_MoveRelative" and "MC_MoveAbsolute":

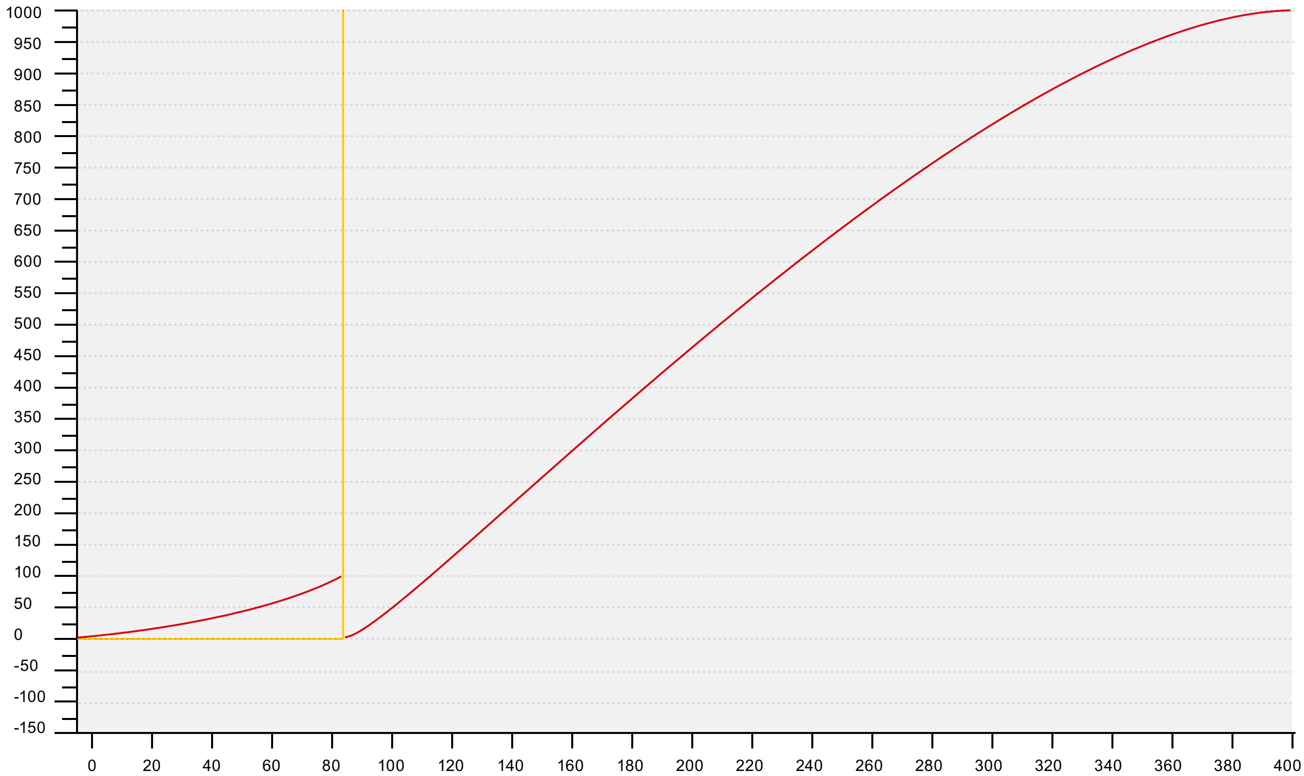
Positioning behavior with passive homing with "MC_MoveRelative"



An MC_MoveRelative command *1000* is started for passive homing.

1. The MC_MoveRelative command and passive homing start at the same time.
2. When the yellow marked homing position *200* is reached, the axis is homed and the position of the axis is set to position *0*.
3. The axis moves by the remaining relative distance to position *800 (1000-200)*.

Positioning behavior with passive homing with "MC_MoveAbsolute"



An MC_MoveAbsolute to position *1000* command is started for passive homing.

1. The MC_MoveAbsolute command and passive homing start at the same time.
2. When the yellow marked homing position *200* is reached, the axis is homed and the position of the axis is set to position *0*.
3. The axis moves to position *1000* in accordance with the new homing point.

4.7.13.5 Direct homing

The actual axis position is set to the value defined at the MC_Home technology function. No reference point shift settings are active. The function does not execute a motion. The axis is homed when the command is executed.

Start of the homing function

MC_Home	*Mode = 3* *Position = x*	Direct homing: The current position is assigned the value of the *Position* input parameter.

After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone= TRUE* .

4.7.13.6 Position correction

A correction value is deducted from the actual position value of the axis. By contrast to other homing modes, the axis maintains its homed state (homed / not homed) in this case.

The position correction function can also be used to manipulate the setpoints of the various coordinate systems (base coordinate system, superimposing coordinate system). This is of significance for superimposing camming in order to be able to generate a reference within a cam disk.

Start of the homing function:

MC_Home	*Mode = 4* *Position = x*	Actual value correction: Position value = (current position) - (parameter position).
	Mode = 6 *Position = x*	Setpoint correction in the base coordinate system: Position value = (current position) - (parameter position).
	Mode = 7 *Position = x*	Setpoint correction in the superimposing coordinate system: Position value = (current position) - (parameter position).

Position correction does not influence the *Statusword.HomingDone* status in the axis technology DB.

4.7.13.7 Motions with non-homed axes

You determine whether absolute positioning is to be available for a non-homed axis in the **Axis > Homing** dialog box.

Optional settings for "Homing required":

- No: Relative and absolute motions are possible. The Software limit switches (Page 149) are monitored.
- Yes: Relative motion only. The software limit switches are not monitored as long as the axis is not homed.

4.7.14 Monitoring functions

The slave values calculated by the synchronization object and the compensation functions on the following axis are monitored in terms of their dynamic activation at the following axis. If a dynamic activation is not possible, corresponding adjusting movements are generated.

The currently set high limits of velocity and acceleration (including jerk) are active at the axis.

If a motion can not be carried out, an attempt is made to follow the values as closely as possible by initiating a compensation motion which is determined by the maximum dynamic values of the axis. The result is a setpoint error.

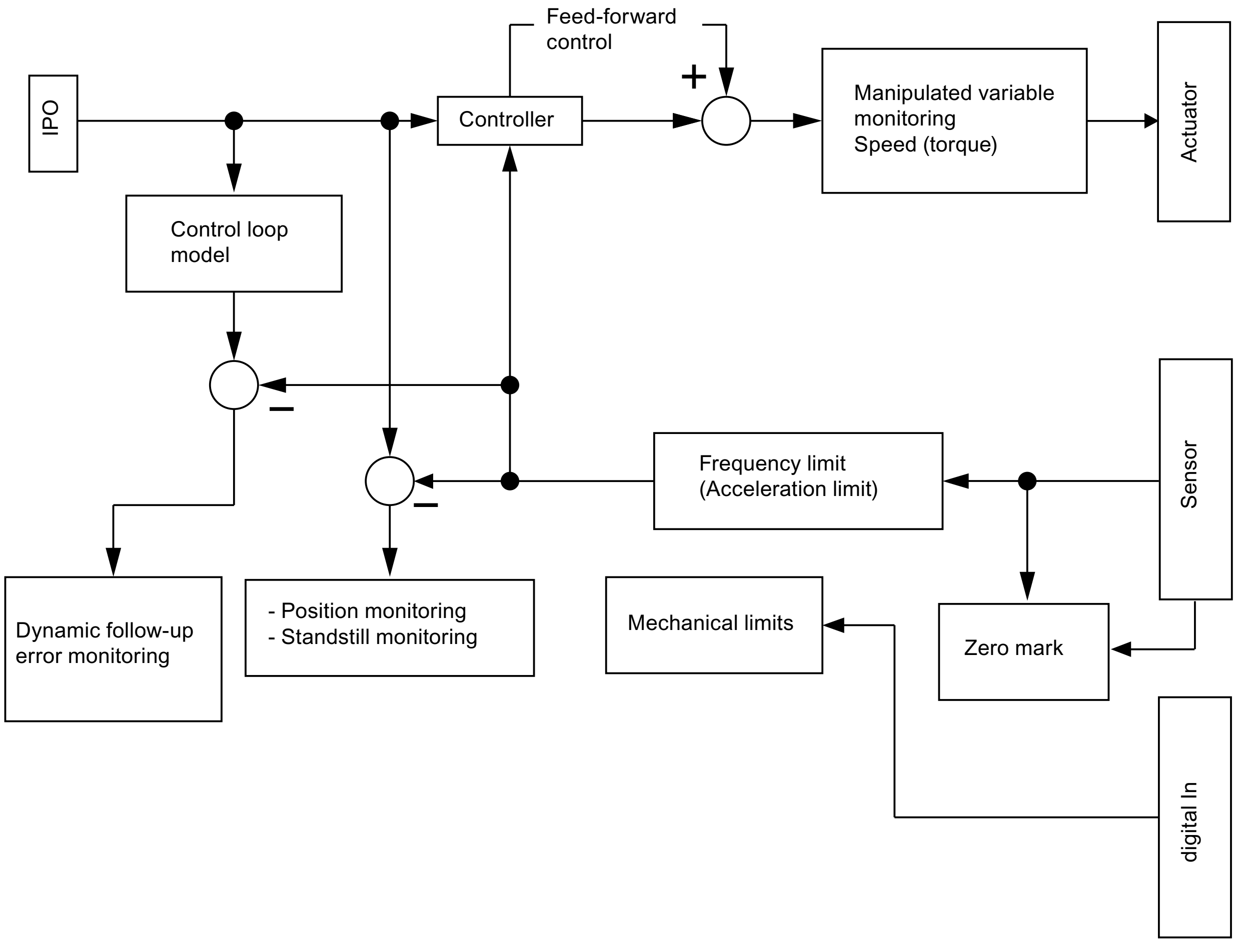
4.7.14.1 Monitoring functions - Overview

Axis monitoring functions you can configure in S7T Config:

Monitoring functions	Speed-controlled axis	Positioning axis	Synchronization axis
Velocity error monitoring (Page 187)	X	-	-
Positioning monitoring (Page 184)	-	X	X
Following error monitoring (Page 185)	-	X	X
Standstill signal (Page 186)	X	X	X
Software limit switch (Page 149)	-	X	X
Hardware limit switch (Page 147)	X	X	X
Synchronization monitoring (Page 321)	-	-	X
Manipulated variable monitoring (Page 187) (always active)	X	X	X
Encoder limit frequency monitoring (Page 919)	X (only with encoder)	X	X

Positioning axis monitoring

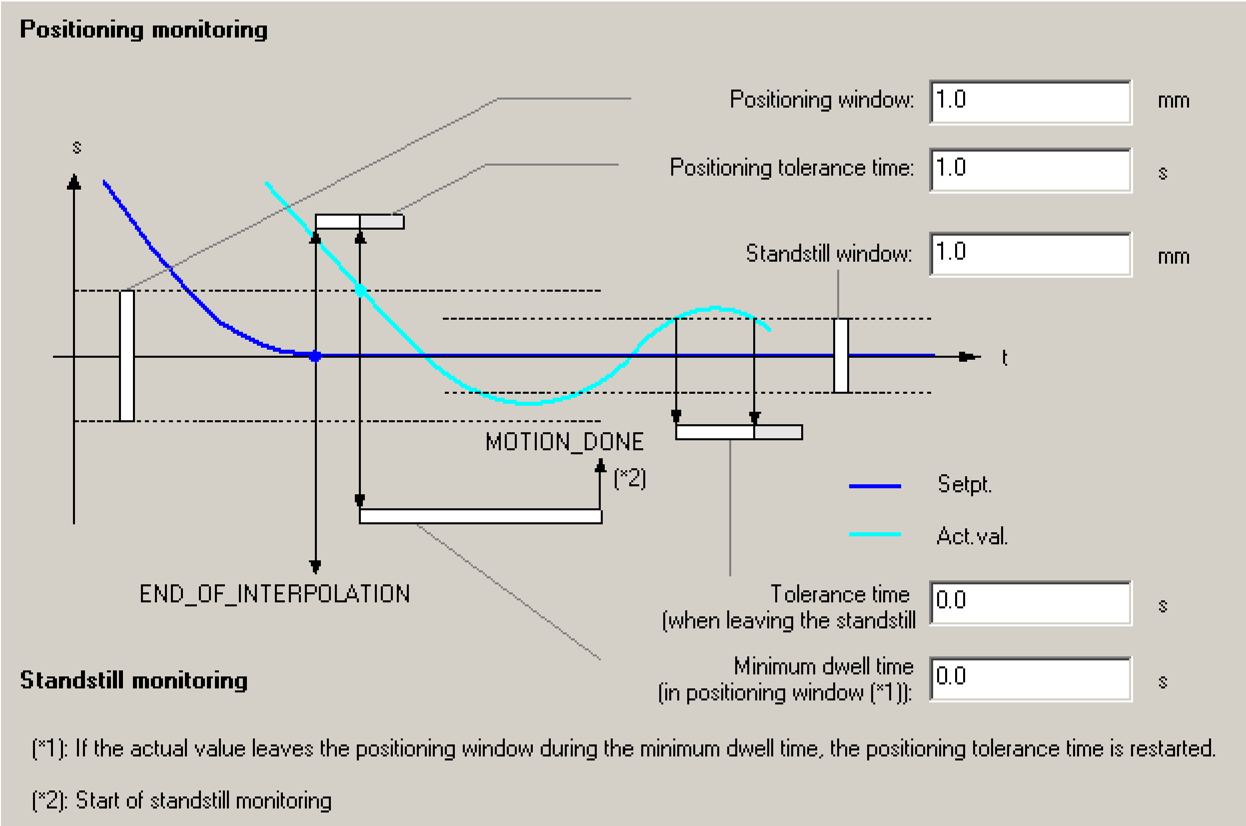
The figure below shows an example of positioning axis monitoring:



Errors are reported in the *ErrorStatus* parameter of the technology DB.

4.7.14.2 Monitoring functions - Positioning and standstill monitoring

Set the limit values for monitoring the positioning of position-controlled and synchronization axes in the "Positioning and standstill monitoring" tab of the **Axis > Monitoring functions** dialog box.



The "Positioning and standstill monitoring" tab shows whether the **Positioning** or **Synchronous operation** technology is set for the axis.

Positioning monitoring

At the end of a positioning motion, the function monitors the axis entry to the target position. For this purpose, a positioning window and a time tolerance within which the end position must be reached are specified. Monitoring is activated at the end of setpoint interpolation.

You can also set a minimum dwell time in the positioning window to expire before the positive feedback of the positioning command is activated. This time can be used for oscillating processes and control loops, for which the tolerance window should be less than the overshoot amplitude.

A positive feedback for the positioning command is output when the actual value reaches the positioning window.

The axis technology DB returns error *8019* if the positioning monitoring function is triggered.

Standstill (zero-speed) monitoring

Standstill monitoring is enabled when the position setpoint of a positioning command equals the value of the target position, and the delay of standstill monitoring activation has expired.

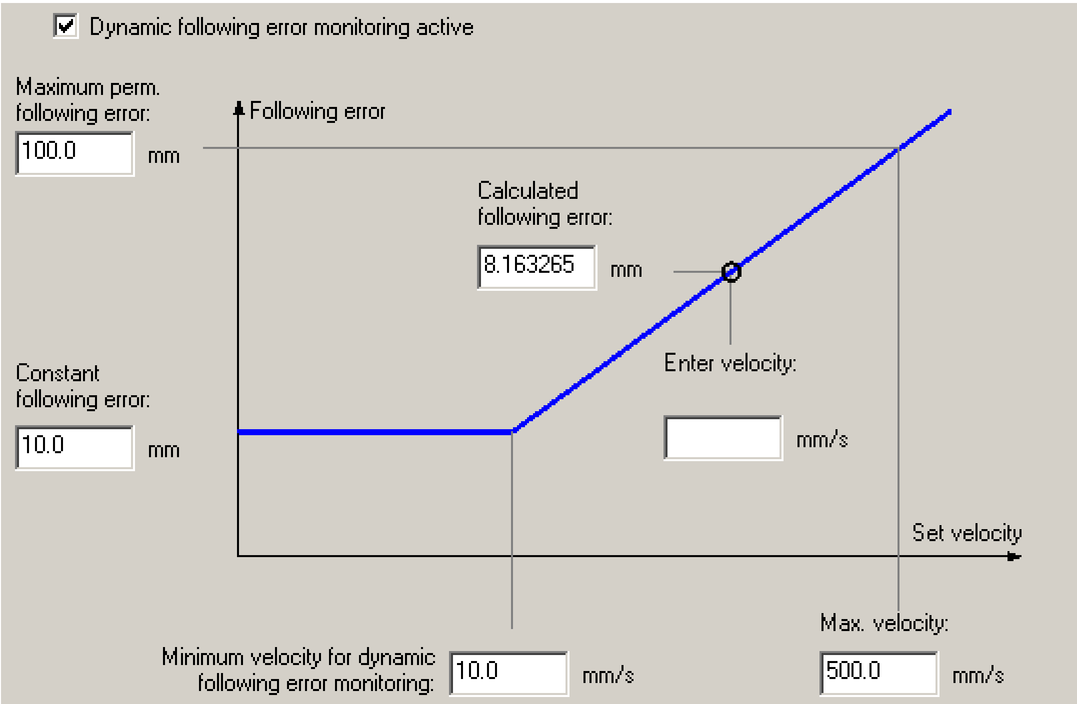
Standstill monitoring is triggered if the axis moves out of the configured standstill tolerance window for any time longer than the configured period. The axis technology DB returns error *8018* if standstill monitoring is triggered.

4.7.14.3 Monitoring functions - "Following error monitoring" tab

Dynamic following error monitoring can be enabled in the "Following error monitoring" tab of the **Axis > Monitoring functions** dialog box.

The following error monitoring on the position-controlled axis is performed using the calculated following error. The axis technology DB returns error *8016* if the offset between the actual position value and the position setpoint exceeds the programmed following error limit. The permitted following error depends on the velocity setpoint of the axis.

At velocities less than the configured minimum the permitted following error is constant and is programmed at the "constant following error" parameter. Above this limit, there is a linear increase of the following error to a maximum value which is defined by the parameter "maximum permitted following error" at maximum velocity. The permissible maximum following error is reached at maximum velocity.



If you specify a value in the "Enter velocity" input box, the corresponding function value is displayed in the "Calculated following error" box. This allows you to control how large the following error may be at the entered speed.

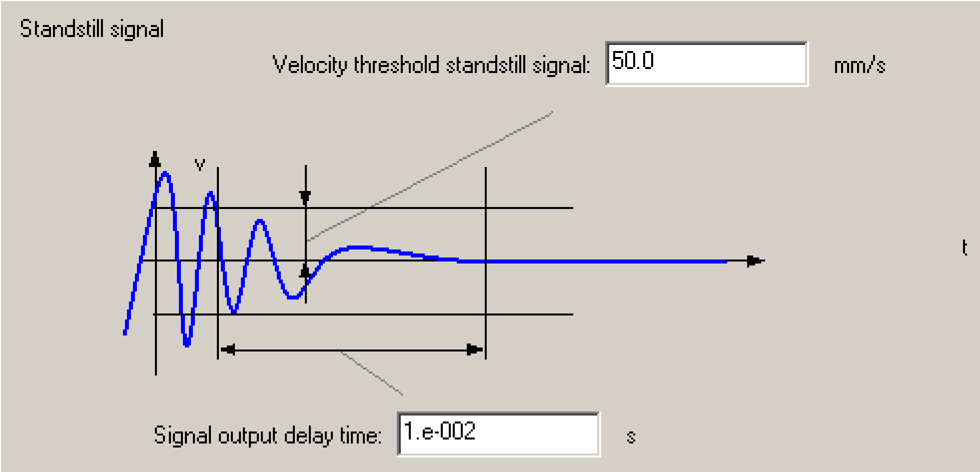
The "Following error monitoring" tab is shown for position-controlled real axes.

4.7.14.4 Monitoring functions - "Standstill signal" tab

The velocity threshold and the "Signal output delay" can be set in the the "Standstill signal" tab of the **Axis > Monitoring functions** dialog box.

The standstill signal sets the *standstill* bit in the status word of the technology DB if the current velocity is below the configured velocity threshold at least for the duration of the set delay time.

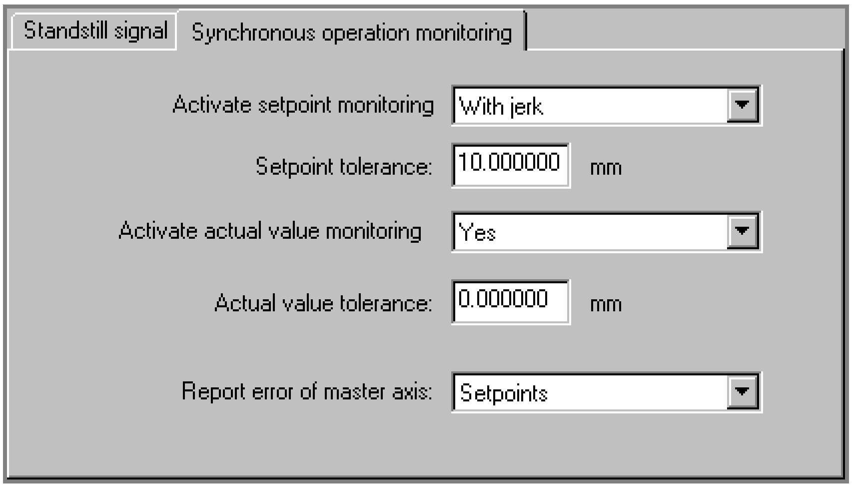
At speed-controlled and positioning axes the override is activated in speed-controlled mode. At the positioning axis, the override is activated when the positioning window is reached.



4.7.14.5 Monitoring functions - "Synchronous operation monitoring" tab

The "Synchronous operation monitoring" tab of the **Axis > Monitoring functions** can be used to enable synchronization error monitoring at the following axis. The "Synchronous operation monitoring" tab is displayed if the **synchronization** technology was set for the axis.

The following settings are possible:



Activate setpoint monitoring:

Deviations may develop between the setpoint of the synchronization correlation and the following axis setpoint which is limited by dynamic values. You can react to setpoint deviations by taking the following measures:

- "no"
  Select "no" if you want to discard monitoring.
- "without jerk"/"with jerk"
  Select one of these settings to activate monitoring (it is not possible to make allowances for the jerk in this version).

A setpoint error at the following axis stops the axis: the following axis technology DB returns error *801A*.

Setpoint tolerance:

Enter the maximum setpoint tolerance.

Activate actual value monitoring:

The actual value monitoring function monitors the difference between the setpoint and actual value of the following axis. Actual value monitoring can be enabled and disabled.

An actual value error at the following axis stops the axis: the following axis technology DB returns error *801A*.

Actual value tolerance:

Enter the maximum actual value tolerance.

Report error of master axis:

Deviations may develop between the setpoint of the synchronization correlation and the setpoint which is limited by dynamic values. You can react to setpoint deviations by taking the following measures:

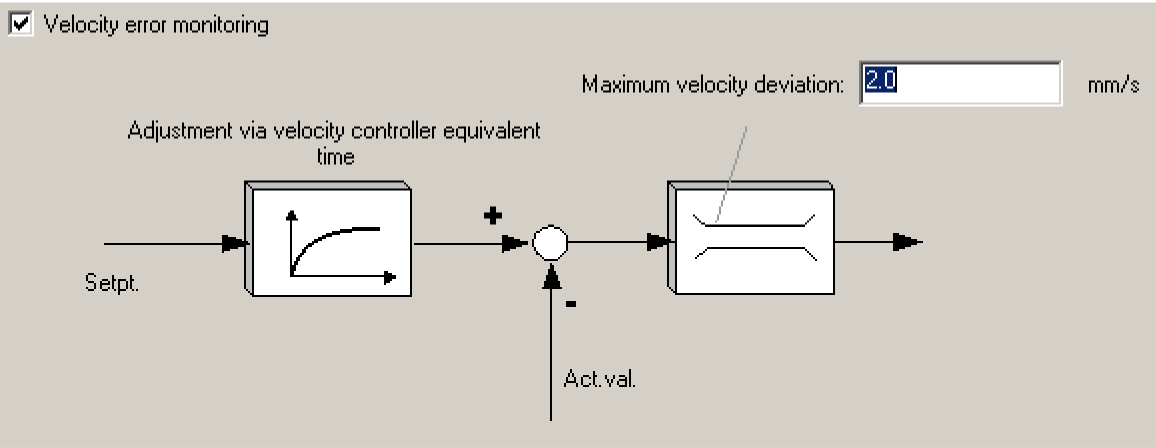
- "no"
  The leading axis is not stopped when a setpoint or actual value error is detected at the following axis.
- "Setpoints"
  A setpoint error at the following axis also stops the leading axis; the leading axis technology DB returns error *801A*.
- "Actual values"
  An actual value error at the following axis also stops the leading axis; the leading axis technology DB returns error *801A*.

4.7.14.6 Monitoring functions - "Velocity monitoring" tab

Enable velocity error monitoring in the "Velocity error monitoring" tab of the **Axis > Monitoring functions** dialog box.

Velocity error monitoring is only relevant to these axes:

- Speed-controlled axis with encoder
- Positioning axes operating in speed-controlled mode ("MC_MoveVelocity" - input parameter *PositionControl = FALSE*)



Velocity error monitoring

Enable velocity error monitoring in this dialog.

Maximum velocity error:

This input box is only visible if velocity error monitoring is enabled.

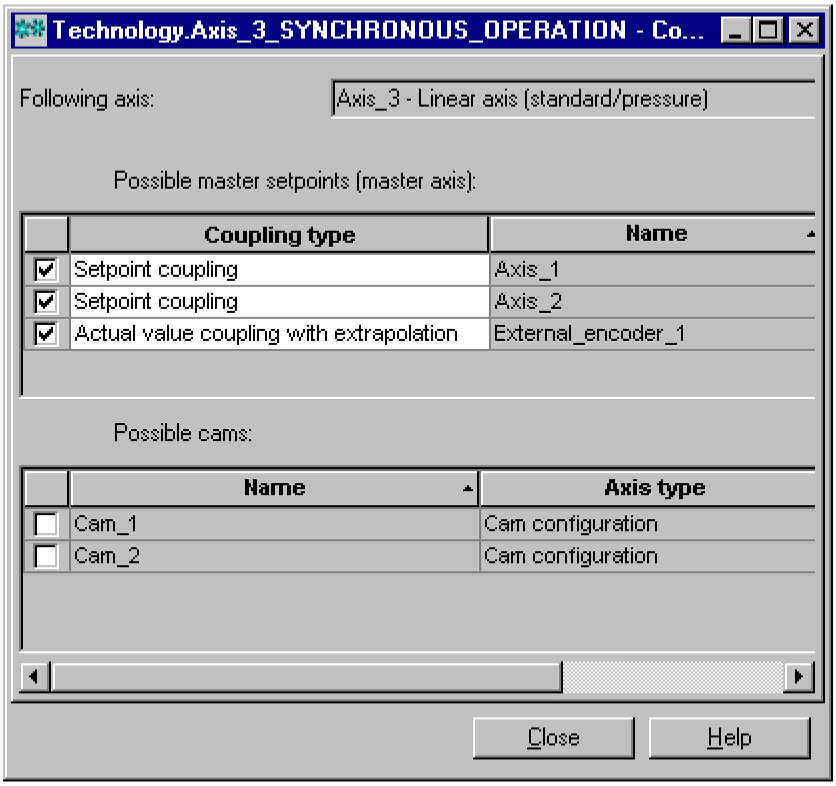
Enter the maximum velocity error in this dialog.

4.7.15 Following object

4.7.15.1 Configuration

Configuration - Synchronous object

Assign master values and cams to the synchronization axis (following axis) in the dialog Synchronous object > Configuration .



following axis:

The name of the following axis (synchronized axis) is displayed in this field.

Possible master values (leading axis):

The master values available in the project, which you can assign to the following axis, are found in this list. The master value can be specified by the following technology objects:

- Axis (real or virtual axis)
- External encoder

In this dialog, take into account all the combinations of leading axis/following axis which you will use in the user program with the technology functions such as "MC_GearIn" or "MC_CamIn".

Possible cams:

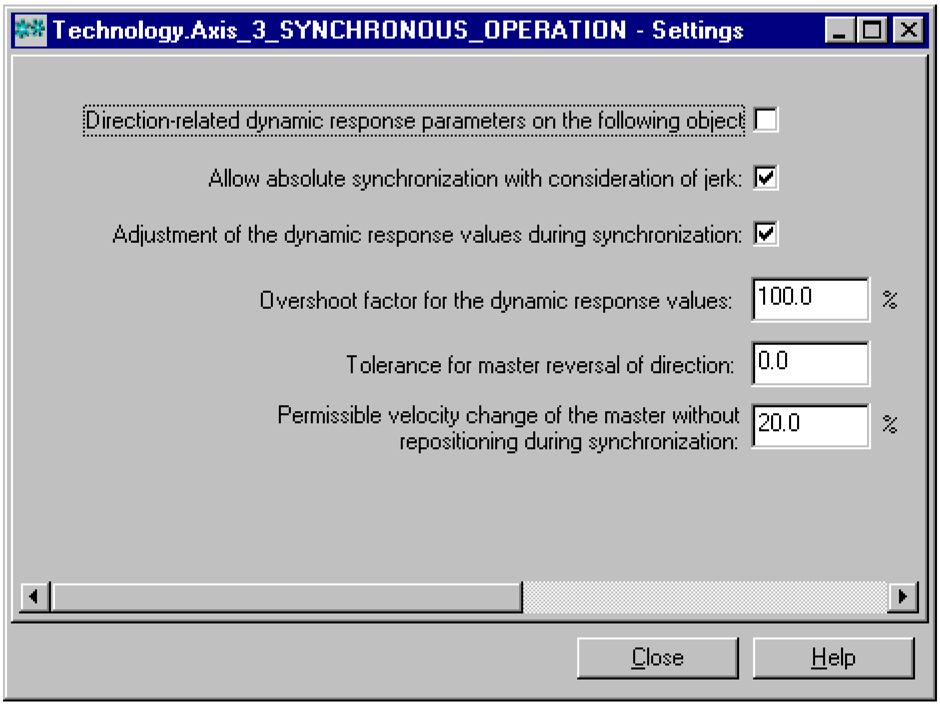
The list includes all configured cams. You can assign cams to the synchronous object for camming.

In this dialog, take into account all the cams which you will use in the user program with the associated synchronization axis.

4.7.15.2 Settings

Settings - Synchronous object

In the dialog **Settings - Synchronous object** define some parameters for the synchronization.



Direction-related dynamic parameters on the synchronous object

Activate the check box if you want to achieve a direction-related effect of the programmed dynamic response values.

Permit absolute synchronization with provision for jerk

Activate this check box if the jerk is to be taken into account for absolute synchronization. This setting is effective only for trailing synchronization.

Adapt the dynamic response values for the synchronization

Activate the check box if the dynamic response values are to be adapted in the synchronous position.

Magnification factor for the dynamic response values

(The field is only visible if the check box "Adapt the dynamic response values for the synchronization" is activated)

Specify the overdrive factor for the adapted dynamic response values here to compensate for a remaining path difference. Enter the value as a percentage (%) in relation to the current master value velocity for the synchronization start.

Tolerance for master direction reversal

Specify a tolerance window for canceling the synchronization for direction reversal of the master values.

Permitted velocity change of the master without restarting for synchronization

In this field enter the maximum permitted change of the master value velocity during synchronization. The value refers to the current master value velocity for the synchronization start

If during synchronization the leading axis accelerates more than the specified value, the following axis cannot synchronize. In this case increase the value for the permitted velocity change of the master.

4.7.15.3 Default setting

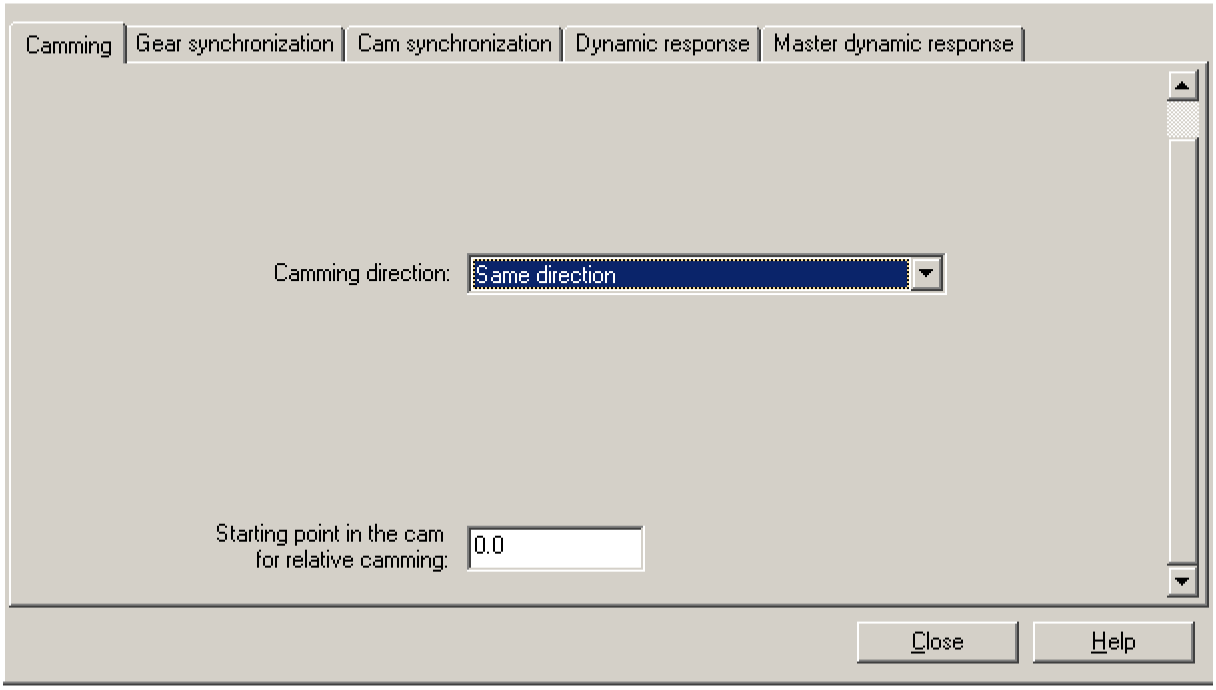
Synchronization object defaults - "Cam synchronization" tab

Set the properties of the synchronization object for the activation of camming in the "Cam synchronization" tab of the **Axis_SYNCHRONOUS_OPERATION > Default** dialog box.

The settings in this dialog box are only active if *Mode = 2* is set at the "MC_CamIn" or "MC_CamInSuperImposed" technology functions.

Camming is characterized by variable coupling between the master setpoint source and the following axes. The coupling is described by a cam disk (transfer function).

Scaling and shift (offset) of camming is possible on both the master setpoint source side and on the following axes side. This enables individual adaptation of a cam disk in terms of its definition range and range of values.



Sense of direction in camming:

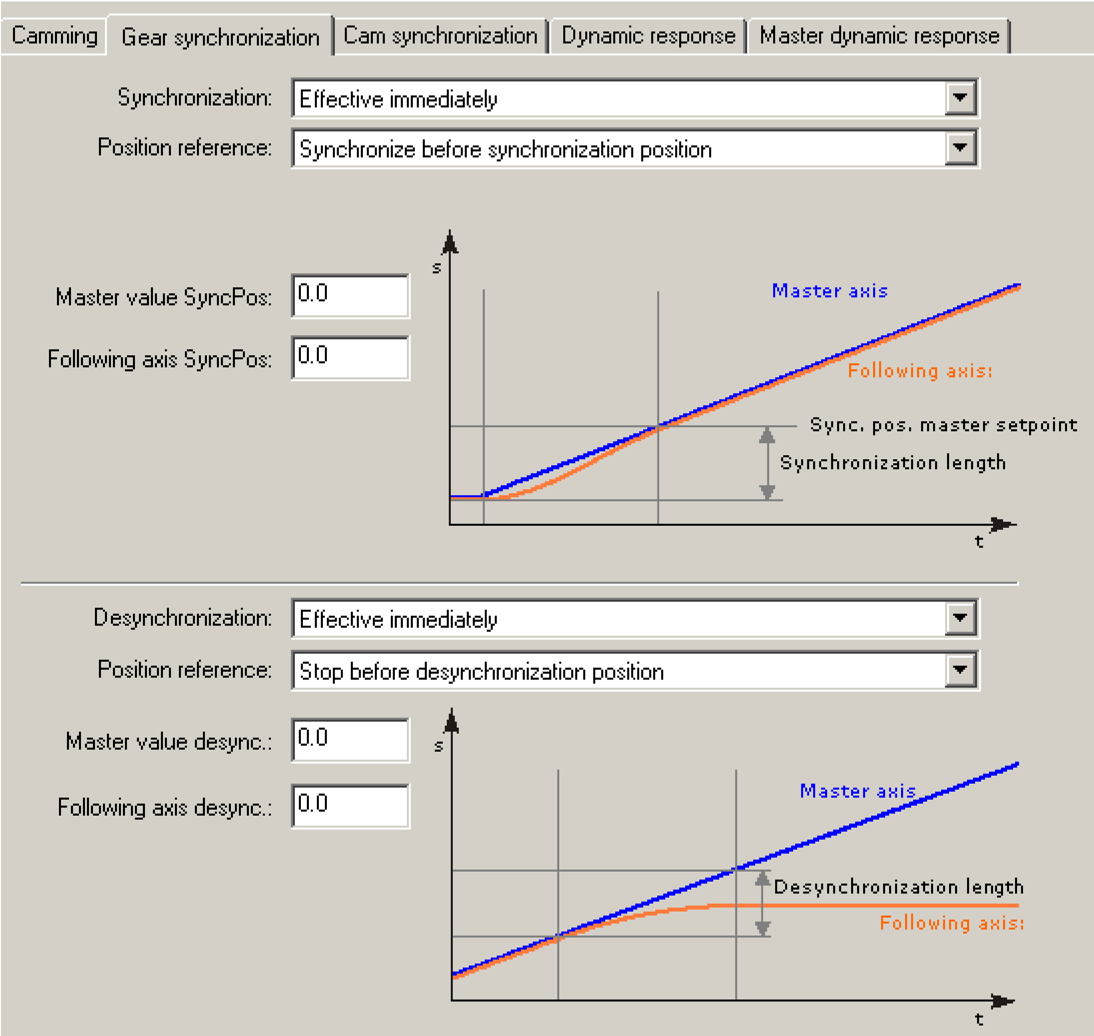
Specify the direction in which the cam disk is executed.

Start point in the cam disk for relative camming:

Define the start point in the cam disk for relative camming as of which the leading axis passes the cam disk.

Synchronization object defaults - "Gear synchronization" tab

Set the synchronization / desynchronization parameters in the "Gear synchronization" tab of the **Axis_SYNCHRONOUS_OPERATION > Default** dialog box. These settings are only relevant in gearing mode. You can set the following parameters:



Synchronization

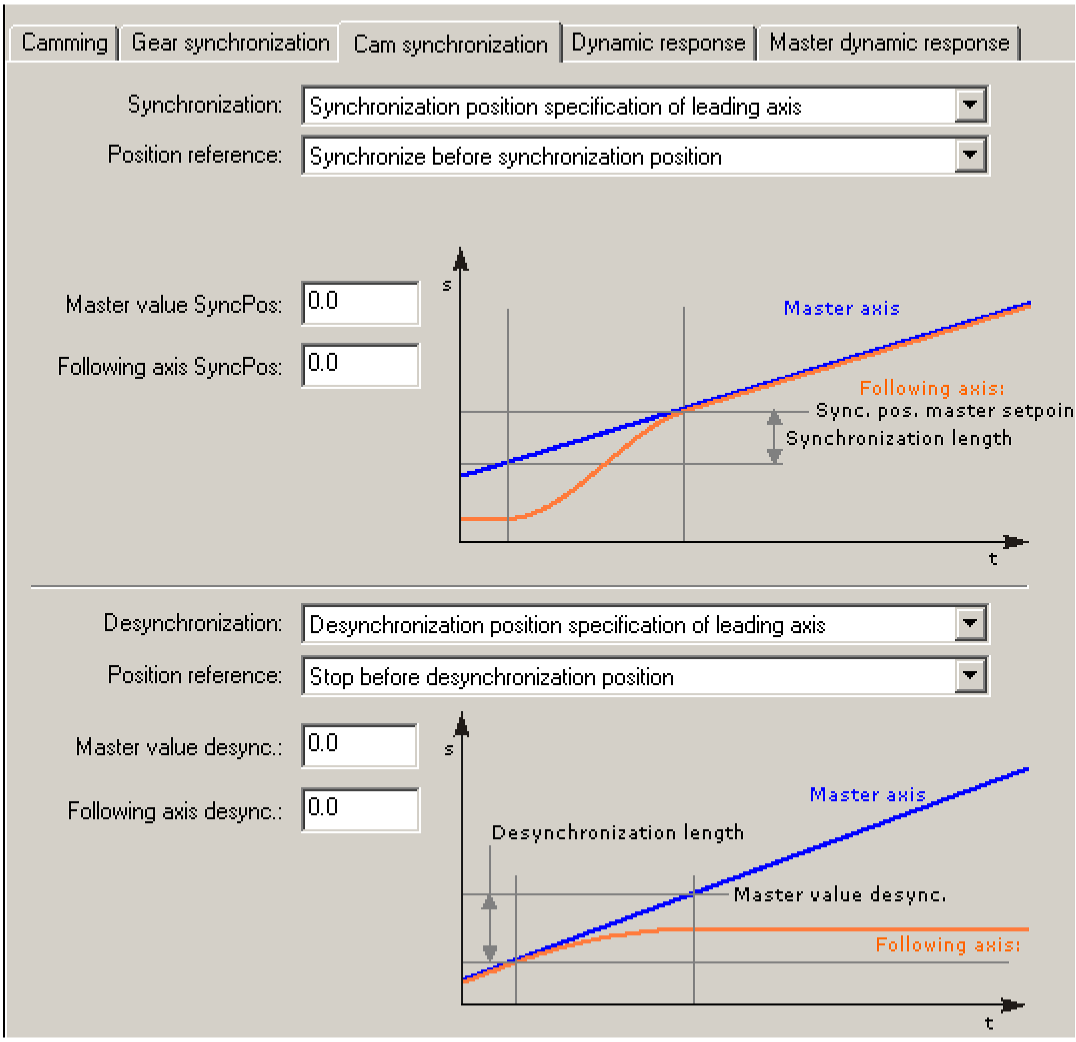
- **Synchronization:**
  Define the time at which the following axis is synchronized with the leading axis.
- **Position reference:**
  Define the position of the synchronization profile relative to the position of the synchronization point.
- **Synchronization direction:** (only for modulo axes)
  Define the synchronization direction.
- **Master value SyncPos:**
  Enter the position of the synchronization point for the leading axis.
- **Following axis SyncPos:**
  Enter the position of the synchronization point for the following axis.

Desynchronization:

- **Desynchronization:**
  Define the time at which synchronization of the following axis with the leading axis is deactivated
- **Position reference:**
  Define the position of the desynchronization profile relative to the position of the desynchronization point.
- **Desync. desync.:**
  Enter the position of the desynchronization point for the leading axis.
- **Desync. desync.:**
  Enter the position of the desynchronization point for the following axis.

Synchronization object defaults - "Cam synchronization" tab

Set the synchronization / desynchronization parameters in the "Cam synchronization" tab of the **Axis_SYNCHRONOUS_OPERATION** dialog box. These settings are only relevant to camming mode. You can set the following parameters:



Synchronization

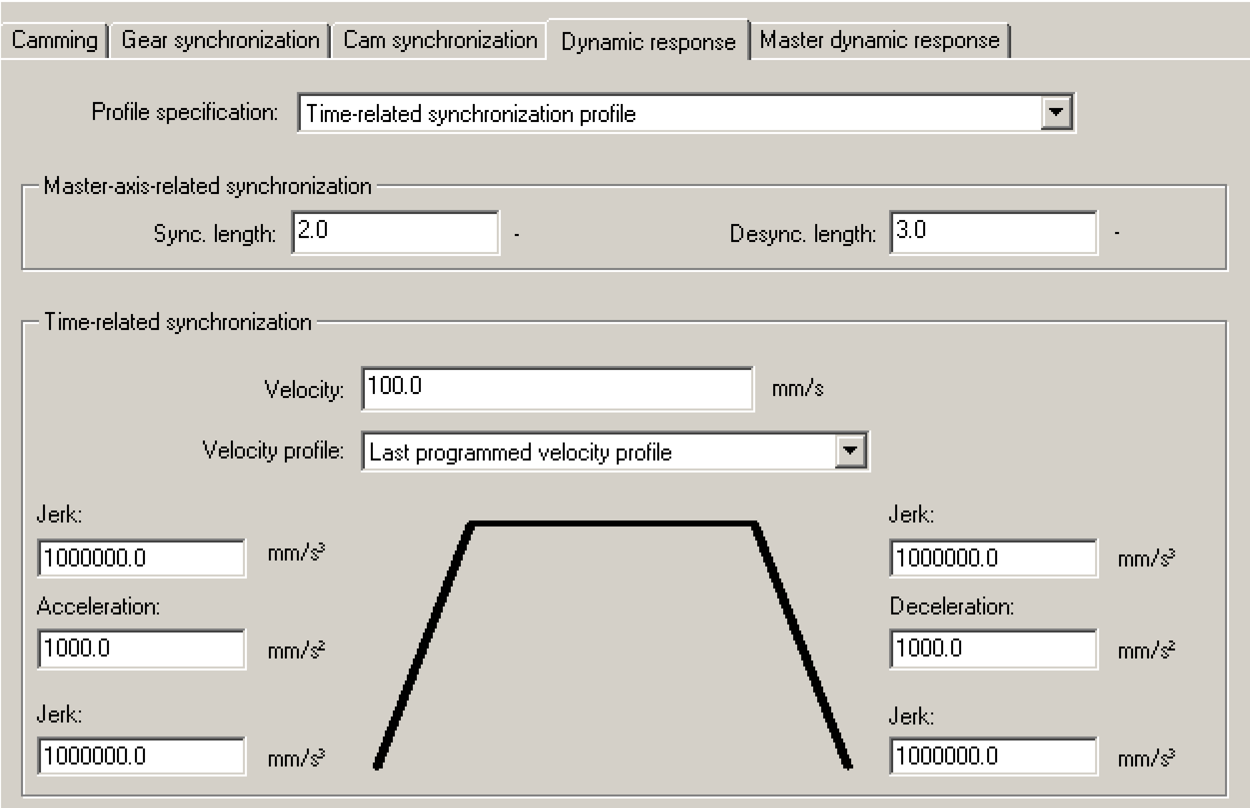
- Synchronization:
  Define the time at which the following axis is synchronized with the leading axis.
- Position reference:
  Define the position of the synchronization profile relative to the position of the synchronization point.
- Synchronization direction: (only for modulo axes)
  Define the synchronization direction.
- Master value SyncPos:
  Enter the position of the synchronization point for the leading axis.
- Following axis SyncPos:
  Enter the position of the synchronization point for the following axis.

Desynchronization

- Desynchronization:
  Define the time at which synchronization of the following axis with the leading axis is deactivated
- Position reference:
  Define the position of the desynchronization profile relative to the position of the desynchronization point.
- Master value desync.:
  Enter the position of the desynchronization point for the leading axis.
- Following axis desync.:
  Enter the position of the desynchronization point for the following axis.

Synchronization object defaults - "Dynamics" tab

Set the basic defaults for synchronization / desynchronization in the Axis_SYNCHRONOUS_OPERATION dialog box, "Dynamics" tab.



Profile setting:

Here you define the reference for the synchronization profile.

- Leading axis-related synchronization
- Time-related synchronization profile
- Last programmed setting

The default "last programmed setting" profile is not available in this version.

Leading axis-related synchronization

- Sync. Length:
  Here you enter the synchronization distance.
- Desync. Length:
  Here you enter the desynchronization distance.

The synchronization / desynchronization length is only evaluated for a synchronization profile related to the leading axis.

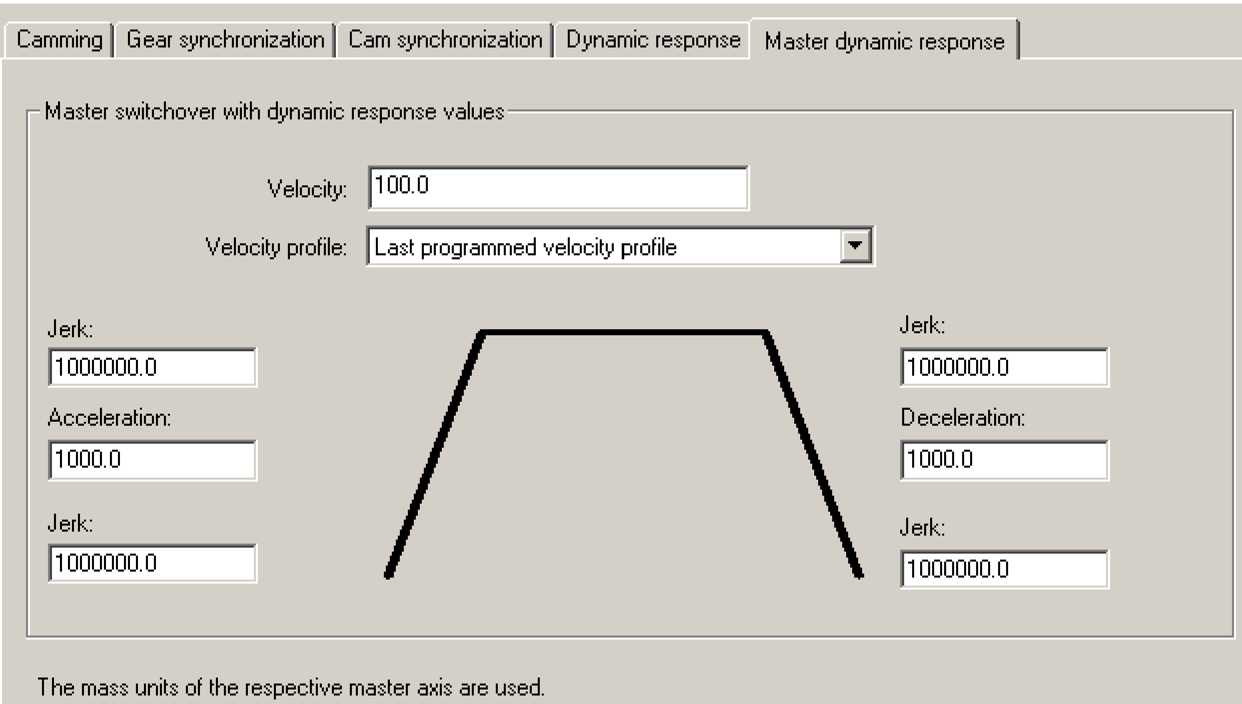
Time-related synchronization

- **Velocity:**
  Here you enter the maximum velocity.
- **Velocity profile:**
  Here you select the velocity profile.
- **Acceleration:**
  Enter the maximum acceleration at this parameter
- **Deceleration:**
  Enter the maximum deceleration at this parameter.
- **Jerk:**
  Enter the maximum jerk at this parameter.

The Velocity profile, Velocity, Acceleration, Deceleration, and Jerk parameters are only evaluated for time-related synchronization profiles.

Synchronization object defaults - "Master dynamic response" tab

Set your dynamic response values for master setpoint changeover in the **Axis_SYNCHRONOUS_OPERATION > Default** dialog box "Master dynamic response" tab.



You can set the following parameters:

- Velocity:
  Here you enter the maximum velocity.
- Velocity profile:
  Here you select the velocity profile.
- Jerk:
  Enter the maximum jerk at this parameter.
- Acceleration:
  Enter the maximum acceleration at this parameter
- Deceleration:
  Enter the maximum deceleration at this parameter.

4.8 Cam Configuration

Any cam disk applied in the user program must have been inserted in S7T Config as "Cam disk" technology object.

Cam disks within the user program are defined by the "MC_CamClear", "MC_CamSectorAdd" and "MC_CamInterpolate" technology functions. You can call technology function "MC_GetCamPoint" to read in the position values of an interpolated cam disk.

The reference to the cam disk created in S7T Config is defined by using the *CamTable* input parameter. The number of the technology data block that was created for the cam disk has to be entered in *CamTable*.

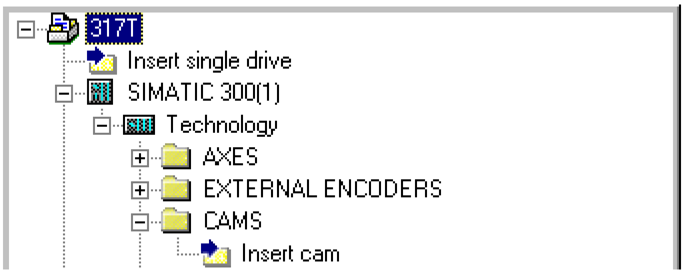
A cam disk application is defined in the user program by programming the "MC_CamIn" and "MC_CamInSuperimposed" technology functions. The cam disk is selected by means of the *CamTable* input parameter. Prerequisite for use is the assignment to a corresponding synchronization configuration.

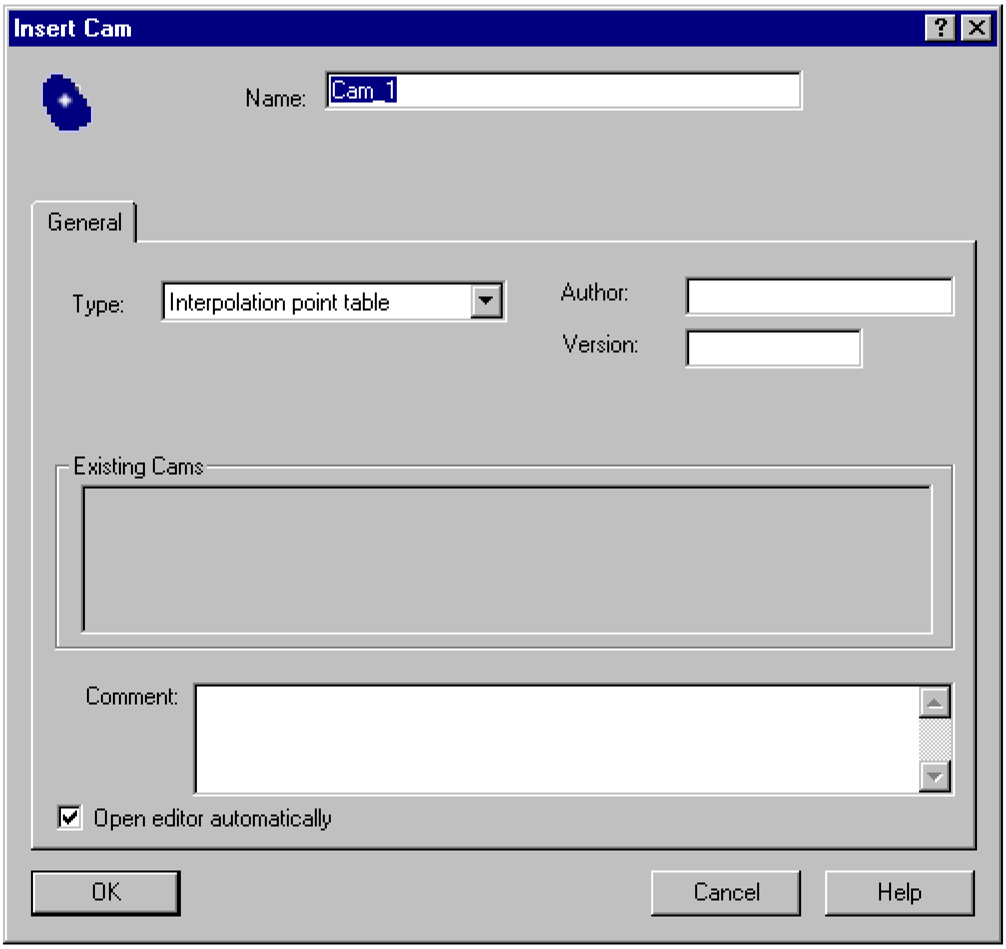
4.8.1 Inserting cams

Requirement

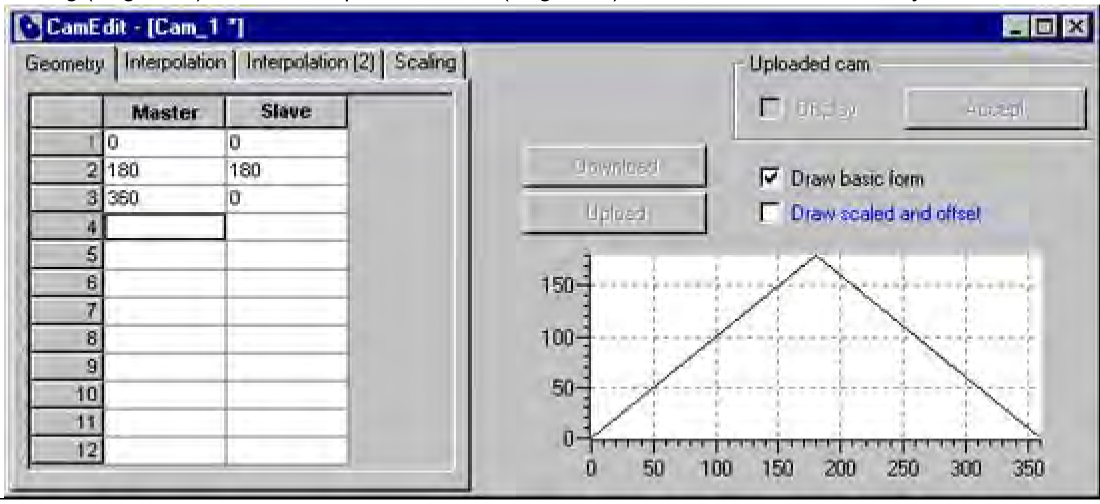
- The data of the Technology CPU were configured in HW Config and compiled.

How to insert a cam disk in S7T Config

Step	Description
1.	To create a new cam disk in S7T Config using the CamEdit tool, double-click "Insert cam disk" in the Navigator of S7T Config.  To create the cam disk using the optional SW package SCOUT CamTool, double-click "Insert cam disk with CamTool". The entry is only visible if the optional package is installed. In the next phases of this example we shall only refer to CamEdit. For a detailed description of the functions and handling of SCOUT CamTool, refer to the "SIMOTION CamTool" manual.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>2.</td><td>Enter the name of the technology object in the "Insert cam disk" dialog box. Fields for the author, the version number and a comment are additionally available.
Select whether the cam is to be defined using an interpolation point table or polynomials. Set the "Open editor automatically" check box in order to automatically open the cam disk configuration dialog box:


Click "OK."
Result: The CamEdit dialog box opens.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>3.</td><td>Define the cam disk by means of the interpolation table or polynomials, depending on the type. Change the scaling (Page 345) and the interpolation mode (Page 342) of the cam disk if necessary.

</td></tr>
<tr><td>4.</td><td>Close CamEdit.</td></tr>
<tr><td></td><td>Result: The technology object is inserted.</td></tr>
</table>

Note

If you want to insert new polynomials or interpolation points after the interpolation, the cam disk has to be deleted beforehand.

The properties of leading and following axes are ignored during interpolation and in the continuity check.

4.8.2 Defining cams

Defining cam disks

In the **Insert Cam** dialog box, you can choose to define the cam based on interpolation points or on polynomials.

- **Definition based on interpolation points**

  Interpolation points are represented in an interpolation point table in the form **P = P(x,y)**. The order in which the value pairs are entered is irrelevant. They are automatically sorted in ascending order in the domain.
  S7T Config interpolates the cams according to the configured interpolation type.

- **Definition based on polynomials/segments**

  The various polynomials are described in accordance with "VDI Directive 2143 (Page 354), Motion Laws for Cam Mechanisms." The maximum degree of the polynomial is 6. A polynomial can also contain a trigonometric function.

Methods for defining cams

Both definition methods, i.e., based on interpolation points or on polynomials, have their advantages and disadvantages. Your application determines which one of these methods will suit the requirements.
A combination of both methods in one cam is only possible in runtime using the MC_CamSectorAdd technology function.

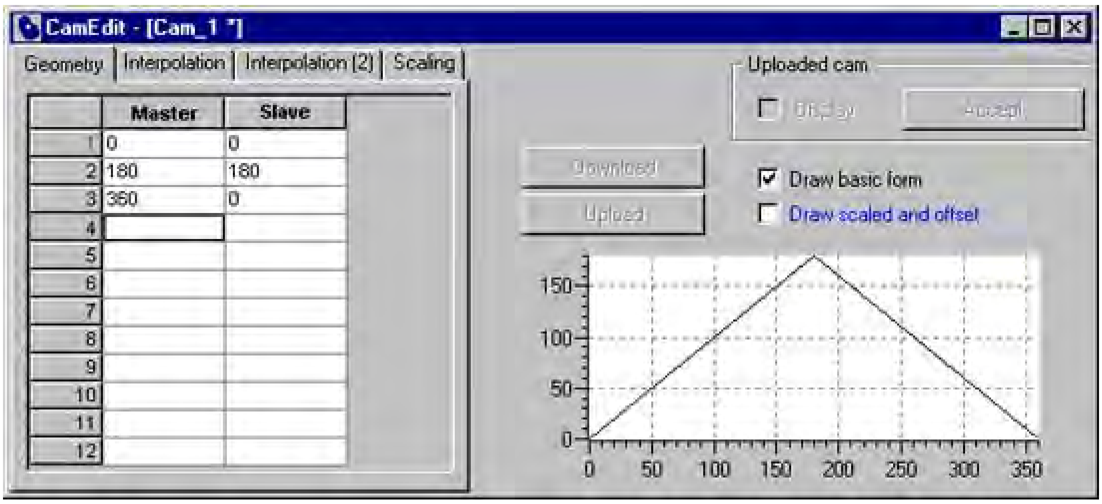
	Definition based on interpolation points	Definition based on polynomials
Advantage	• Simple definition • Any algorithms can be mapped by interpolation points • Plotting by Teach-In • Simple interface to HMI	• Low data volume for the definition • Standard transitions in accordance with VDI... • Extremely precise contour, continuous transitions
Disadvantages	• Large number of interpolation points required for smooth contour	• Requires complex calculation of coefficients

4.8.3 Creating cams with CamEdit

The **CamEdit** dialog box can be used to perform the following tasks:

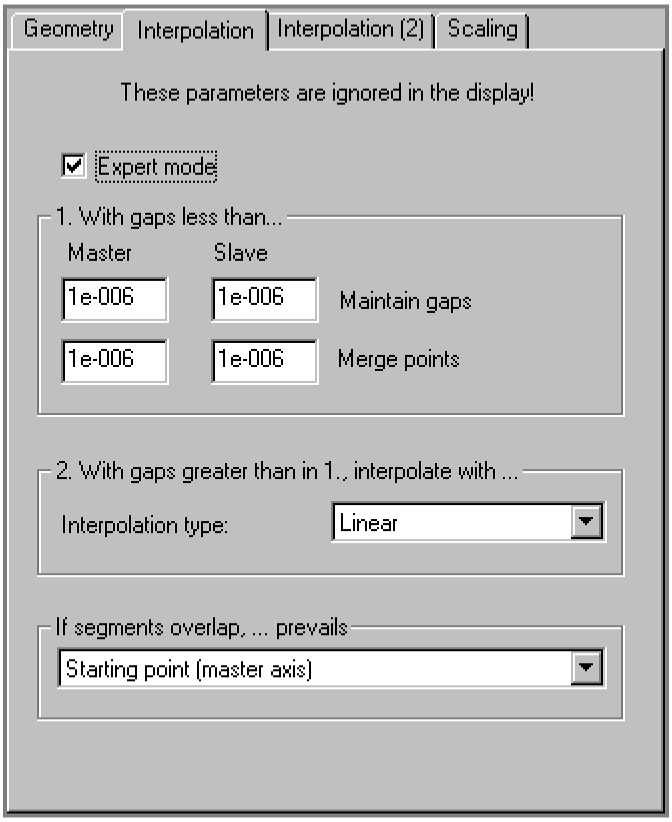
- Define the cam disk by entering their segments or interpolation points
- Create cam disks with defined motion transitions using the VDI wizard.
- Edit the cams of the cam disk
- Download the cams of a cam disk to the PLC (only possible if the cam disk is not in use)
- Read the cams from the PLC
- View the cam's geometry. Compare the cam on the programming device / PC with the cam in the target device (scaled / non-scaled display).

The interpolation settings in CamEdit will be ignored if made in technology function "MC_CamInterpolate". The cam can not be scaled and shifted in the user program.



4.8.3.1 Interpolation

Select the "Interpolation" tab of the **CamEdit** dialog box to define whether to retain gaps between the segments of the cam disk, or whether to interpolate the cam between the end points of particular segments. Gaps are inconsistencies in the curve characteristic. They occur whenever cam segments do not overlap. If the individual segments overlap, you can choose which cam segment is to have priority.



The settings shown can be made by setting the "Expert mode" check box:

With gaps up to a length of…

This function is only available if you are in expert mode.

- **Retain gaps**
  Select the "Retain gaps" row of the "Master" or "Slave" fields to define the length of gaps to be retained between the cam disk segments of the "master axis" and "slave axis".
- **Join points**
  Select the "Join points" row of the "Master" or "Slave" fields to define the maximum gap length between the "master axis" and "slave axis" segments you want to join at their end points.

If the gap length is greater than the length defined in 1., interpolate with ...

At "Interpolation mode", you can select the mode of interpolation between gaps with a length greater than the previously defined maximum.

- Linear (linear interpolation)
  Continuous joining of gaps by inserting a linear segment between the interpolation points.
- Cubic splines (interpolation with cubic splines)
  Continuous differentiable joining of gaps. The curve runs through the specified interpolation points.
- Bezier splines (interpolation with Bezier splines)
  The approximation trend runs along the defined interpolation points.

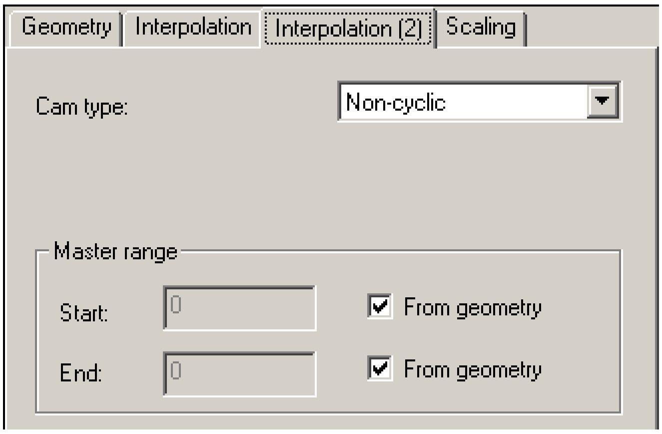
If segments overlap, priority is taken by ...

Select which of the overlapping cam disk segments should take priority. This function is only available if you are in expert mode.

- Start point (master axis)
  The cam profile continues at the start point of the next segment.
- End point (master axis)
  The cam profile of the next segment starts at the end point of the current segment.
- Chronological sequence
  The cam profile is determined by the chronological order of the insertion of segments. The last segment inserted takes highest priority, and the first segment inserted takes lowest priority.

4.8.3.2 Interpolation(2)

Set the type and range of the leading axis values of the cam disk in the "Interpolation(2)" tab of CamEdit. These settings form the interpolation conditions.



Cam type

Set the mode of operation of the cam disk in the drop-down list box:

- Non-cyclic
  The cam disk is not to be used in cyclic mode
- Cyclic absolute
  The cam disk is to be operated in cyclic camming mode, absolute to the following axis
- Cyclic relative
  The cam disk is to be operated in cyclic camming mode relative to the following axis.

Master range (range of leading axis values)

Here you set the range of leading axis values:

- Start point
  Defined by a number or by setting the check mark. With this setting the start point is derived from the existing geometry of the cam disk.
- ENd point
  Defined by a number or by setting the check mark. With this setting the end point is derived from the existing geometry of the cam disk.

4.8.3.3 Scaling and shift

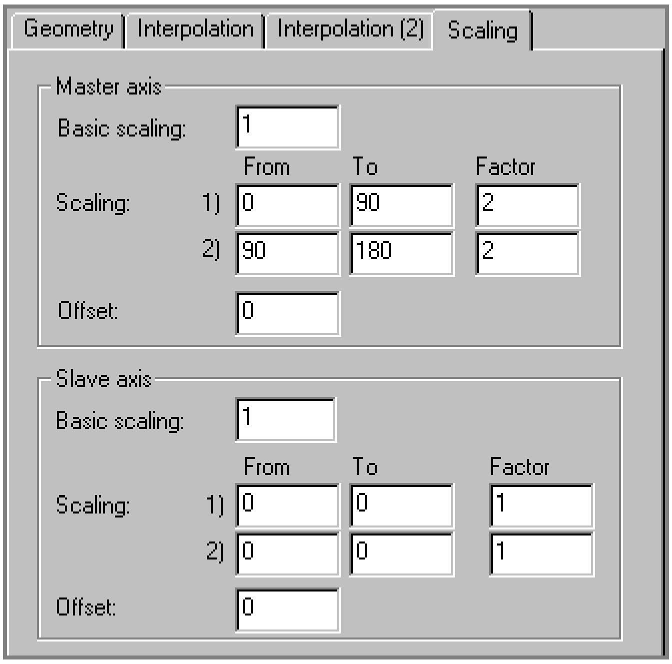
In CamEdit, you can scale (stretch or compressing) and shift the cam disk profile. The scaling and shift can be applied to the values of the leading and following axes.

You can scale either the entire cam disk, or a range defined by the start and end points:

- You can define the basic scaling and shift for the entire cam disk
- The range scaling lets you scale and shift selected cam segments Define the range by means of a start and end point.

The scaling and offset of the cam disk can not be modified by the user program.

Select the "Scaling" tab in the **CamEdit** dialog box to define the scaling and shift of the leading and following axes.



Settings scaling and shift within the range of leading axis values (master axis):

- Basic scaling
  Enter the scaling coefficient for the leading axis in the "Basic scaling of the master axis". The coordinate origin always forms the center of scaling.
  Do not use any scaling values greater than *5*. Greater values cause a roughness in the cam profile.
- Scaling 1) and Scaling 2)
  Use the scaling dialogs to define the scaling of up to two user-specific ranges of the "master axis" (range of leading axis values). Use "from" and "to" to specify the range that you want to scale. Enter the scaling factor under factor. You may not overlap two scaled ranges. The center of scaling forms the start point of the scaling range.
- Shift
  Under "Shift", you can enter a shift coefficient for the master axis (leading axis values). The shift refers to the already scaled cam or to the basic form if you have not scaled the range.

Settings scaling and shift within the range of following axis values (slave axis):

- Basic scaling
  Enter the scaling coefficient for the following axis in the "Basic scaling of the slave axis". The coordinate origin always forms the center of scaling. Do not use any scaling values greater than *5*. Greater values cause a roughness in the cam profile.
- Scaling 1) and Scaling 2)
  Use the scaling dialogs to define the scaling of up to two user-specific ranges of the "slave axis" (range of following axis values). Use "from" and "to" to specify the range that you want to scale. Enter the scaling factor under factor. You may not overlap two scaled ranges. The center of scaling forms the start point of the scaling range.
- Shift
  Under "Shift" you can enter a shift coefficient for the "slave axis" (following axis values). The shift refers to the already scaled cam or, if you have not scaled the range, to the basic form.

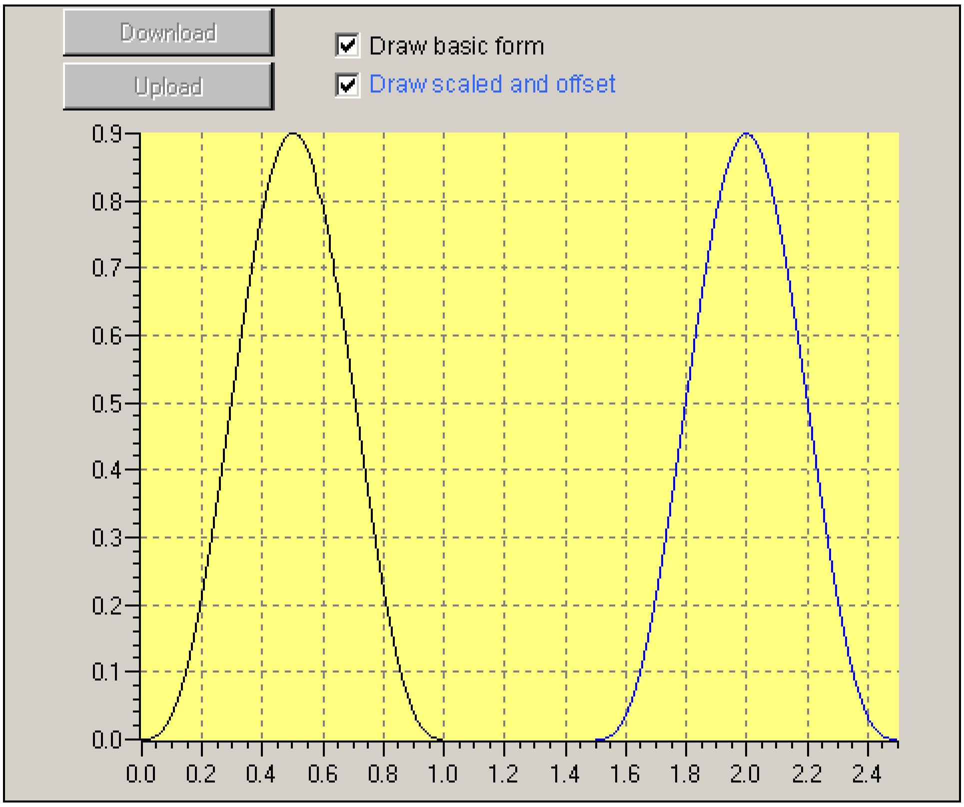
Note

If a scaled and/or shifted cam disk is used in more than one object, the scaling and shift settings affect all relevant objects.

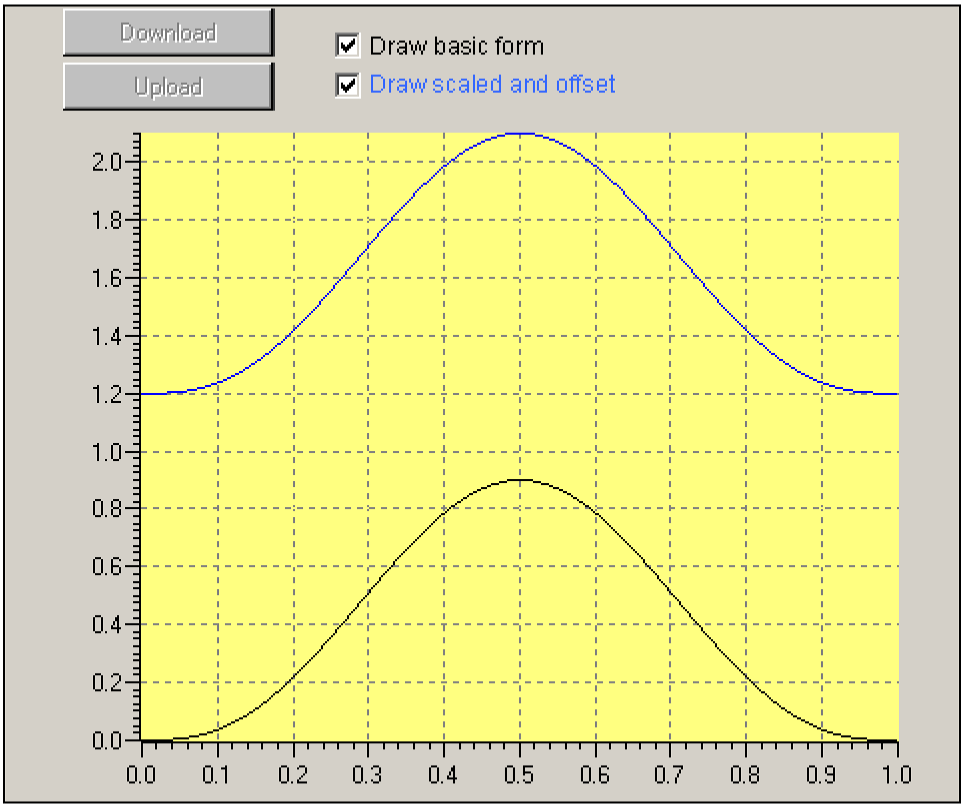
Examples of cam scaling and shift

The examples below demonstrate the effects of scaling and shift on the cams. The original cam is represented by a black line. The blue line represents a scaled or shifted cam.

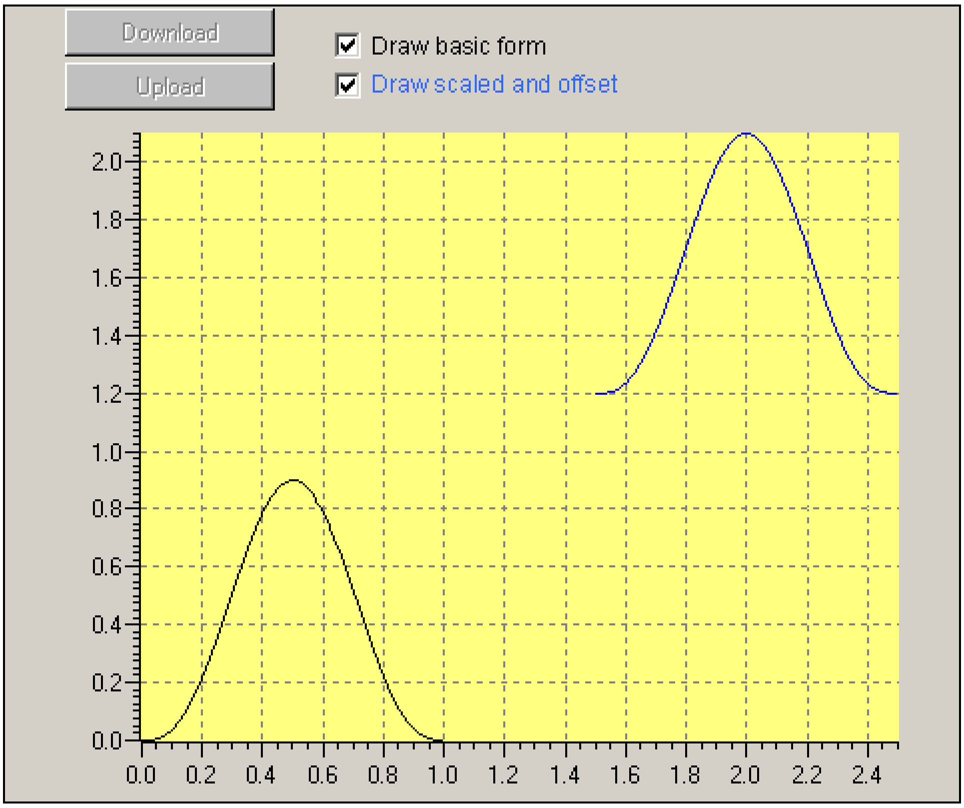
Shift of the leading axis values:



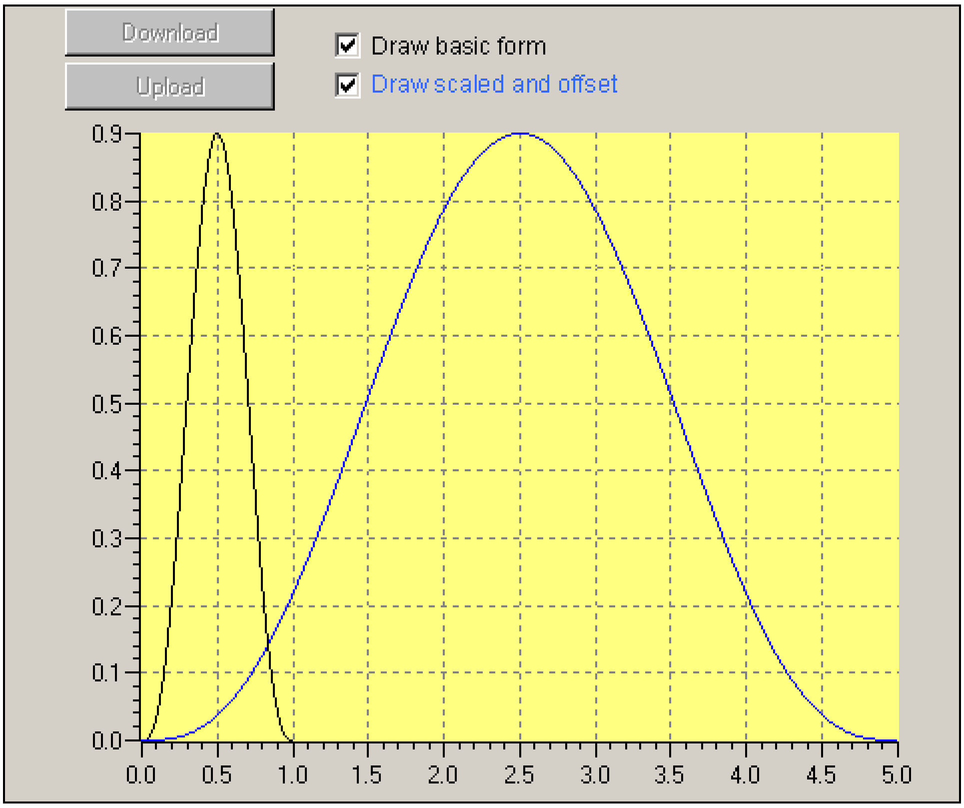
Shift of the following axis values:



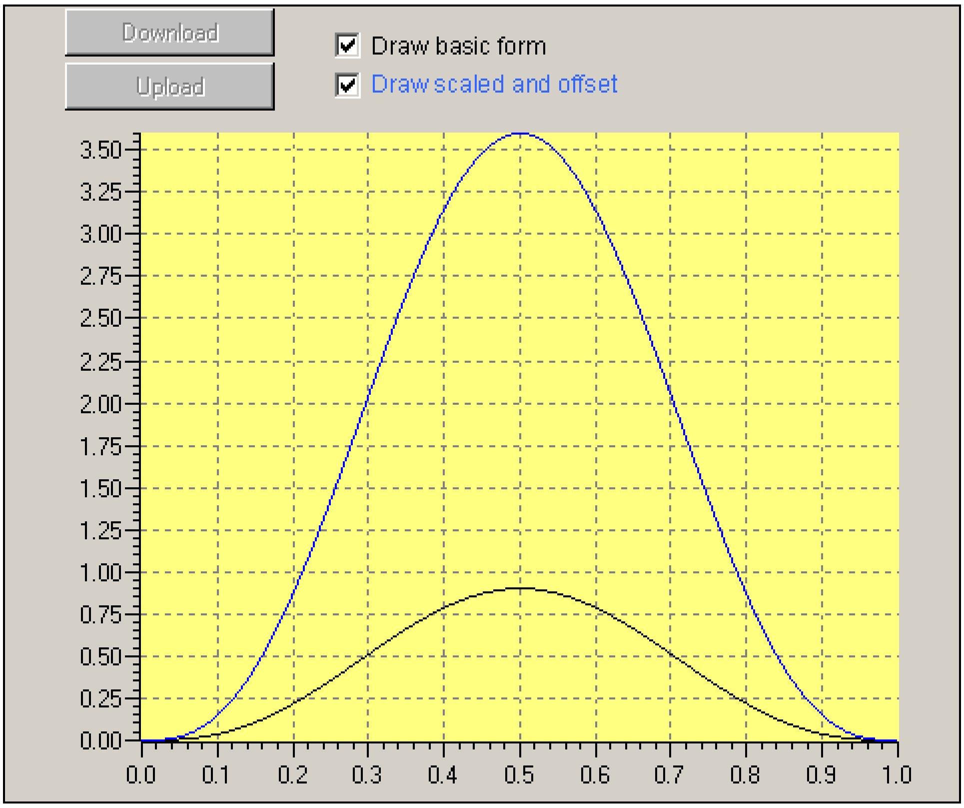
Shift of the leading axis and following axis values:



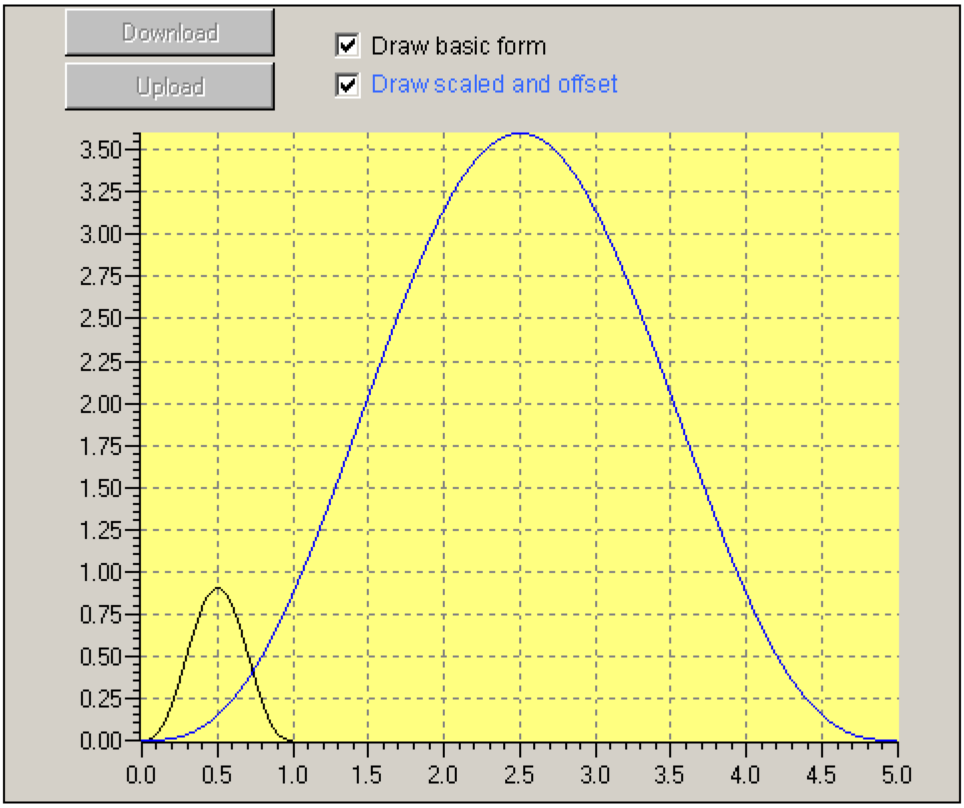
Scaling of the leading axis values:



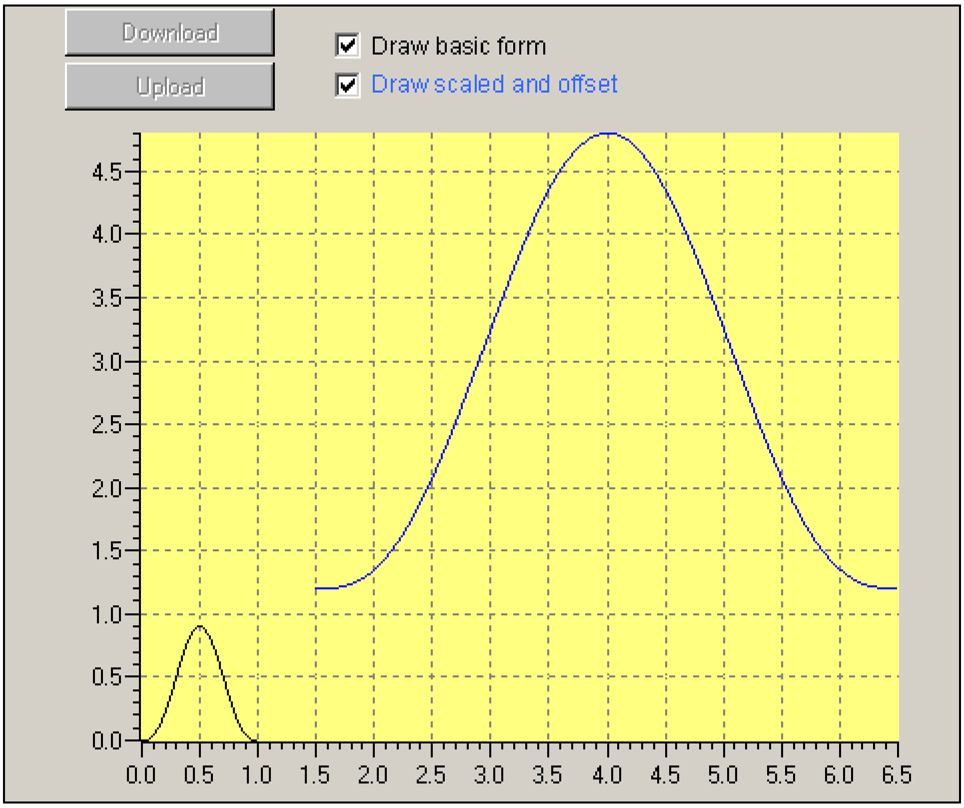
Scaling of the following axis values:



Scaling of the leading axis and following axis values:



Shift of the leading axis and following axis values
Scaling of the leading axis and following axis values:



4.8.4 Creating cams using CamTool

You can also use the SCOUTCamTool to create and edit cam disks. This tool must be purchased separately. The SCOUTCamTool has the following benefits:

- Precise, graphic visualization of the cam
- Quick and easy cam definition by means of drag-and-drop of cam elements
- Quick and easy cam tuning by means of "dragging at the profile"
- Simultaneous visualization of the position, velocity, acceleration and jerk profile has an immediate effect on maximum velocity, the motor torque required and on mechanical load.
- Tuning the velocity, acceleration or jerk parameters of the cam

The SCOUTCamTool can be totally integrated in the user interface of S7T Config.

For detailed information on the functions and handling of this SW, refer to the "SIMOTIONCamTool" manual.

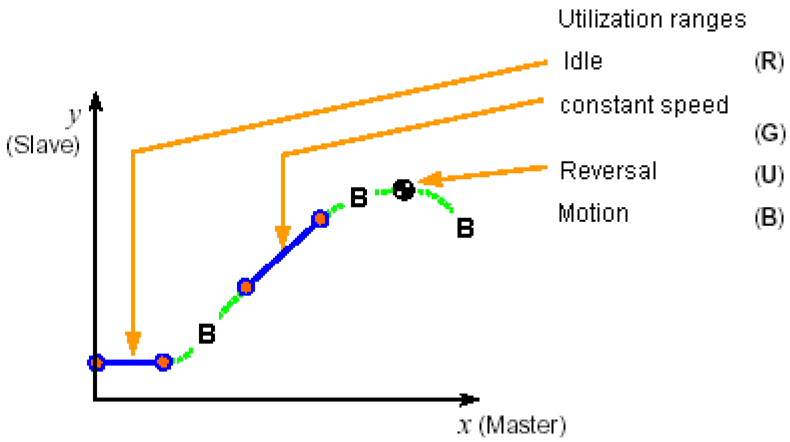
4.8.5 Motion laws in accordance with VDI

4.8.5.1 Working ranges and motion transitions

The VDI concept distinguishes between **working ranges** and **motion transitions**:

- Working ranges correspond to sequences in a process. The VDI concept distinguishes between four working ranges:
- Motion transitions represent transitions between working ranges. Although these are not directly relevant to the process, they must satisfy certain conditions such as constant velocity and acceleration.

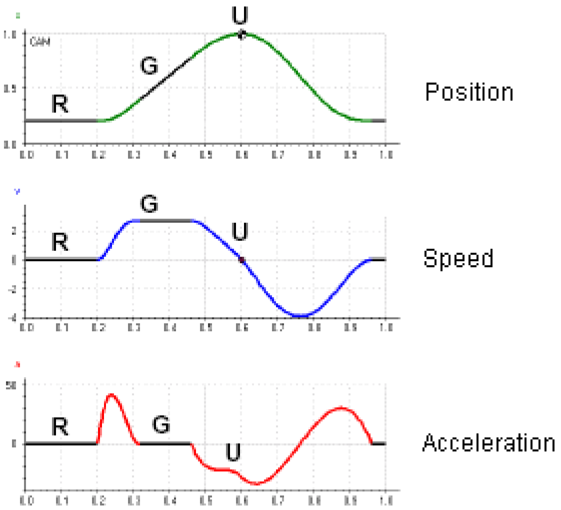
Working ranges



The VDI concept distinguishes between the following working ranges:

	Working range	Velocity (v)	Acceleration (a)
R	Dwell	= 0	= 0
G	Constant velocity	≠ 0	= 0
U	Reversal	= 0	≠ 0
B	Motion	≠ 0	≠ 0

Example of a cam disk with three working ranges



Motion transitions

Possible motion transitions in the various working ranges:

	R	G	U	B
R	v=0 a=0 v=0 a=0	v=0 a=0 v≠0 a=0	v=0 a=0 v=0 a≠0	v=0 a=0 v≠0 a≠0
G	v≠0 a≠0 v=0 a=0	v=0 a=0 v≠0 a=0	v=0 a≠0 v=0 a≠0	v≠0 a=0 v≠0 a≠0
U	v=0 a≠0 v=0 a=0	v=0 a≠0 v≠0 a=0	v=0 a≠0 v=0 a≠0	v=0 a≠0 v≠0 a≠0
B	v≠0 a≠0 v=0 a=0	v≠0 a≠0 v≠0 a=0	v≠0 a≠0 v=0 a≠0	v≠0 a≠0 v≠0 a≠0

Note

The **VDI Wizard** supports the creation of cam disks in S7T Config. You can open it in the **CamEdit** dialog box.

4.8.5.2 Defining a cam disk with segments

Definition of working ranges

The **working ranges** of a motion task are usually specified by the process.

Example:

1. A tool waits on a production line for a part to pass by (dwell state).
2. The tool is synchronized to the part and performs a task on the work piece (constant velocity).
3. The tool then returns to the waiting position (reversal).

The process restarts from the beginning.

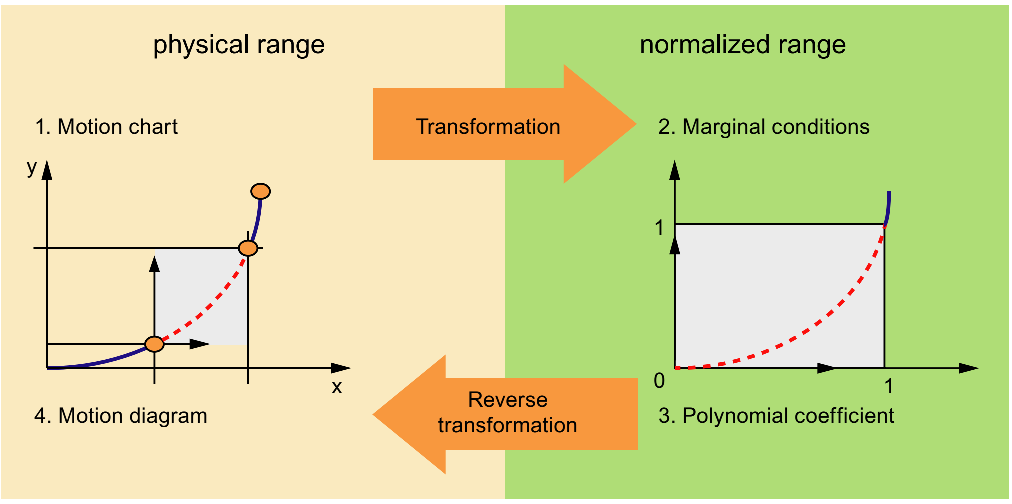
In order to implement this sequence, the cam disk must be segmented in the first step according to the working ranges.

Creating a motion transition

In the next step you define the motion transitions which meet certain conditions, for example, jerk-free motion.

- Start by transforming the motion transition to the normalized range.
- Make allowances for conditions such as the position, velocity, and acceleration at the segment limits.
- In order to apply the polynomial defined in this way, it must be transformed back into the real range.

Creating motion transitions to VDI:



References

- VDI Guideline 2143, Bl. 1: Motion Laws for Cam Mechanisms - Basic Theory Düsseldorf: VDI-Verlag, 1980
- Volmer, J. (edited.): Mechanism Design - Cam Mechanisms, 2. Printed in Berlin Published by Technik Verlag, 1989

4.9 Configuring output cams

Any output cam to be used in the user program must have been be inserted in S7T Config as "Output cam" technology object.

You program the object within the user program using the MC_CamSwitch (position-based/switching output cam) or MC_CamSwitchTime technology function (time-based output cam). The assignment to the output cam created in S7T Config is defined by setting the corresponding Technology DB number at input *CamSwitch*.

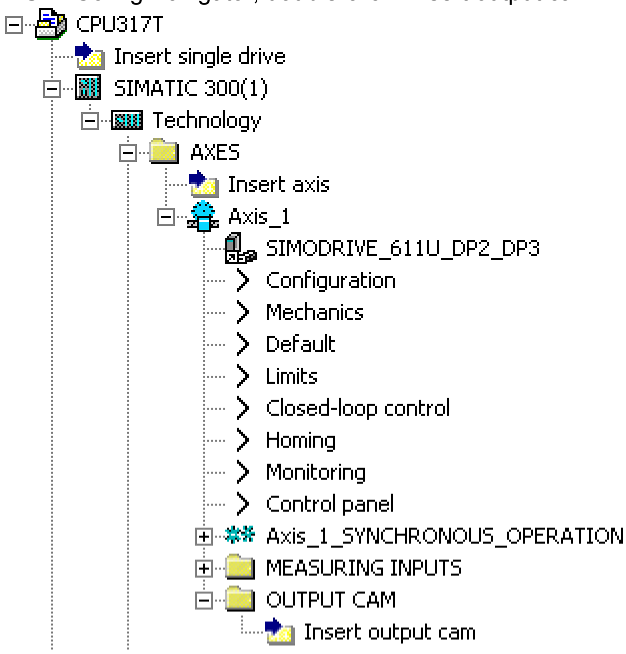
The ON / OFF conditions are configured directly at the block.

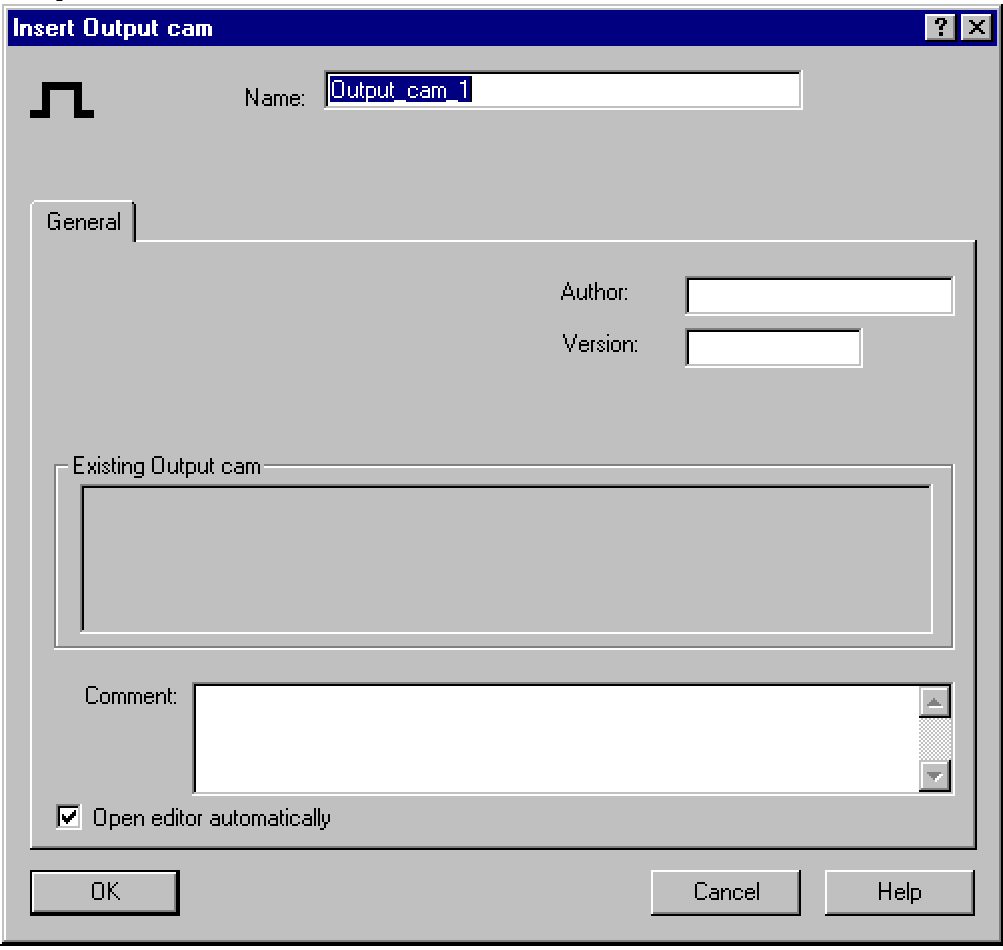
4.9.1 Inserting output cams

Prerequisite

- The data of the Technology CPU were configured in HW Config and compiled.
- An axis or external encoder was created in S7T Config.

How to insert an output cam in S7T Config

Step	Description
1.	In S7T Config Navigator, double-click "Insert output cam". 

Step	Description
2.	Enter the name of the technology object in the "Insert output cam" dialog box. Fields for the author, the version number and a comment are additionally available. Activate the "Open editor automatically" check box in order to automatically open the output cam configuration dialog box: 
3.	Click "OK" to confirm your entries. Result: The technology object is inserted.

Further procedures

Configure the output cam in the **Output cam > Configuration**.

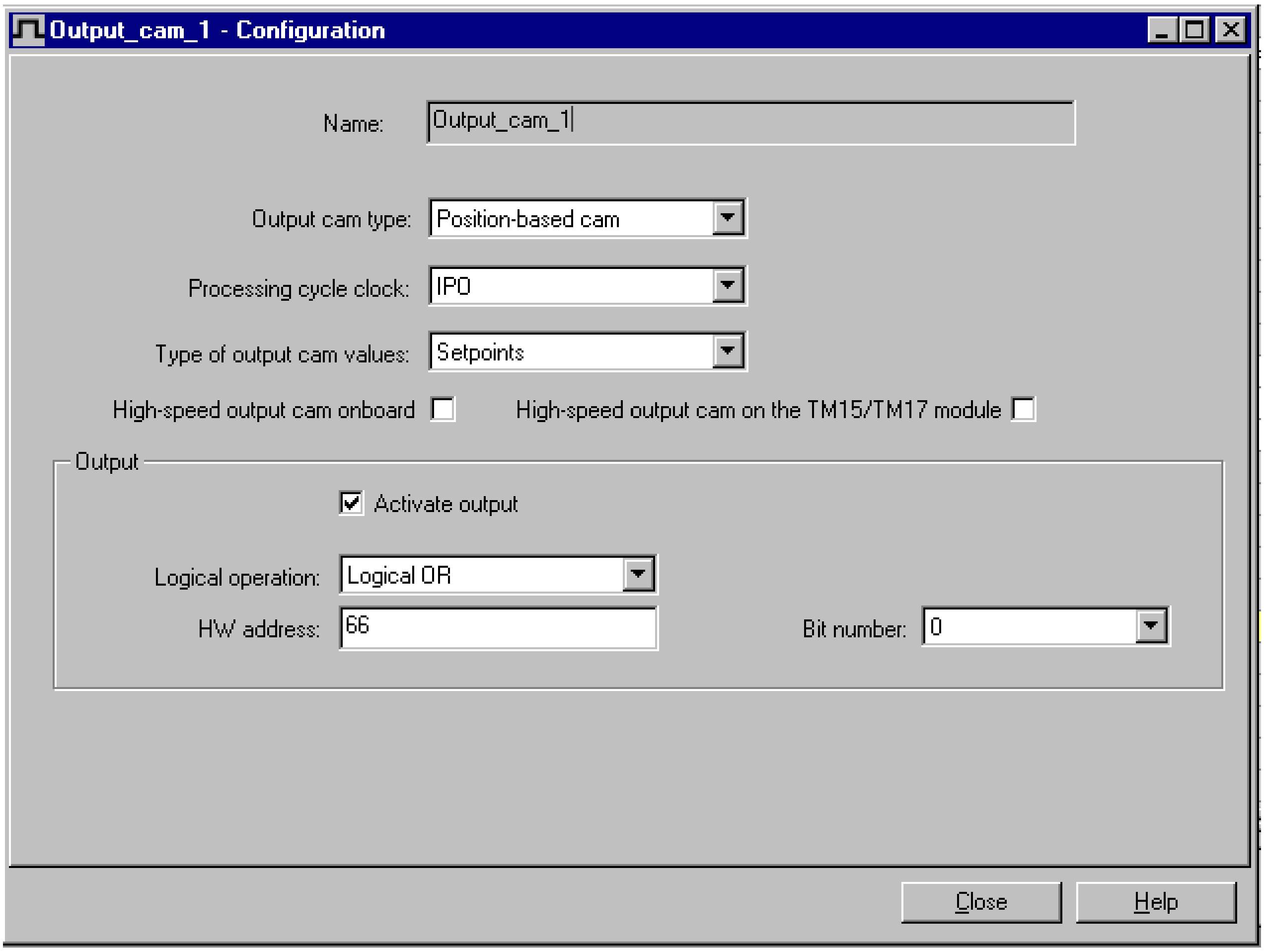
4.9.2 Configuration - Output cam

You can set the following items for the "Output Cam" technology object in the **Output cam > Configuration** dialog box:

- Output cam type: Position-based cam, time-based cam, or switching cam
- The system cycle in which the cam is calculated
- Reference value of the output cam: Setpoints or actual values

  Rule for the setpoint reference:

  If the cam is not being processed within the position control cycle, it must be run within the same cycle as the axis.
- Operating the output cam as high-speed output cam (Page 362)



Note

Output cams also take effect at non-homed axes.

Output

Set the "Enable output" check box to assign a HW address and a logic operation to the "Output Cam" technology object. If the output is disabled the output cam can only be evaluated in the software.

HW address

The output cam is assigned only to one output. This may be an integrated digital output of the Technology CPU, a digital output on DP(DRIVE), for example, at an ET200 station, or the output of a TM15 or TM17 High-Feature module.

Several output cams can be connected to the same output.

The switching accuracy of the output cam is determined by:

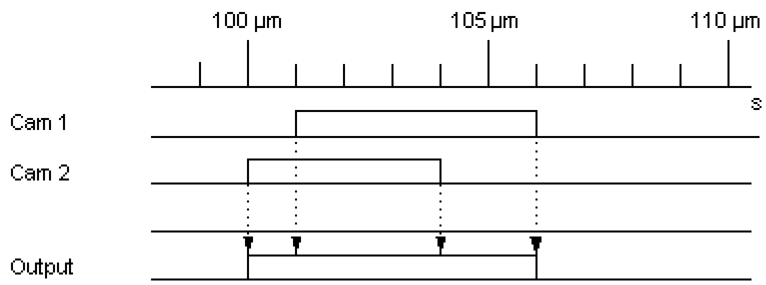
- Output accuracy of the I/O
- Priority of the output cam in the system cycles
- Compensation for constant delay times

Logical operation

You can determine whether to interconnect the output cam with the output by means of logical AND operation or by logical OR operation at the **Output cam > Configuration** dialog box.

That is, all ORed output cams will be grouped and then logically linked at the output with the output cams linked by AND logic.

OR logic with two output cams



4.9.3 High-speed output cams

You can activate the "High-speed output cams" function in the **Output cam > Configuration** dialog box.

Only the integrated outputs of the Technology CPU or the outputs of a TM15 or TM17 High Feature modules support high-speed output cams. Only these are supported by internal hardware functions.

Note

High-speed and normal output cams may not share the same hardware address and bit number. Always use separate addresses

---

Output cam output is performed independent of the processing cycle via interrupt.

The status of output cams are calculated based on the IPO cycle or on the position controller cycle. When using integrated outputs, select the position controller cycle as the processing cycle in the dialog **Output cam > Configuration** if you require up-to-date status displays. When TM15 and TM17 output cams are used, the processing cycle does not affect the status calculations.

4.10 Configuring cam tracks

Before a cam track can be used in the user program, the cam track first has to be inserted in S7T Config as a "Cam track" technology object at an axis or at an external encoder.

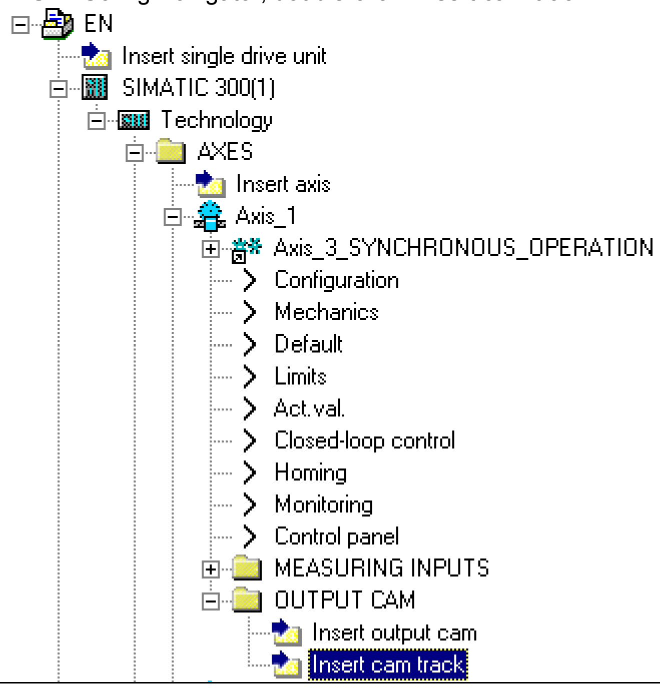
By calling the technology function "MC_CamTrack" in the user program, you can use the cam track configured in S7T Config. The assignment to the cam track created in S7T Config is defined by entering the technology DB number at the *CamTrack* input parameter.

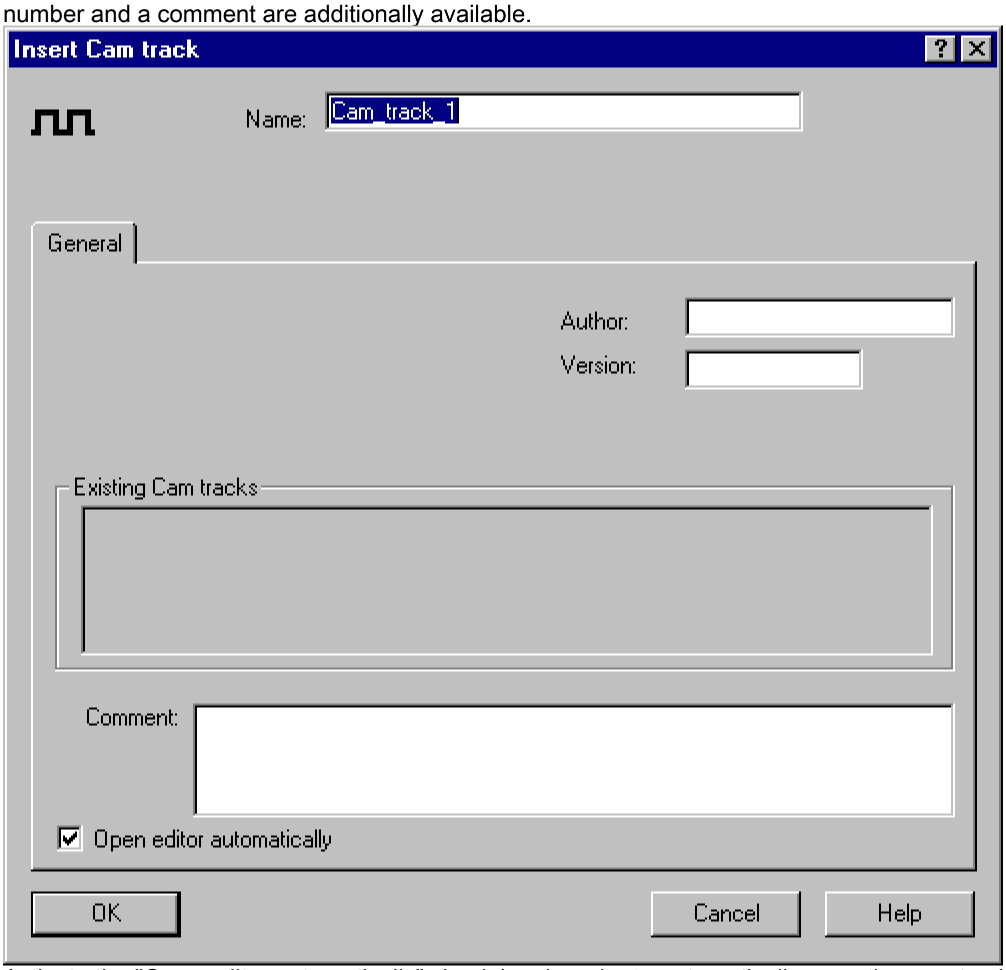
4.10.1 Inserting cam tracks

Requirement

- The data of the Technology CPU were configured in HW Config and compiled.
- An axis or external encoder was inserted in S7T Config.

How to insert a cam track in S7T Config

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1.</td><td>In S7T Config Navigator, double-click "Insert cam track".

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

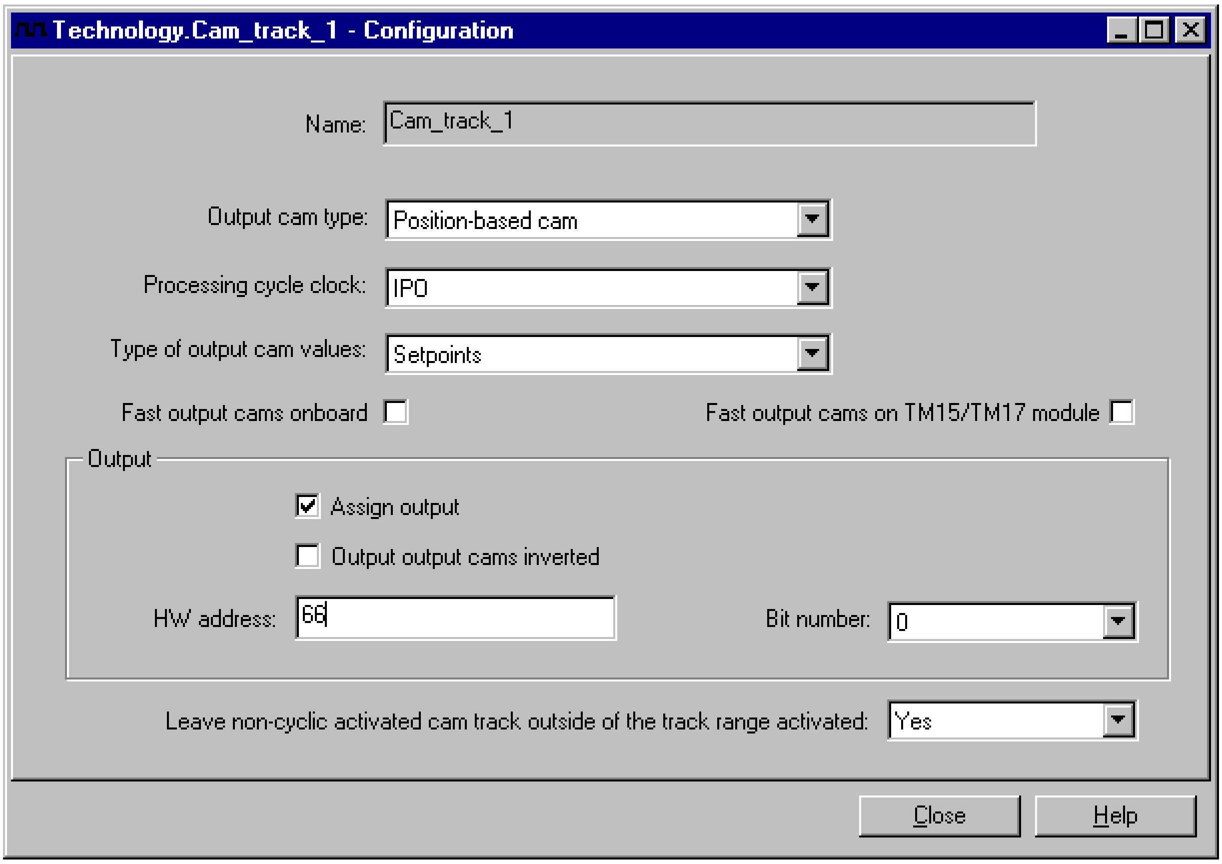
Step	Description
2.	Enter the name of the technology object in the "Insert output cam" dialog box. Fields for the author, the version number and a comment are additionally available.  Activate the "Open editor automatically" check box in order to automatically open the cam track configuration dialog box:
3.	Click "OK" to confirm your entries. Result: The technology object is inserted.

Further procedures

Configure the cam track in the **Cam track > Configuration** dialog box.

4.10.2 Configuration - Cam track

The **Cam track > Configuration** dialog box can be used to set the following parameters for the "Cam track" technology object:



- Name:

  Shows the name of the technology object defined in the S7T Config Navigator.
- Output cam type:

  Specify the type of the individual output cam here (position-based cam, time-based cam or time-based cam with maximum ON length). The set cam type applies to all the individual output cams of the cam track.
- Execution cycle:

  The execution cycle is used to specify the cycle in which the cam track is to be executed (IPO, IPO2 or servo clock / position controller cycle).
- Type of output cam value:

  Specify whether the setpoints or the actual values of the axis or of the external encoders are to be used for the current output cam switching points.

  Rule for the setpoint reference: If the cam track is not being processed within the position control cycle, it must be run within the same cycle as the axis.
- Integrated high-speed output cams / high-speed output cams on TM15/TM17-Modul

  Select one of the two check boxes if you want to switch the output of the cam track as high-speed output cams (Page 362). You can select between high-speed output cams of the integrated I/Os of the technology CPU and high-speed output cams of a TM15/TM17-Modul.

Release

If the ""Assign output" check box is activated, further selection possibilities are available:

- Output output cam inverted

  Select this check box if the switching state of the cam track is to be output in inverted form to a hardware output. This setting does not affect the switching state of the individual output cams.
- HW address / bit number

  Enter the hardware address and the bit number of the hardware output of the cam track in these fields. A single output is assigned to a cam track. This may be an integrated digital output of the Technology CPU, a digital output on DP(DRIVE), for example, at an ET200 station, or the output of a TM15 or TM17 High-Feature module.
- Non-cyclic activated cam track can be activated outside the track range.
  - Drop-down list box "Yes"

    The cam track remains active if the axis or the external encoder travels beyond the cam track. If the axis or the external encoder then travels back into the range of the cam track, the configured cams switch again. Use the technology function "MC_CamTrack" in the *Mode = 1*, to lock and/or deactivate the cam track.
  - Drop-down list box "No"

    The cam track is deactivated if the axis or the external encoder travels beyond the cam track. A re-entry into the cam track range does not trigger any switch processes of the cam track.

Note

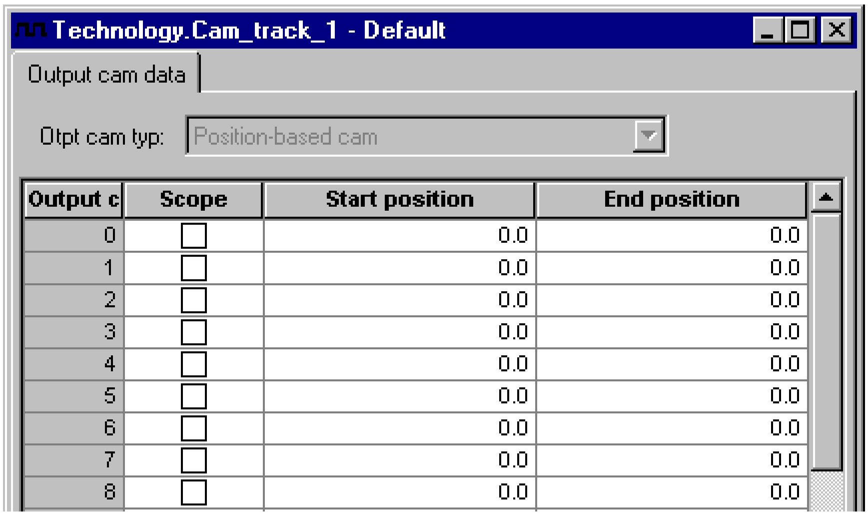
Cam tracks also take effect at non-homed axes.

4.10.3 Default - cam track

Specify the output cam data of the individual output cams of your cam track in the **Cam track > Default** dialog box.

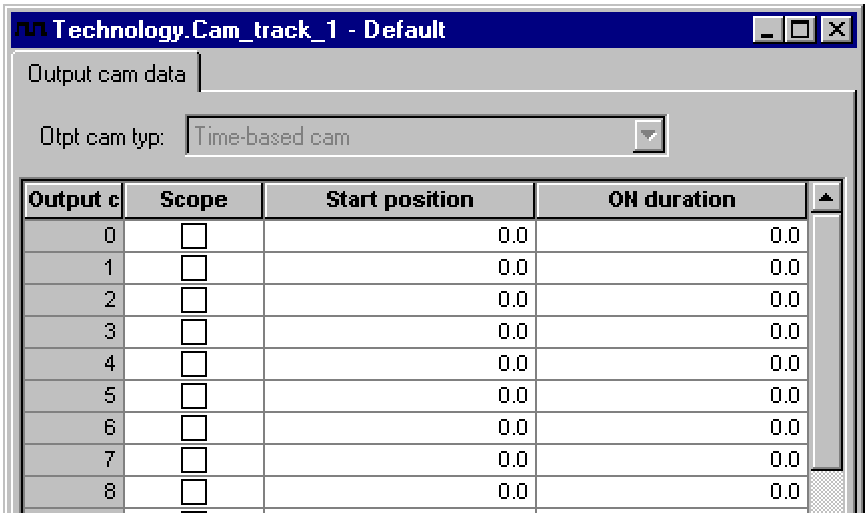
Depending on the cam type different input fields are available for the individual output cams 0 to 31:

Position-based cam



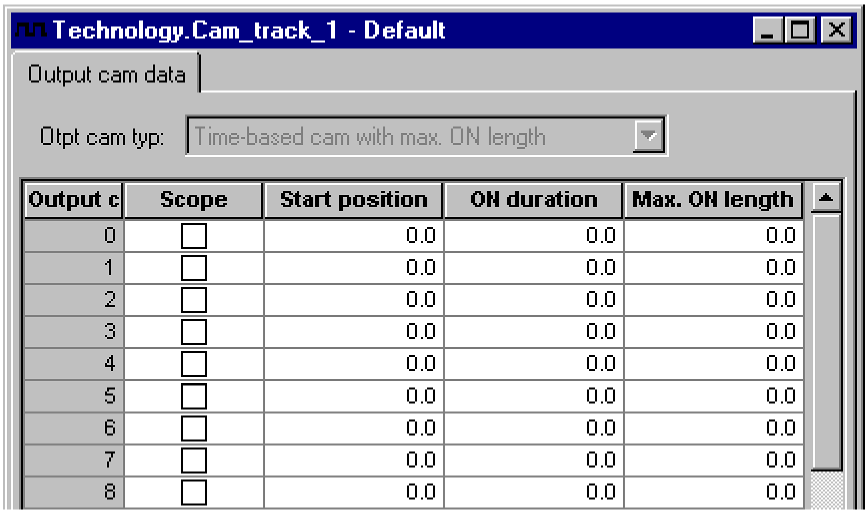
- Validity
  If the check box is activated, the individual output cam is taken into consideration when editing the cam track. Set individual output cams to invalid if the individual output cam is not required or lies outside the track length.
- Start position
  Define the start position of the individual output cam, referenced to the cam track beginning, in this input box.
- End position
  Define the end position of the individual output cam, referenced to the cam track beginning, in this input box. If the direction of movement is positive, the individual output cam switches on when the start position is reached and off again at the end position. If the direction of movement is negative, the individual output cam switches on when the end position is reached and off again at the start position.

Time-based cam



- Validity
  If the check box is activated, the individual output cam is taken into consideration when editing the cam track. Set individual output cams to invalid if the individual output cam is not required or lies outside the track length.
- Start position
  Define the start position of the individual output cam, referenced to the cam track beginning, in this input box.
- ON time
  Define the ON time of the individual output cam in this input box. The individual output cam is switched on when the start position is reached switched off again when the ON time expires.

Time-based output cam with maximum ON length



- Validity
  If the check box is activated, the individual output cam is taken into consideration when editing the cam track. Set individual output cams to invalid if the individual output cam is not required or lies outside the track length.
- Start position
  Define the start position of the individual output cam, referenced to the cam track beginning, in this input box.
- ON time
  Define the ON time of the individual output cam in this input box.
- Max. ON length
  Specify a maximum ON length for the individual output cam in this input box. The individual output cam switches on when the start position is reached. After the ON time has expired, or after the ON length has been exceeded, the individual output cam switches off again.

Note

To read out the cam data of the cam track, use the technology function "MC_ReadCamTrackData" in the user program. The technology function "MC_WriteCamTrackData" can be used to write the modified cam data back to the cam track.

4.11 Configuring measuring inputs

Any measuring input deployed in the user program must be inserted in S7T Config as a "Measuring input" technology object.

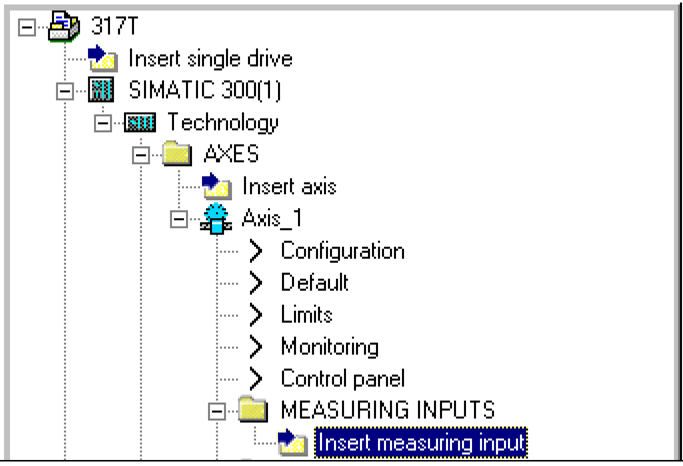
You program the measuring inputs within the user program using the "MC_MeasuringInput" technology function. These functions are used to enable and disable the measuring input according to the various operating mode settings, and define its measurement range. The assignment to the measuring input created in S7T Config is defined by entering the technology DB number at the *MeasuringInput* input parameter.

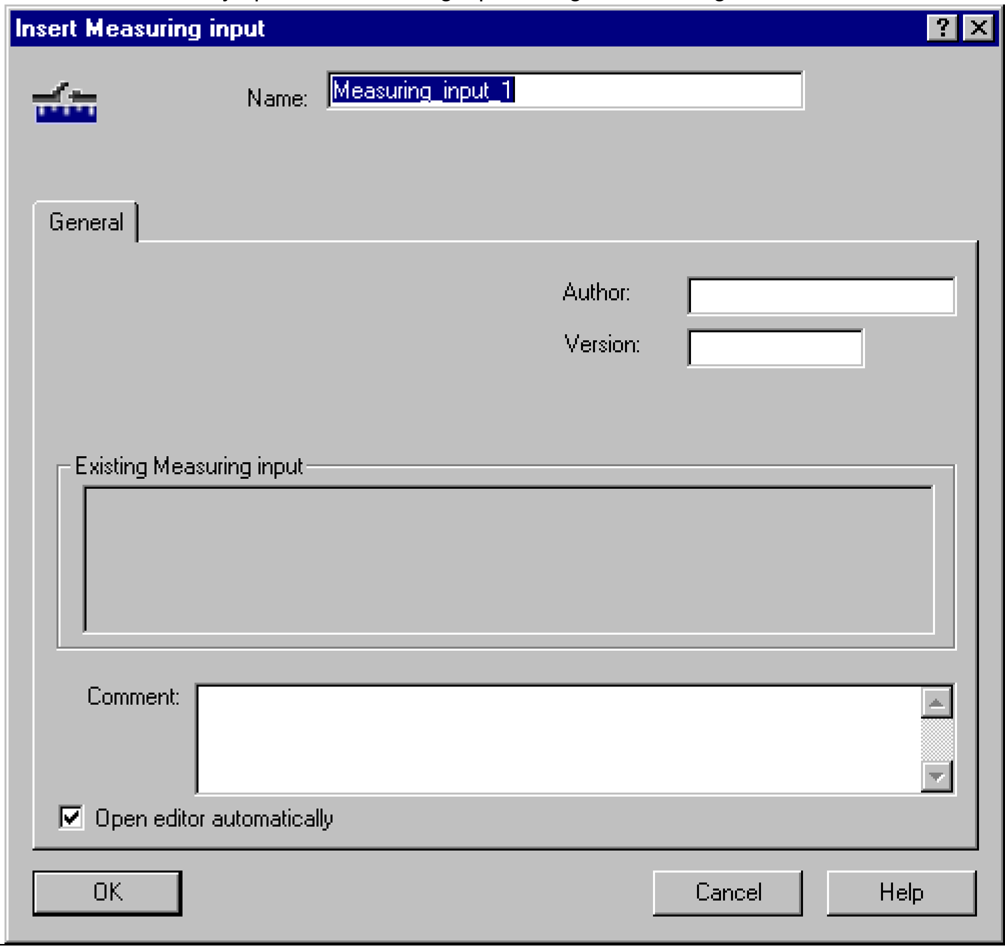
4.11.1 Inserting a measuring input

Prerequisite

- The data of the Technology CPU were configured in HW Config and compiled.
- An axis or external encoder was inserted in S7T Config.

How to insert a measuring input in S7T Config

Step	Description
1.	In the S7T Config navigator, double-click "Insert measuring input". 

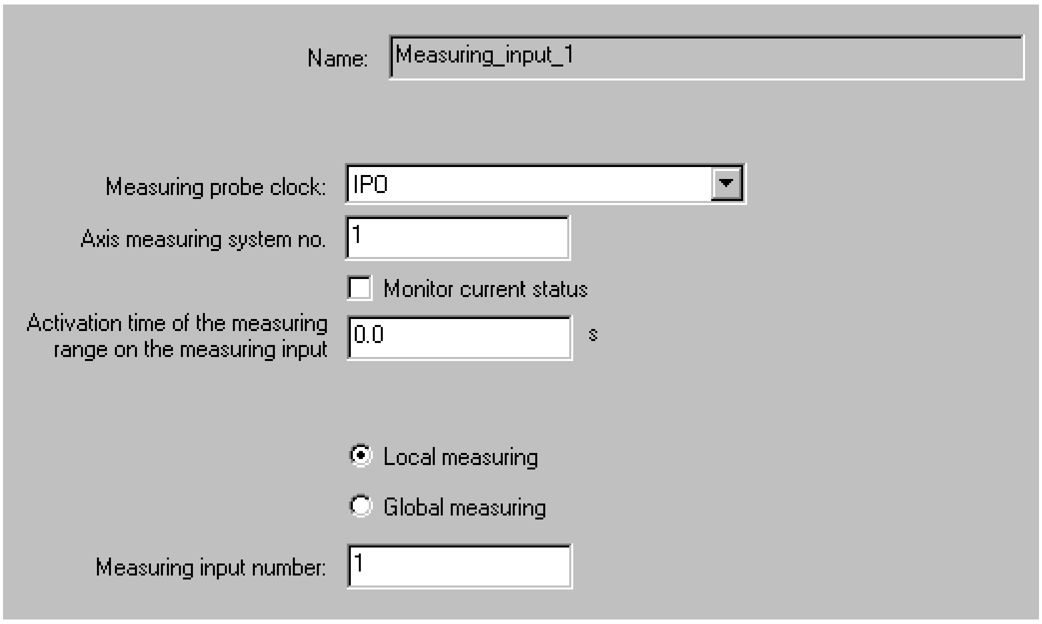
<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>2.</td><td>Enter the name of the technology object in the "Insert cam measuring input" dialog box. Fields for the author, the version number and a comment are additionally available. Set the "Open editor automatically" check box in order to automatically open the measuring input configuration dialog box:

</td></tr>
<tr><td>3.</td><td>Click "OK" to confirm your entries.
Result: The technology object is inserted.</td></tr>
</table>

Further procedures

Configure the measuring input in the **Measuring Input> Configuration** dialog box.

4.11.2 Configuration - Measuring input

Set the following properties of the "Measuring input" technology object in the **Measuring input > Configuration** dialog box:



Measuring input clock:

Use the drop-down list box to select the system cycle during which the measurement result is interpolated and processed.

Axis measuring system no.:

Enter the number of the used encoder system in this input box. "Encoder system 1" is used by default. An encoder system can be assigned to several measuring inputs.

Monitor current status:

If the check box is activated, short pulses (shorter than the position control cycle) will be suppressed at the measuring input.

If *Mode = 1* (measuring with positive edge) is selected at the "MC_MeasuringInput" technology function, the measuring input is not activated until the input of the measuring input has had the signal state *0* for at least one position control cycle.

Activation time of the measuring range at the measuring input:

Enter an activation time for the activation and deactivation of the active range here in seconds. This time can be used, for example, to compensate the runtimes during activation at the DP(DRIVE) and at the drive. The accuracy of the activation depends on the position control cycle clock.

Local measuring:

Select the "Local measuring" option if the encoder system and the input of the measuring input are located at the same rive or the same system.

- **Measuring input number:** (Input box is visible when "Global measuring" is used)
  Enter the number of the measuring input at the drive or at the system here.

Global measuring:

Select the "Global measuring" option if the input of the measuring input is located at a Terminal Module TM15/TM17.

- HW address: (Input box is visible when "Global measuring" is used)
  Enter the hardware address of the Terminal Module TM15 / TM 17 here.
- Bit number: (Drop-down list box is visible when "Global measuring" is used)
  Select the bit number of the input of the measuring input used respectively here.

Note

The measuring input must be located either on the drive component used, or on a TM15/TM17 High Feature module. Other digital inputs can not be used as measuring inputs. The measuring input can only be connected to the drive component containing the encoder input. If the encoder input is connected to SIMODRIVE, for example, you can only connect the measuring input to IM 174/ADI4 if the IM 174/ADI4 is also connected to SIMODRIVE.

4.12 Configuring external encoders

Encoders supported by the "External encoders" technology object:

- Analog absolute encoder (sensor analog) wired to an analog input module of ET 200M or ET 200S
- Incremental encoder (rectangular TTL) wired to analog drive interface IM 174/ADI4
- Absolute encoder (SSI) wired to analog drive interface IM 174/ADI4
- Incremental or absolute encoders wired to the encoder input of a DP drive
- Absolute encoder SIMODRIVE Sensor Isochron (message frame 81)

The hardware of any external encoders applied in the user program must be configured in HW Config, and the external encoder must be inserted as "External encoder" technology object in S7T Config. For information on configuring the encoder system in HW Config, refer to the description of the drive or of the SIMODRIVE sensor.

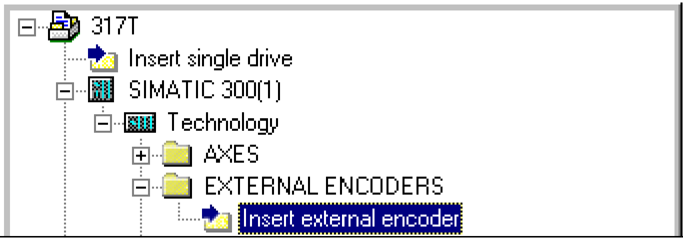
You program the measuring inputs within the user program using the "MC_ExternalEncoder" technology function. This function allows you to enable and disable the external encoder and the homing settings. You define the reference to the external encoder created in S7T Config by setting the technology DB number at the *Axis* input.

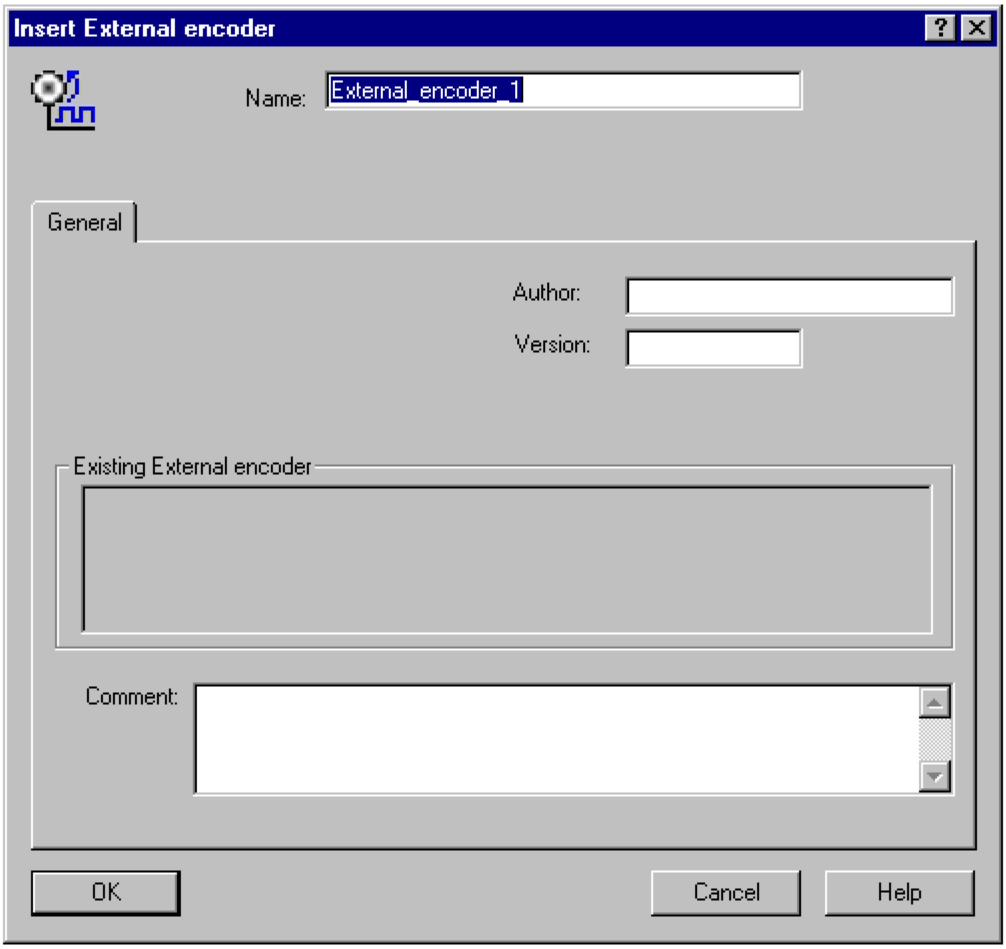
4.12.1 Inserting external encoders

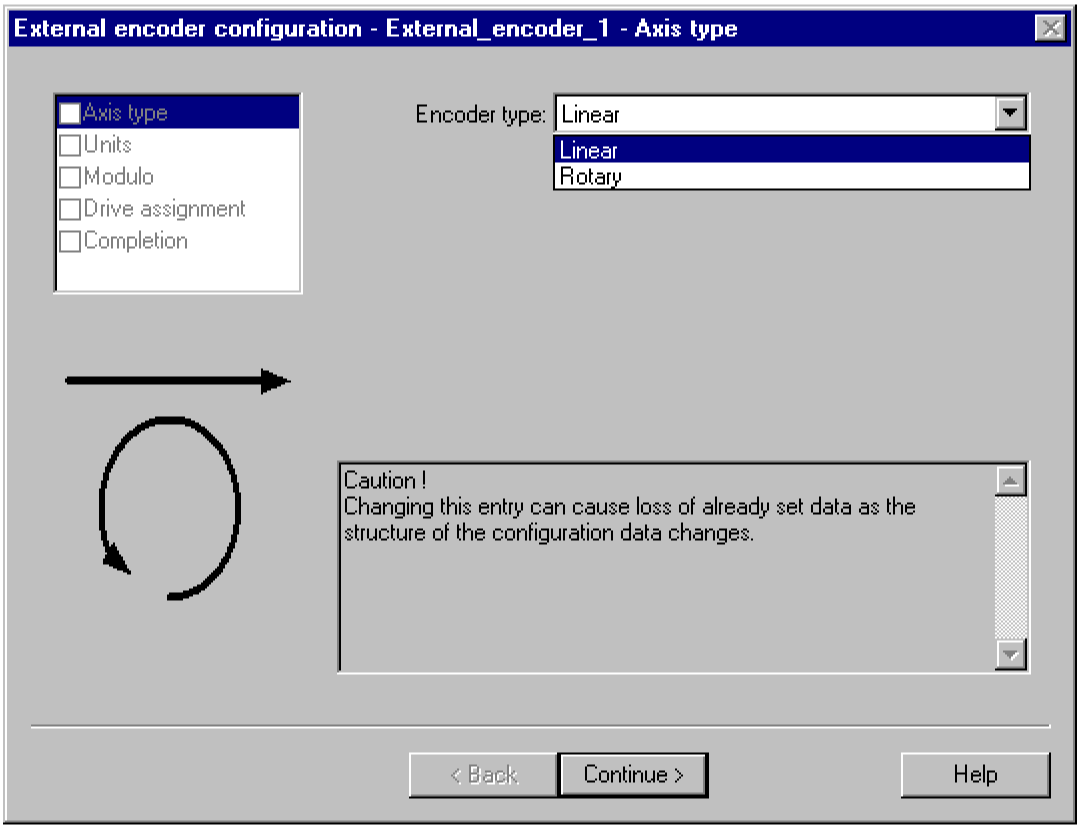
Requirement

- The Technology CPU was configured in HW Config
- You configured a drive component with a free encoder system, or a PROFIBUS encoder SIMODRIVE sensor in HW Config. A PROFIBUS encoder SIMODRIVE sensor must be operated in "isochronous mode". For details, refer to the product information, or to the encoder documentation.
- The hardware configuration was compiled in HW Config and saved.

How to insert an external encoder in S7T Config

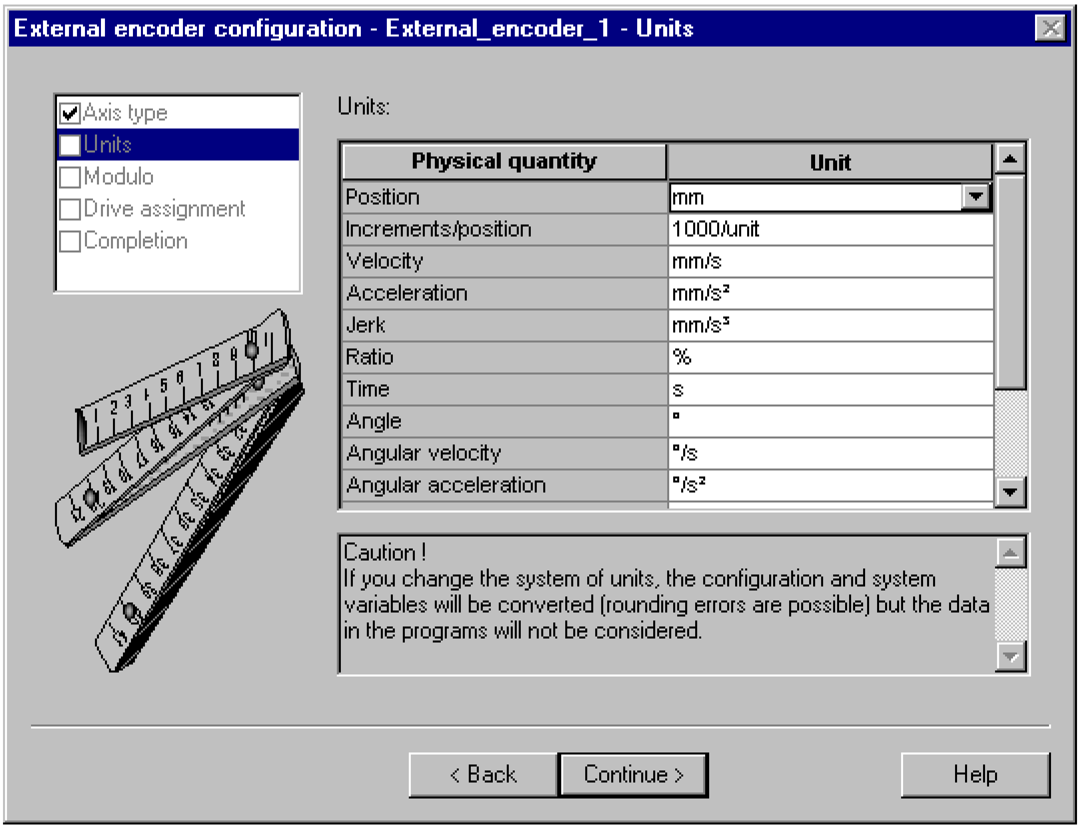
Step	Description
1.	Open the Navigator of S7T Config, and then double-click "Insert external encoder". 

Step	Description
2.	On the next dialog box, type in the technology object name, and optionally include the name of the author, the version number and a comment.  Click "OK" to confirm your entries.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>3.</td><td>The "External encoder configuration" wizard opens.
Select the encoder type "Linear" or "Rotary":


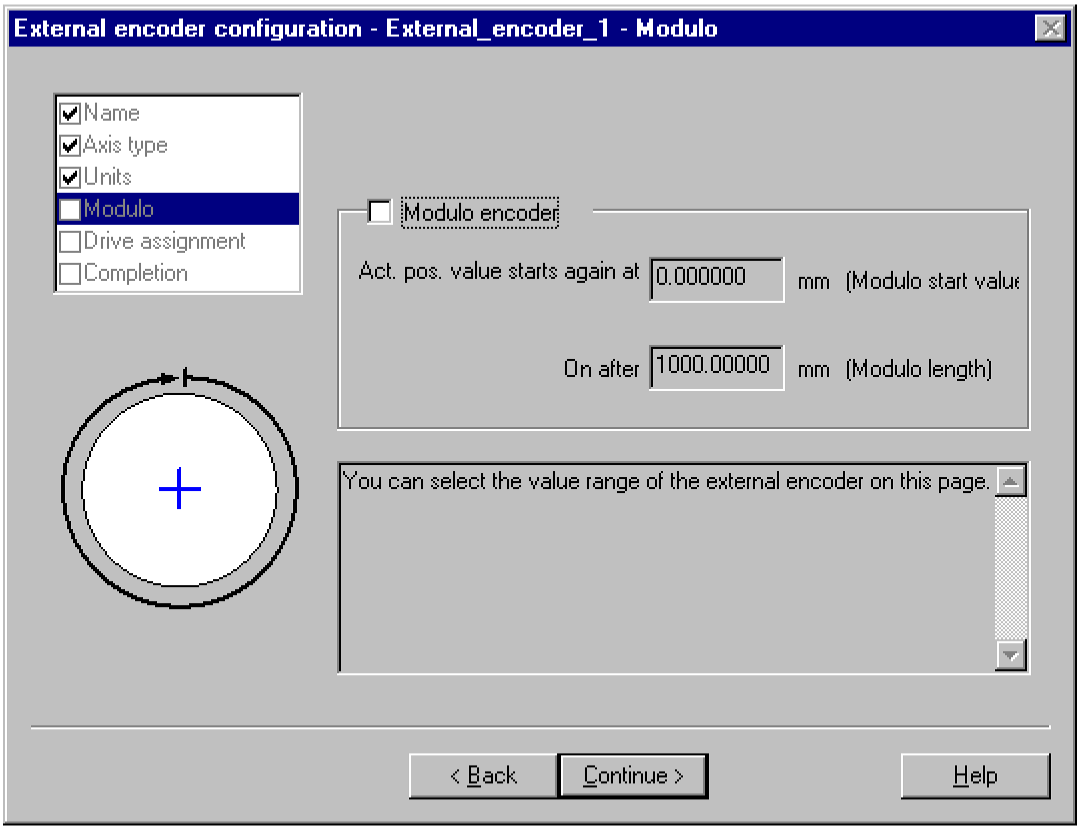
Confirm your entry by clicking "Continue".</td></tr>
</table>

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>4.</td><td>
The "External encoder configuration - Units" dialog box opens.
Select the units, for example "degrees".

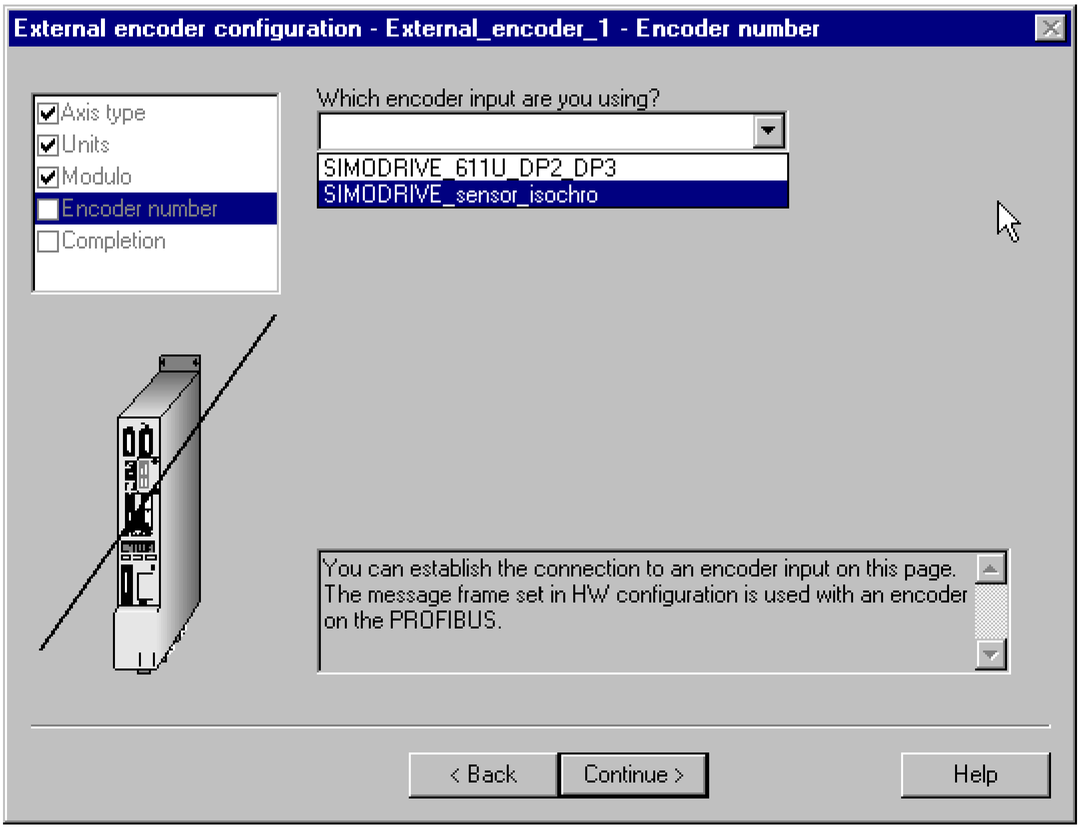


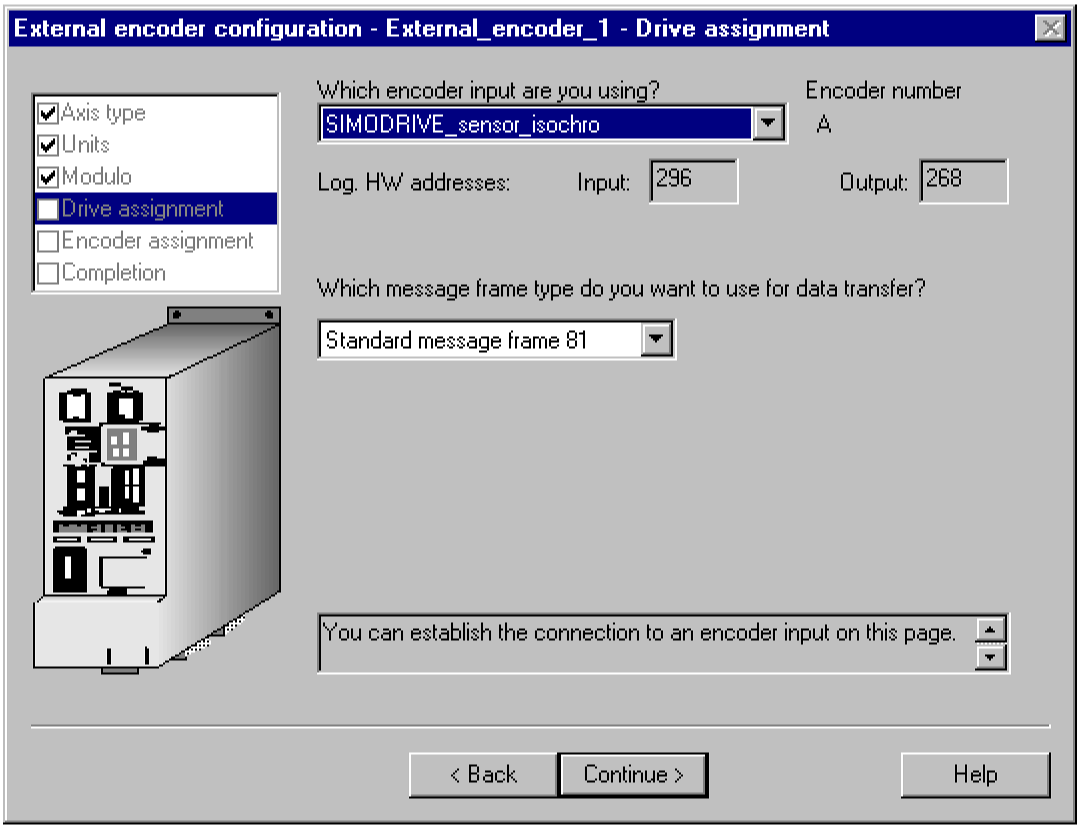
Confirm your entries by clicking "Continue".
</td></tr>
</table>

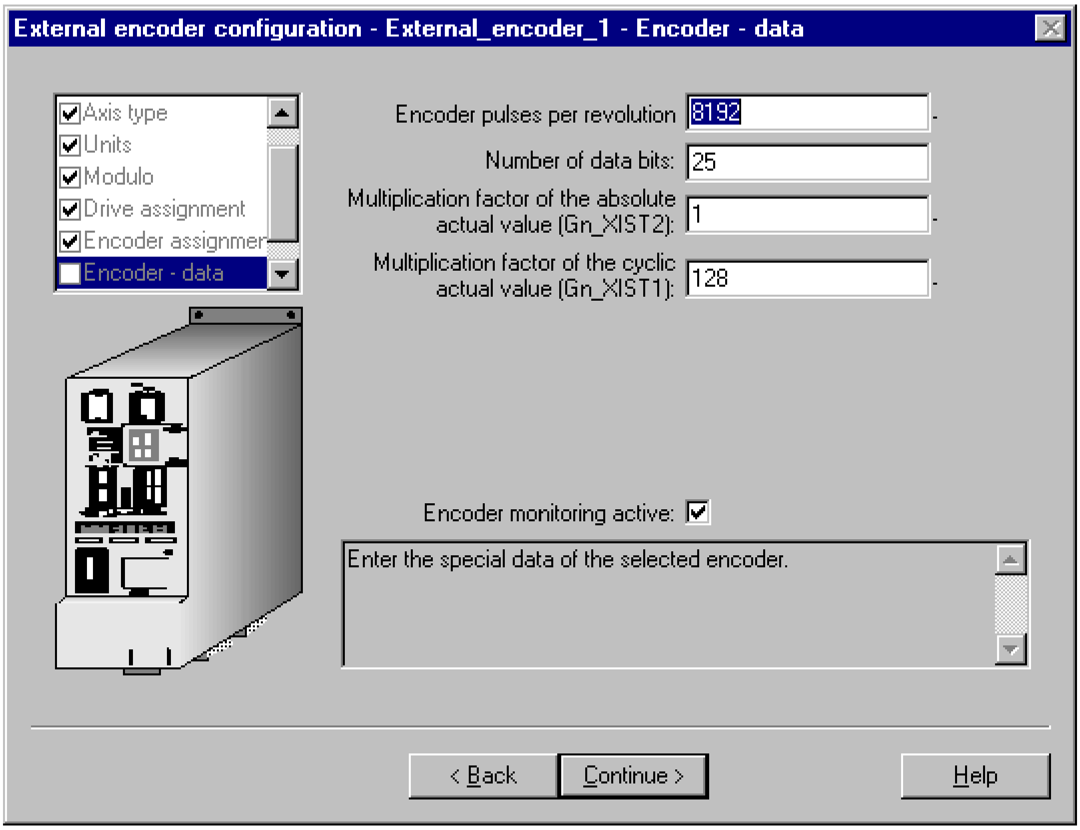
<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>5.</td><td>The "External encoder configuration - Modulo" dialog box opens.



If you want to use the external encoder as a modulo encoder, meaning that its position values are referenced to a traversing range that is defined by the modulo start value and the modulo length, activate the check box. Enter the values for the modulo start value and the modulo length.</td></tr>
<tr><td></td><td>The address information will be read from HW Config.</td></tr>
</table>

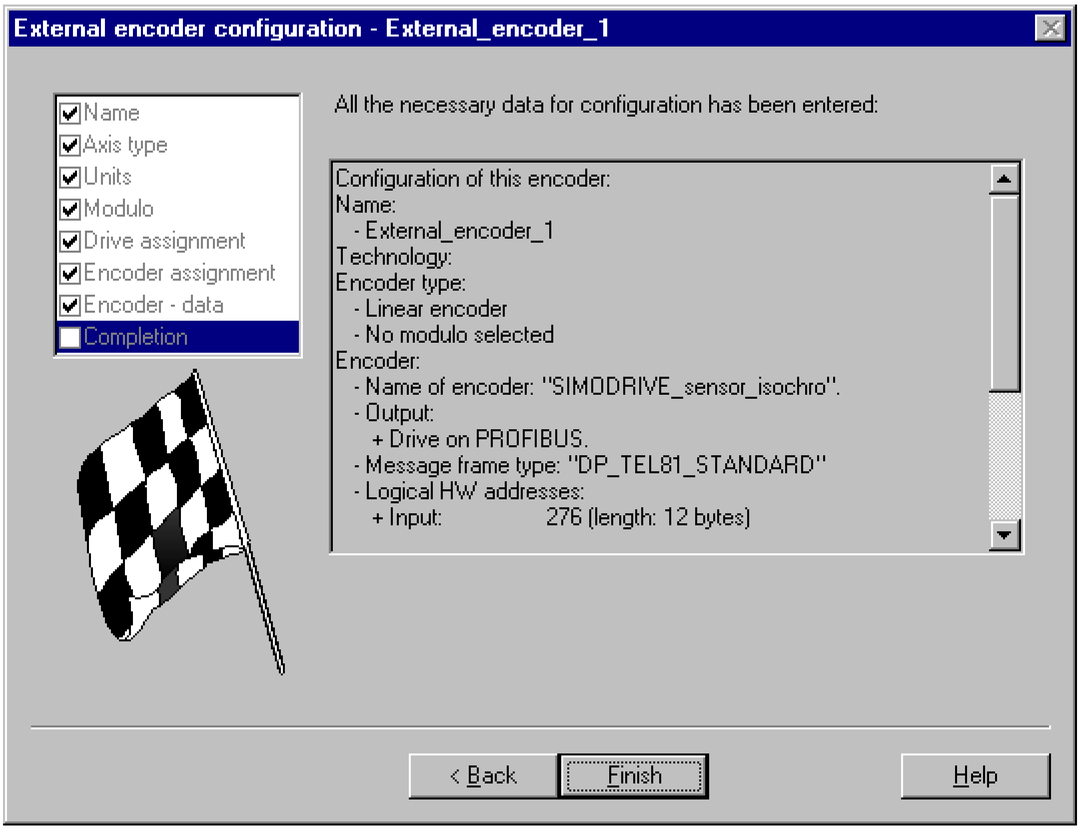
Step	Description
6.	The "External encoder configuration - Encoder number" dialog box is opened if several encoder systems are available. If you configure only one drive or one encoder system in HW Config, the dialog box shown in step 7 is opened immediately.  From the drop-down list, select the encoder system, or the drive component to which the encoder system is connected, that is to be used as an external encoder.

Step	Description
7.	After the encoder has been selected, the dialog box outputs additional information and a drop-down list for the selection of the message frame type.  Message frame type: The message frame type selected for the encoder system of the drives must correspond with the setting in HW Config (see alsoSelecting the message frame type (Page 113)). • SIMODRIVE sensor settings When using the PROFIBUS encoder SIMODRIVE sensor, always select "Standard telegram 81" as the message frame type. • IM 174/ADI4 settings Always select the "Standard message frame 3" type for IM 174/ADI4. The "Encoder number" parameter is displayed in a drop-down list box in the selection dialog for IM 174/ADI4. Select the encoder number you programmed for this encoder at IM 174/ADI4 from this drop-down list box. Encoder type / mode and measuring system: From the "Encoder type", "Encoder mode" and "Measuring system" drop-down list boxes, select the settings used to configure the drive component or encoder. For details, refer to the supplementary descriptions of SIMODRIVE 611 universal, MASTERDRIVES MC, IM 174/ADI4 and SIMODRIVE Sensor, or to the device manuals. Example of settings at the SIMODRIVE Sensor: • Encoder type: "Absolute encoder cyclic absolute" • Encoder mode: "SSI" • Measuring system: "Rotary encoder system" Confirm your entries by clicking "Continue".

<table>
<tr><th>Step</th><th colspan="4">Description</th></tr>
<tr><td rowspan="7">8.</td><td colspan="4">The "External encoder configuration - Encoder data" dialog box opens. The content of this dialog box depends on the selected encoder type. The diagram shows the dialog box, for example, for the SIMODRIVE Sensor encoder.


IM 174/ADI4 settings
Enter the encoder data at IM 174/ADI4 which you configured in HW Config.
Settings of the encoder systems on SIMODRIVE, MASTERDRIVES, etc.
Enter the same encoder data you set in the drive configuration.
SIMODRIVE sensor settings
Enter the following values for the SIMODRIVE sensor:
</td></tr>
<tr><td colspan="2"></td><td>Single-turn encoder</td><td>Multiturn encoder</td></tr>
<tr><td rowspan="4">Parameters</td><td>Encoder lines</td><td>8192</td><td>8192</td></tr>
<tr><td>Multiplication factor of the absolute actual value</td><td>1</td><td>1</td></tr>
<tr><td>Multiplication factor of the cyclic actual value</td><td>524288</td><td>128</td></tr>
<tr><td>Number of data bits</td><td>13</td><td>25</td></tr>
<tr><td colspan="4">Confirm your entry by clicking "Continue".</td></tr>
</table>

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>10.</td><td>The next dialog box below shows a summary of the configuration.

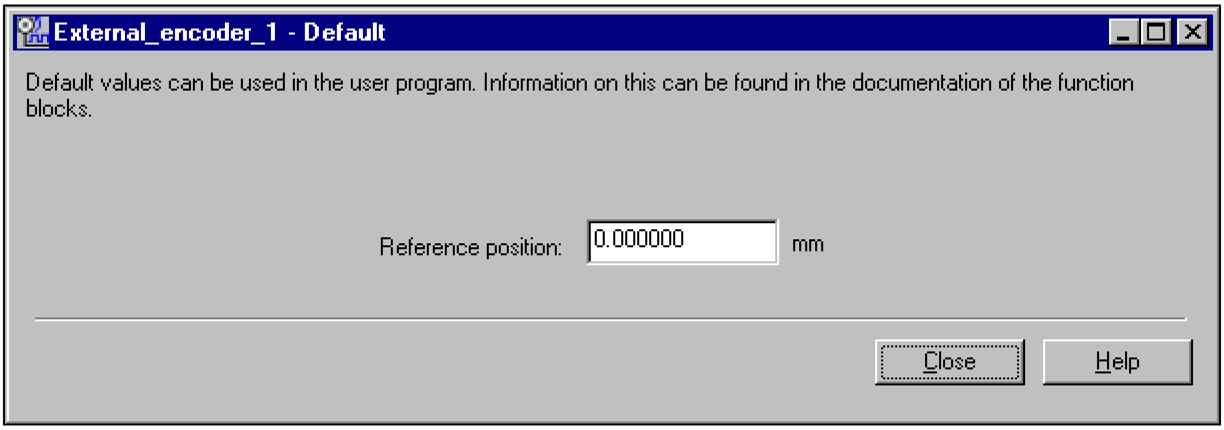


Close the wizard by clicking "Apply".

Result: The technology object will be inserted.</td></tr>
</table>

4.12.2 Defaults - External encoder

Open the **External encoder > Default** dialog box to enter the **Reference position** of the external encoder. Use the reference position to define the position within the coordinate system of the "External encoder" technology object. This position setting can be used for synchronization using the "MC_ExternalEncoder" technology function in Mode *4* and *5*.



4.12.3 External encoder - Synchronization with incremental encoders

The Technology CPU supports various synchronization modes for external incremental encoders. You can set the reference position of the external encoder in the Expert List of S7T Config.

During synchronizing of the encoder the value of the reference point coordinate is assigned to the current encoder position:

- After the encoder zero mark is reached (default setting)
- On reaching the external zero mark (set in the expert list)

The homing mode of the external encoder can be programmed in the expert list at the *TypeOfAxis.Encoder_1.IncHomingEncoder.passiveHomingMode* parameter.

Options:

- *CAM_AND_ZM_PASSIVE(1)*

  Passive homing mode, with reference cam and next encoder zero mark
- *ZM_PASSIVE(2)*

  Passive homing mode, with next encoder zero mark
- *CAM_PASSIVE(3)*

  Passive homing mode, with external zero mark
- *DEFAULT_PASSIVE(4)*

  Either the *ZM_PASSIVE(2)* (with encoder zero mark) or the *CAM_PASSIVE(3)* (without encoder mark) homing modes are used, depending on the encoder type.

In addition, set the expected reference point approach direction at configuration parameter *TypeOfAxis.Encoder_1.IncHomingEncoder.passiveApproachDirection*.

Options available:

- *APPROACH_NEGATIVE_PASSIVE(0)*

  Homing with approach to encoder zero mark in negative direction of movement.
- APPROACH_POSITIVE_PASSIVE(1)

  Homing with approach to encoder zero mark in positive direction of movement.
- EDGE_POS_SIDE_NEG_PASSIVE(2)

  Homing with approach to external zero mark in positive direction of movement.
- EDGE_POS_SIDE_POS_PASSIVE(3)

  Homing with approach to external zero mark in negative direction of movement.
- EDGE_NEG_SIDE_NEG_PASSIVE(4)

  Homing start after the external zero mark was passed in positive direction of movement.
- EDGE_NEG_SIDE_POS_PASSIVE(5)

  Homing start after the external zero mark was passed in negative direction of movement
- ACTUAL_DIRECTION_PASSIVE(6)

  Homing at the next edge of the external zero mark or encoder zero mark

The selected approach direction depends on the homing mode. The table below shows the relationship between the approach direction and homing mode:

Approach direction	Homing mode			
	CAM_AND_ZM_ PASSIVE(1)	*ZM_ PASSIVE(2)*	*CAM_ PASSIVE(3)*	*DEFAULT_ PASSIVE(4)*
APPROACH_NEGATIVE_PASSIVE(0)	X	X	F	Gx
APPROACH_POSITIVE_PASSIVE(1)	X	X	F	Gx
EDGE_POS_SIDE_NEG_PASSIVE(2)	F	F	X	Gy
EDGE_POS_SIDE_POS_PASSIVE(3)	F	F	X	Gy
EDGE_NEG_SIDE_NEG_PASSIVE(4)	F	F	X	Gy
EDGE_NEG_SIDE_POS_PASSIVE(5)	F	F	X	Gy
ACTUAL_DIRECTION_PASSIVE(6)	X	X	N	Gx
X - Valid configuration F - Invalid configuration N - Function not supported Gx - Encoder with zero mark (TTL, sine/cosine) Gy - Encoder without zero mark (Endat)				

Note

Note that certain drive components do no support the reference point modes listed above. For further details refer to the corresponding documentation.

Homing mode *DEFAULT_PASSIVE(4)* supports the selection of the approach direction according to the encoder mode setting in configuration data. You can only select approach directions *0, 1, 6* if you define an encoder with zero mark in your configuration. You can only select the approach directions *2, 3, 4 and 5* if you define an encoder without zero mark. Any other combination will lead to configuration errors.

If homing mode *CAM_AND_ZM_PASSIVE(1)* is selected, the reference cam can be wired directly to a hardware input of the Technology CPU (for example, at address *66*), or to an input of the distributed I/O on DP(DRIVE). The address is set at the following configuration data:

- Bit address

  TypeOfAxis.Encoder_1.IncHomingEncoder.passiveBitNumberBero
- Byte address

  TypeOfAxis.Encoder_1.IncHomingEncoder.passiveLogAdressBero

If the *CAM_PASSIVE(3)* homing mode is selected, the external zero mark must be wired to the relevant drive component of the corresponding axis.

4.12.4 Setting the standstill signal

The standstill signal for the external encoder can be set in the configuration data in the Expert List of S7T Config:

- The "Velocity threshold standstill signal" is defined in configuration parameter

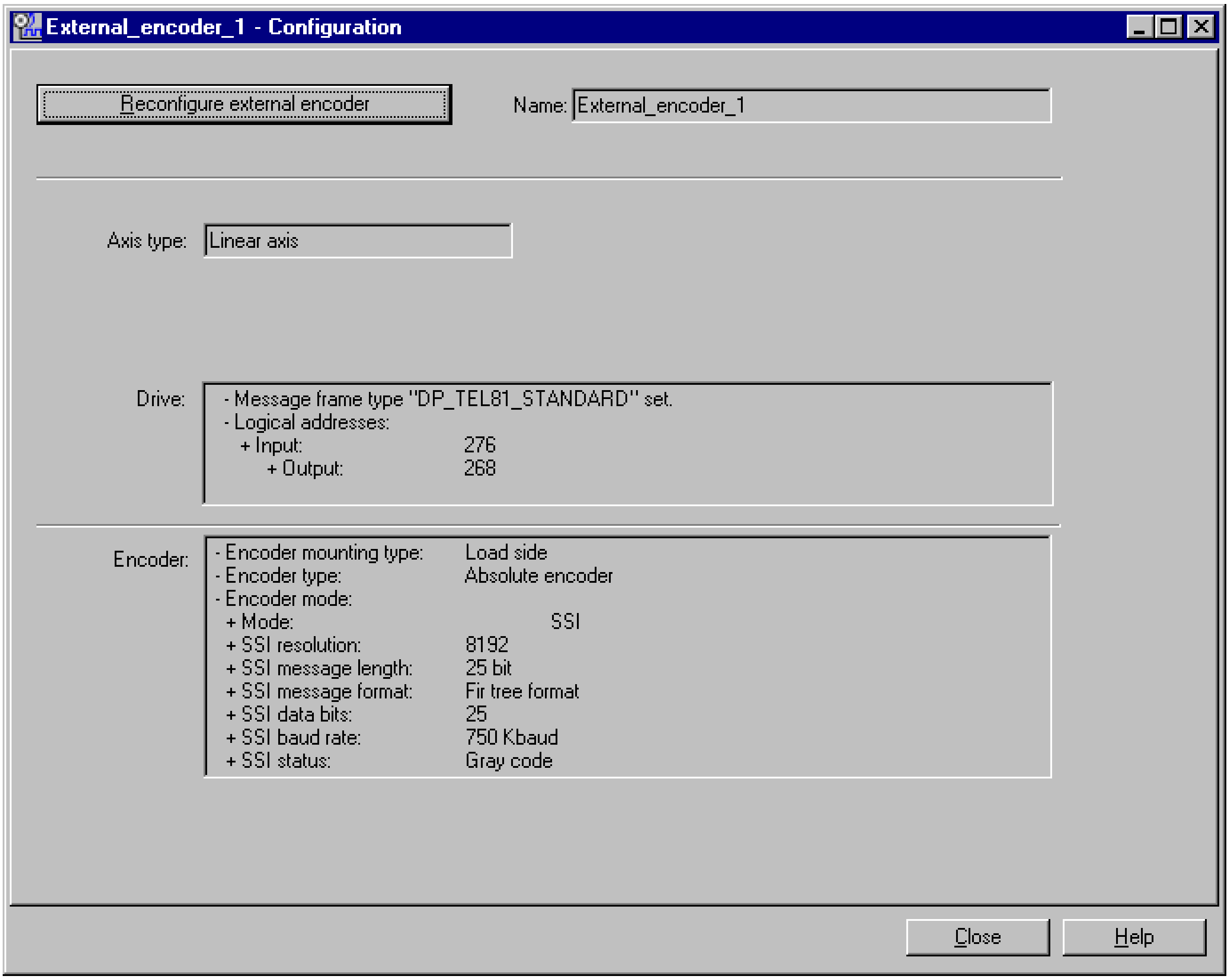
  TypeOfAxis.StandStillSignal.maxVeloStandStill
- The "Signal output delay" is defined in configuration parameter

  TypeOfAxis.StandStillSignal.delayTimeToActivate

4.12.5 Configuration

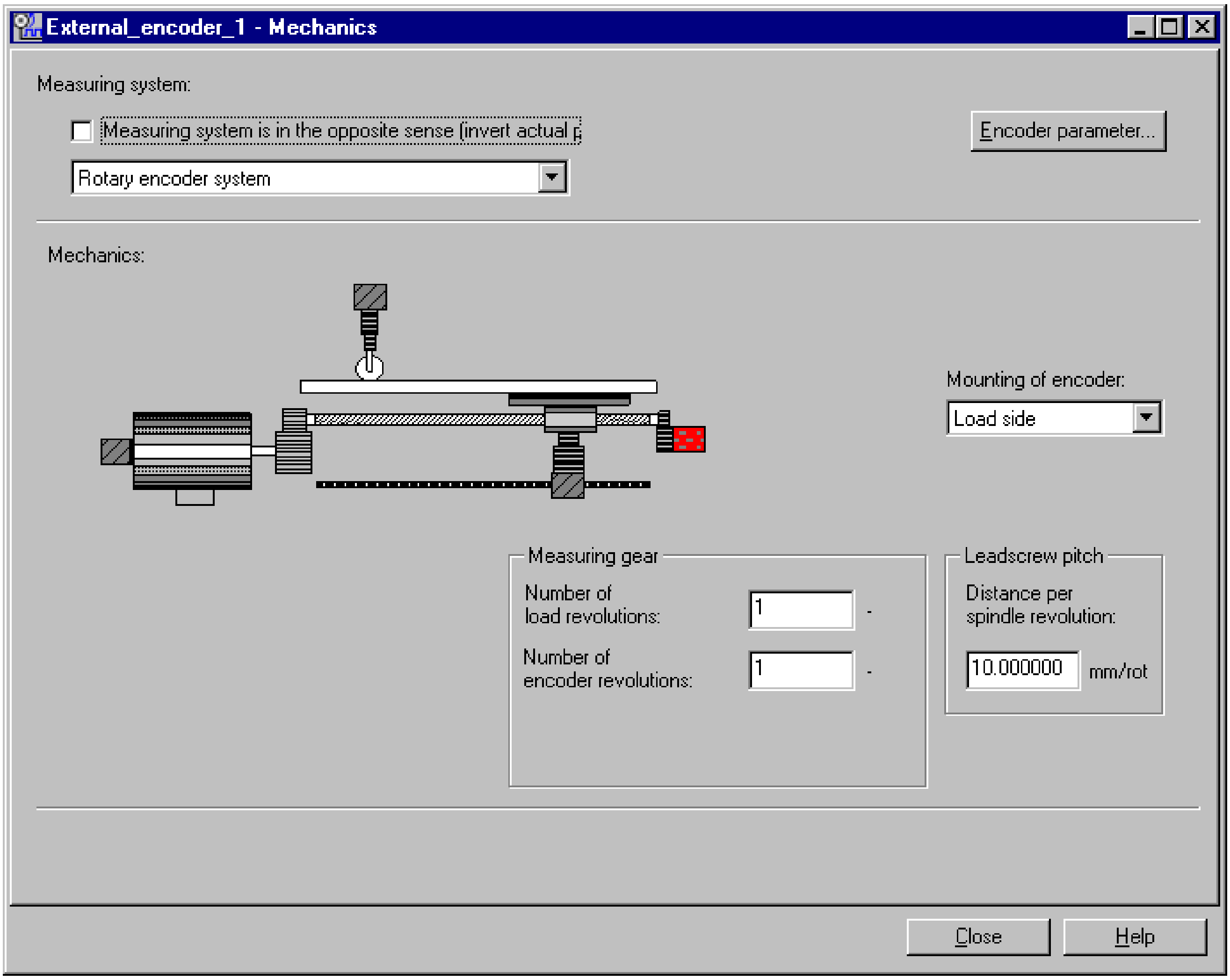
The **External encoder > Configuration** dialog box shows the drive and encoder settings.

Click "Reconfigure external encoder" in order to open the encoder wizard and edit the settings.



4.12.6 Mechanics

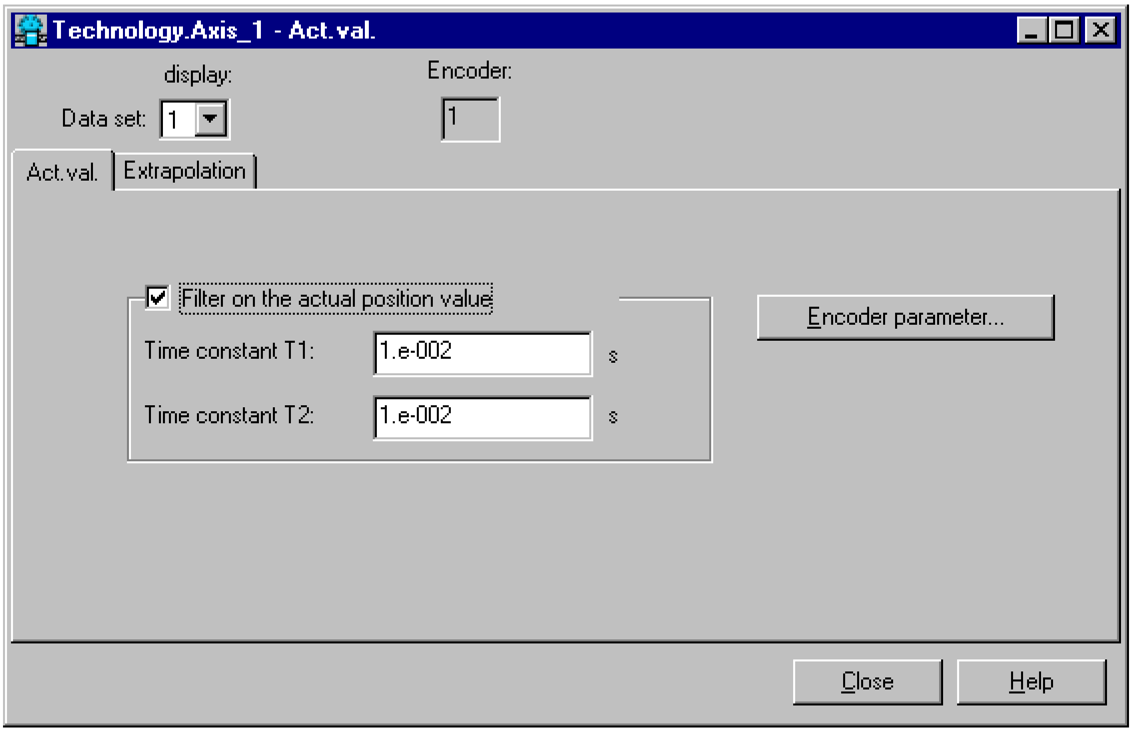
You define the mechanical parameters of the external encoder in the **External encoder > Mechanics** dialog box.



4.12.7 actual value

4.12.7.1 Actual value - "Actual value" tab

Activate actual value filtering in the dialog **Axis / External encoder > Actual value** on the "Actual value" tab and set the appropriate time constants.



Filter on the actual position value

Activate the check box if you would like to activate the filtering of the actual position value.

Time constant T1

Here, you set time constant T1 of the PT2 position filter in the actual value system.

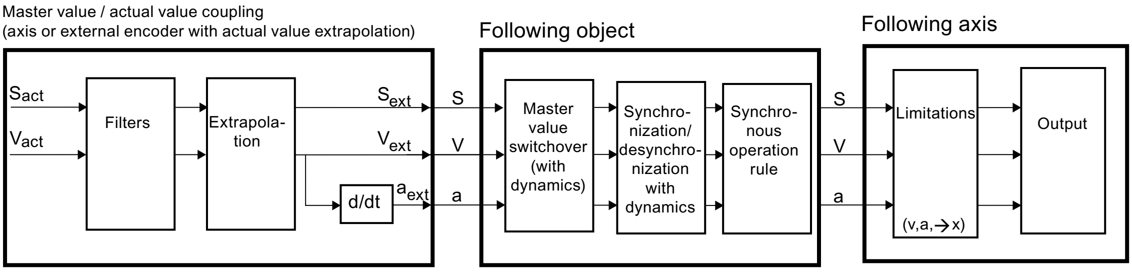
Time constant T2

Here, you set time constant T2 of the PT2 position filter in the actual value system.

"Encoder parameter" button

Opens a dialog box in which the encoder data are displayed.

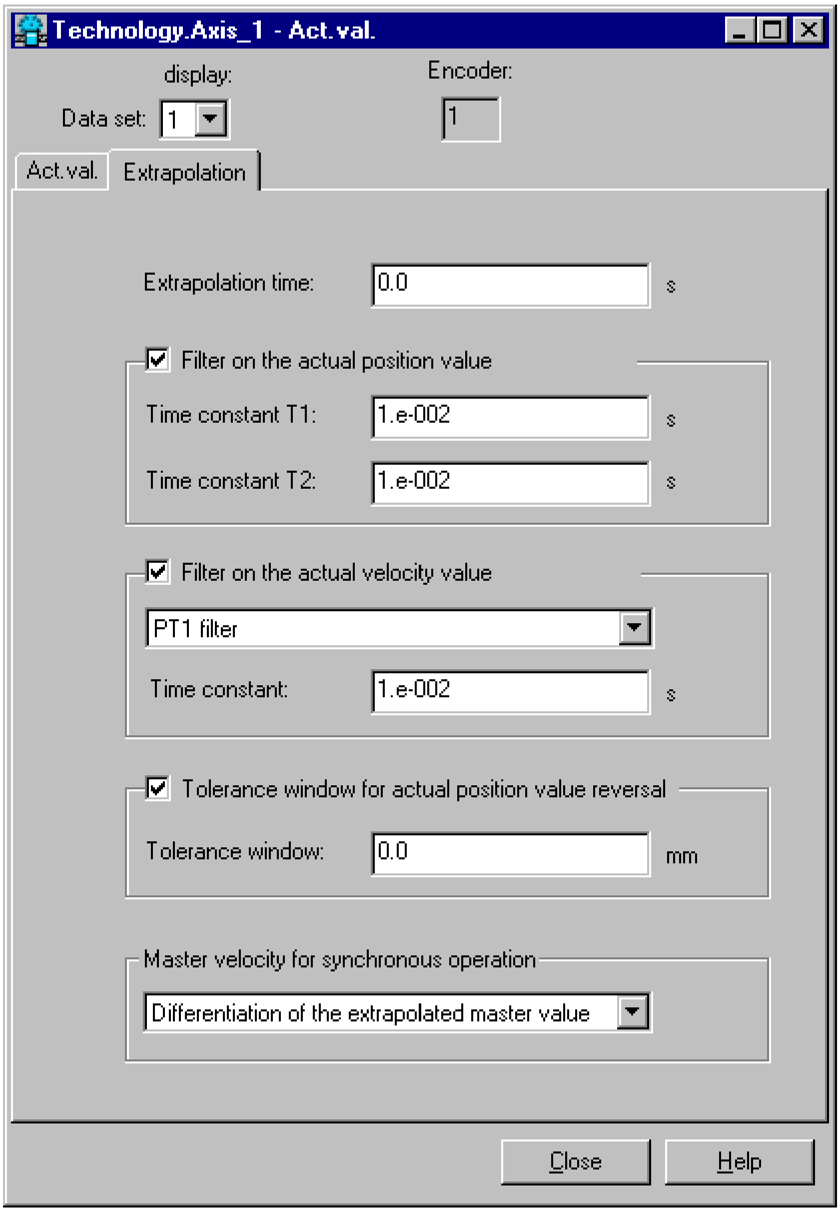
4.12.7.2 Actual value - "Extrapolation" tab



If there is a synchronous operation interconnection within a control, the synchronous operation takes into account the position, velocity, and acceleration of the master value position.

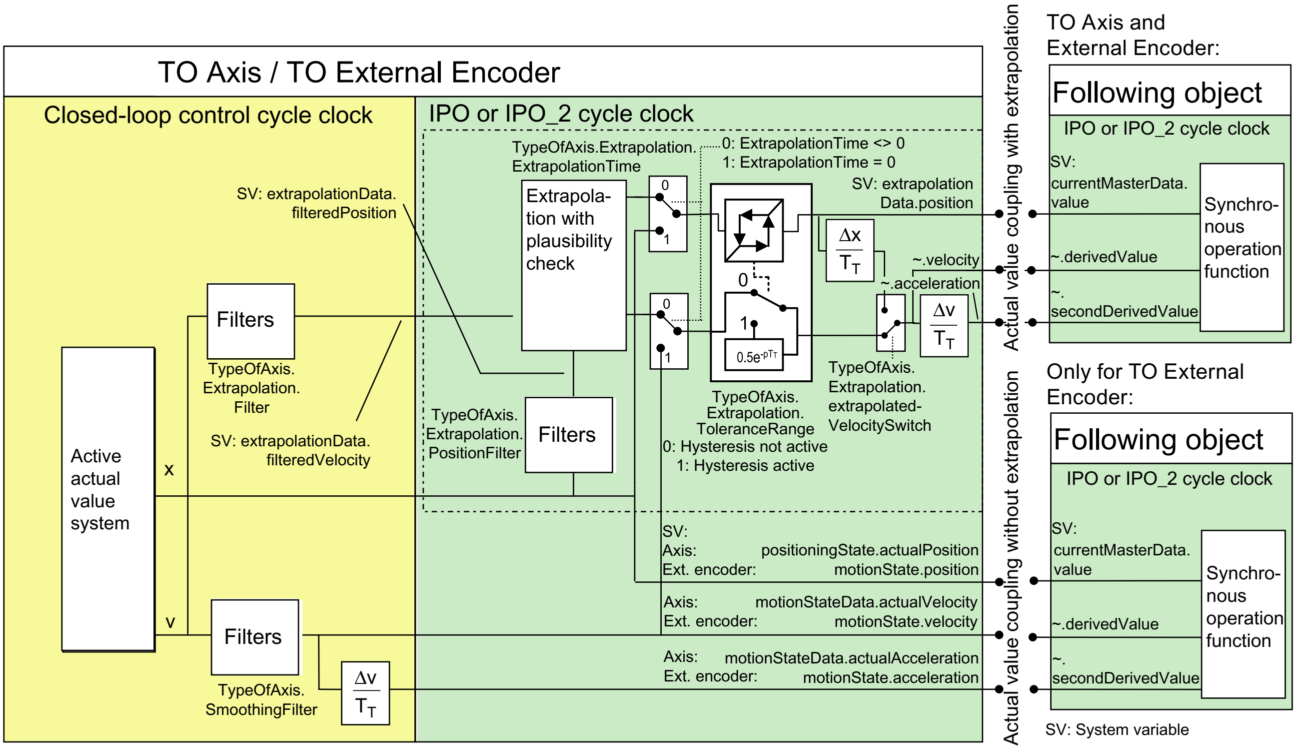
If an actual encoder value is used as the master value, it is useful to extrapolate the measured actual value for the synchronous operation in order to compensate for dead times. Dead times result within the system when measuring actual values, e.g. due to the bus communication and the system processing times.

The extrapolation is set in S7T Config on the leading axis or on the external encoder in the dialog **Axis / External encoder > Actual value** in the "Extrapolation" tab.



The parameters of this dialog can also be read from or written to the user program via technology parameters.

Actual value coupling with extrapolation (axis and external encoder)



Extrapolation time

(Parameter *1110;* configuration data *TypeOfAxis.Extrapolation.ExtrapolationTime*)

Here, you set the time for the extrapolation. No extrapolation if the value entered is *0.0*.

Filter on the actual position value

(Parameter *1130* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.enable*)

Activate the check box if you would like to extrapolate the actual position value.

Time constant T1

(Parameter *1131* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T1*)

Here, you set time constant T1 of the PT2 filter for the extrapolation in the actual value system.

Time constant T2

(Parameter *1132* configuration data *TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T2*)

Here, you set time constant T2 of the PT2 position filter in the actual value system.

The filter acts on the actual position for the extrapolation. The velocity for the extrapolation is taken over from the actual values of the axis or external encoder before application of the smoothing filter (*TypeOfAxis.smoothingFilter*).

Filter on the actual velocity value

(Parameter *1112* configuration data *TypeOfAxis.Extrapolation.Filter.enable*)

Activate the check box if you would like to extrapolate the actual velocity value.

Here, you select the filter for the extrapolation velocity in the drop-down list.

(Parameter *1111* configuration data *TypeOfAxis.Extrapolation.Filter.Mode*)

Time constant

(Parameter *1113* configuration data *TypeOfAxis.Extrapolation.Filter.timeConstant*)

Here, you enter the time constant for the filter.

The position is extrapolated based on the filtered or averaged velocity value. Averaging is via the "time constant".

Tolerance window for actual position value inversion

(Parameter *1114* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.enable*)

Here, you can activate the tolerance window for actual position value inversion

Tolerance window

(Parameter *1115* configuration data *TypeOfAxis.Extrapolation.ToleranceRange.Value*)

Enter the size of the tolerance window here.

If the master value is superimposed with high-frequency noise signals that the synchronous operation cannot follow, this can cause the dynamic response limits to be exceeded or the master value to briefly change directions during synchronization.

In this case, define a tolerance window to prevent the dynamic response limits from being exceeded on the following axis or to prevent direction changes during synchronization.

Master velocity for synchronous operation

(Parameter *1116* configuration data *TypeOfAxis Extrapolation.extrapolatedVelocitySwitch*)

In the drop-down list, select whether the velocity is to be applied for the extrapolation or if the extrapolated master position value is to be differentiated.

Checking the extrapolated and filtered values

The extrapolated and filtered values can be checked in the following system variables:

- *extrapolationdata.position*
- *extrapolationdata.velocity*
- *extrapolationdata.filteredposition*
- *extrapolationdata.filteredvelocity*
- *extrapolationdata.acceleration*

Support of encoders with NIST evaluation

With encoders with NIST evaluation, the speed determined by the encoder and the resulting velocity can be accepted by the encoder. A calculation of the speed and velocity by the integrated technology is not necessary in this case. Two methods of transmission are available:

- **Transmission in the PROFIdrive message frame**

  Setting via the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_PROFIDRIVE_NIST_*

- **Transmission in the I/O area**

  Setting via the configuration data element

  *TypeOfAxis.NumberOfEncoders.Encoder_n.EncoderValueType = POSITION_AND_DIRECT_NIST*In this case, 4000H corresponds to 100%. The address is set in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.logAdress*, and the reference value in the configuration data element *TypeOfAxis.NumberOfEncoders.Encoder_n.nistConfig.referenceValue*.

4.12.8 Homing

4.12.8.1 Homing - "Passive homing" tab

Passive homing can be configured in the "Passive homing" tab of the **Axis > Homing** dialog box of S7T Config. The difference compared to active homing is that the required homing motion is not initiated by a homing command.

Three homing modes are available for "Passive homing":

- Reference cam and encoder zero mark
- External zero mark only
- Encoder zero mark only

Start of the homing function:

MC_Home	*Mode = 2* *Position = x*	The current position is assigned the value of the *Position* input parameter at the reference point.

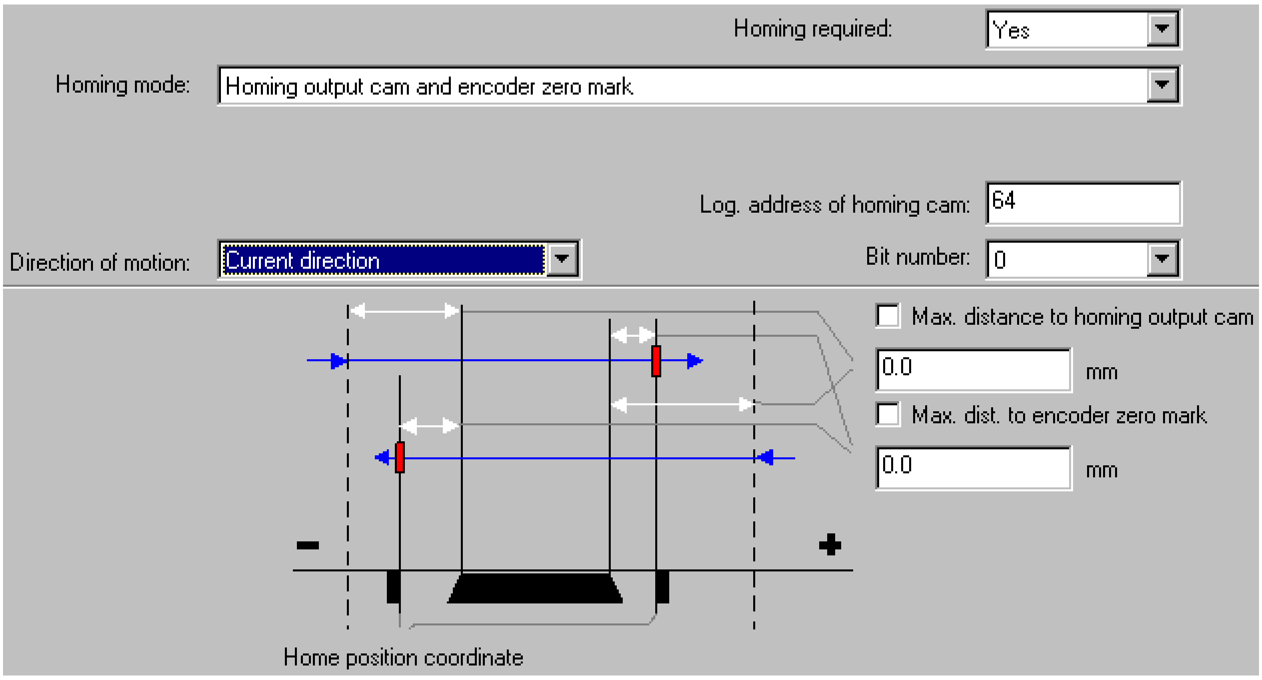
After homing is successfully completed, the axis technology DB returns the status *Statusword.HomingDone= TRUE*.

Note

Note that not all drive components support all homing mode or measured signal evaluation functions. For details, refer to the documentation of the drive components used.

Triggering of encoder zero marks or reference cam monitoring during passive homing operations is indicated by a corresponding error message at the technology DB. The current axis motion is terminated in this case.

Passive homing mode with "Reference cam and encoder zero mark"

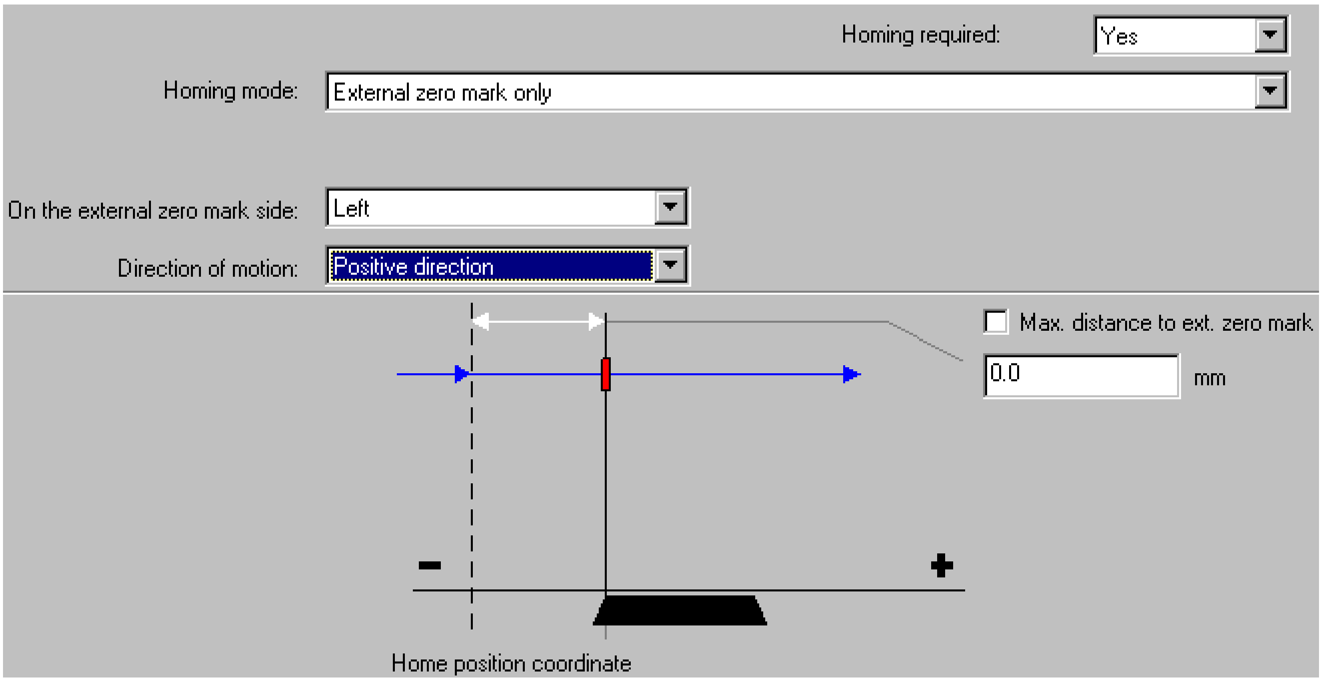


After the axis has passed the reference cam, the next encoder zero mark triggers axis homing. The axis position is set in the reference point to the value defined at the *Position* input parameter of the "MC_Home" technology function.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Reference cam and encoder zero mark"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.
Logical address of reference cam	[byte address]	Logical byte address of the reference cam You can connect the reference cam to the integrated inputs of the Technology CPU, or to the IO on DP(DRIVE).
Bit number	[Number of the bit]	Bit address of the signal used for the reference cam
Maximum distance to homing output cam	deactivated	The distance to reference cam detection is not monitored
	Activated	Monitoring of the distance between the start of the homing function and detection of the reference cam. If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	Monitoring of the distance an axis travels between the reference cam and the encoder zero mark If the difference in this distance exceeds the configured value, the corresponding technology DB returns error *801D*. The homing function is canceled.

Passive homing mode with "External zero mark only"



Axis homing starts with the detection of the external zero mark. The axis is set to the value of the *Position* input parameter of the "MC_Home" technology function.

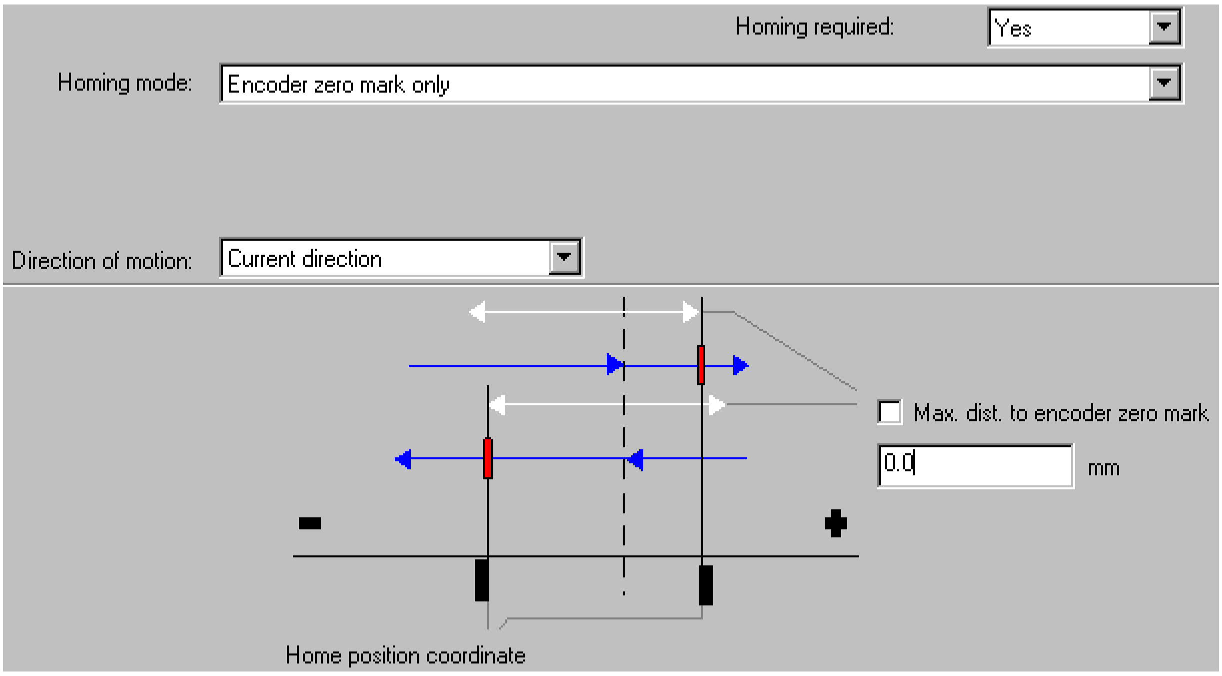
Meaning of the individual parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "External zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the external zero mark.
	Negative direction	The axis is only homed with negative approach to the external zero mark.
	Current direction	The axis is homed when it reaches the next external zero mark.
On the side of the external zero mark	left	Direction of movement: Positive direction Axis homing is triggered at the positive edge. Direction of movement: Negative direction Axis homing is triggered at the negative edge.
	right	Direction of movement: Positive direction Axis homing is triggered at the negative edge. Direction of movement: Negative direction Axis homing is triggered at the positive edge.
Max. distance to external zero mark	deactivated	Distance to go to the external zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the external zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Note

- For homing with "external zero mark", connect the external zero mark as digital measuring input to the drive component.
- In order to be able to execute the reference point approach in "external zero mark" homing mode as required, you should always set values at "Direction of movement" and "on the side of the external zero mark" which are compatible with drive configuration or functionality. The "on the side of the external zero mark" and "Direction of movement" parameters are not used to configure the measuring function in the drive and only reflect their functionality.
- For information on the configuration of external zero mark detection, refer to the relevant product information or to the drive manuals.

Passive homing mode with "Encoder zero mark only"



Homing mode with "Zero mark only" is used for axes, for example, which are equipped with an encoder that outputs only one zero mark signal within the entire traversing range of the axis

Axis homing starts with the detection of the encoder zero mark. The position value of the axis is set to the value of the *Position* input parameter of the "MC_Home" technology function after the encoder zero mark is detected.

Meaning of the various parameters:

Parameters	Value	Explanation
Homing required	Yes	The axis must be homed in order to execute absolute motion commands.
	No	The axis does not have to be homed in order to execute absolute motion commands.
Homing mode	-	In this case: "Encoder zero mark only"
Direction of motion	Positive direction	The axis is only homed with positive approach direction to the encoder zero mark.
	Negative direction	The axis is only homed with negative approach direction to the encoder zero mark.
	Current direction	The axis is homed when it reaches the next encoder zero mark.
Maximum distance to encoder zero mark	deactivated	Distance to go to the encoder zero mark is not monitored
	Activated	The function monitors the distance an axis travels between the start of its homing function and detection of the encoder zero mark If the difference in this distance exceeds the configured distance the corresponding axis technology DB indicates error *801D*. The homing function is canceled.

Passive homing mode with "Default"

When a new axis is created in S7T Config, the *Default* homing mode is preset.

Homing mode with *encoder zero mark only* is used if the configured encoder provides a zero mark. Homing mode external zero mark only is used if the encoder does not provide a zero mark.

4.13 Copying the configuration data from another station

The next steps show how you can reuse the HW and technology configuration data and the user program in a second station of the same project.

Conditions for the next actions:

- The HW configuration of the source station is successfully compiled and saved
- The configuration data of the technology in S7T Config are compiled and saved in the source project
- SIMATIC Manager is open (S7T Config, HW Config, etc. are closed)

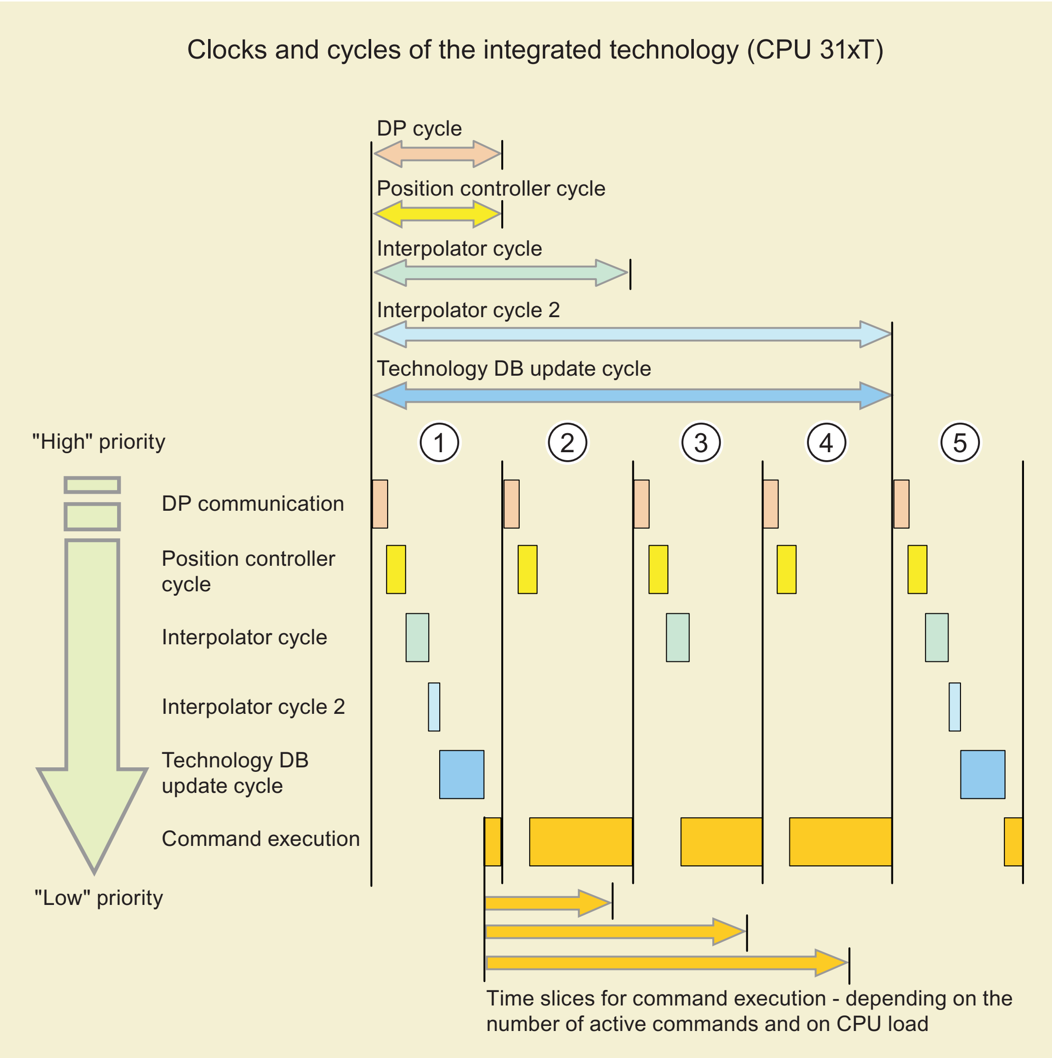
Step	Description
1.	Select the source station in SIMATIC Manager and then select **Edit > Copy.**
2.	Select the destination project, and then select the **Edit > Paste** command. After insertion, a new station with an unambiguous name is shown in the project. All STEP 7 components were copied to this new station.
3.	Open the hardware configuration of the new station. Add it to the configuration of the DP(DRIVE) interface, and of the MPI/DP interface as required, because the logical network links or the networks are not included in the copy operation. You should therefore create new network objects, and then adapt the settings for isochronous operation of the configured drive component.
4.	Save and compile the HW configuration. Close HW Config.
5.	Select "Technology objects" in SIMATIC Manager, and then run "Technology Objects Management" by selecting the **Edit > Open object** command. Adapt the technology DBs you generated as required.

You have now created a new station which contains the data of the source station (save for the difference in names.)

Programming 5

5.1 Time pattern of CPU 31xT

The diagram below shows the cycles of integrated technology, and their influences on job processing of the technology objects for CPU 31xT. The practical example shows a sequence covering several position control cycles.



The diagram represents time requirements of the various cycles in time slice units. The time slice requirements vary, depending on the number of technology objects configured. The time slices of the interpolator cycle, of interpolator cycle 2 and of the update cycle of the technology DBs are not processed in all position control cycles.

DP communication, the position control cycle, the interpolator cycle, interpolator cycle 2 and the Technology DB update cycle are executed with descending priority.

Command execution utilizes the interval between the position control cycles. The smaller this interval, the more time is required for command execution.

Cycles selected in the example shown:

- Position control cycle = *1*

  Corresponds with a 1:1 ratio between the DP cycle and the position control cycle
- Interpolator cycle = *2*

  Corresponds with a 1:2 ratio between the DP cycle and the position control cycle
- Interpolator cycle 2 = *2*

  Corresponds with a 1:2 ratio between the interpolator cycle and interpolator cycle 2
- Update cycle of the technology DBs = multiple of the interpolator cycle

  Factor 2 in the example shown

The cycles shown are synchronized with the DP cycle, or with DP communication.

- DP cycle

  The DP cycle determines the intervals for data exchange with the DP I/O on DP(DRIVE). Data are exchanged within the DP communication time slice. The constant bus cycle time at the DP(DRIVE) must be activated at the corresponding parameters in HW Config. The integrated technology in combination with the isochronous components on DP(DRIVE) form the basis of isochronous operation.
- Position control cycle

  The system also computes the axis position control within the time slice of the position control cycle. The time requirements of this time slice remain near constant while the Technology CPU is in operation.

  The position control cycle clock determines the interval between the position control cycles.

  The value of the position control cycle clock can be set to a multiple of the DP cycle in S7T Config. A clock ratio of 1:1 in order to enhance performance in terms of technology object processing is advisable.
- Interpolator cycle

  The time slice of the interpolator cycle is also used to compute the control variables. The interpolator cycle clock determines the interval between two interpolator cycles. Time slice requirements may increase briefly at the start of one or several commands.
- Interpolator cycle 2

  Interpolator cycle 2 handles the same tasks as the interpolator cycle and can be used for technology objects of lower priority classes.

- **Updating technology DBs**

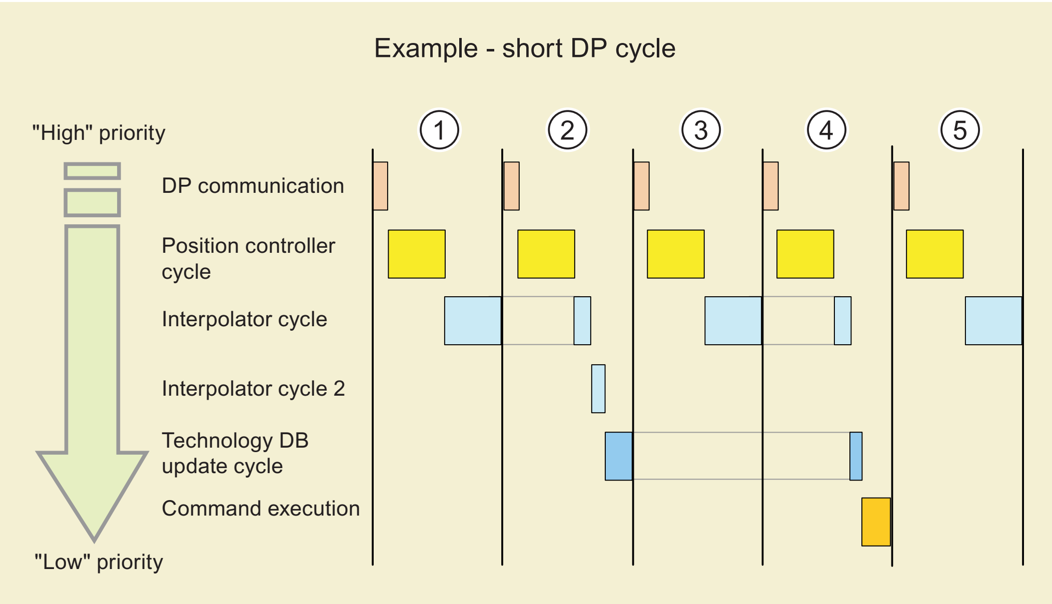
  The technology DBs are updated within the time slice. Integrated technology with firmware version V3.2.x or higher supports the distribution of technology DB updates to different cycles. For further information refer to "Updating the technology data blocks (Page 791)". At firmware versions before V3.2.x all the technology DBs are updated within this same time slice.

  The update cycle of the technology DBs can be set to a multiple of the interpolator cycle in S7T Config. The technology synchronization interrupt OB is called after each update of the Technology DBs.

Command processing

The commands of integrated technology are processed and monitored at indefinite cycles. The time required to process current commands depends on the number of active commands and on CPU load. Both the average and maximum command processing times can be read from the *CmdLoopDuration* and *MaxLoopDuration* tags of the MCDevice technology DB. Integrated technology with firmware version V3.2.x or higher supports the start of axis commands within the interpolator cycle. Subsequent command monitoring operations are again handled by means of the command processing functions of the integrated technology.

The example above shows that cycle 1 only provides a small time slice for command processing. In addition to DP communication and the position control cycle, the system also processes the interpolator cycle and interpolator cycle 2, and the update of the Technology DBs. This method provides additional time for command processing in the next cycles.



The example shows the time profiles with insufficient DP cycle length. The clock ratios as shown in the previous example apply:

- Cycle 1

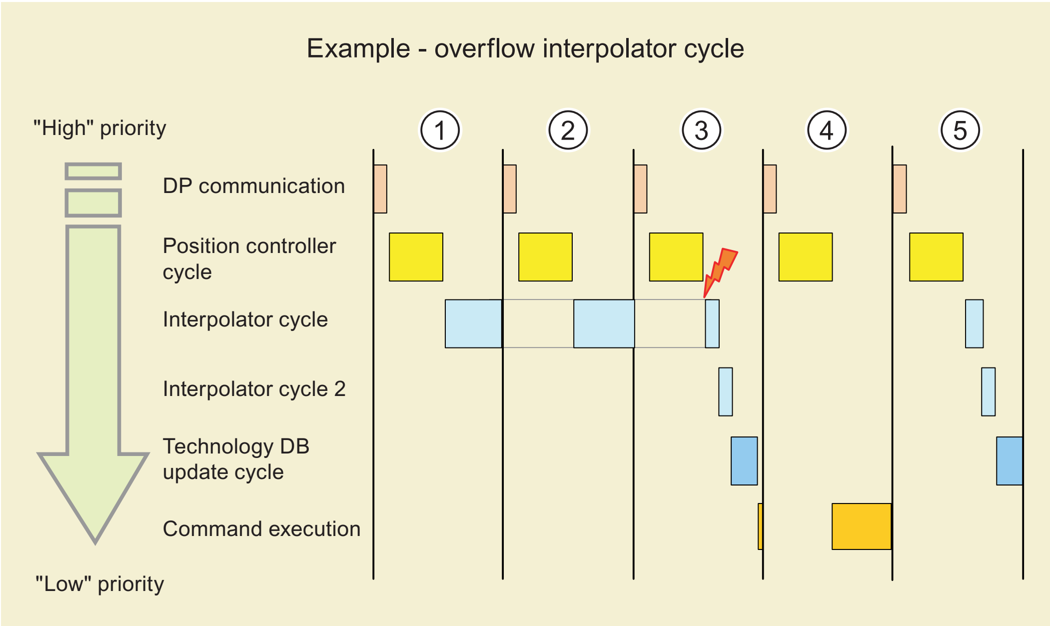
  The time required for the interpolator cycle exceeds the time available in cycle 1. Interpolator cycle 2, including the update of Technology DBs, cannot be started within this cycle. The start will be postponed to the next cycle.
- Cycle 2

  DP communication and the position control cycle interrupt the interpolator cycle. The time slice of the interpolator cycle resumes after the interrupt is cleared. Next, the system executes interpolator cycle clock 2, and initiates the technology DB update.
- Cycle 3

  The Technology DB update is interrupted in order to execute tasks of higher priority (DP communication, position control cycle / interpolator cycle). Same as in cycle 1, the interpolator cycle cannot be terminated within this cycle.
- Cycle 4

  The time slice of the interpolator cycle is resumed and concluded on completion of the position control cycle. Next, the system executes the time slice of interpolator cycle 2 and completes the update of the Technology DBs. This leaves only a narrow time slice for command processing.
- Cycle 5

  The sequence restarts at Cycle 5 as explained in Cycles 1 to 4.



The command start may briefly increase time requirements of the interpolator cycle. The diagram shows a state which may lead to an "overflow" of the interpolator cycle. The same clock ratios apply as in the previous examples:

- Cycle 1

  The time required for the interpolator cycle exceeds the time available in cycle 1. Interpolator cycle 2, including the update of Technology DBs, cannot be started within this cycle. The start will be postponed to a later cycle.
- Cycle 2

  DP communication and the position control cycle interrupt the interpolator cycle. The time slice of the interpolator cycle resumes after the interrupt is cleared.
- Cycle 3

  The time slice of the interpolator cycle is interrupted again in order to execute DP communication and the position control cycle. The system is still busy executing the current interpolator cycle, and calls a new interpolator cycle. This action leads to an "overflow" of the interpolator cycle. The Technology CPU goes into stop or tolerates this overflow, depending on the system clock settings in S7T Config. In this example, at least one overflow will be tolerated and processing is continued, meaning that the interpolator cycle will be completed. Next, the system executes interpolator cycle 2 and updates the Technology DBs. This leaves only a narrow time slice for command processing.
- Cycle 4

  The interpolator cycle which should have been started in cycle 3 is skipped in this cycle, that is, the system executes the time slice for command processing after the position control cycle is completed.
- Cycle 5

  Again, time requirements of the interpolator cycle are slight. The interpolator cycle, interpolator cycle 2 and the Technology DB update can be processed in cycle 5.

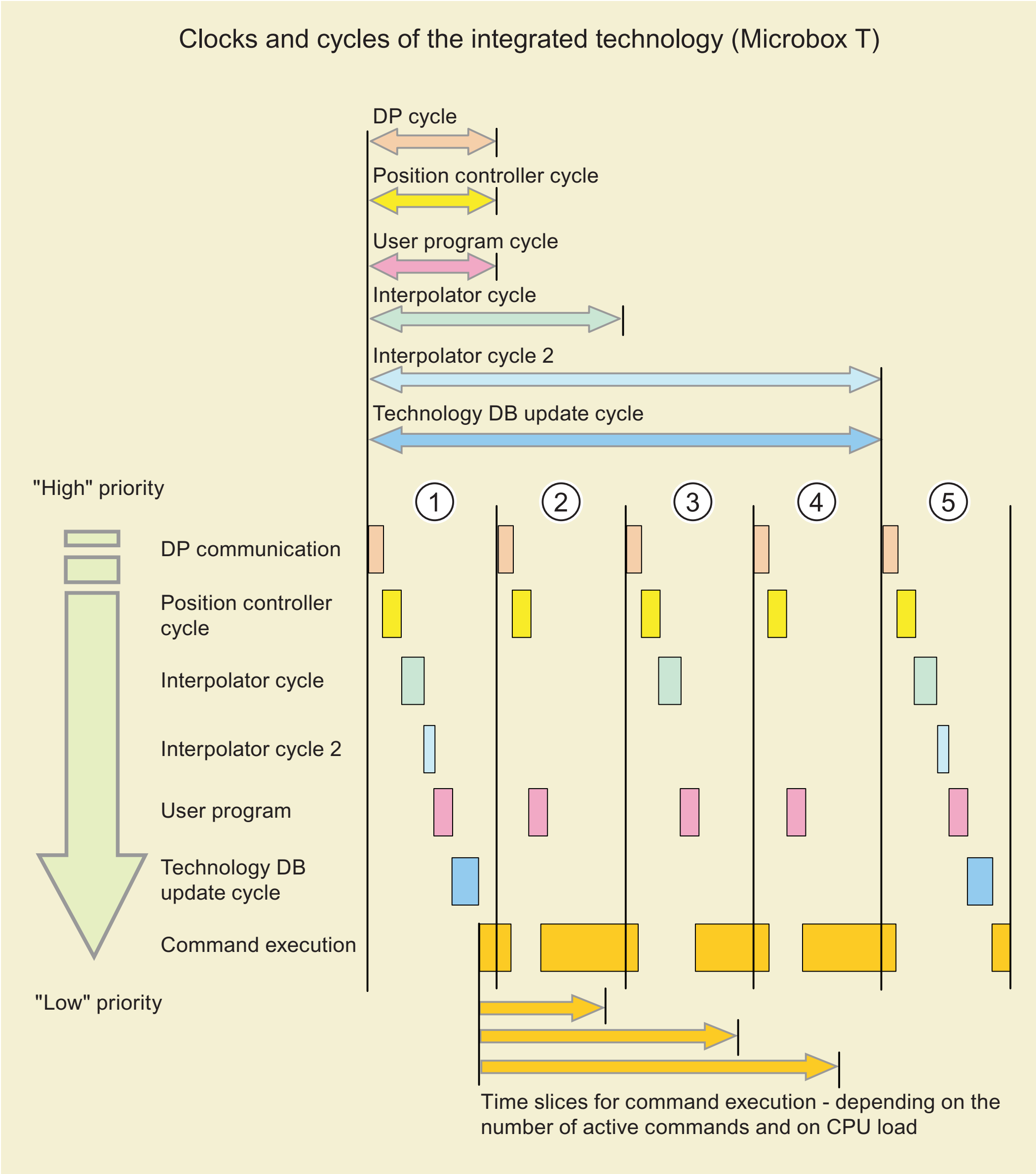
Note

A narrow margin in the cycle time settings may lead to an overflow of interpolator cycle 2.

The number of tolerated overflows can be set interpolator cycle and interpolator cycle 2 in the "System clock cycles" dialog of S7T Config (**Target system > Set System Clock Cycles** command). The Technology CPU goes into STOP if the tolerated number of overflow events is exceeded.

5.2 Time pattern of WinLC T (MICROBOX T)

The diagram below shows the cycles of integrated technology and their influences on the execution of technology object commands for MICROBOX T. The practical example shows a sequence covering several position control cycles.



The diagram represents time requirements of the various cycles in time slice units. The time slice requirements vary, depending on the number of technology objects configured. The time slices of the interpolator cycle, of interpolator cycle 2 and of the update cycle of the technology DBs are not processed in all position control cycles.

DP communication, the position control cycle, the interpolator cycle, interpolator cycle 2 and the Technology DB update cycle are executed with descending priority. At the MICROBOX T, the time for DP communication is also available for integrated technology cycles, with the exception of the position control cycle.

Command execution utilizes the interval between the position control cycles. The smaller this interval, the more time is required for command execution.

Cycles selected in the example shown:

- Position control cycle = *1*

  Corresponds with a 1:1 ratio between the DP cycle and the position control cycle
- Interpolator cycle = *2*

  Corresponds with a 1:2 ratio between the DP cycle and the position control cycle
- Interpolator cycle 2 = *2*

  Corresponds with a 1:2 ratio between the interpolator cycle and interpolator cycle 2
- Update cycle of the technology DBs = multiple of the interpolator cycle
  Interpolator cycle 2 in the example shown

The cycles shown are synchronized with the DP cycle, or with DP communication.

- DP cycle

  The DP cycle determines the intervals for data exchange with the DP I/O on DP(DRIVE). Data are exchanged within the DP communication time slice. The constant bus cycle time at the DP(DRIVE) must be activated at the corresponding parameters in HW Config. The integrated technology in combination with the isochronous components on DP(DRIVE) form the basis of isochronous operation.
- Position control cycle

  The system also computes the axis position control within the time slice of the position control cycle. The time requirements of this time slice remain near constant while the Technology CPU is in operation.

  The position control cycle clock determines the interval between the position control cycles.

  The value of the position control cycle clock can be set to a multiple of the DP cycle in S7T Config. A clock ratio of 1:1 in order to enhance performance in terms of technology object processing is recommended.
- Interpolator cycle

  The time slice of the interpolator cycle is also used to compute the control variables. The interpolator cycle clock determines the interval between two interpolator cycles. Time slice requirements may increase briefly at the start of one or several commands.
- Interpolator cycle 2

  Interpolator cycle 2 handles the same tasks as the interpolator cycle and can be used for technology objects of lower priority classes.

- User Program

  The user program and all interrupt OBs from WinLC T (MICROBOX T) are executed within the time slice shown. The length of the time slice can be set on the WinLC T panel to a percentage of the DP cycle time. You can set values between 10 % and 50 %.

  The OB 1 cycle is executed within this time slice. However, it is not started synchronously with the DP cycle. Once the time slice of the user program has ended, current program execution will be interrupted, and is resumed at the start of the next time slice.

  The time slice of the user program is called at each cycle clock of the DP cycle. The WinLC T status changes to STOP if the time slice cannot be executed three times in succession.

- Updating technology DBs

  The technology DBs are updated within the time slice. Integrated technology with firmware version V3.2.x or higher supports the distribution of technology DB updates to different cycles (see also "Updating the technology DBs"). At firmware versions before V3.2.x all the technology DBs are updated within this same time slice.

  The update cycle of the technology DBs can be set to a multiple of the interpolator cycle in S7T Config. The technology synchronization interrupt OB is called after each update of the Technology DBs.

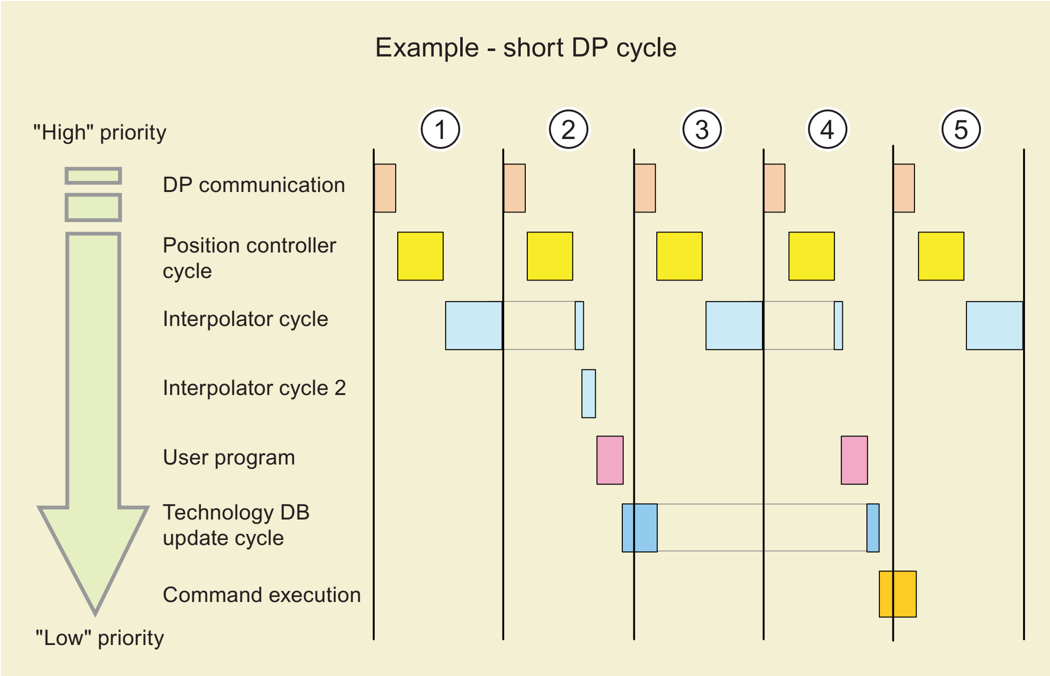
- Command processing

  The commands of integrated technology are processed and monitored at indefinite cycles. The time required to process current commands depends on the number of active commands and on CPU load. Both the average and maximum command processing times can be read from the *CmdLoopDuration* and *MaxLoopDuration* tags of the MCDevice technology DB. Integrated technology with firmware version V3.2.x or higher supports the start of axis commands within the interpolator cycle. Subsequent command monitoring operations are again handled by means of the command processing functions of the integrated technology.

- Windows / Idling cycle

  The displayed time slice is available to Windows application. If this time slice is not further used by Windows applications, a idling slice is inserted up until the next DP communication.

The example above shows that cycle 1 only provides a small time slice for command processing. In addition to DP communication and the position control cycle, the system also processes the interpolator cycle and interpolator cycle 2, and the update of the Technology DBs. This method provides additional time for command processing in the next cycles.



The example shows the time profiles with insufficient DP cycle length. The clock ratios as shown in the previous example apply:

- Cycle 1

  The time required for the interpolator cycle exceeds the time available in Cycle 1. Interpolator cycle 2, including the update of Technology DBs, cannot be started within this cycle. The start will be postponed to the next cycle.

- Cycle 2

  DP communication and the position control cycle interrupt the interpolator cycle. The time slice of the interpolator cycle resumes after the interrupt is cleared. Next, the system executes interpolator cycle clock 2, and initiates the technology DB update.

- Cycle 3

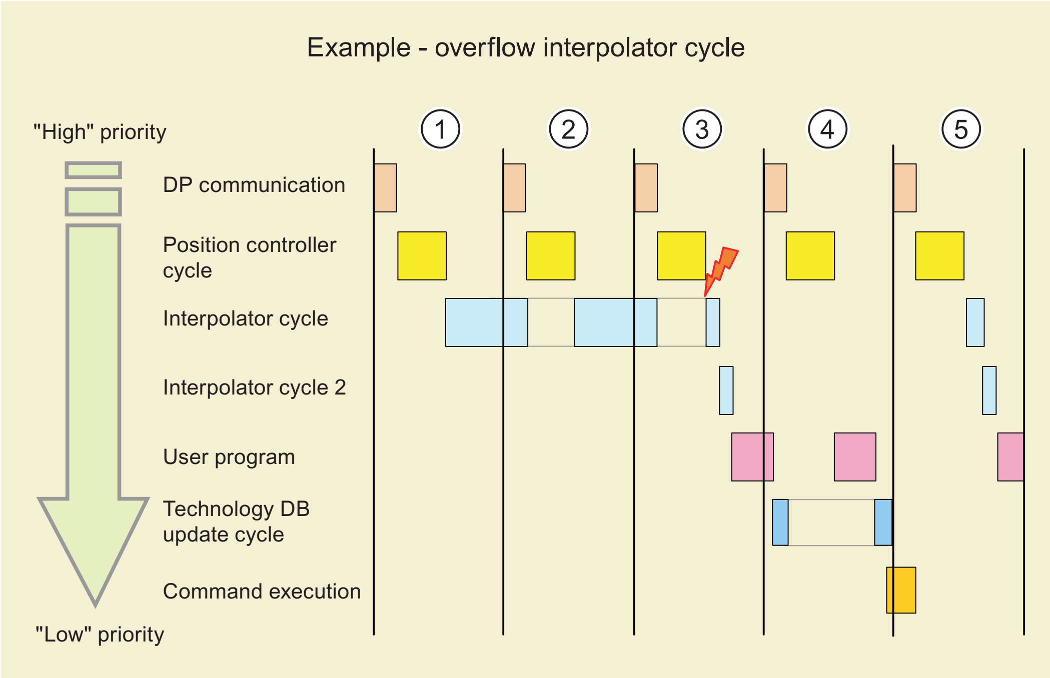
  The Technology DB update is interrupted in order to execute tasks of higher priority (DP communication, position control cycle / interpolator cycle). Same as in cycle 1, the interpolator cycle cannot be terminated within this cycle.

- Cycle 4

  The time slice of the interpolator cycle is resumed and concluded on completion of the position control cycle. Next, the system executes the time slice of interpolator cycle 2 and completes the update of the Technology DBs. This leaves only a narrow time slice for command processing.

- Cycle 5

  The sequence restarts at Cycle 5 as explained in Cycles 1 to 4.



The command start may briefly increase time requirements of the interpolator cycle. The diagram shows a state which may lead to an "overflow" of the interpolator cycle. The same clock ratios apply as in the previous examples:

- Cycle 1

  The time required for the interpolator cycle exceeds the time available in cycle 1. Interpolator cycle 2, including the update of Technology DBs, cannot be started within this cycle. The start will be postponed to a later cycle.

- Cycle 2

  DP communication and the position control cycle interrupt the interpolator cycle. The time slice of the interpolator cycle resumes after the interrupt is cleared.

- Cycle 3

  The time slice of the interpolator cycle is interrupted again in order to execute DP communication and the position control cycle. The system is still busy executing the current interpolator cycle, and calls a new interpolator cycle. This action leads to an "overflow" of the interpolator cycle. The Technology CPU goes into stop or tolerates this overflow, depending on the system clock settings in S7T Config. In this example, at least one overflow will be tolerated and processing is continued, meaning that the interpolator cycle will be completed. Next, the system executes interpolator cycle 2 and updates the Technology DBs. This leaves only a narrow time slice for command processing.

- Cycle 4

  The interpolator cycle which should have been started in Cycle 3 is not applicable in this cycle, that is, the system executes the time slice for command processing after the position control cycle is completed.

- Cycle 5

  Again, time requirements of the interpolator cycle are slight. The interpolator cycle, interpolator cycle 2 and the Technology DB update can be processed in cycle 5.

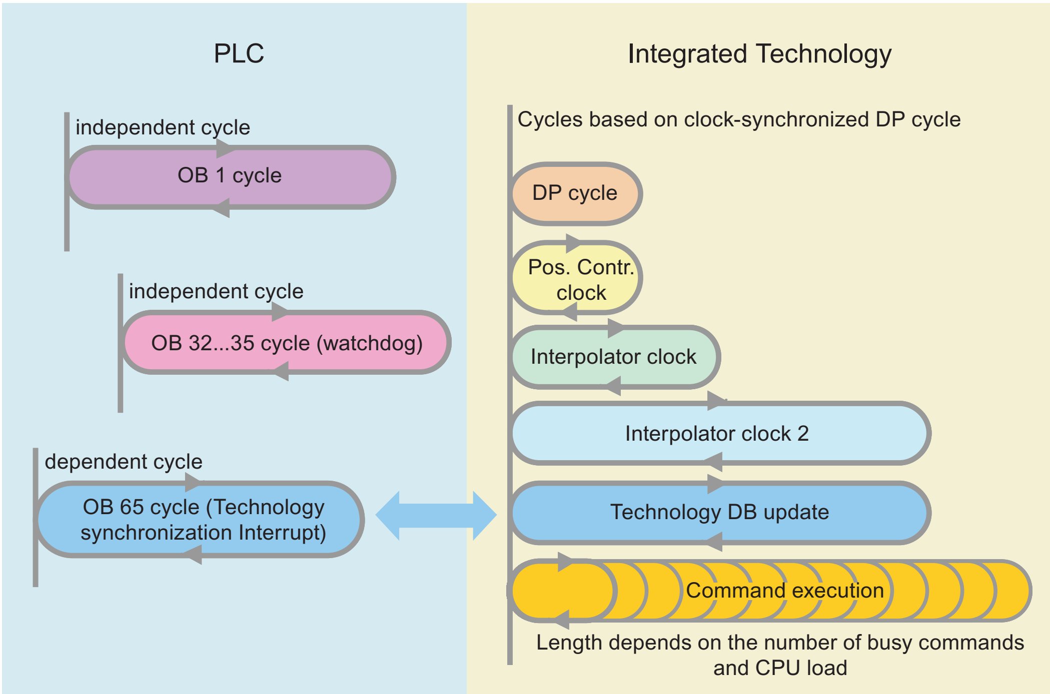
Note

A narrow margin in the cycle time settings may lead to an overflow of interpolator cycle 2.

The number of tolerated overflows can be set interpolator cycle and interpolator cycle 2 in the "System clock cycles" dialog of S7T Config (**Target system > Set System Clock Cycles** command). The Technology CPU goes into STOP if the tolerated number of overflow events is exceeded.

5.3 Cycles of the Technology CPU

The diagram below shows the correlation between the various cycles and clocks of the Technology CPU:



Controller cycles

- **OB1 cycle**

  The length of the OB1 cycle is determined by the number of instructions to be executed. The OB1 cycle operates independent of the cycles of integrated technology.

- **OB32 to OB35 cycle**

  The watchdog interrupt OBs are called within a time pattern of 1 s to 60 s. The watchdog interrupts are called independent of the clocks and cycles of the integrated technology.

- **OB65 cycle**

  The technology synchronization interrupt OB is called after each update of the Technology DBs. The technology synchronization interrupt is coupled to the integrated technology.

Cycles of the integrated technology

The cycles of the integrated technology are synchronized with the DP cycle and DP communication.

- **DP cycle**

  Data are exchanged with the DP I/O on DP(DRIVE) within the DP cycle.

- **Position control cycle**

  The system also computes the axis position control within the time slice of the position control cycle.

- **Interpolator cycle**

  The interpolator cycle is primarily used to calculate control variables.

- **Interpolator cycle 2**

  Interpolator cycle 2 handles the same tasks as the interpolator cycle and can be used for technology objects of lower priority classes.

- **Technology DB update cycle**

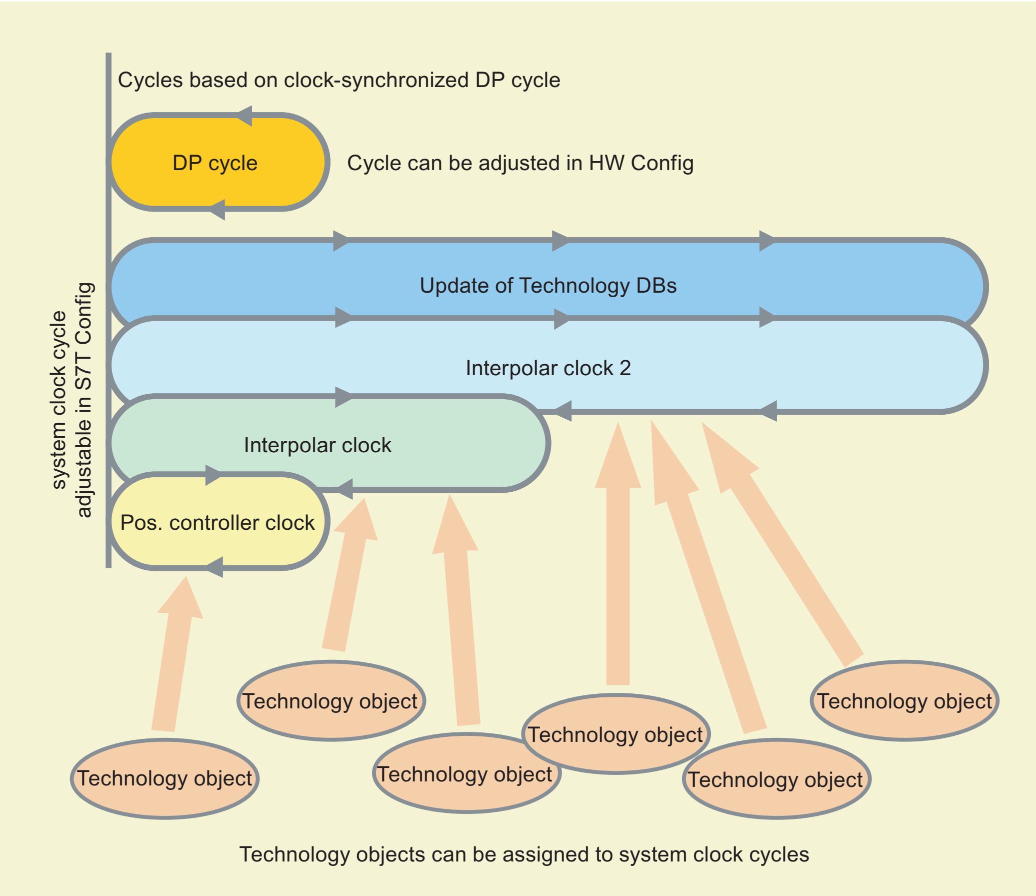
  The technology DBs are updated within this cycle.

Command processing

The commands of integrated technology are processed and monitored at indefinite cycles. The time required to process current commands depends on the number of active commands and on CPU load.

5.4 Assigning system clocks

You can improve utilization of system resources by assigning system clocks to the technology objects.



You can assign the following clocks to the technology objects:

Motion control task	High priority	...	Low priority
Technology object	Position control cycle	Interpolator cycle	Interpolator cycle 2
Speed-controlled axis	-	Standard	X
Positioning axis	-	Standard	X
Synchronization axis	-	Standard	X
External encoder	-	Standard	X
Output cams	X	X	X
Measuring inputs	X	X	X
"X" = allowed "-" = not allowed			

You can set the basic clock for Technology DB updates to a multiple of the interpolator cycle.

Note

The following combinations are possible when selecting different execution cycles for an output cam and the associated axis:

- Axis and output cam in the same cycle clock (interpolator cycle clock or interpolator cycle clock 2)
- Axis in the interpolator cycle clock or interpolator cycle clock 2 and output cam in the position control cycle.
- Axis in the interpolator cycle clock and actual value output cam in interpolator cycle clock 2.

The technology system clock of axes, external encoders, output cams and measuring inputs are set in the **Configuration** dialog box of S7T Config.

The synchronization axis and the corresponding synchronization object must be operated within the same execution cycle. Any change of the processing cycle of a synchronization axis also has to be applied to the synchronization object:

1. In Navigator of S7T Config, select the synchronization object of the corresponding synchronization axis.
2. Select the shortcut menu command **Expert > Expert list**
3. You can change the processing cycle clock in the *Execution.executionlevel* configuration data element.

Change the default setting:

- if you detect command processing times of excess length
  The execution times are available in the *MaxLoopDuration* and *CmdLoopDuration* variables of the MCDevice technology DB.
- If you detect excess load on the Technology CPU
  The load on the Technology CPU can be determined in online mode by selecting the **Target system > Device diagnostics > System load** menu command.

Assign the "axis" and "external encoder" technology objects with low-priority tasks to "interpolator cycle 2", and the "output cam" and "measuring input" technology objects to the "interpolator cycle" or "interpolator cycle 2". Assign the "interpolator cycle" or the "position control cycle" to the "Output cam" and "Measuring input" technology objects with high-priority tasks.

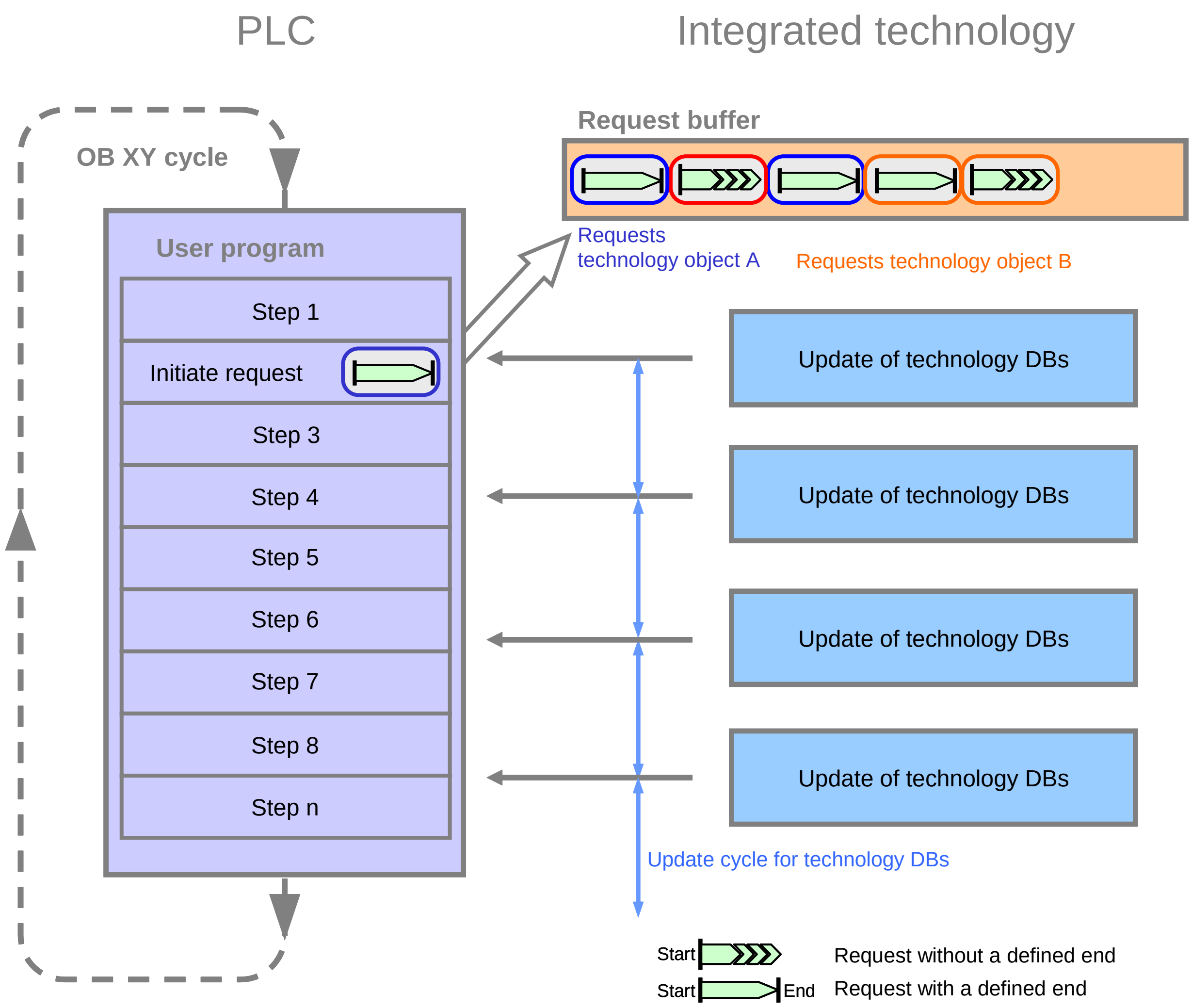
If the command monitoring time is exceeded you can reduce command execution time by taking the following measures:

- Set a longer DP cycle or position control cycle.
- Extend the technology DB update cycle, or distribute the update of individual technology DBs to different cycles (for further information refer to "Updating the technology DBs (Page 791)".)
- Assign interpolator cycle 2 to the low-priority technology objects. This reduces load on the interpolator cycle and increases the time slices for command execution.
- Select the expert mode in Technology Objects Management and set "IPO synchronous" command start for specific technology objects This setting forces the command to be started within the time slice of the interpolator cycle in order to reduce load on the time slice for command execution

5.5 Sequence and programming model

Sequence model

The diagram below shows the command interface for the PLC and integrated technology.



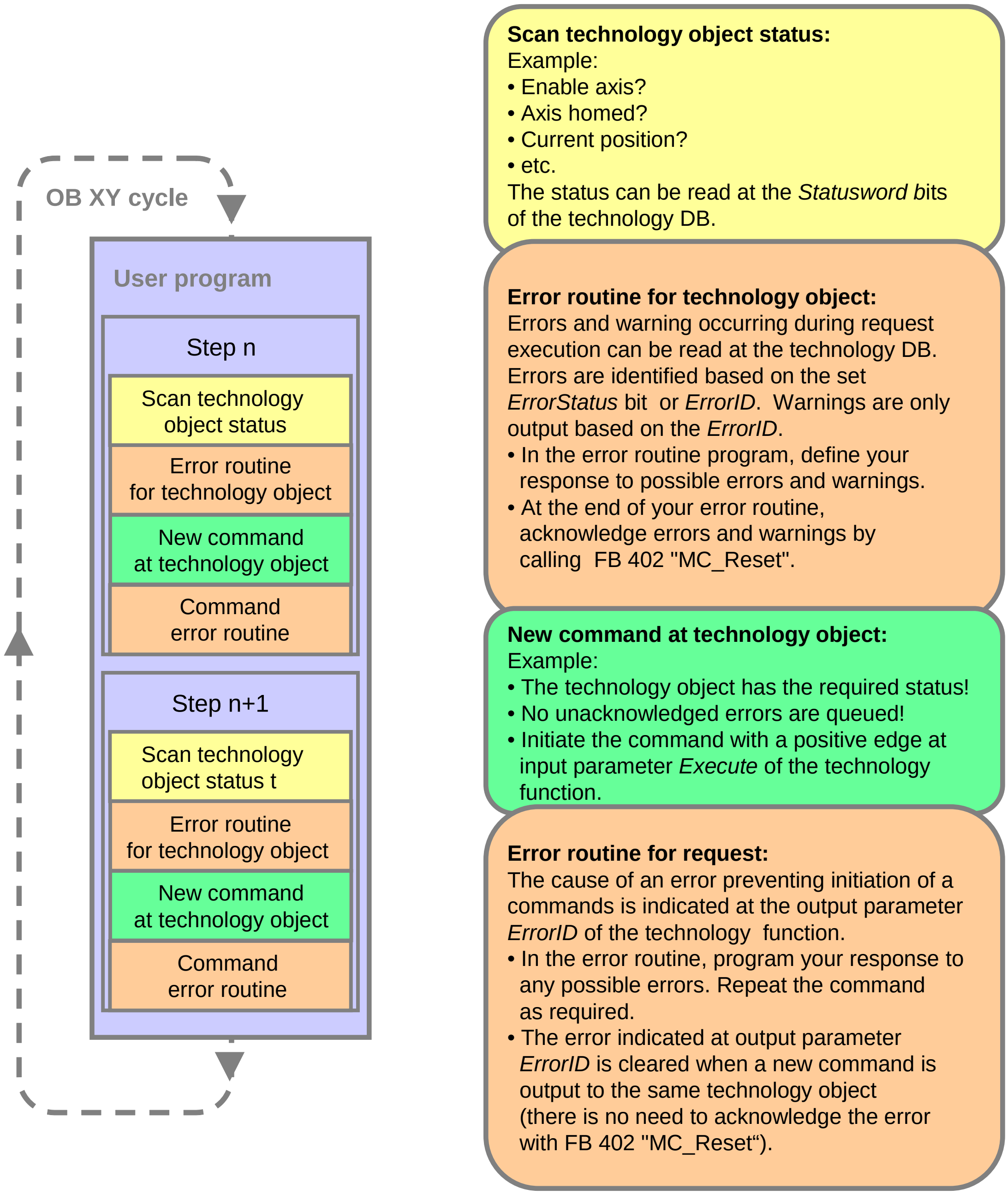
The sequential user program (OB 1 or the watchdog interrupt OBs) initiates a command and enters this in the command buffer of integrated technology. "IPO synchronous" and "non-IPO synchronous" commands are saved to separate input buffers. Each technology object is capable of handling several commands in parallel without having to wait for discrete commands to be processed.

Both the status and error messages, including the actual values of the technology object, are updated alongside with the Technology DBs in the PLC. Updates run asynchronously to the OB XY cycle, meaning that a new actual value is returned in each scan cycle of the user program.

NOTICE
Do not call up technology functions in the restart OBs. The integrated technology can neither accept nor execute commands in the restart phase.

Programming model

The diagram below shows the structure of a user program. The step sequence is independent of processing in OB 1 or in a watchdog interrupt cycle (OB 32 to OB 35).



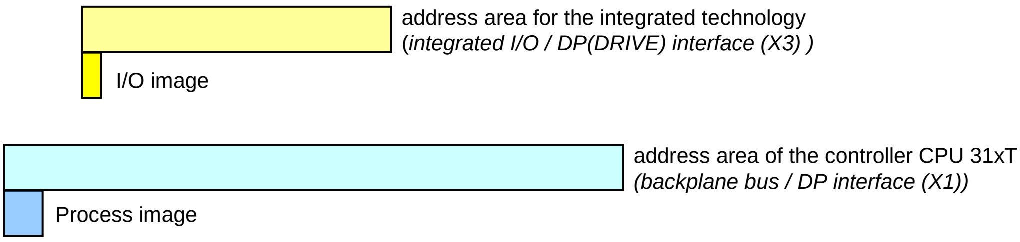
Each command step consists of these subroutines:

- Request of the technology object status
- Evaluation of technology object errors
- Initiation of a new command for the technology object
- Evaluation of command errors

The programming model shown serves as a guideline. The programming model can be customized to suit user requirements, taking into account the time patterns and sequence models shown earlier.

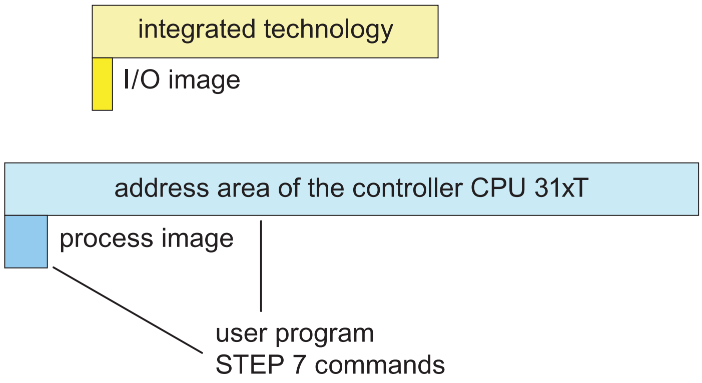
5.6 Accessing address spaces of CPU 31xT

CPU 31xT features separate address areas for the PLC and for integrated technology.



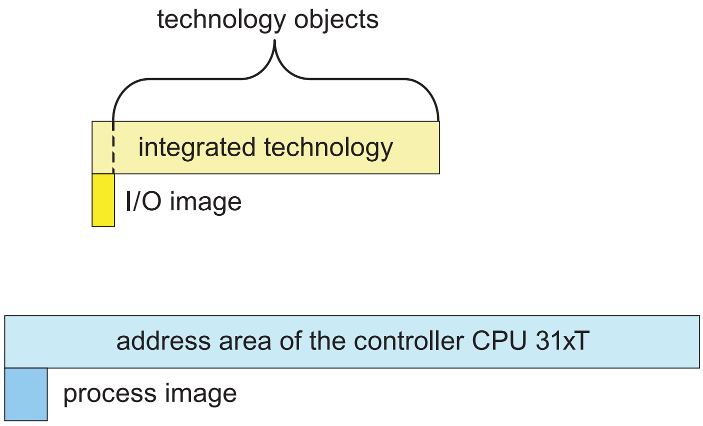
Address areas	Addresses	
	CPU 315T-2DP	CPU 317T-2 DP
CPU 31xT PLC	*0* to *2047*	*0* to *8191*
Process image	*0* to *127*	*0* to *255*
Integrated Technology	*0* to *1023*	*0* to *1023*
I/O image	*0* to *63*	*0* to *63*

Access via the PLC



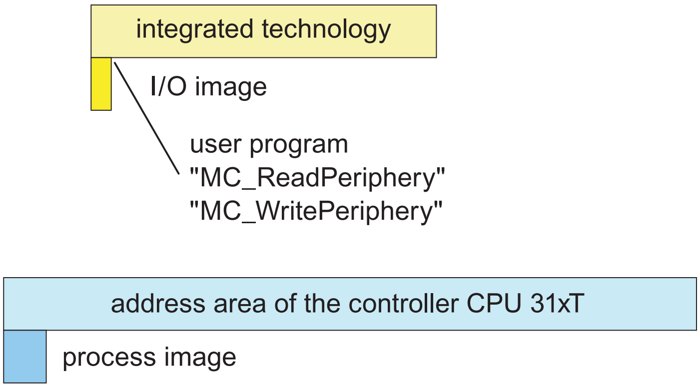
The user program can be used to access the entire address space of the PLC using STEP 7 commands.

Access by means of technology objects



Assign the addresses to be accessed by means of the technology objects to address space *64* to *1023*. Technology objects cannot access addresses outside this space.

Access via the "MC_ReadPeriphery" and "MC_WritePeriphery" technology functions.



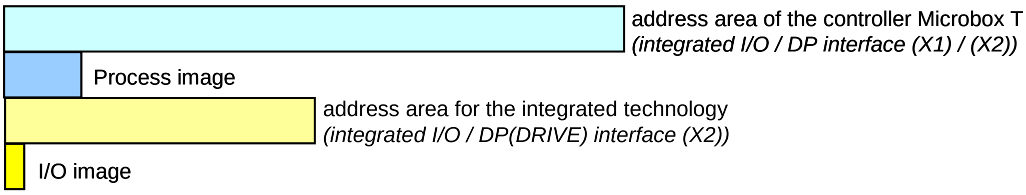
You can use the "MC_ReadPeriphery" and "MC_WritePeriphery" technology functions to access the I/O image of the integrated technology (address area *0* to *63*).

Place the addresses that you would like to access with the "MC_ReadPeriphery" and "MC_WritePeriphery" technology functions completely in this address area. These technology functions cannot access addresses outside this space.

Addresses of the DP(DRIVE) interface (X3) or the integrated I/O can be assigned to the I/O image.

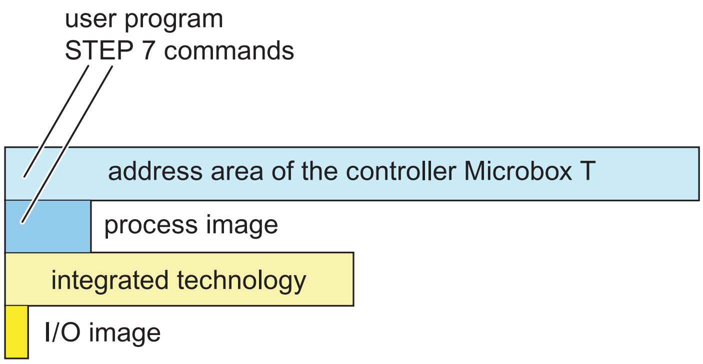
5.7 Accessing address spaces of MICROBOX T

MICROBOX T provides an address space which is shared by the controller and integrated technology. This address space can be assigned addresses of the DP interface (X1) and DP(DRIVE) interface (X2), and addresses of integrated I/O.



Address areas	MICROBOX 420-T addresses
MICROBOX T PLC	*0* to *2047*
Process image (default setting *)	*0* to *511*
Integrated technology	*0* to *2047*
I/O image	*0* to *63*
*) The size of the process image can be set in HW Config.	

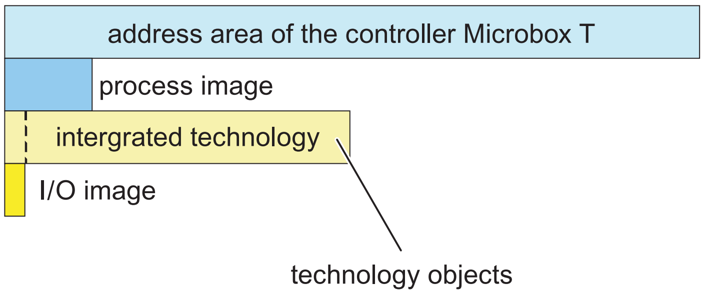
Access via the PLC



The user program can always access the entire address space of MICROBOX T by means of STEP 7 commands. These addresses, however, must be assigned to the DP interface (X1) in HW Config.

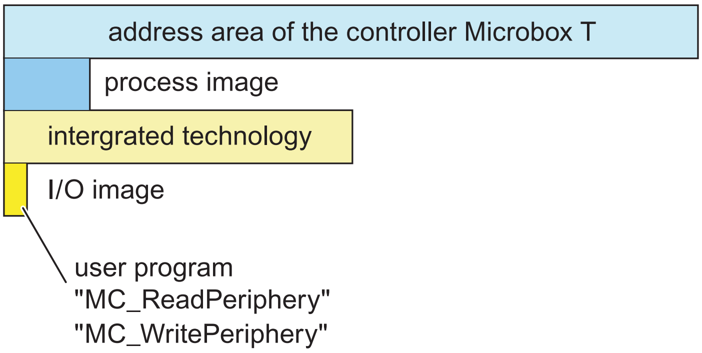
Direct access to addresses of the integrated I/O or addresses that have been assigned to the DP(DRIVE) interface (X2), is not possible. If these addresses are accessed in the user program the controller reacts as though these addresses were not in use physically.

Access by means of technology objects



Assign the addresses to be accessed by means of the technology objects to address space *64* to *1023.* Technology objects cannot access addresses outside this space. The technology objects can only access addresses that were assigned in HW Config to the DP(DRIVE) interface (X2) or to the integrated I/O.

Access via the "MC_ReadPeriphery" and "MC_WritePeriphery" technology functions.



You can use the "MC_ReadPeriphery" and "MC_WritePeriphery" technology functions to access the I/O image of the integrated technology (address area *0* to *63*).

Place the addresses that you would like to access with the "MC_ReadPeriphery" and "MC_WritePeriphery" technology functions completely in this address area. These technology functions cannot access addresses outside this space.

Addresses of the DP(DRIVE) interface (X2) or the integrated I/O can be assigned to the I/O image.

Note

- " MC_ReadPeriphery"
  If the "MC_ReadPeriphery" technology function is used to access address areas or subareas that have been assigned to the DP interface (X1), invalid values are read for these addresses.
- "MC_WritePeriphery"
  If the "MC_WritePeriphery" technology function is used to access address areas or subareas that have been assigned to the DP interface (X1), writing for these addresses is not performed.

5.8 Monitoring active commands

You can monitor active commands by reading the output parameters of the technology functions. The output parameters (status outputs) of the technology functions indicate the command state.

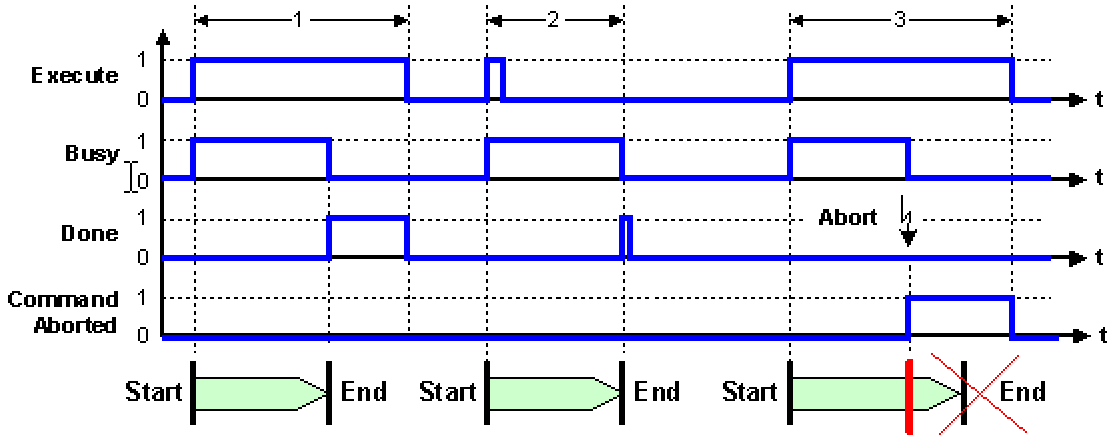
The value at output parameter *Busy* is *TRUE as long as a command is active;* parameter *Busy* returns the value *FALSE* after the command has been completed.

The remaining output parameters indicate the status at least for the duration of one cycle. These status messages are displayed and saved as long as input parameter *Execute* = *TRUE* (see also Applications 1 and 3 in the figure below).

Technology functions with output parameter *Done*

Commands assigned output parameter *Done* have a defined termination. Output parameter *Done* returns the value *TRUE* to signal successful completion of the command (see application 1).

Output parameter *CommandAborted* is set if the command is canceled prematurely; output parameter *Done* is not set (also refer to Application 3). The cancellation may be triggered if a new command is output to the same technology object, or by an error event.



Technology functions without output parameter *Done*

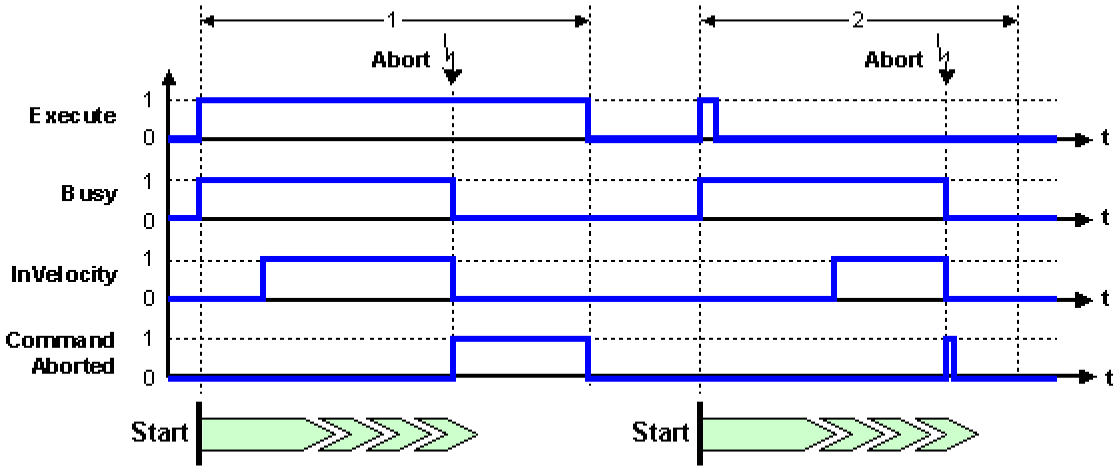
Commands which do not feature the output parameter *Done* do not have a defined termination. The action or motion continues until a cancel command is received or an error is detected.

These technology functions return a status message which indicates the initial reaching of the required operating state (*InVelocity, InSync, InGear, DataValid* etc.)

Example - Technology function "MC_MoveVelocity" (also refer to the diagram below)

Each command is started with input parameter *Execute = 1.* Output parameter *Busy* indicates the active state of the command. The status at output parameter *InVelocity* changes to *TRUE* after the required velocity has been reached for the first time. The command would be permanently active if not canceled (*Busy = TRUE*)

The active command is canceled (*Busy = FALSE, InVelocity = FALSE* and *CommandAborted = TRUE*) if a new command is initiated at the same technology object or if an error is detected.



Cancellation of commands by the same technology function

It may be necessary to override an active command with a new command of the same type (for example, replacing Approach to position *100* by Approach to position *200*), depending on the application.

Note

Use different instance DBs in this case. You can no longer monitor the active command at the output parameters of the technology function if you are using the same instance DB. Start of the new command was prevented due to an error, for example. You can avoid this problem by using different instance DBs.

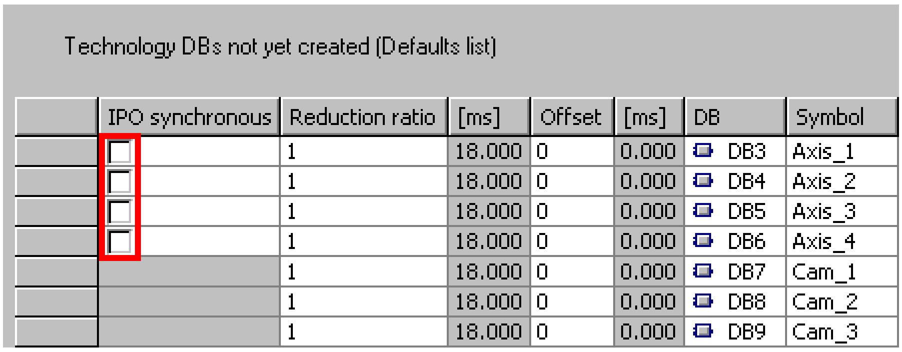
Each one of the instance DBs is assigned a separate command buffer. A rapid sequence of commands with the same instance DB may lead to consistency problems at the command buffer. New commands are thus rejected with *ErrorID = 804C* as long as the command buffer is being processed. This also prevents command monitoring at the output parameters of the technology function.

5.9 Start of axis commands in IPO synchronous mode

Valid for Integrated Technology with firmware V3.2.x

The integrated technology executes commands by default within the time gaps after the interpolator cycles and before the next position control cycle. Short cycle times may lead to the distribution of command processing, and therefore of the command start, to several position control cycles.

Integrated technology V3.2.x or higher supports the "IPO synchronous" start of single axis commands. The "IPO synchronous" property can be set for the corresponding "Axis" technology object in "Technology Objects Management". Start by selecting the **View > Expert mode** command.


Technology DBs not yet created (Defaults list)

	IPO synchronous	Reduction ratio	[ms]	Offset	[ms]	DB	Symbol
	☐	1	18.000	0	0.000	DB3	Axis_1
	☐	1	18.000	0	0.000	DB4	Axis_2
	☐	1	18.000	0	0.000	DB5	Axis_3
	☐	1	18.000	0	0.000	DB6	Axis_4
		1	18.000	0	0.000	DB7	Cam_1
		1	18.000	0	0.000	DB8	Cam_2
		1	18.000	0	0.000	DB9	Cam_3

The "IPO synchronous" property allows the deterministic, faster on average start of axis commands.

Axis commands which can be started "IPO synchronously":

MC_Power	MC_Home (not for absolute adjustment)	MC_Stop
MC_Halt	MC_ChangeDataset	MC_MoveAbsolute
MC_MoveRelative	MC_MoveAdditive	MC_MoveSuperImposed
MC_MoveVelocity	MC_MoveToEndPos (job tracking is also "IPO synchronous" until the end position is reached until the end position is reached, job tracking is "IPO synchronous", too)	MC_GearIn
MC_CamIn	MC_GearOut	MC_CamOut
MC_Phasing	MC_SetTorqueLimit	MC_SetCharacteristic
MC_GearInSuperImposed	MC_CamInSuperImposed	MC_GearOutSuperImposed
MC_CamOutSuperImposed	MC_PhasingSuperImposed	

Unless otherwise specified in the table, only the command start is carried out in "IPO synchronous" mode. Commands continue to be tracked during standard command processing.

Note

Every "IPO synchronous" command start extends the processing of the interpolator cycle by approx. 300 µs. Avoid a large number of concurrent "IPO synchronous" commands to prevent an IPO overflow.

Sequence of "IPO synchronous" and "non-IPO synchronous" commands

Not all commands for the "Axis" technology object can be executed in "IPO synchronous" mode. "IPO synchronous" and "non-IPO synchronous" commands are saved to separate input buffers. Conditions of mixing "IPO synchronous" and "non-IPO synchronous" commands:

- All "IPO synchronous" commands of an axis are processed in the order of their startup.
- All the "non-IPO synchronous" commands for an axis are processed in the order of their startup.

Note that the order of start and processing sequences may differ when you mix "IPO synchronous" and "non-IPO synchronous" commands.

Note

To define a fixed order of command processing:

Analyze the *Done, InGear, InSync* or *InClamping* output parameters of the commands before you transfer a new command to the TO.

5.10 Errors and warnings at the technology function

The technology functions indicate any errors at the output parameters *Error*, *ErrorID* or *CommandAborted*. If the program fails to correctly update all output parameters due to insufficient length of an instance DB you can evaluate this error by reading the BIE-Bit bit.

Warnings and errors at output parameter *Error* or *ErrorID*

Output parameter *Error = TRUE* indicates that the technology function was unable to initiate the command. The cause of error is indicated by the value at output parameter *ErrorID* (*ErrorID = 8xxx*).

Warnings are also relevant to command initiation. Output parameter *ErrorID* returns a *Wert 00xx* to indicate the cause of a warning. Output parameter *Error* shows *FALSE*.

If the technology function reports an error, you need to call it either with the correct parameters or at a different time, provided that function is allowed.

It is neither required, nor is it possible to acknowledge the errors or warnings. The error remains active until the parameters *Enable* and *Execute* have been reset.

Output parameter CommandAborted

Output parameter *CommandAborted* reports the cancellation of an active command which was triggered during command execution by a new command or by an error. If *ErrorStatus* of the technology DB does not report an error, the command was canceled by a subsequent command. An error reported at *ErrorStatus* indicates that an error in integrated technology during command execution has canceled the command.

Error analysis with the BIE-Bit

BIE-Bit (binary result bit) has the value *0* when an active error exists or the output parameters of the technology function could not be updated correctly.

Failure of the update of the output parameters is caused by an instance DB of insufficient length in the controller.

The example below shows how you can evaluate the BRB bit in the user program:

```
CALL  "MC_Power" , DB401
 Axis    :=1
 Enable  :=E5.0
 Mode    :=0
 StopMode:=0
 Status  :=M100.0
 Busy    :=M100.1
 Error   :=M100.2
 ErrorID :=MW102
UN    BIE
=     A     16.0
```

The BIE-Bit is toggled from *1* to *0* if an error is detected. Output *Q16.0* is assigned the value *TRUE*.

Note

In order to validate the output parameters, you should always evaluate the BIE-Bit immediately after the call of the technology function.

In the next step, check the *Error* and *ErrorID* parameters.

5.11 Errors at the technology DB - MCDevice, Trace

Invalid "MCDevice" or "Trace" technology DBs are reported with *ErrorID 8008* at the corresponding Technology DB. An error message is also output if the technology DB was replaced with a new instance (*ErrorID 8009*).

Error indications defined at technology DB "MCDevice" and "Trace":

- *ErrorID* variable

  The *ErrorID* variable returns the ErrorID of the most recently detected error (*ErrorID = 8xxx*).

  The value of this variable is cleared by acknowledging it with "MC_Reset" (*Restart = FALSE*).

- *ErrorBuffer[0..2] variable*(not available for the "Trace" technology DB)

  The first three errors are saved to the variable array *ErrorBuffer[0..2]*. The first error is written to ARRAY element *1*, the second to ARRAY element *2*, and so forth.

  The value of this variable is cleared by acknowledging it with "MC_Reset" (*Restart = FALSE*).

Acknowledging errors

Eliminate the cause of any error detected in the *ErrorID* variable (*ErrorID = 8xxx*). Errors indicated at the "MCDevice" and "Trace" technology DB cannot be acknowledged.

Siemens on the Internet (www.siemens.com)

5.12 Errors and warnings at the technology DB – Axes, external encoders

Any errors and warnings occurring at the technology object during command execution are reported at the technology DB. This also applies to the transfer of inconsistent or contradictory dynamic values.

Error indications defined at the Technology DB for axes and external encoders:

- *Statusword.Error* variable

  The value at this variable is *TRUE* if at least one error has occurred. The error can be analyzed at the *ErrorID* or *ErrorBuffer[0..2]* variable by means of the ErrorID.

- *Statusword.Errorstop* variable

  The axis is/was stopped due to an error event; the technology object may be disabled.

  If the value at the *Statusword.Errorstop* and *Statusword.Error* variables is *TRUE* you can analyze the error by reading the *ErrorID* or *ErrorBuffer[0..2]*.

- *ErrorID* variable

  The *ErrorID* variable displays the ErrorID of the most recently detected error (*ErrorID = 8xxx*) or warning (*ErrorID = 00xx*) returned by the technology object.

  The value of this variable is cleared by acknowledging it with "MC_Reset" (*Restart = FALSE*).

- *ErrorBuffer[0..2]* variable

  The first three error or warning events (since last acknowledgment) are saved to the *ErrorBuffer[0..2]* variable array. The first error is written to ARRAY element *1*, the second to ARRAY element *2*, and so forth.

  The value of this variable is cleared by acknowledging it with "MC_Reset" (*Restart = FALSE*).

- *ErrorStatus.xxx* variable

  The *ErrorStatus.xxx* variable is used to indicate specific axis error information in bit format (note the comments in the technology DB).

Acknowledging warnings

Warnings at the *(ErrorID = 00xx)* technology DB may, but do not necessarily have to be acknowledged by calling "MC_Reset".

However, observe the data volume at the *ErrorBuffer[0..2]* variable You can keep the *ErrorBuffer[0..2]* variable clear for any incoming error messages by acknowledging the warnings in due time.

Acknowledging errors

In order to acknowledge all errors, you first eliminate their cause and then acknowledge the errors by executing "MC_Reset" (*Restart = FALSE*). You can then re-enable the technology object by calling the "MC_Power" technology function.

⚠ CAUTION
Several unacknowledged errors may cause the CPU to go into STOP.

5.13 Errors and warnings at the technology DB - Cam disk, measuring input, output cam

Any errors and warnings occurring at the technology object during command execution are reported at the technology DB. This also applies to the transfer of inconsistent or contradictory data.

Error indications defined at the technology DB for cam disks, measuring inputs and output cams:

- *ErrorID* variable

  The *ErrorID* variable displays the ErrorID of the most recently detected error (*ErrorID = 8xxx*) or warning (*ErrorID = 00xx*) of the technology object.

  The value of this variable is cleared by acknowledging it with "MC_Reset" (*Restart = FALSE*).

- *ErrorBuffer[0..2]* variable

  The first three error or warning events (since last acknowledgment) are saved to the *ErrorBuffer[0..2]* variable array. The first error is written to ARRAY element *1*, the second to ARRAY element *2*, and so forth.

  The value of this variable is cleared by acknowledging it with "MC_Reset" (*Restart = FALSE*).

Acknowledging warnings

Warnings at the technology DB *(ErrorID = 00xx)* may, but do not necessarily have to be acknowledged by calling "MC_Reset".

However, observe the data volume at the *ErrorBuffer[0..2]* variable. You can keep the *ErrorBuffer[0..2]* variable clear for any incoming error messages by acknowledging the warnings in due time.

Acknowledging errors

In order to acknowledge all errors, you first eliminate their cause and then acknowledge the errors by executing "MC_Reset" (*Restart = FALSE*).

⚠ CAUTION
Several unacknowledged errors may cause the CPU to go into STOP.

5.14 Response of virtual axes

Although virtual axes feature a control variable, they neither feature a position control system, nor a drive, nor an encoder interface. The actual value of the virtual axis is always set equal to the setpoint. The absence of the position control system, and of the drive and encoder interface, leads to certain particularities during command processing which are described below.

Particular response of the variable *Statusword.DriveEnabled*

The status of the *Statusword.DriveEnabled* variables of the corresponding technology DB remains *TRUE* if the enable status of a virtual axis is reset at the input parameter *Enable* of the "MC_Power".

The *Statusword.DriveEnabled* variable for a virtual axis is always *TRUE*.

Editing parameters when the Technology CPU is in RUN

- Restart
  As virtual axis do not feature a position control and drive and encoder interfaces, it is often not required to perform a restart in order to activate parameter changes, regardless of technical specifications in the help systems of the dialog boxes or in the list of technology parameters.
  The *Statusword.RequestRestart* variable of the corresponding technology DBs shows whether or not a restart is required.
- Parameter changes
  All axis parameters are listed both in the list of technology parameters and in the dialog boxes of the DB-Param.

  Take into account that it is not possible to read or edit all the parameters of virtual axes.

5.15 Symbolic programming with FC 400 "DB2INT"

Use FC 400 "DB2INT" of the "S7-Tech" library if you want to use the symbolic name of the technology block to call the technology functions.

The example below shows an application with FC 400 "DB2INT":

```
CALL  "DB2Int"
 DBref  :="Achse_1_"
 RET_VAL:=#TECH_DB

CALL  "MC_Power" , DB401
 Axis    :=#TECH_DB
 Enable  :=
 Mode    :=
 StopMode:=
 Status  :=
 Busy    :=
 Error   :=
 ErrorID :=
```

1. Define a variable of the type INT (variable "TECH_DB" in this case).
2. Call FC 400 "DB2INT" by executing the "CALL" command.
3. Set the symbolic name of the technology DB at input parameter *Dbref* .
4. Set the defined variable value at output parameter *RET_VAL* .
5. Next, call the technology function by executing the "CALL" command (here, FB401 technology function "MC_Power").
6. Set the defined variable at input parameter *Axis* (for other technology functions, at input parameter *Master*, *Slave*, *CamSwitch*, *MeasureInput* or *CamTable*).

5.16 Programming axis-specific parameter changes

The parameters of the technology objects set in S7T Config can be changed in the user program by calling technology function "MC_WriteParameter" while the Technology CPU is in RUN. These changes are active until the next POWER OFF or CPU memory reset.

Certain parameters are not activated unless the addressed technology object has been initialized (restart.) This is only possible when the technology object is disabled. The parameters set in S7T Config are valid again after power is cycled OFF and ON at the Technology CPU.

You basically have two options of editing parameters:

- By editing the corresponding input parameter data.
- The parameter changes are made at the instance DB of the technology function. You can select different instance DBs to edit different parameters.

Creating a new instance DB for technology function "MC_WriteParameter"

1. Select the block folder in SIMATIC Manager.
2. Select Insert **Insert > S7 block > Data block** to insert a new data block.
3. Assign the required name, and then select the "Instance DB" type and the corresponding FB 407.
4. You can also define a symbolic name and enter a symbol comment.
5. Close DB creation by clicking "OK".

Changing parameters by editing input parameter data

In order to change a technology object parameter you require its parameter number, and information about the type, the access mode, and the permitted range of values. This information is available in the parameter list. However, it is advisable to use the configuration view of the instance DB, because there you can also search for specific parameters.

1. Open the instance DB in SIMATIC Manager.

   The "DB-Param" tool opens and shows the content of the instance DB in the configuration view.
2. Select the technology object from the drop-down list in order to edit its parameters.
3. Select the relevant group and the parameter from the tree view.
4. Position the mouse cursor on this parameter. The tooltip shows you the parameter number, the type, and the access mode ("read, write".) Note down these values for setting the input parameters of the technology function. You can also output the list of technology parameters from the Online Help to a printer.
5. You can view the permitted parameter values in the "Value" field of the drop-down list or in the tooltip.
6. Close the configuration view. Confirm or reject your changes in the subsequent dialog box. The configured content of the instance DB is irrelevant when data are set at the input parameters of the technology function.
7. Call the "MC_WriteParameter" technology function in the user program using the instance DB and then set the input parameter values you have noted down. The parameter assignment command is initiated by a positive edge at the *Execute* input.

The advantage of this procedure is that you only need one instance DB. However, the draw-back is that you can only edit one parameter at a time, and that you have to wait for the response of the MC_WriteParameter technology function before can you edit the next parameter.

Editing parameter using different instance DBs

Each parameter change requires the creation and configuration of a new instance DB.

1. Open the instance DB in SIMATIC Manager.

   The "DB-Param" tool opens and shows the content of the instance DB in the configuration view.
2. From the drop-down list, select the technology object which contains the parameter you want to edit using this instance DB.
3. Select the relevant group and the parameter from the tree view.
4. Select the required value from the "Value" drop-down list, or type in the value.
5. Close the dialog box and confirm your changes in the next dialog box.
6. Call the "MC_WriteParameter" technology function in the user program using the relevant instance DB.
7. Set **only** the *Execute* input parameter. A positive at input parameter *Execute* initiates the command to activate the parameters with the values of the instance DB.

Repeat this procedure for all parameter changes.

The positive aspect of this procedure is that you can comfortably set the parameters using DB-Param. It is also of advantage that you can edit several parameters in immediate succession. However, the negative aspect is that you always require several DBs or a multiple instance DB.

6 Technology functions

6.1 Overview

6.1.1 Overview - Programming

Valid for Integrated Technology with firmware version V4.1.x

Requirement

Motion control commands can be programmed with the help of function blocks, if:

- The station configuration in HW Config was saved and compiled
- The technology objects were inserted and configured in S7T Config
- The technology DBs were generated in the "Technology Objects Management" dialog box of S7T Config

Define the reference to the technology object by means of the relevant technology DB number.

Function blocks

Programmable function blocks:

Function	No.	Name	Description
Single axis	FB 401	MC_Power (Page 443)	Enable/disable axis
	FB 403	MC_Home (Page 454)	Home/set axis
	FB 404	MC_Stop (Page 465)	Stop axis and prevent new motion commands
	FB 405	MC_Halt (Page 470)	Normal stop
	FB 409	MC_ChangeDataset (Page 475)	Change data set
	FB 410	MC_MoveAbsolute (Page 481)	Absolute positioning
	FB 411	MC_MoveRelative (Page 495)	Relative positioning
	FB 412	MC_MoveAdditive (Page 509)	Relative positioning to the current target position
	FB 413	MC_MoveSuperImposed (Page 515)	Superimposed positioning
	FB 414	MC_MoveVelocity (Page 521)	Motion with speed preset
	FB 415	MC_MoveToEndPos (Page 530)	Move to fixed endstop / terminals
	FB 437	MC_SetTorqueLimit (Page 537)	Activate/deactivate torque limiting
	FB 439	MC_SetCharacteristic (Page 542)	Activate valve characteristic
Synchronous operation	FB 420	MC_GearIn (Page 546)	Start gearing
	FB 440	MC_GearInSuperImposed (Page 561)	Start superimposed gearing
	FB 422	MC_GearOut (Page 556)	End gearing
	FB 442	MC_GearOutSuperImposed (Page 572)	End superimposed gearing
	FB 421	MC_CamIn (Page 576)	Start camming
	FB 441	MC_CamInSuperImposed (Page 596)	Start superimposed camming
	FB 423	MC_CamOut (Page 591)	End camming
	FB 443	MC_CamOutSuperImposed (Page 605)	End superimposed camming
	FB 424	MC_Phasing (Page 609)	Change phase shift between the leading axis and following axis
	FB 444	MC_PhasingSuperImposed (Page 614)	Change superimposed phase shift
Advanced functions	FB 430	MC_CamSwitch (Page 646)	Position-based cam / uni-directional output cam
	FB 431	MC_CamSwitchTime (Page 653)	Time-based cam
	FB 461	MC_CamTrack (Page 658)	Activate cam track
	FB 462	MC_ReadCamTrackData (Page 664)	Read out cam track
	FB 463	MC_WriteCamTrackData (Page 668)	Write cam track
	FB 432	MC_ExternalEncoder (Page 677)	External encoders
	FB 433	MC_MeasuringInput (Page 672)	Measuring inputs
Cam disc	FB 434	MC_CamClear (Page 621)	Clear cam
	FB 435	MC_CamSectorAdd	Add cam sector
	FB 436	MC_CamInterpolate (Page 633)	Interpolate cam
	FB 438	MC_GetCamPoint	Read points from cam

Function	No.	Name	Description
Basic function	FB 402	MC_Reset (Page 682)	Acknowledge errors/interrupts
	FB 460	MC_ActivateTO (Page 686)	Activate / deactivate technology object
	FB 457	MC_ActivateDPSlave (Page 692)	Activate / deactivate DP slave
	FB 406	MC_ReadSysParameter (Page 695)	Read parameter
	FB 407	MC_WriteParameter (Page 699)	Change parameter
	FB 450	MC_ReadPeriphery (Page 708)	Read technology I/O
	FB 451	MC_WritePeriphery (Page 714)	Write technology I/O
	FB 453	MC_ReadRecord (Page 719)	Read a data record
	FB 454	MC_WriteRecord (Page 724)	Write data record
	FB 455	MC_ReadDriveParameter (Page 728)	Read drive parameter
	FB 456	MC_WriteDriveParameter (Page 733)	Write drive parameter

6.1.2 List of technology functions, sorted by numbers

Valid for Integrated Technology with firmware version V4.1.x

No.	Name	Function	Description
FB 401	MC_Power (Page 443)	Single axis	Enable/disable axis
FB 402	MC_Reset (Page 682)	Basic function	Acknowledge errors/interrupts
FB 403	MC_Home (Page 454)	Single axis	Home/set axis
FB 404	MC_Stop (Page 465)	Single axis	Stop axis and prevent new motion commands
FB 405	MC_Halt (Page 470)	Single axis	Normal stop
FB 406	MC_ReadSysParameter (Page 695)	Basic function	Read parameter
FB 407	MC_WriteParameter (Page 699)	Basic function	Change parameter
FB 409	MC_ChangeDataset (Page 475)	Single axis	Change data set
FB 410	MC_MoveAbsolute (Page 481)	Single axis	Absolute positioning
FB 411	MC_MoveRelative (Page 495)	Single axis	relative positioning
FB 412	MC_MoveAdditive (Page 509)	Single axis	Relative positioning to current target position
FB 413	MC_MoveSuperImposed (Page 515)	Single axis	Superimposed positioning
FB 414	MC_MoveVelocity (Page 521)	Single axis	Motion with speed preset
FB 415	MC_MoveToEndPos (Page 530)	Single axis	Move to fixed endstop / terminals
FB 420	MC_GearIn (Page 546)	Synchronous operation	Start gearing
FB 421	MC_CamIn (Page 576)	Synchronous operation	Start camming
FB 422	MC_GearOut (Page 556)	Synchronous operation	End gearing
FB 423	MC_CamOut (Page 591)	Synchronous operation	End camming

No.	Name	Function	Description
FB 424	MC_Phasing (Page 609)	Synchronous operation	Change phase shift between the leading axis and following axis
FB 430	MC_CamSwitch (Page 646)	Extended function	Position-based cam / uni-directional output cam
FB 431	MC_CamSwitchTime (Page 653)	Extended function	Time-based cam
FB 432	MC_ExternalEncoder (Page 677)	Extended function	External encoders
FB 433	MC_MeasuringInput (Page 672)	Extended function	Measuring inputs
FB 434	MC_CamClear (Page 621)	Cam disc	Clear cam
FB 435	MC_CamSectorAdd	Cam disc	Add cam sector
FB 436	MC_CamInterpolate (Page 633)	Cam disc	Interpolate cam
FB 437	MC_SetTorqueLimit (Page 537)	Single axis	Activate/deactivate torque limiting
FB 438	MC_GetCamPoint	Cam disc	Read points from cam
FB 439	MC_SetCharacteristic (Page 542)	Single axis	Activate valve characteristic
FB 440	MC_GearInSuperImposed (Page 561)	Synchronous operation	Start superimposed gearing
FB 441	MC_CamInSuperImposed (Page 596)	Synchronous operation	Start superimposed camming
FB 442	MC_GearOutSuperImposed (Page 572)	Synchronous operation	End superimposed gearing
FB 443	MC_CamOutSuperImposed (Page 605)	Synchronous operation	End superimposed camming
FB 444	MC_PhasingSuperImposed (Page 614)	Synchronous operation	Change superimposed phase shift
FB 450	MC_ReadPeriphery (Page 708)	Basic function	Read technology I/O
FB 451	MC_WritePeriphery (Page 714)	Basic function	Write technology I/O
FB 453	MC_ReadRecord (Page 719)	Basic function	Read a data record
FB 454	MC_WriteRecord (Page 724)	Basic function	Write data record
FB 455	MC_ReadDriveParameter (Page 728)	Basic function	Read drive parameter
FB 456	MC_WriteDriveParameter (Page 733)	Basic function	Write drive parameter
FB 457	MC_ActivateDPSlave (Page 692)	Basic function	Activate / deactivate technology object
FB 460	MC_ActivateTO (Page 686)	Basic function	Activate / deactivate DP slave
FB 461	MC_CamTrack (Page 658)	Extended function	Activate cam track
FB 462	MC_ReadCamTrackData (Page 664)	Extended function	Read out cam track
FB 463	MC_WriteCamTrackData (Page 668)	Extended function	Write cam track

Note

If you want to use existing FBs from other projects, you can rename the FBs in SIMATIC Manager. Note that the documentation refers to the default FB numbers.

6.1.3 List of technology functions, sorted alphabetically

Valid for Integrated Technology with firmware version V4.1.x

Name	No.	Function	Description
MC_ActivateDPSlave (Page 692)	FB 457	Basic function	Activate / deactivate DP slave
MC_ActivateTO (Page 686)	FB 460	Basic function	Activate / deactivate technology object
MC_CamClear (Page 621)	FB 434	Cam disc	Clear cam
MC_CamIn (Page 576)	FB 421	Synchronous operation	Start camming
MC_CamInSuperImposed (Page 596)	FB 441	Synchronous operation	Start superimposed camming
MC_CamInterpolate (Page 633)	FB 436	Cam disc	Interpolate cam
MC_CamOut (Page 591)	FB 423	Synchronous operation	End camming
MC_CamOutSuperImposed (Page 605)	FB 443	Synchronous operation	End superimposed camming
MC_CamSectorAdd	FB 435	Cam disc	Add cam sector
MC_CamSwitch (Page 646)	FB 430	Extended function	Position-based cam / uni-directional output cam
MC_CamSwitchTime (Page 653)	FB 431	Extended function	Time-based cam
MC_CamTrack (Page 658)	FB 460	Extended function	Activate cam track
MC_ChangeDataset (Page 475)	FB 409	Single axis	Change data set
MC_ExternalEncoder (Page 677)	FB 432	Extended function	External encoders
MC_GearIn (Page 546)	FB 420	Synchronous operation	Start gearing
MC_GearInSuperImposed (Page 561)	FB 440	Synchronous operation	Start superimposed gearing
MC_GearOut (Page 556)	FB 422	Synchronous operation	End gearing
MC_GearOutSuperImposed (Page 572)	FB 442	Synchronous operation	End superimposed gearing
MC_GetCamPoint	FB 438	Cam disc	Read points from cam
MC_Halt (Page 470)	FB 405	Single axis	Normal stop
MC_Home (Page 454)	FB 403	Single axis	Home/set axis
MC_MeasuringInput (Page 672)	FB 433	Extended function	Measuring inputs
MC_MoveAbsolute (Page 481)	FB 410	Single axis	Absolute positioning
MC_MoveAdditive (Page 509)	FB 412	Single axis	Relative positioning to current target position
MC_MoveRelative (Page 495)	FB 411	Single axis	relative positioning
MC_MoveSuperImposed (Page 515)	FB 413	Single axis	Superimposed positioning
MC_MoveToEndPos (Page 530)	FB 415	Single axis	Move to fixed endstop / terminals
MC_MoveVelocity (Page 521)	FB 414	Single axis	Motion with speed preset
MC_Phasing (Page 609)	FB 424	Synchronous operation	Change phase shift between the leading axis and following axis

Name	No.	Function	Description
MC_PhasingSuperImposed (Page 614)	FB 444	Synchronous operation	Change superimposed phase shift
MC_Power (Page 443)	FB 401	Single axis	Enable/disable axis
MC_ReadCamTrackData (Page 664)	FB 462	Extended function	Read out cam track
MC_ReadDriveParameter (Page 728)	FB 455	Basic function	Read drive parameter
MC_ReadPeriphery (Page 708)	FB 450	Basic function	Read technology I/O
MC_ReadRecord (Page 719)	FB 453	Basic function	Read a data record
MC_ReadSysParameter (Page 695)	FB 406	Basic function	Read parameter
MC_Reset (Page 682)	FB 402	Basic function	Acknowledge error
MC_SetCharacteristic (Page 542)	FB 439	Single axis	Activate valve characteristic
MC_SetTorqueLimit (Page 537)	FB 437	Single axis	Activate/deactivate torque limiting
MC_Stop (Page 465)	FB 404	Single axis	Stop axis and prevent new motion commands
MC_WriteCamTrackData (Page 668)	FB 463	Extended function	Write cam track
MC_WriteDriveParameter (Page 733)	FB 456	Basic function	Write drive parameter
MC_WriteParameter (Page 699)	FB 407	Basic function	Change technology object parameters
MC_WritePeriphery (Page 714)	FB 451	Basic function	Write technology I/O
MC_WriteRecord (Page 724)	FB 454	Basic function	Write data record

6.2 Technology functions - Single axes

6.2.1 FB401 MC_Power - Disable/enable axis

Enabling / disabling an axis with FB 401 "MC_Power"

Valid for Integrated Technology with firmware version V4.1.x

Purpose

- Use the "MC_Power" technology function to enable or disable an axis.
- Use the technology function to specify the operating mode of the axis.
- Use the technology function to specify the emergency program for handling the axis when it is disabled or at a CPU transition to STOP.
- Use the technology function to control the integrated brake control of the drivers (for example when operating "hanging axes"). The brake control of the following drives is supported:
  – SIMODRIVE 611U universal
  – SINAMICS S120
  – MASTERDRIVES Motion Control Plus

Supported for

- Speed-controlled axes
- Positioning axes
- Synchronization axes

Prerequisites

- There may be no active errors at the technology DB which prevent enabling of the axis.
- The technology function may only be active once per axis.

Overriding commands

MC_Power commands can not be canceled by any other command.

An MC_Power command does not cancel any other commands with *Enable = TRUE* .

An MC_Power command cancels all other commands to this technology object with *Enable = FALSE*.

New command - active single command (3) (Page 743)

Input parameters

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td>Axis</td><td>INT</td><td>0</td><td colspan="2">Number of the technology DB</td></tr>
<tr><td>Enable</td><td>BOOL</td><td>FALSE</td><td colspan="2">The system attempts to enable the axis as long as Enable = TRUE. The axis is enabled if there are no errors preventing that.
If both the Statusword.Error and Statusword.Errorstop variables are TRUE in the corresponding technology DB, this means that a pending error is preventing the axis enable.
All active commands are canceled if Enable = FALSE. A torque limit set with the "MC_SetTorqueLimit" technology function is maintained after removal of the enable.</td></tr>
<tr><td rowspan="7">Mode</td><td rowspan="7">INT</td><td rowspan="7">0</td><td colspan="2">Axis mode:</td></tr>
<tr><td>Value = 0:</td><td>Default (according to the axis configuration)
Speed-controlled axis:
Speed-controlled
(speed-controlled axis without encoder)
Positioning axis and synchronization axis:
Position- or speed-controlled
("MC_MoveVelocity" PositionControl = FALSE)</td></tr>
<tr><td>Value = 1:</td><td>Enable axis through speed control:
Speed-controlled axis:
Speed-controlled
(speed-controlled axis without encoder)
Positioning axis and synchronization axis:
Axis is enabled speed-controlled</td></tr>
<tr><td>Value = 2:</td><td>Following mode with enabled power unit</td></tr>
<tr><td>Value = 3:</td><td>Following mode with disabled power unit</td></tr>
<tr><td>Value = 4:</td><td>Simulation mode</td></tr>
<tr><td>Value = 5:</td><td>Enable axis for position-controlled operation
(for positioning and synchronous axes only)</td></tr>
</table>

Parameters	Data Type	Initial value	Description	
StopMode	INT	*0*	Value = *0*:	*DefaultStop* (Emergency-stop deceleration)
			Value = *1*:	*FastStop* (hardware limit of acceleration)
			Value = *2*:	*TimeStop* Axis stop within the configured emergency stop time (STOP time)
			Value = *3*:	*OFF2Stop* (only at hydraulic axis) The control signal of the axis is changed with a configured ramp to the programmed substitute control signal value (input parameter *QOutputValue)*. Motion commands are canceled with "Enable signals missing" error. Any configured enable output is reset. You can set the ramp in S7T Config at system variable *userdefaultqfaxis.maxderivative.qoutput*. You can alternatively change the ramp value using technology function "MC_WriteParameter", parameter *5009*. The value is defined percent/second. Default is 100%/sec.
			Value = *4*:	*GearStop* The synchronous axis maintains synchronous operation until the leading axis has reached a standstill.
			Value = *5*	*Rampenhalt* (only at real electrical axes) Stop the axis by using the ramp function generator
			Value = *6*	*Schnellhalt* (only at real electrical axes) Stop the axis by using the fast stop ramp
QOutputValue	REAL	*0.0*	Control signal substitute value for Q valve. The substitution value of the control signal is defined as a percentage of the maximum setpoint voltage output for the valve (corresponds with the percentage of the valve opening.) The values can be set within the range from *-100.0 %* to *+100.0 %.* The substitute value is activated if he axis is disabled by setting input parameter *Enable = FALSE* and *StopMode = 3* is selected. Set the substitution value of the control signal to prevent movement of the hydraulic drive when the axis is disabled. A modified value for *QOutputValue* is activated at the positive and negative edges at the *Enable* input parameter. If an invalid value is detected at the negative edge of *Enable* at input parameter *QOutputValue* the "MC_Power" technology function outputs ErrorID *808B* for the duration of one cycle. "In this case, "MC_Power" uses a value for *QOutputValue* which was valid at the time the positive edge was output at *Enable*.	
FOutputValue	REAL	*0.0*	This input parameter is not yet supported by the integrated technology with firmware V3.2.x.	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Status	BOOL	*FALSE*	Axis enable status:	
			FALSE	No enable signal. Axis disabled The axis neither existing, nor accepts new motion commands. The drive power unit is disabled. If the axis is locked the *Status* can change with delay to *FALSE*.
			TRUE	Enabled The axis is only enabled if cyclic communication between the control and drive is active and the actual position value of the active encoder is valid. Check *Statusword.CyclicInterface* and *Statusword.EncoderValid* at the technology DB. The enable signal is reset if an error preventing enabling occurs (indicated by *Statusword.Error = TRUE* and *Statusword.Errorstop = TRUE*at the relevant technology DB.
Busy	BOOL	*FALSE*	*TRUE*	The command is being executed
Error	BOOL	*FALSE*	*FALSE*	Command initiation **without error**.
			TRUE	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
ErrorID	WORD	*0*	ErrorID (Page 443) of the *Error* output parameter.	

NOTICE

An axis switched off due to an error is re-enabled with *Enable = TRUE* after the error has been eliminated and acknowledged.

Note

The "MC_Power" technology function influences the following status bits of the *Statusword* variable at the technology DB:

- *DriveEnable*
- *FollowUpControl*
- *Simulation*

Recommended procedure for disabling an axis

To disable an axis:

1. Stop the drive
2. Disable the axis after the drive has stopped (*Enable = FALSE*).

Parameter *Mode* - Mode of operation

The drive is enabled with a positive edge at input *Enable.*

Axis technology	Modes of operation supported	
	Real axes	Virtual axes
Speed-controlled axis	*Mode* = *0*: speed-controlled *Mode* = *3:* Following mode *Mode* = *4:* Simulation	*Mode* = *0:* speed-controlled
Positioning axis	*Mode* = *0:* Position-controlled *Mode = 3:* Following mode *Mode* = *4:* Simulation	*Mode* = *0:* Position-controlled
Synchronized axis	*Mode* = *0:* Position-controlled *Mode* = *3:* Following mode *Mode* = *4:* Simulation	*Mode* = *0:* Position-controlled

Parameter *StopMode* - Emergency program

An emergency program is started if a drive is disabled while it is in RUN (negative edge at MC_Power) or when the CPU goes into STOP. The reaction of the axis can be defined at input *StopMode*. The *StopMode* is determined at the negative or positive edge at input parameter *Enable* .

Parameter *StopMode* is disabled if torque reduction is active. In this case, the axis is disabled immediately. The drive is brought to a standstill based on its internal deceleration ramp settings.

An emergency program started after the CPU goes into STOP must be executed within the "maximum shutdown time". The CPU goes into STOP on expiration of the "maximum shutdown time". Instead of stopping according to the selected *StopMode*, the drives will stop as defined in the drive configuration.

The "maximum shutdown time" can be configured in the "System clocks" dialog box of S7T Config (**Target system > Set System Clocks** menu command).

All errors must be acknowledged before the restart by calling the "MC_Reset" technology function.

StopMode	Reaction of the axis
Value = *0*: *DefaultStop*	The axis is ramped down by activating the "Emergency stop deceleration" function. The "Emergency stop deceleration" function can be defined in the S7T Config in the **Axis > Limits** dialog box, "Dynamic response" tab, in the "Stop with preprogrammed deceleration ramp" input box.
Value = *1*: *FastStop*	The axis is ramped down with maximum deceleration (**S7T Config > Limits > "Dynamic response" tab > Hardware limits**).
Value = *2*: *TimeStop*	The axis is ramped down within the "Stop time" configured for the emergency-stop (**S7T Config > Default > "Dynamics" tab > Stop time**).
Value = *3*: *OFF2Stop*	The drive is disabled by a pulse inhibit signal and trails to standstill (OFF2). The missing enable signal is indicated by *8040* in the technology DB of the axis.
Value = *4*: *GearStop*	Synchronized axes maintain their status until the leading axis has reached a standstill. The axis is then disabled. In any other case, the axis is ramped down based on the default axis configuration (for speed-controlled axes, for example).
Value = *5*: *Rampenhalt*	Stop the axis by using the ramp function generator
Value = *6*: *Schnellhalt*	Stop the axis by using the fast stop ramp

Note

The selected *StopMode* is irrelevant if an axis is operating in following mode, as the axis can not be decelerated dynamically! A rotating axis will therefore trail to a standstill within an indefinite time.

NOTICE

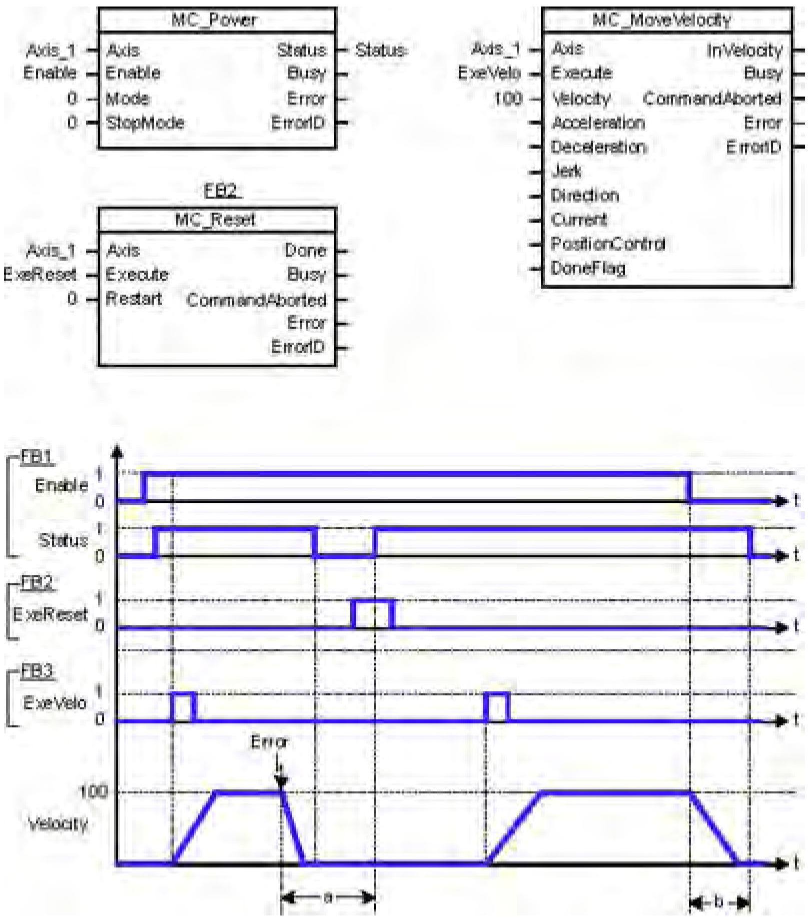
When *StopMode = 4* and camming is set, the following axis may restart before the leading axis has reached a standstill!

MC_Power, MC_Reset - example

The example shows the interaction between the technology functions "MC_Power" (axis enable), "MC_Reset" (error acknowledgment) and a technology function for axis motion (here "MC_MoveVelocity").

a) The axis was enabled and accelerated to a speed of 100. An error resets the axis enable signal. The error is eliminated and acknowledged with "MC_Reset". The axis is enabled again.

b) The axis was accelerated to a speed of 100. The *Enable* signal is cleared from technology function "MC_Power". The axis now ramps down at a defined deceleration rate and is then disabled.

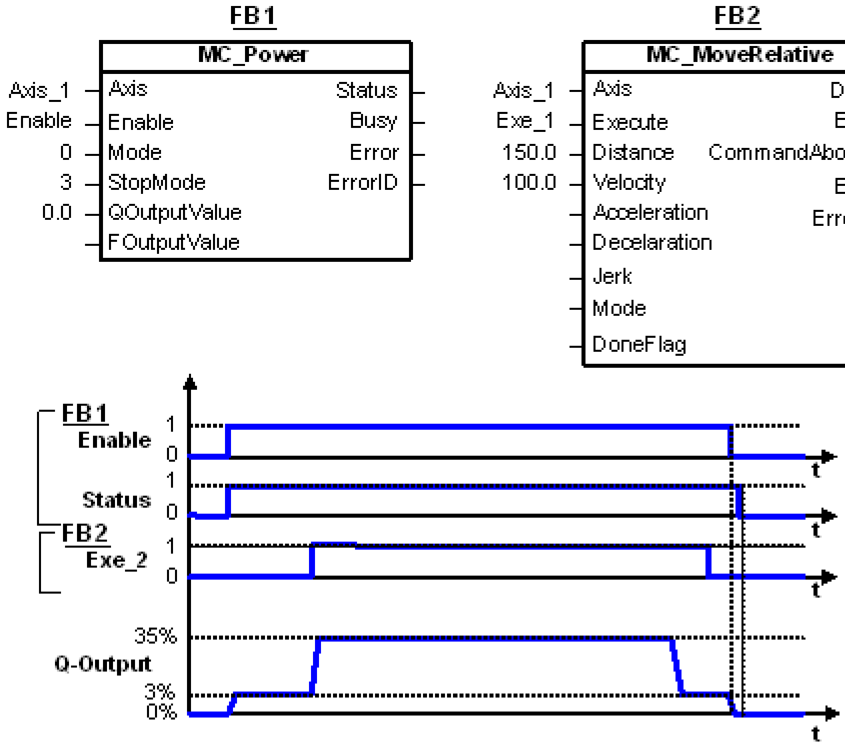


MC_Power - Example - "Hydraulic axis"

A proportional valve is to be used to run two axes. Both axes have different valve profiles. Velocity v = *0* at "axis 1" is proportional to valve position *3 %*. Velocity v = *0* at "axis 2" is proportional to valve position *0 %*.

In this example, "Axis 1" is enabled for position control. A master signal of *3 %* is therefore output to hold the axis at its position. "Axis 1" is positioned relative by a small value. At the end of positioning, the axis is again "held" at its position with v = *0*. When "Axis 1" is disabled, the value of the manipulated variable is ramped down to the programmed substitute value *QOutputValue = 0 %*.

The manipulated variable will have the correct value after "Axis 2" is enabled by the call of MC_Power.

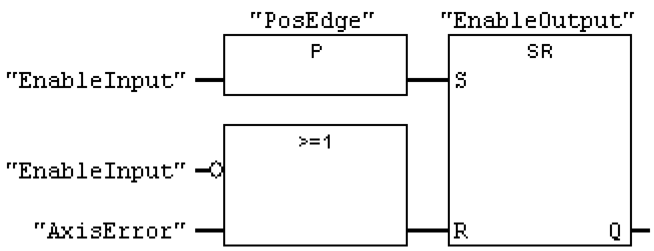


Edge-triggered drive enable function

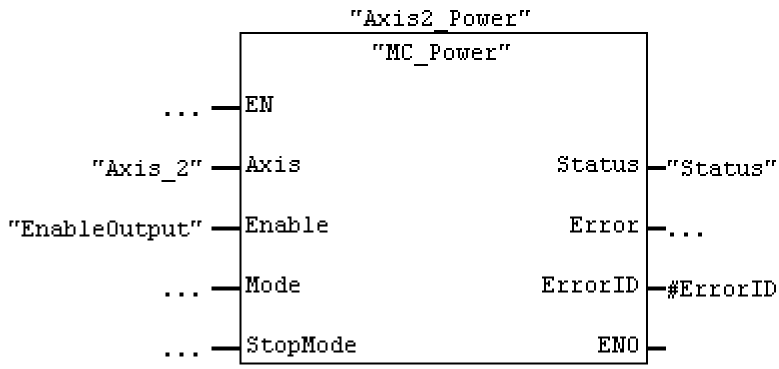
Technology function "MC_Power" enables the drive, based on the logical status at input *Enable*. Logic operations you can use in order to enable the drive only at the positive edge of a control signal:

Network 1 - Control logic for edge evaluation

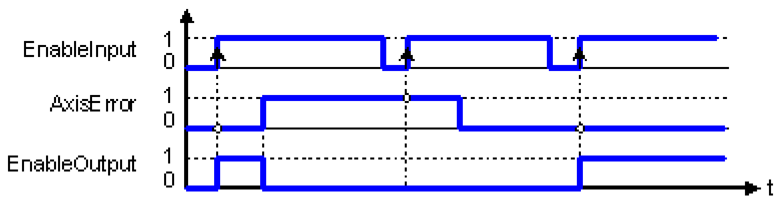
EnableInput	Control input for the axis enable signal
AxisError	*0* No error *1* An error is active at the axis
EnableOutput	Output signal (to (to "MC_Power"; input *Enable*)



Network 2 - Axis enable with "MC_Power"



Corresponding signal diagram



MC_Power - ErrorIDs

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware up to V3.1.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project / software
8005	Command canceled because command memory is in use by another process.	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at these technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done, CommandAborted* or *Error* is *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Concerns all input parameters of data type REAL, and the input parameters *Mode* or *StopMode*. Example: An invalid value was entered at *Mode* or *Stopmode*, for example, *Mode = 129*.
8044	Command not supported by the technology object.	For example, using the "MC_Power" at an output cam.
8045	Command not allowed in current state	Possible causes: • The axis was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804C	Command output rate too high.	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
804E	Only one instance allowed per axis	The technology function may only be active at one instance per axis. Another "MC_Power" (*Busy = TRUE*) command is already active at the specified axis. Use only one instance, or terminate the active command.
804F	Change of the axis at "MC_Power" is not allowed.	An axis was disabled (Enable *1->0*) while an axis other than the one set at the enable parameter was set at input parameter *Axis*. You can only disable an axis which was actually enabled by this instance of "MC_Power".

ErrorID	Error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB.	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis* . Download the current technology to the target system, or change the DB number at input parameter *Axis*.. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.2 FB403 MC_Home - Home/set axis

Homing / setting axes with FB403 "MC_Home"

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware up to V3.2.x

Purpose

- The "MC_Home" technology function establishes a position-based correlation between the control and the mechanical system by means of a measuring system.
- Active and passive homing of an axis

  The homing mode is always defined at the technology function The homing mode and direction parameters must be set in S7T Config
- Setting a position value

  The assigned position value is an absolute value which allows absolute and relative motions.
- Relative correction / offset of the actual value
- Correction of the setpoint in the base coordinate system and superimposed coordinate system. Only relative offset is supported
- Absolute encoder adjustment.

Supported for

- Positioning axes
- Synchronization axes

Note

Virtual axes only support *Mode* = *3, 4, 6* and *7*.

Axes with incremental encoder only support *Mode = 0, 1, 2, 3, 4, 6* and *7*.

Axes with absolute encoder only support *Mode = 3, 4, 5, 6* and *7* .

You can only set *Mode* = *3, 4, 6* and *7* if "no mode" was selected in the homing configuration of the axis.

The axis status remains unchanged (homed/not homed) in *Mode* = *4, 6* and *7*.

Mode = 2, 3 and *5* can only be used if the axis is not operating in "speed-controlled mode" ("MC_MoveVelocity" - *PositionControl = TRUE)*.

Prerequisites

- The axis must be enabled for position-controlled operation for *Mode* = *0, 1* and *2*.
- No MC_Stop command may be active when the axis is operated in *Mode = 0* and *1*.
- The drive interface IM 174/ADI4 does not support the simultaneous execution of MC_MeasuringInput and MC_Home commands.

Interaction of commands

New command - active single command (2) (Page 741)
New command – active commands (2) (Page 746)

Input parameters

Parameters	Data Type	Initial value	Description
Axis	INT	*0*	Number of the technology DB
Execute	BOOL	*FALSE*	Start of the command at the positive edge
Position	REAL	*0.0*	Absolute position when reaching the home position, or position setpoint or position correction value.
Mode	INT	*0*	Homing mode:
			Value = *0*: Active homing: Reference point approach in accordance with the axis configuration
			Value = *1*: Active homing: Reference point approach in accordance with the axis configuration The value of input parameter *Position* is used as reference point coordinate.
			Value = *2*: Passive homing: Homing according to the configuration in S7T Config. The value of input parameter *Position* is used as reference point coordinate.
			Value = *3*: Direct homing: The current position is assigned the value of the *Position* input parameter.
			Value = *4*: Correction of the actual position: (New actual position = Current actual position - *Position* parameter).
			Value = *5*: Absolute encoder adjustment: The current position is assigned the value of the *Position* input parameter. The position shift which results from the absolute encoder adjustment is retained in case of power failure. The axis must be at a standstill in order to allow precise adjustment. Any correction of the position when operating in *Mode = 3, 4, 6* or *7* will be retained after absolute encoder adjustment. The axis must be operated in position-controlled mode for absolute encoder adjustment. For details, refer to the description of input parameter *PostionControl* for "MC_MoveVelocity".
			Value = *6*: Setpoint correction in the base coordinate system: Any active superimposed motion is based on the corrected position setpoint. (new position setpoint = current base position – input parameter *Position*)

Parameters	Data Type	Initial value	Description	
			Value = *7*:	Setpoint correction in the superimposing coordinate system: (new position setpoint = current superimposing position – *Position* input parameter) Note that any position corrections (*Mode = 3, 4, 6* or *7*) are activated in addition to the absolute encoder adjustment. Only the absolute encoder adjustment is activated after POWER OFF or restart ("MC_Reset", *Restart = TRUE*).
			Value = *8*	Cancel passive homing A started passive homing command (*Mode = 2*) is canceled.
			"MC_Home" can also be executed in *Mode = 2* to *7* without enabled axis. If operating an axis with absolute encoder, the value at the *Statusword.EncoderValid* parameter of the technology DB must be *TRUE* (this signal is set with delay after axis restart) before the MC_Home command is initiated.	
DoneFlag	INT	0	DoneFlag (Page 750) generation in the MCDevice DB	

Notes on *Mode* = *7*.

The superimposing coordinate system can be modified by superimposing motions or homing commands ("MC_Home" *Mode* = *7*). The coordinates of the superimposed coordinate system are transferred to the basic coordinate system according to the settings at *TypeOfAxis.DecodingConfig.transferSuperimposed* configuration parameter. The superimposing coordinate system is reset in this operation.

Set this configuration parameter to determine the time at which the coordinates of the superimposing coordinate system should be activated at the base coordinate system.

Settings at the *TypeOfAxis.DecodingConfig.transferSuperimposed* configuration parameter:

- *TRANSFER_STANDSTILL* (default):
  - Activation after the axis is at a standstill - the axis must be desynchronized
  - Activation when the axis changes to following mode ("MC_Power" *Mode* = *2* or *3*)
  - Activation at axis restart ("Reset" with *Restart = TRUE*).
  - Activation when commands are output to override the basic motion
- *TRANSFER_MERGE*:
  - Activation when the axis changes to following mode ("MC_Power" *Mode = 2* or *3*)
  - Activation at axis restart ("Reset" with *Restart = TRUE*).
  - Activation when commands are output to override the basic motion
- *TRANSFER_RESET*:
  - Activation when the axis changes to following mode ("MC_Power" *Mode = 2* or *3*)
  - Activation at axis restart ("Reset" with *Restart = TRUE*).

NOTICE
Caution: In *Mode = 7*, the override response of the modules will differ. Superimposing motions are no longer canceled by basic motions (exceptions). *Mode = 7* is always executed in the base coordinate system for axes operating in following mode.

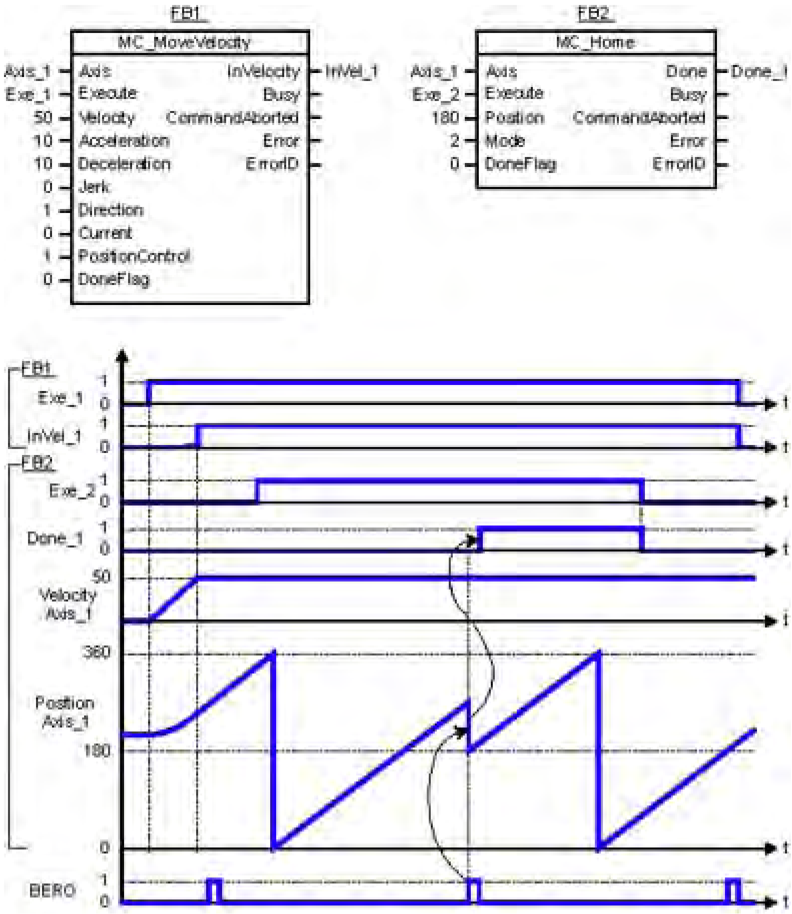
Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command completed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE:*	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 454) of the *Error* output parameter.	

MC_Home - Example - "Passive homing"

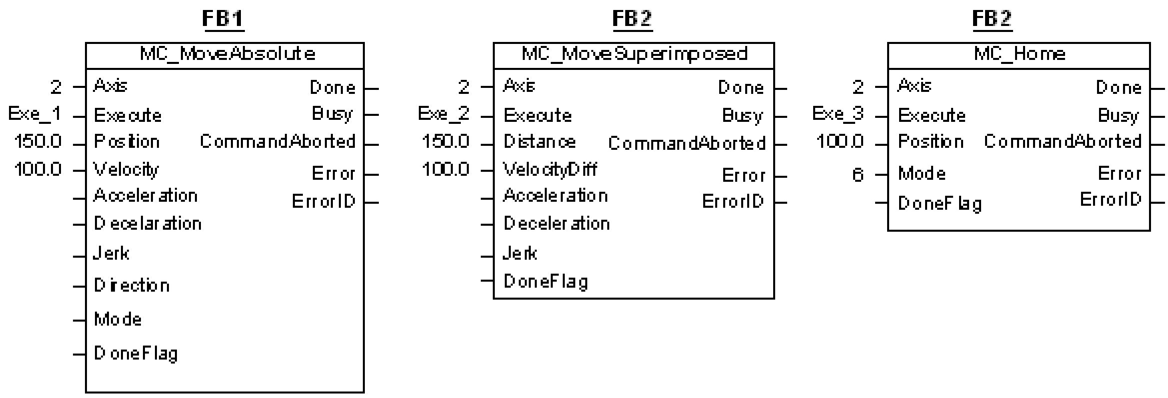
The example demonstrates the reaction to passive (on-the-fly) homing.

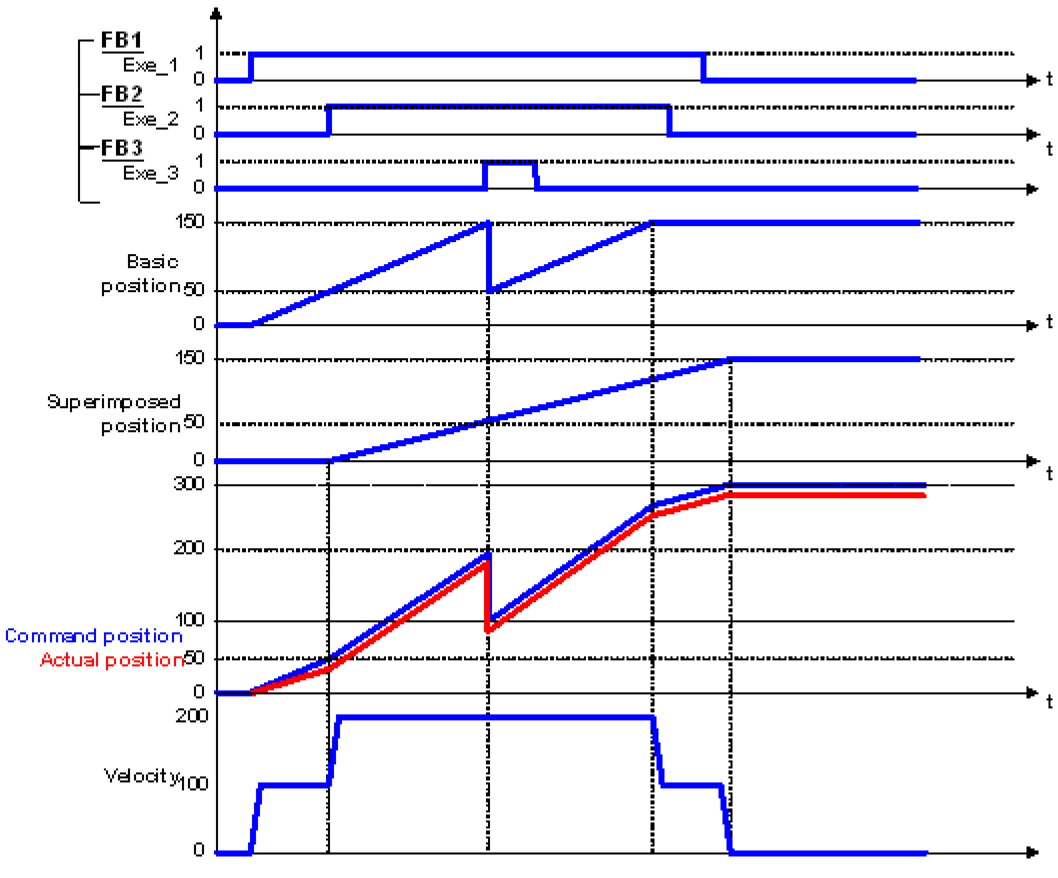
"Axis_1" is started by calling "MC_MoveVelocity". Passive homing is enabled ("MC_Home"; *Mode* = 2). At the next synchronization event (in this case: edge at the external zero mark) the actual axis position is set to *180°*.



MC_Home - Example "Mode 6"

A position setpoint correction only affects the base coordinate system, meaning that superimposed synchronism is not affected by this change. The example shows the start of "MC_MoveAbsolute" with the successive, briefly delayed start of "MC_MoveSuperImposed". The position in the base coordinate system is corrected by *100* while the axis is in motion.



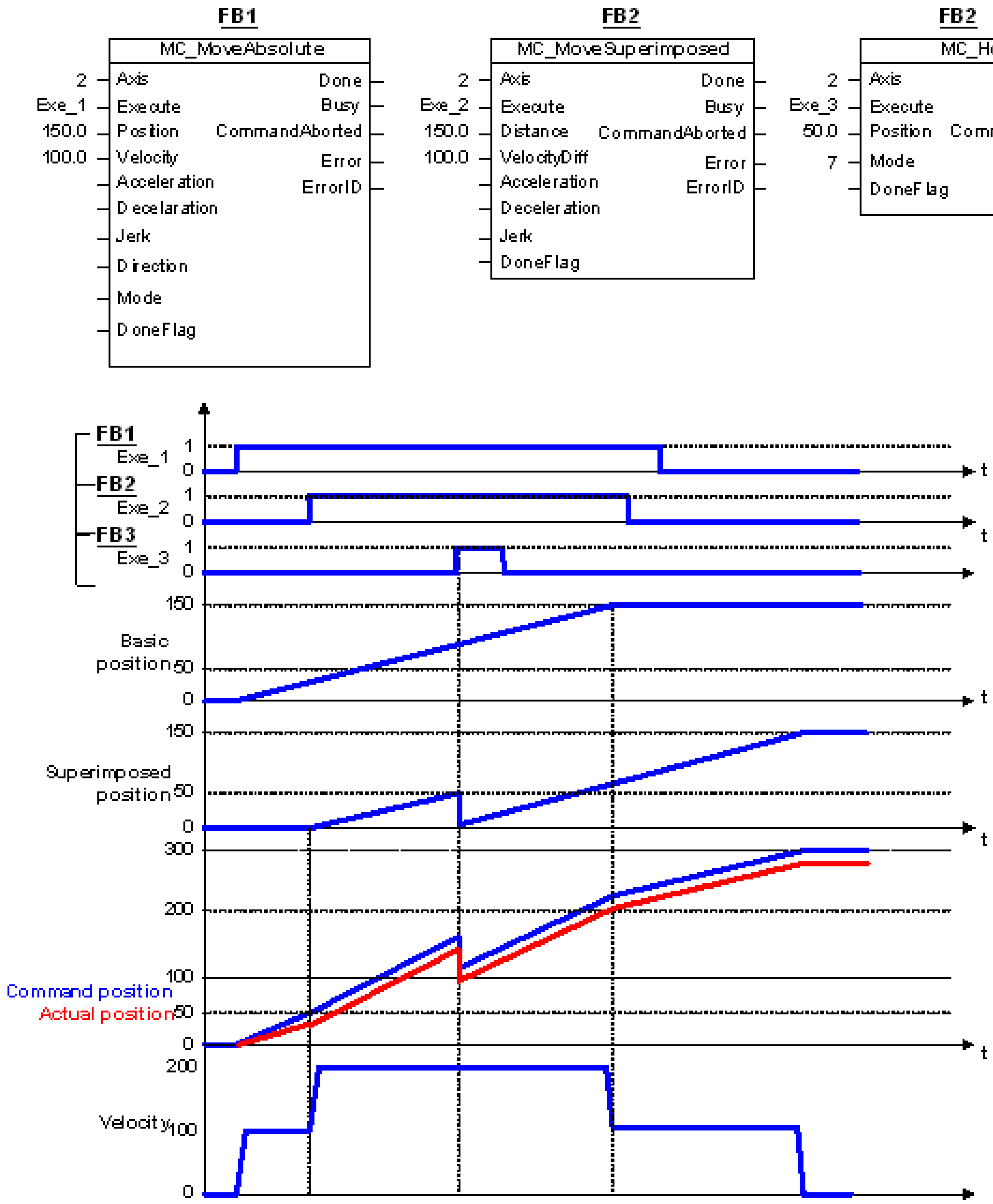


MC_Home - Example "Mode 7"

The position change initiated by a superimposing command is transferred by default from the superimposing to the base coordinate system after the axis has reached a standstill. This response can be configured at the axis configuration parameter *TypeOfAxis.DecodingConfig.transferSuperimposed*. *Transfer_Reset* is set instead of the default setting in the next example. The superimposing coordinate system is therefore only transferred to the base coordinate system after the axis enable signals are canceled.

The example shows the start of "MC_MoveAbsolute" with the successive, briefly delayed start of "MC_MoveSuperImposed". Superimposing positioning is to be corrected by *50*.

Mode *6* or *7* corrections adjust the position setpoint of the relevant coordinate system. The actual position is followed based on the evaluation of the actual following error.


FB1
MC_MoveAbsolute
2
Axis
Done
Exe_1
Execute
Busy
150.0
Position
CommandAborted
100.0
Velocity
Error
Acceleration
ErrorID
Decelaration
Jerk
Direction
Mode
DoneFlag
FB2
MC_MoveSuperimposed
2
Axis
Done
Exe_2
Execute
Busy
150.0
Distance
CommandAborted
100.0
VelocityDiff
Error
Acceleration
ErrorID
Deceleration
Jerk
DoneFlag
FB2
MC_Home
2
Axis
Done
Exe_3
Execute
Busy
50.0
Position
CommandAborted
7
Mode
Error
DoneFlag
ErrorID
FB1
Exe_1
FB2
Exe_2
FB3
Exe_3
Basic position
Superimposed position
Command position
Actual position
Velocity
t

MC_Home - ErrorIDs

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware up to V3.2.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at these next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8022	No actual value at the axis / external encoder	Encoder or data bus not ready, for example
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *DoneFlag* input parameters.
8044	Command not supported by the technology object	Sending a homing command to a speed-controlled axis or to an output cam.

ErrorID	Error message	Description / to correct or avoid errors
8045	Command not allowed in current state	Possible causes: • MC_Stop command active • The axis is not operated in position-controlled mode during absolute encoder adjustment. For details, refer to the description of input parameter *PostionControl* for "MC_MoveVelocity". • Active homing is requested (*Mode = 0, 1*) and an MC_MoveToEndPos command is active. • The selected axis is in speed-controlled mode ("MC_MoveVelocity", *PositionControl =FALSE*) • Power is disabled with servo interlocking (for example due to an emergency stop). • A parameter of the technology object was changed. This change to the parameter requires a restart, but the restart has not yet been carried out. • The "Collect changes" button is active in the expert list for the technology object (value of system variable *activationmodechangedconfigdata = collect_changed_config_data).* • A command for canceling passive homing (*Mode = 8*) was started although no passive homing command is active; or the passive homing command was already completed.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
804D	Invalid axis type	This is a virtual axis. Virtual axes do not support Modes *0, 1, 2* and *5*.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DB x, both in OB 1 as well as in O B35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB.	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.3 FB404 MC_Stop - Stopping an axis and preventing new motion commands

Stopping an axis with FB404 "MC_Stop"

Valid for Integrated Technology with firmware V3.2.x or higher

This section describes applications with firmware up and including V3.1.x

Purpose

- The "MC_Stop" technology function stops all motions and brings the axis to a standstill. The standstill position is not defined.
- The command is completed (*Done = TRUE*) after the axis has stopped and *Execute = FALSE* is set. That is, startup of the axis is prevented as long as *Execute = TRUE*. This also applies if the axis was disabled and re-enabled in the meantime with the MC_Power" technology function.
- Define the dynamic response of the axis to stop commands at the input parameters *Jerk* and *Deceleration*.

Supported for

- Speed-controlled axes
- Positioning axes
- Synchronization axes

Prerequisites

Dynamic stopping requires that the axis is enabled for position-/speed-controlled operation. The axis cannot be ramped down dynamically if operated in following mode or if it is in disabled state!

Interaction of commands

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Note

You should not disable the axis by means of "MC_Power" as long as the "MC_STOP" command is active. However, if this does happen, the STOP condition at "MC_Power" cancels the STOP motion you programmed at "MC_Stop". Dynamic axis motions are locked as long as *Execute = TRUE* at "MC_Stop", irrespective of its activation with "MC_Power".

Input parameters

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td>Axis</td><td>INT</td><td>0</td><td colspan="2">Number of the technology DB</td></tr>
<tr><td>Execute</td><td>BOOL</td><td>FALSE</td><td colspan="2">Start of the command at the positive edge</td></tr>
<tr><td rowspan="4">Deceleration</td><td rowspan="4">REAL</td><td rowspan="4">-1.0</td><td colspan="2">Deceleration (decreasing energy in the motor):</td></tr>
<tr><td>Value > 0:</td><td>Use the defined value</td></tr>
<tr><td>Value = 0:</td><td>Not permitted</td></tr>
<tr><td>Value < 0:</td><td>Use default</td></tr>
<tr><td rowspan="4">Jerk</td><td rowspan="4">REAL</td><td rowspan="4">-1.0</td><td colspan="2">Jerk:</td></tr>
<tr><td>Value > 0:</td><td>Use the defined value</td></tr>
<tr><td>Value = 0:</td><td>Use trapezoidal motion profile</td></tr>
<tr><td>Value < 0:</td><td>Use default</td></tr>
<tr><td>DoneFlag</td><td>INT</td><td>0</td><td colspan="2">DoneFlag (Page 750) generation in the MCDevice DB</td></tr>
</table>

The axis is stopped at the maximum deceleration if the value set at *Deceleration* > *+1E+12*.

The axis is stopped with maximum jerk if the value set at *Jerk* > *+1E+12*.

WARNING

"MC_Stop" is **not** executed if a numerical value which cannot be represented as REAL value is set at the *Deceleration* or *Jerk* input parameters.

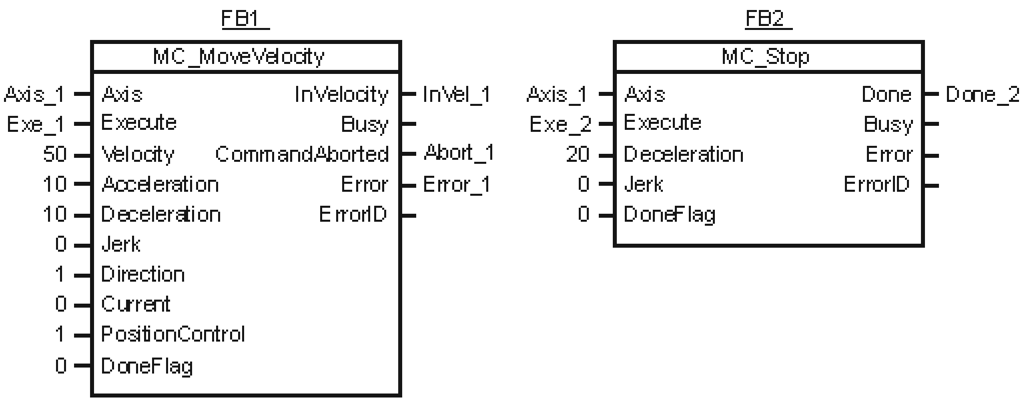
Output parameters (status outputs)

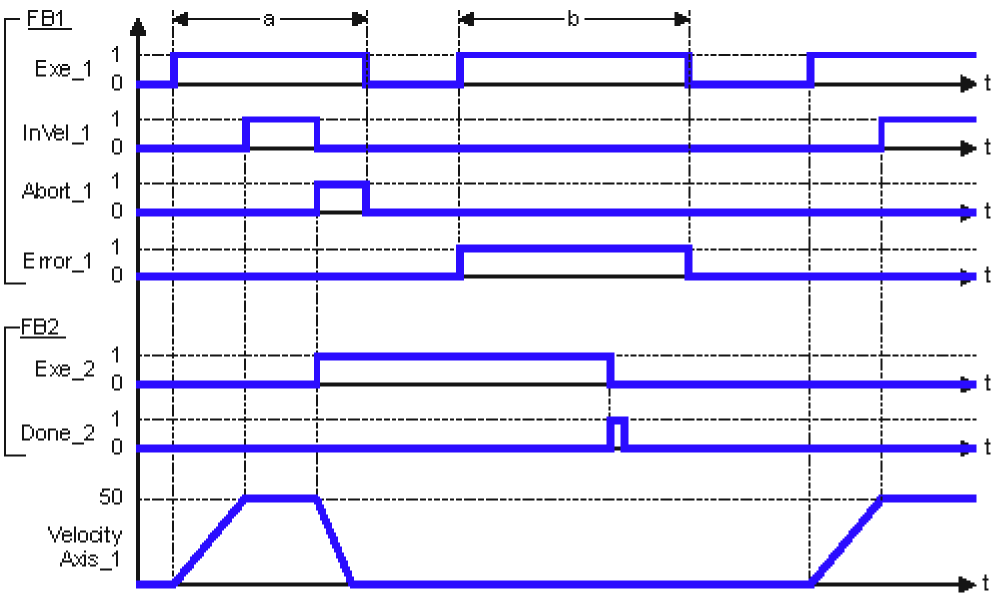
<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td>Done</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: Zero velocity reached and input Execute = FALSE (set for the duration of one cycle)</td></tr>
<tr><td>Busy</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: Command in process</td></tr>
<tr><td>CommandAborted</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: The command was canceled by another MC_Stop command.</td></tr>
<tr><td rowspan="2">Error</td><td rowspan="2">BOOL</td><td rowspan="2">FALSE</td><td>TRUE:</td><td>Command initiation with error. The command is not executed. For information about the cause, refer to the ErrorID.</td></tr>
<tr><td>FALSE:</td><td>Command initiation without error.</td></tr>
<tr><td>ErrorID</td><td>WORD</td><td>0</td><td colspan="2">ErrorID (Page 465) of the Error output parameter.</td></tr>
</table>

MC_Stop - example

a) A rotary axis is ramped down by the call of technology function "MC_Stop".

b) The axis rejects motion commands as long as the *Execute = TRUE* parameter is set at the "MC_Stop" technology function.. The "MC_MoveVelocity" technology function reports an error to indicate the active MC_Stop command.





MC_Stop - ErrorIDs

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware up to V3.1.x

ErrorID	Warning message	Description / to correct or avoid errors
0000	No error	-
0021	Dynamic response values are limited	The dynamic values of the command (*Deceleration* or *Jerk*) are limited because they exceed configured limits
0028	Illegal parameter value was ignored	Invalid value at an input parameter. The value is ignored and the default is used instead. Check the input parameter values and correct the invalid value.

ErrorID	Error message	Description / to correct or avoid errors
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at these next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *DoneFlag* input parameter.
8044	Command not supported by the technology object	"MC_Stop" command to an output cam, for example.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.4 FB405 MC_Halt - Normal stop

Normal stop with FB 405 "MC_Halt"

Purpose

- The "MC_Halt" technology function stops all motions and brings the axis to a standstill. The standstill position is not defined.
- Define the dynamic response of the moving axis at the input parameters *Jerk* and*Deceleration*.
- The command is completed after the axis has reached a standstill, or if it is canceled by a new motion command.

Supported for

- Speed-controlled axes
- Positioning axes
- Synchronization axes

Prerequisites

- The axis is enabled for speed- or position-controlled operation
- No active MC_Stop command

Interaction of commands

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Deceleration	REAL	*-1.0*	Deceleration (decreasing energy in the motor):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not allowed (with the exception of an axis in stop state)
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk:	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

Output parameters (status outputs)

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td>Done</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: Zero velocity reached
If a new superimposed command is initiated while the MC_Halt command is being executed, the active MC_Halt command can be terminated with Done = TRUE , even though a superimposed motion is still active.
Further information is available in New command - active single command (2) (Page 741).</td></tr>
<tr><td>Busy</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: Command in process</td></tr>
<tr><td>CommandAborted</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: The command was canceled by another command or as a result of error during its execution.
If no error is displayed in the ErrorStatus of the technology DB, the command was canceled by a subsequent command.
If an error is indicated in the ErrorStatus tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.</td></tr>
<tr><td rowspan="2">Error</td><td rowspan="2">BOOL</td><td rowspan="2">FALSE</td><td>TRUE:</td><td>Command initiation with error. The command is not executed. For information about the cause, refer to the ErrorID.</td></tr>
<tr><td>FALSE:</td><td>Command initiation without error.</td></tr>
<tr><td>ErrorID</td><td>WORD</td><td>0</td><td colspan="2">ErrorID (Page 470) of the Error output parameter.</td></tr>
</table>

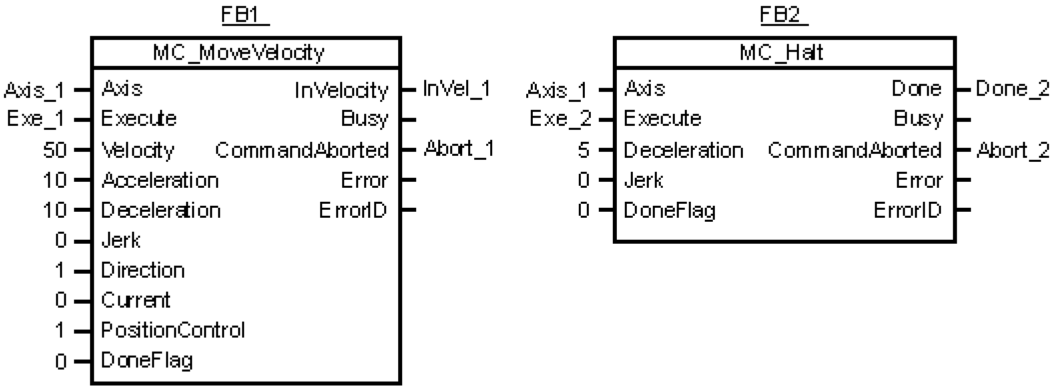
Note

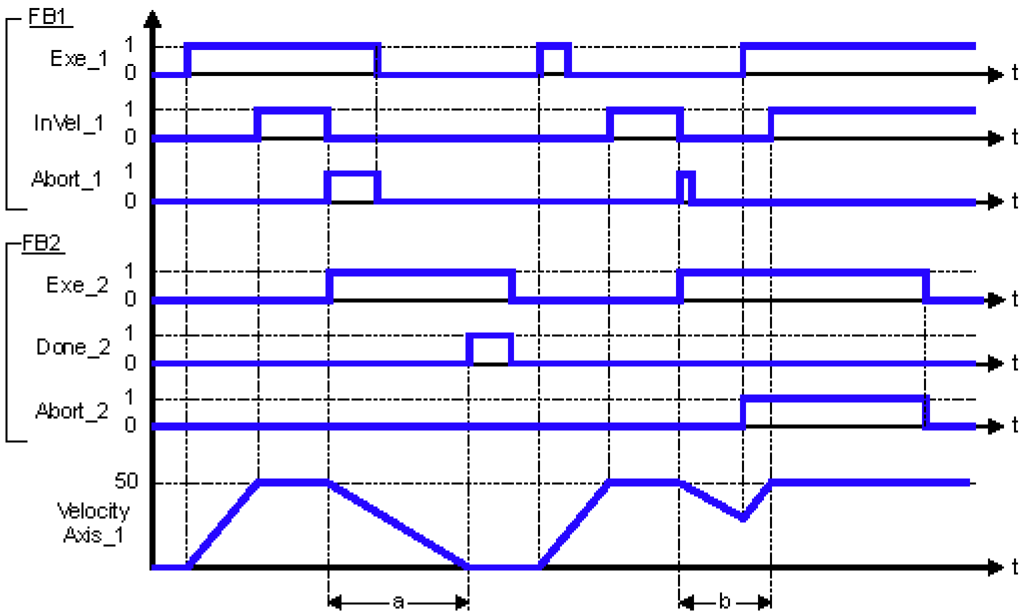
Note that the Technology CPU reduces any active acceleration, based on dynamic parameter settings (*Jerk*). The axis velocity may still increase for this reason after the MC_Halt command has been initiated.

MC_Halt - example

a) A rotary axis is halted by the call of the "MC_Halt" technology function.

b) Another motion command overrides the active MC_Halt command. By contrast to "MC_Stop", this action is supported for the "MC_Halt" command.





MC_Halt - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at these next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *DoneFlag* input parameter.
8044	Command not supported by the technology object	Sending "MC_Halt" to an external encoder, for example.
8045	Command not allowed in current state	MC_Stopactive.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called in the instance DBx, in OB 1, and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*.. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.5 FB409 MC_ChangeDataset - Data record changeover

Data record changeover with FB409 "MC_ChangeDataset"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The technology function can be used to change between the data records of an axis.
- Several data sets can be used, for example to:
  - Change over controller data while the system is running.
  - changeover the encoder used while the system is in run (motor encoder, machine encoder, ...).

Supported for

- Speed-controlled axes
- Positioning axes
- Synchronization axes

Prerequisites

- The corresponding data records must have been configured at the relevant technology object.
- The encoders must be interconnected with the technology object in order to allow the changeover. The encoders must return valid values at the time of changeover. You can verify this by calling the "MC_ReadSysParameter" technology function with parameter number *4050*.

Interaction of commands

New command - active single command (3) (Page 743)

Input parameters

Parameters	Data type	Initial value	Description	
Axis	INT	*0*	Number of the axis technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
SyncEncoder	BOOL	*FALSE*	*FALSE*:	without encoder synchronization
			TRUE:	With encoder synchronization
			Input parameter *SyncEncoder* is only activated if different encoders were defined in the data records.	
Dataset	INT	*1*	Number of the data record to be enabled	
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: The data record was changed The technology function immediately returns Done = TRUE *Done = TRUE* if there is no difference between the defined and the active data records.
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information on the cause, refer to the *ErrorID*.
			FALSE: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 475) of the *Error* output parameter.

Conditions of data record changeover

The time at which the data record is changed is defined by the configuration variable *TypeOfAxis.NumberOfDataSets.changeMode*. The variable can assume the following values:

- NEVER:
  Data record changeover not allowed. An "MC_ChangeDataset" command was terminated due to an error (*ErrorID = 8045*).
- *IN_POSITION*: (default setting in S7T Config after an axis was inserted)
  The data record is changed after the active motion has reached the programmed positioning window. The axis must be enabled for position-controlled operation. The data record is changed immediately after an axis is re-enabled.
  The data record is not changed at synchronized following axes (*InSync = TRUE* or *InGear = TRUE*). Set the *IN_STANDSTILL* or *IMMEDIATELY* value for following axes.
- *IN_STANDSTILL*:
  The data record is changed after the the relevant axis has returned the standstill. The data record is changed immediately after the axis has stopped
- *IMMEDIATELY:*
  The data record is changed immediately

You can modify this response in the expert list of S7T Config You could also modify the configuration variable by means of the "MC_WriteParameter" technology function.

Note

Differences in the data records in terms of parameters which have an influence on the generation of control variables may also cause a step response of the axis in standstill state. This status has an impact on gear ratio parameters or on all controller parameters, for example.

Encoder synchronization

If the current encoder does not match the encoder of the new data record the encoders are synchronized according to the settings at input parameter *SyncEncoder* . The encoder synchronization sets the actual position value of the current encoder at the new encoder. The data record is changed over when synchronization is successfully completed.

Both encoders must rotate at the same speed in order to ensure precise synchronization. Slip between the encoders or one of the encoders being in idle state will prevent the precise interconnection of the encoder with the active process. Synchronization of the encoders therefore fails, due to the difference between encoder values which develops in the period between encoder synchronization and data record changeover.

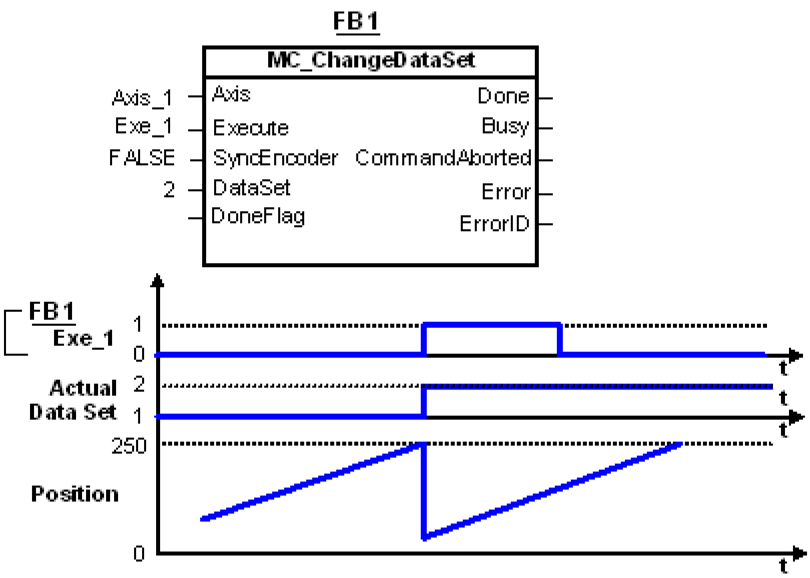
The encoder synchronization is repeated cyclically if the data record cannot be changed over immediately.

Note

If you change over the encoder alongside with the data record without synchronizing the encoders, the axis may perform a compensating movement as a result of the different encoder positions.

MC_ChangeDataset - Example - "Encoder changeover"

In the example below we change over the data record at the axis. The position value of the second measurement system is then used without encoder synchronization.

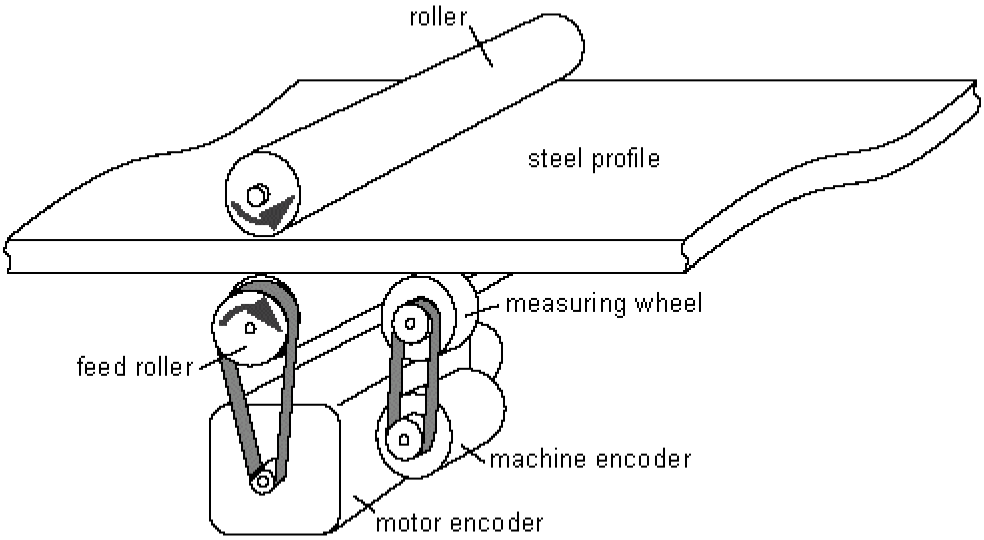


MC_ChangeDataset - Example - "Application"

The diagram below shows the encoder changeover as a typical application of technology function "MC_ChangeDataset".

Continuous steel strip is to be cut to equal length in a processing machine. An additional machine encoder with measuring wheel is integrated in order to eliminate any measuring errors caused by slip of the feed rollers.

The measuring wheel does not return a position value when a new steel strip is fed into the machine, positioning must therefore rely on the motor encoder. A sensor registers the position when the measuring wheel detects the position of the steel strip. When the sensor signal is detected, "MC_ChangeDataset" toggles from motor encoder to machine encoder mode (while the system is in operation and with encoder synchronization). After the encoder changeover, the steel strip can be fed precisely for further processing in position-controlled mode.



MC_ChangeDataset - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at these next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Concerns input parameter *Dataset* or *DoneFlag*.
8044	Command not supported by the technology object	Sending a command to an output cam, for example
8045	Command not allowed in current state	Example: • The value at configuration variable *TypeOfAxis.NumberOfDataSets.changeMode = NEVER*. Data record changeover is not possible. • The technology object is currently performing a restart initiated by "MC_Reset". • The encoder of the new data record does not return valid values Such an error may occur immediately after a STOP to RUN transition, axis restart, or hardware failure.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
804D	Invalid axis type	This is a virtual axis. Virtual axes do not support this function.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.

6.2.6 FB410 MC_MoveAbsolute - Absolute positioning

Absolute positioning with FB410 "MC_MoveAbsolute"

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

Purpose

- The "MC_MoveAbsolute" technological function starts the positioning motion of an axis to an absolute position.
- You define the dynamic response of the axis motion with the input parameters *Velocity*, *Jerk*, *Acceleration* , and *Deceleration*.
- The function is terminated when the target position is reached.
- You can preset the direction of rotation of modulo axes.
- Define whether to trigger an active motion, or whether to append or overlay the motion at input parameter *Mode*.

Supported for

- Positioning axes
- Synchronization axes

Prerequisites

- Axis is enabled for position-controlled operation
- The axis is homed if "Homing required" was selected in the configuration
- No active MC_Stop command
- Valid for appended and superimposing motions (*Mode = 1, 2*) gilt:
  – Neither "MC_CamIn", nor "MC_GearIn" were started.
  – No active basic synchronism

Interaction of commands

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Input parameters

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td>Axis</td><td>INT</td><td>0</td><td colspan="2">Number of the technology DB</td></tr>
<tr><td>Execute</td><td>BOOL</td><td>FALSE</td><td colspan="2">Start of the command at the positive edge</td></tr>
<tr><td>Position</td><td>REAL</td><td>0.0</td><td colspan="2">Target position (negative or positive)</td></tr>
<tr><td rowspan="4">Velocity</td><td rowspan="4">REAL</td><td rowspan="4">-1.0</td><td colspan="2">Maximum velocity (not always reached):</td></tr>
<tr><td>Value > 0:</td><td>Use the defined value</td></tr>
<tr><td>Value = 0:</td><td>Not permitted</td></tr>
<tr><td>Value < 0:</td><td>Use default</td></tr>
<tr><td rowspan="4">Acceleration</td><td rowspan="4">REAL</td><td rowspan="4">-1.0</td><td colspan="2">Acceleration (increasing motor power):</td></tr>
<tr><td>Value > 0:</td><td>Use the defined value</td></tr>
<tr><td>Value = 0:</td><td>Not permitted</td></tr>
<tr><td>Value < 0:</td><td>Use default</td></tr>
<tr><td rowspan="4">Deceleration</td><td rowspan="4">REAL</td><td rowspan="4">-1.0</td><td colspan="2">Deceleration (decreasing energy in the motor):</td></tr>
<tr><td>Value > 0:</td><td>Use the defined value</td></tr>
<tr><td>Value = 0:</td><td>Not permitted</td></tr>
<tr><td>Value < 0:</td><td>Use default</td></tr>
<tr><td rowspan="4">Jerk</td><td rowspan="4">REAL</td><td rowspan="4">-1.0</td><td colspan="2">Jerk:</td></tr>
<tr><td>Value > 0:</td><td>Use the defined value</td></tr>
<tr><td>Value = 0:</td><td>Use trapezoidal motion profile</td></tr>
<tr><td>Value < 0:</td><td>Use default</td></tr>
<tr><td rowspan="6">Direction</td><td rowspan="6">INT</td><td rowspan="6">0</td><td colspan="2">Direction preset for modulo axes:</td></tr>
<tr><td>Value = 0:</td><td>Default in S7T Config *</td></tr>
<tr><td>Value = 1:</td><td>Positive direction of rotation</td></tr>
<tr><td>Value = 2:</td><td>Shortest distance</td></tr>
<tr><td>Value = 3:</td><td>Negative direction of rotation</td></tr>
<tr><td>Value = 4:</td><td>Current direction of rotation
(last used direction of rotation)</td></tr>
<tr><td rowspan="5">Mode</td><td rowspan="5">INT</td><td rowspan="5">0</td><td colspan="2">Override mode:</td></tr>
<tr><td>Value = 0:</td><td>Override motion:
The current motion is canceled</td></tr>
<tr><td>Value = 1:</td><td>Append motion:
The motion command is written to the command buffer. The axis stops at the motion transition</td></tr>
<tr><td>Value = 2:</td><td>Overlay motion:
The motion command is written to the command buffer. The motion transition is overlaid.</td></tr>
<tr><td colspan="2">The "Overlay motion" override mode is not available for modulo axes if the following conditions are met:
Input parameter Direction = 4 or
Input parameter Direction = 0 and
if "Direction = last programmed target direction" was set in the
Axis > Dynamic response
dialog box of S7T Config.</td></tr>
<tr><td>DoneFlag</td><td>INT</td><td>0</td><td colspan="2">DoneFlag (Page 750) generation in the MCDevice DB</td></tr>
<tr><td colspan="5">* It is not advisable to select the direction by setting "Determine based on sign of velocity setpoint" in the defaults of S7T Config, because input parameter Velocity does not support negative velocity setpoints.</td></tr>
</table>

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Target position reached
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 481) of the *Error* output parameter.

MC_MoveAbsolute - Example -"Override motion"

The signal profile below shows the overriding characteristic of (*Mode = 0*) technology function "MC_MoveAbsolute".

Phase "a"

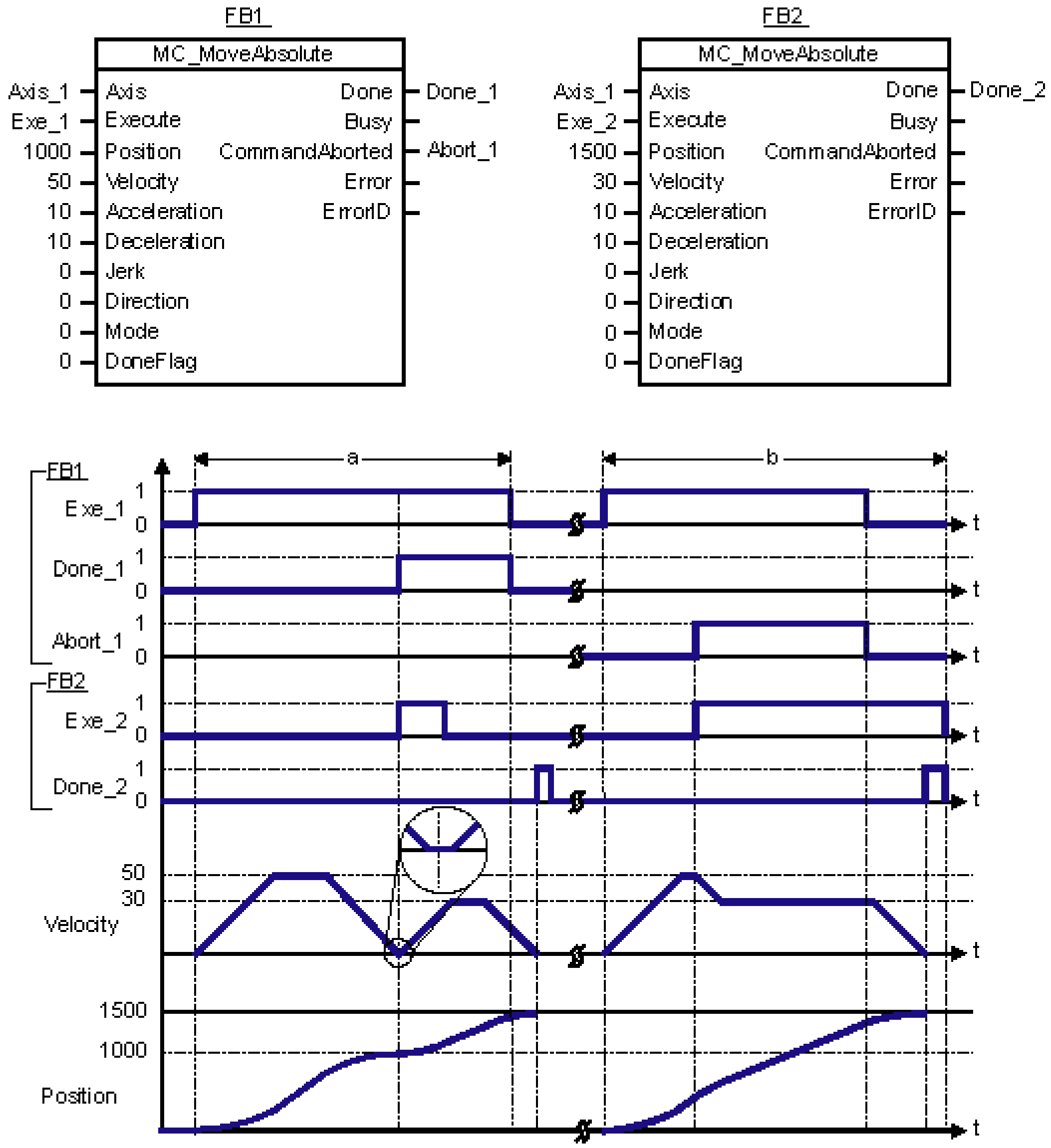
The first rising edge at input parameter *Execute* (*Exe_1)* of FB1 starts absolute positioning of the axis (*Axis_1*). *Done_1* signals that the absolute target position of *1000* has been reached.

Another positioning command is started on FB2 simultaneously with *Done_1 = TRUE* (input parameter *Execute (Exe_2)).*). Reaction times associated with command execution will cause a brief standstill of the axis. Refer to the zoomed view. After this brief standstill, the axis (*Axis_1*) moves to absolute position *1500*. *Done_2* reports that the axis has reached the target position

Phase "b"

A second positive edge at input parameter *Execute* (*Exe_1)* of FB1 starts absolute positioning of the axis (*Axis_1*).

A further positioning command is started at FB2 before the axis has reached target position *1000* (input parameter *Execute* (*Exe_2)). Abort_1* reports cancellation of the active positioning command at FB1. The axis (*Axis_1*) moves at the defined *Deceleration = 10* to velocity *Velocity = 30. Done_2* at FB2 reports the axis at target position *1500*.


FB1
MC_MoveAbsolute
Axis_1
Exe_1
1000
50
10
10
0
0
0
0
Axis
Execute
Position
Velocity
Acceleration
Deceleration
Jerk
Direction
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_1
Abort_1
FB2
MC_MoveAbsolute
Axis_1
Exe_2
1500
30
10
10
0
0
0
0
Done_2
a
b
Exe_1
Done_1
Abort_1
Exe_2
Done_2
Velocity
Position
50
30
1500
1000
t

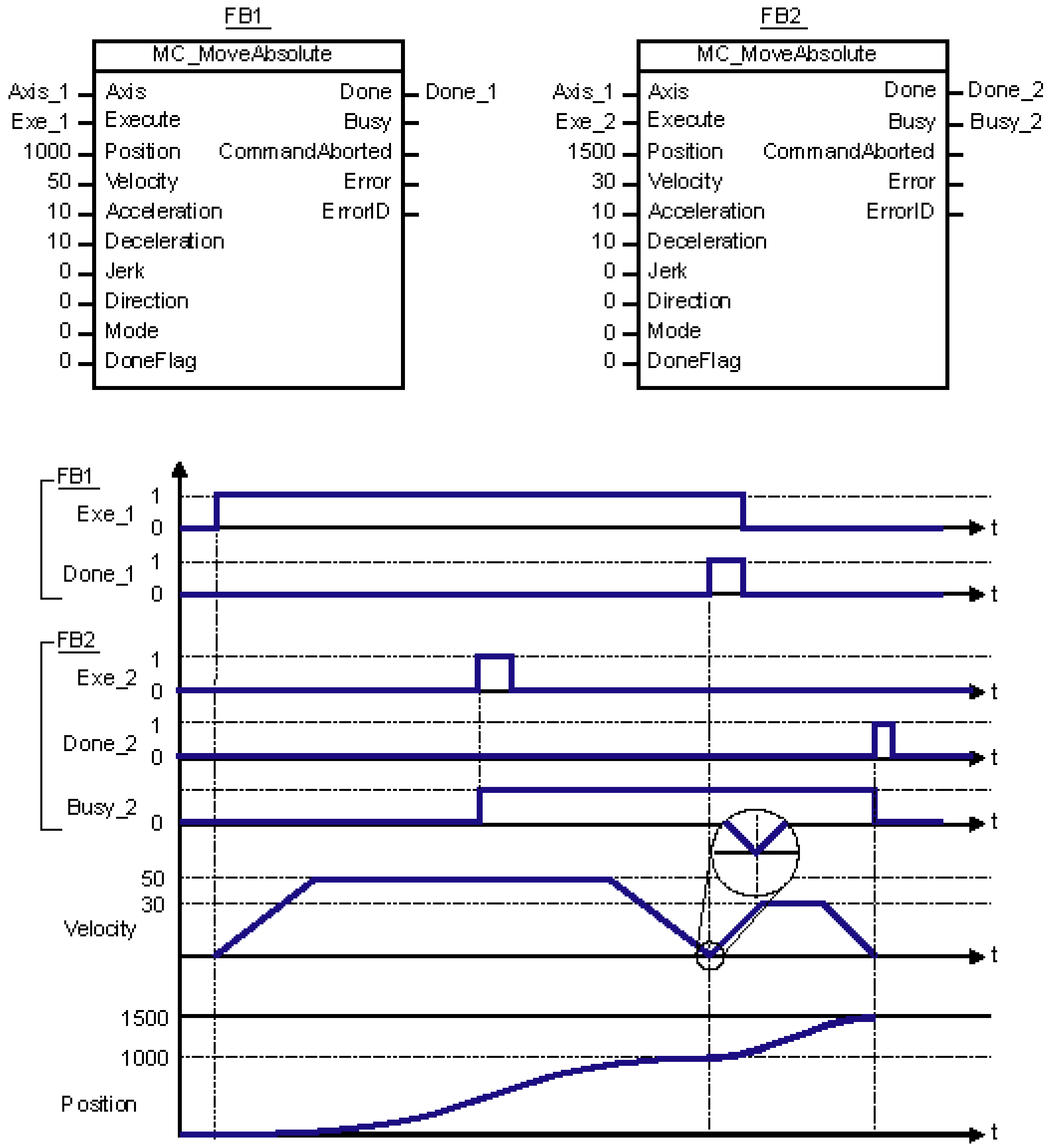
MC_MoveAbsolute - Example - "Append motion"

The signal profile below shows the "appending" characteristic of technology function "MC_MoveAbsolute" (*Mode = 1*).

Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_1*) of FB1. The axis (*Axis_1*) accelerates to its final velocity *50* at the defined *Acceleration = 10*. Before the axis has reached its target position (*Position = 1000*) a further MC_MoveAbsolute command is output (positive edge *Exe_2* at FB2). *Busy_2* reports that the new command is active.

Instead of being canceled, the current positioning command is "appended" by setting *Mode = 1* at FB2. "MC_MoveAbsolute" decelerates the axis with *Deceleration = 10* and then approaches target position *1000* according to the initiated command. Completion of the command is reported with *Done_1*.

After target position *1000* is reached, the system appends the second MC_MoveAbsolute command without time gap for immediate execution (see the zoom view). The axis (*Axis_1*) accelerates to its final velocity along the set acceleration ramp (*= 1500*). *Done_2* reports that the axis has reached the target position; *Busy_2* changes to *FALSE.*


FB1
MC_MoveAbsolute
Axis_1 Axis Done Done_1
Exe_1 Execute Busy
1000 Position CommandAborted
50 Velocity Error
10 Acceleration ErrorID
10 Deceleration
0 Jerk
0 Direction
0 Mode
0 DoneFlag
FB2
MC_MoveAbsolute
Axis_1 Axis Done Done_2
Exe_2 Execute Busy Busy_2
1500 Position CommandAborted
30 Velocity Error
10 Acceleration ErrorID
10 Deceleration
0 Jerk
0 Direction
0 Mode
0 DoneFlag
FB1
Exe_1
Done_1
FB2
Exe_2
Done_2
Busy_2
Velocity
50
30
1500
1000
Position
1
0
t

MC_MoveAbsolute - Example - "Overlay motion 1"

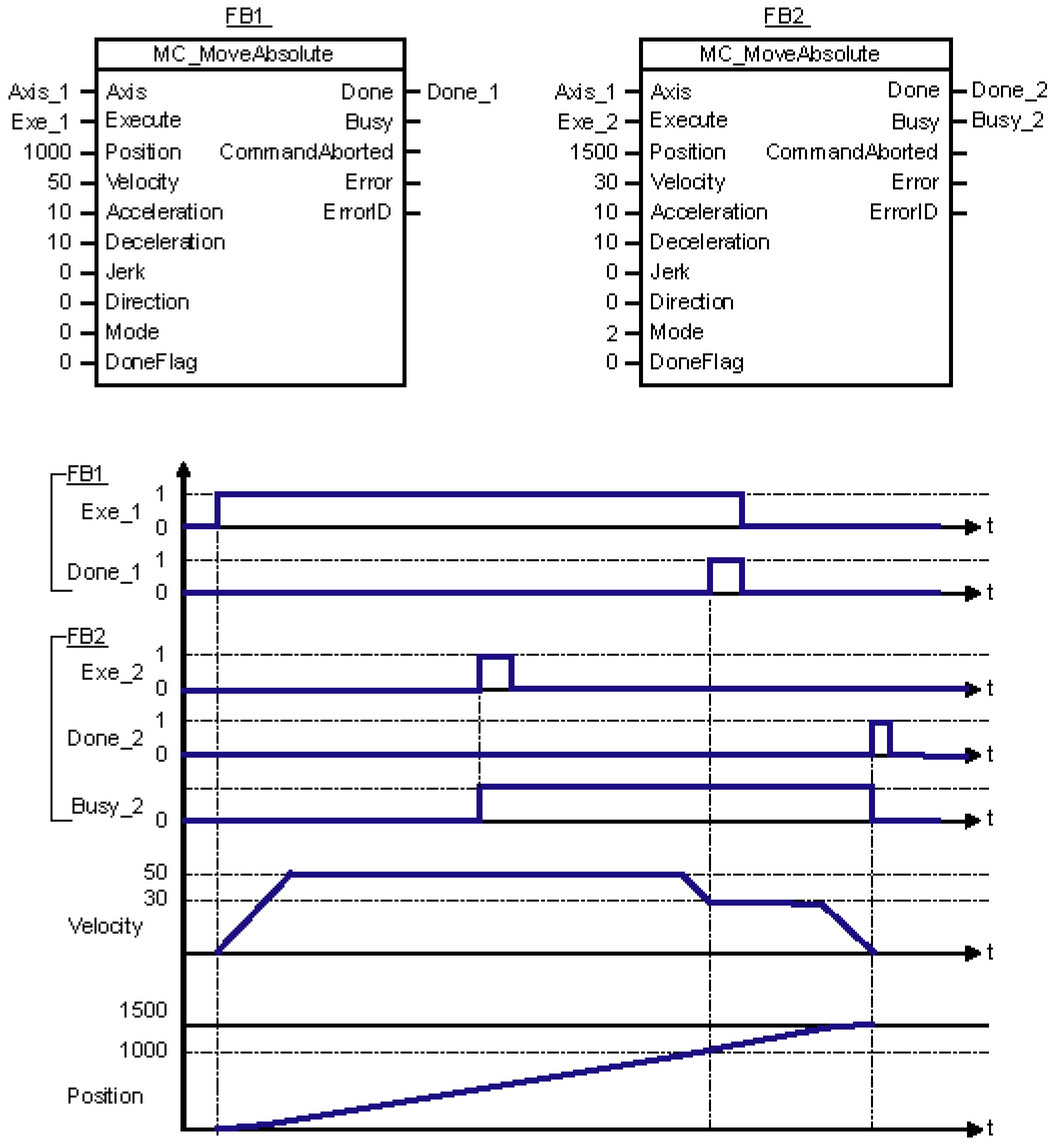
The signal profile shown below shows the "overlay" characteristic of technology function "MC_MoveAbsolute" (*Mode = 2)* in situations where the current velocity exceeds the new velocity.

Current velocity > new velocity

Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_1*) of FB1. The axis (*Axis_1*) accelerates to its final velocity *50* at the defined *Acceleration = 10*. Before the axis has reached its target position (*Position = 1000*) a further MC_MoveAbsolute command is output (positive edge *Exe_2* at FB2). *Busy_2* reports that the new command is active.

Instead of being canceled, the current positioning command is "overlaid" by setting *Mode = 2* at FB2. "MC_MoveAbsolute" decelerates the axis with *Deceleration = 10* so that the axis has reached its final velocity *30* set by the overlaying MC_MoveAbsolute command at target position *1000*. Completion of the command is reported with at FB1 with *Done_1*.

After having reached target position *1000*, the axis (*Axis_1*) continues its approach to target position *1500* at velocity *30* and *Deceleration = 10*. *Done_2* reports at FB2 that the axis has reached the target position; *Busy_2* changes to *FALSE.*


FB1
MC_MoveAbsolute
Axis_1
Exe_1
1000
50
10
10
0
0
0
0
Axis
Execute
Position
Velocity
Acceleration
Deceleration
Jerk
Direction
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_1
FB2
MC_MoveAbsolute
Axis_1
Exe_2
1500
30
10
10
0
0
2
0
Done_2
Busy_2
FB1
Exe_1
Done_1
FB2
Exe_2
Done_2
Busy_2
Velocity
50
30
1500
1000
Position
t

MC_MoveAbsolute - Example - "Overlay motion 2"

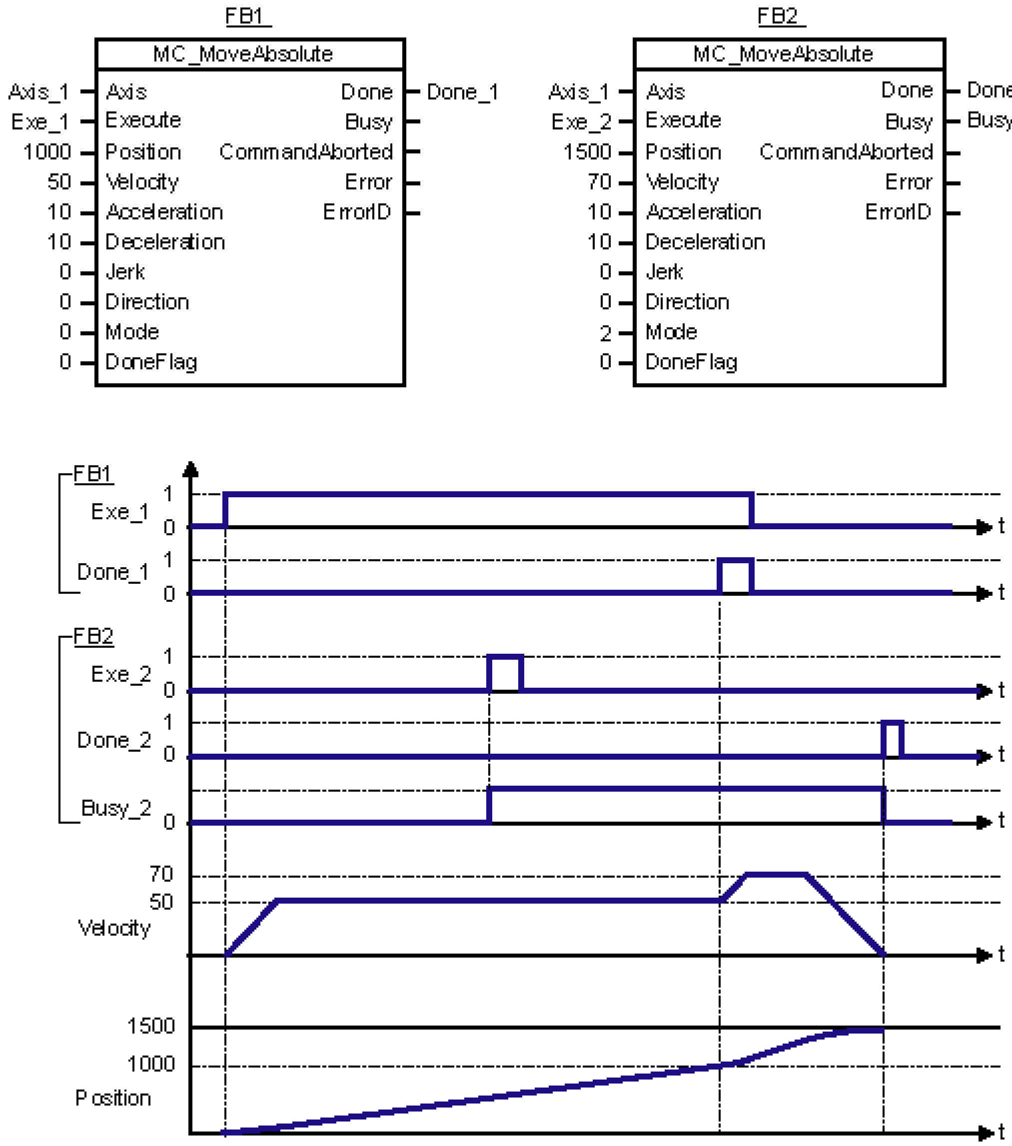
The signal profile shown below shows the "overlay" characteristic of technology function "MC_MoveAbsolute" (*Mode = 2*) in situations where the current velocity is less than the new velocity.

Current velocity < new velocity

Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_1*) of FB1. The axis (*Axis_1*) accelerates to its final velocity *50* at the defined *Acceleration = 10*. Before the axis has reached its target position (*Position = 1000*) a further MC_MoveAbsolute command is output (positive edge *Exe_2* at FB2). *Busy_2* reports that the new command is active.

Instead of being canceled, the current positioning command is "overlaid" by setting *Mode = 2* at FB2. "MC_MoveAbsolute" approaches target position *1000* at the velocity *50* set by the first command. Completion of the command is reported with at FB1 with *Done_1*.

After having reached target position *1000*, the axis accelerates (*Axis_1*) with *Acceleration = 10* to the final speed *70* set by the second command. The axis approaches target position *1500* with *Deceleration = 10*. *Done_2* reports at FB2 that the axis has reached the target position; *Busy_2* changes to *FALSE.*


FB1
MC_MoveAbsolute
Axis_1
Exe_1
1000
50
10
10
0
0
0
0
Axis
Execute
Position
Velocity
Acceleration
Deceleration
Jerk
Direction
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_1
FB2
MC_MoveAbsolute
Axis_1
Exe_2
1500
70
10
10
0
0
2
0
Done_2
Busy_2
Exe_1
Done_1
Exe_2
Done_2
Busy_2
70
50
Velocity
1500
1000
Position
t

MC_MoveAbsolute - Example - "Overlay motion 3"

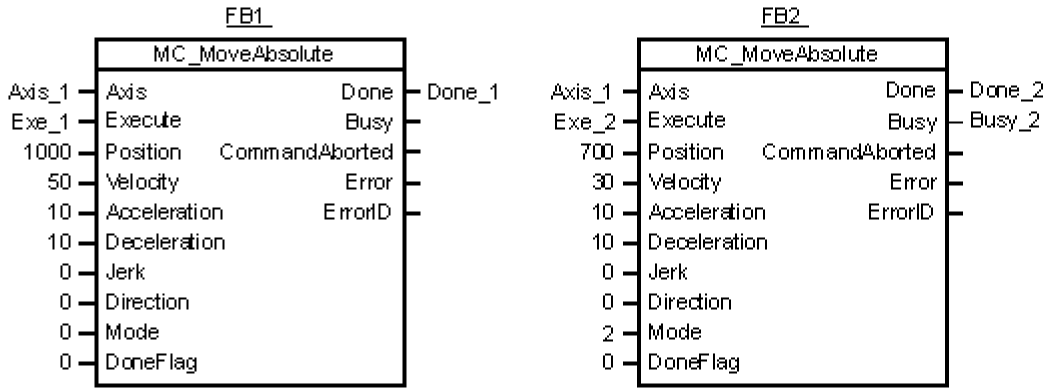
The signal profile below shows the "overlaying" characteristic of technology function "MC_MoveAbsolute" (*Mode = 2*) with direction reversal.

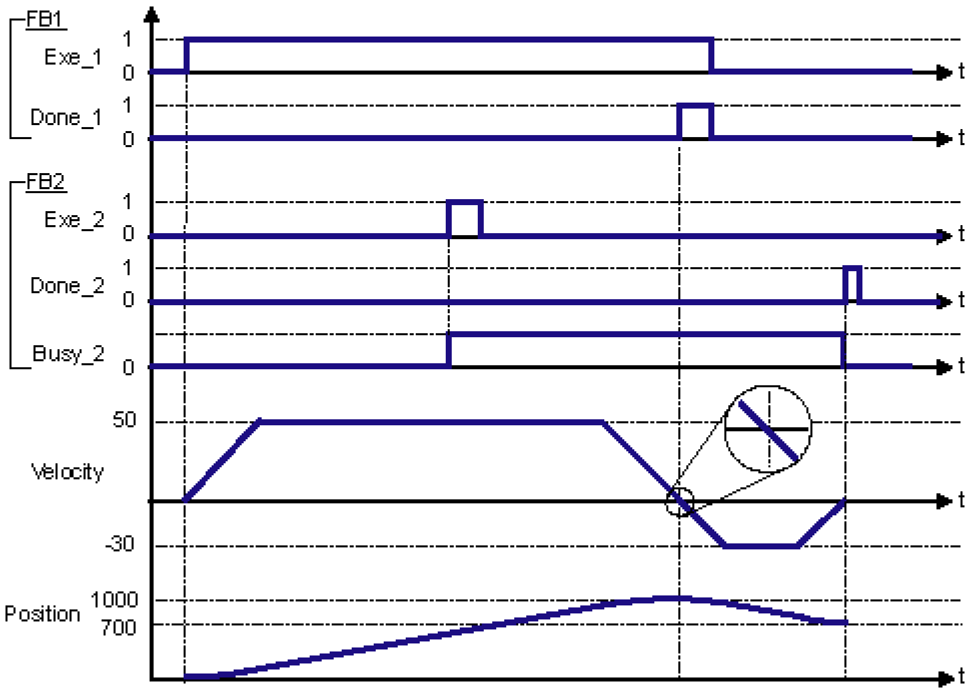
Direction reversed

Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_1*) of FB1. The axis (*Axis_1*) accelerates to its final velocity *50* at the defined *Acceleration = 10*. Before the axis has reached its target position (*Position = 1000*) a further MC_MoveAbsolute command is output (positive edge *Exe_2* at FB2). *Busy_2* reports that the new command is active.

Instead of being canceled, the current positioning command is "overlaid" by setting *Mode = 2* at FB2. The value of target position *1000* of the currently active MC_MoveAbsolute command is higher than target position *700* of the next MC_MoveAbsolute command. You therefore have to reverse the direction. "MC_MoveAbsolute" decelerates the axis with *Deceleration = 10* so that the axis has reached velocity *0* at target position *1000*. Completion of the command is reported with at FB1 with *Done_1*.

After having reached target position *1000*, the axis decelerates (*Axis_1*) with *Acceleration = 10* to velocity *-30*. The zoom view shows that there are no gaps at the transition. The axis approaches target position *700* with *Deceleration = 10*. *Done_2* reports at FB2 that the axis has reached the target position; *Busy_2* changes to *FALSE.*


FB1
MC_MoveAbsolute
Axis_1 Axis
Exe_1 Execute
1000 Position
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 Direction
0 Mode
0 DoneFlag
Done Done_1
Busy
CommandAborted
Error
ErrorID
FB2
MC_MoveAbsolute
Axis_1 Axis
Exe_2 Execute
700 Position
30 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 Direction
2 Mode
0 DoneFlag
Done Done_2
Busy Busy_2
CommandAborted
Error
ErrorID


FB1
Exe_1
Done_1
FB2
Exe_2
Done_2
Busy_2
Velocity
Position
1
0
50
-30
1000
700
t

MC_MoveAbsolute - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Concerns all input parameters of data type REAL, or the input parameters *Mode, Direction* or *DoneFlag*. Example: *Mode =2* and *Direction = 4.*
8044	Command not supported by the technology object	Sending a command to an output cam, for example.
8045	Command not allowed in current state	"MC_Stop" is active
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8055	Latching motion not allowed in current state	A latching command cannot be accepted at the current axis state. Reasons: • A basic synchronization or superimposing synchronization command is being executed at the axis • "MC_Halt" is active
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.7 FB411 MC_MoveRelative - Relative positioning

Relative positioning with FB 411 "MC_MoveRelative"

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

Purpose

- The "MC_MoveRelative" technology function starts the positioning motion of an axis at a position relative to the start position.
  If the axis was already in motion at the start of command execution and a "motion override" was set with *Mode = 0*, the system uses the start position which is available internally at the start of command execution. Allowances must be made for the reaction time between the execution of a technology function and of a command, based on the load and cycle time.
- You define the dynamic response of the axis motion with the input parameters *Velocity*, *Jerk*, *Acceleration*, and *Deceleration*.
- Define whether to trigger an active motion, or whether to append or overlay the motion at input parameter *Mode*.

Supported for

- Positioning axes
- Synchronization axes

Prerequisites

- Axis is enabled for position-controlled operation
- No active MC_Stop command
- Rule for saving and overlaying motions (*Mode = 1, 2*):
  - Neither "MC_CamIn", nor "MC_GearIn" were started.
  - No active basic synchronism

Interaction of commands

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Distance	REAL	*0.0*	Positioning distance (negative or positive)	
Velocity	REAL	*-1.0*	Maximum velocity (not always reached):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Acceleration (increasing motor power):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Deceleration (decreasing energy in the motor):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk:	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
Mode	INT	*0*	Override mode:	
			Value = *0*:	Override motion: The current motion is canceled
			Value = *1*:	Append motion: The motion command is written to the command buffer. The axis stops at the motion transition.
			Value = *2*:	Overlay motion: The motion command is written to the command buffer. The motion transition is overlaid.
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Target position reached	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 495) of the *Error* output parameter.	

MC_MoveRelative - Example - "Override motion"

The signal profile below shows the overriding characteristic of (*Mode = 0*) technology function "MC_MoveRelative".

Phase "a"

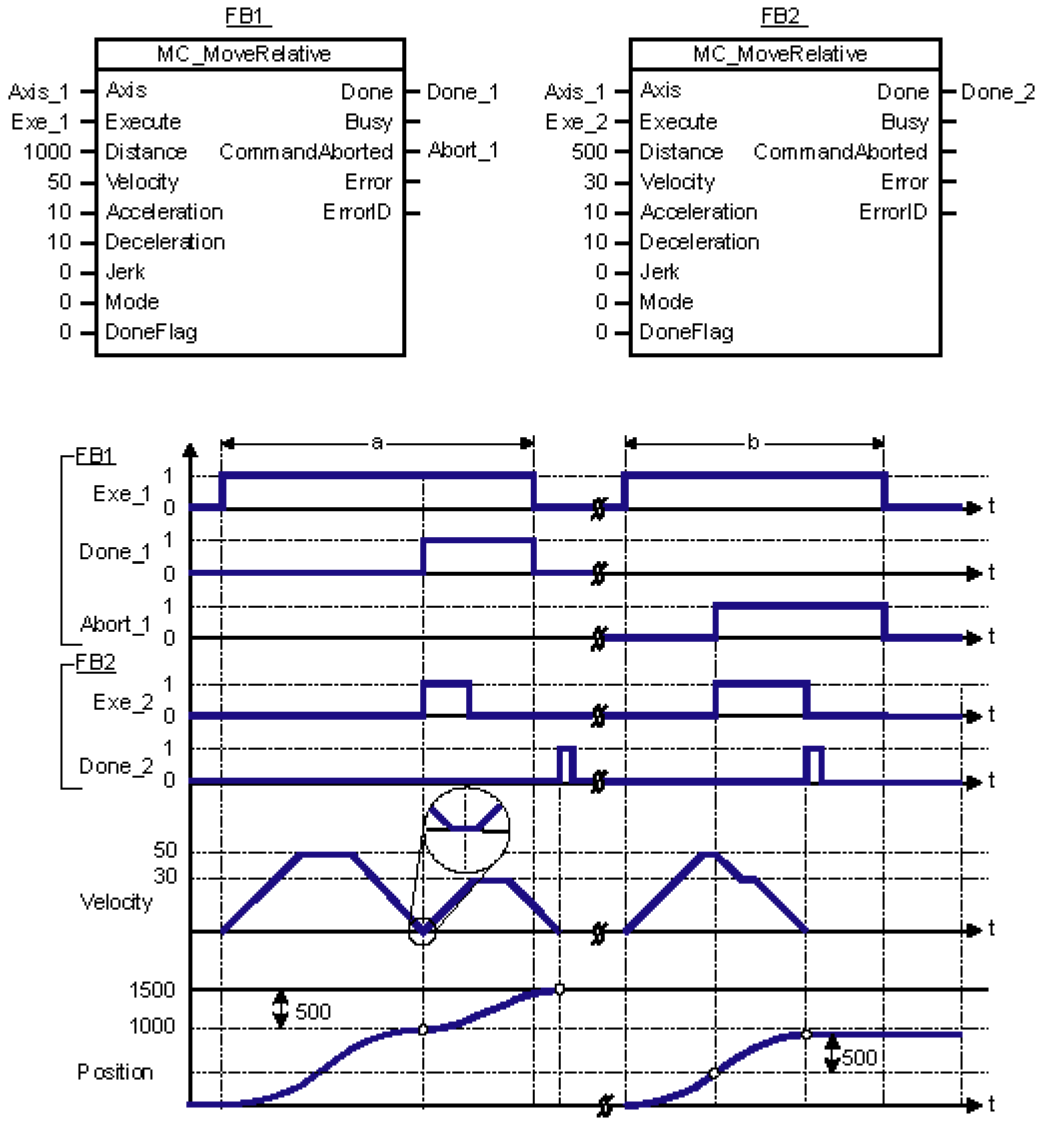
The first positive edge at input parameter *Execute* (*Exe_1)* of FB1 starts relative positioning of the axis (*Axis_1*). *Done_1* changes to *TRUE* after the axis has reached *Distance = 1000*.

A second relative positioning command is started at FB2 in time with *Done_1 = TRUE* (input parameter *Execute (Exe_2)).* Reaction times during command execution will cause a brief standstill of the axis. Refer to the zoomed view. After a brief stop, the axis (*Axis_1*) continues to move by a distance of *500*. *Done_2* reports completion of the command.

Phase "b"

A further positive edge at input parameter *Execute (Exe_1)* of FB1 starts relative positioning of the axis (*Axis_1*).

A further positioning command is started at FB2 before the axis has reached target position *1000* (input parameter *Execute (Exe_2)). Abort_1* reports cancellation of the active positioning command at FB1. The axis (*Axis_1*) moves at the defined *Deceleration = 10* to velocity *Velocity = 30. Done_2* at FB2 reports that the axis has travelled the *Distance = 500.*


FB1
MC_MoveRelative
Axis_1 Axis Done Done_1
Exe_1 Execute Busy
1000 Distance CommandAborted Abort_1
50 Velocity Error
10 Acceleration ErrorID
10 Deceleration
0 Jerk
0 Mode
0 DoneFlag
FB2
MC_MoveRelative
Axis_1 Axis Done Done_2
Exe_2 Execute Busy
500 Distance CommandAborted
30 Velocity Error
10 Acceleration ErrorID
10 Deceleration
0 Jerk
0 Mode
0 DoneFlag
FB1
Exe_1
Done_1
Abort_1
FB2
Exe_2
Done_2
Velocity
Position
a
b
1
0
50
30
1500
1000
500
500
t

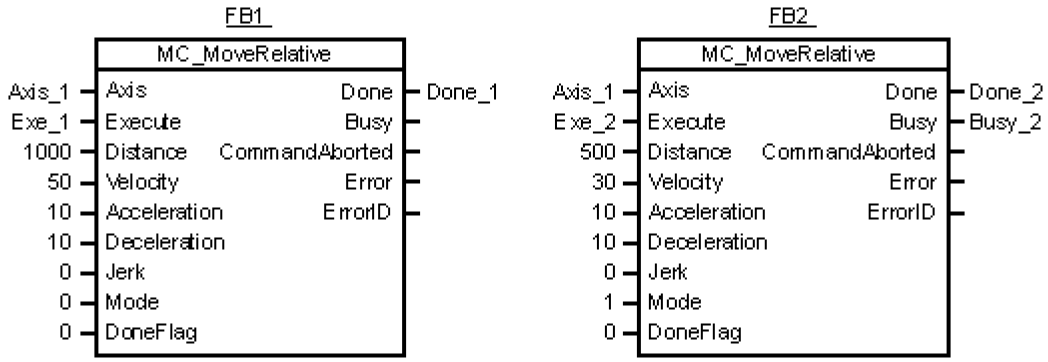
MC_MoveRelative - Example - "Append motion"

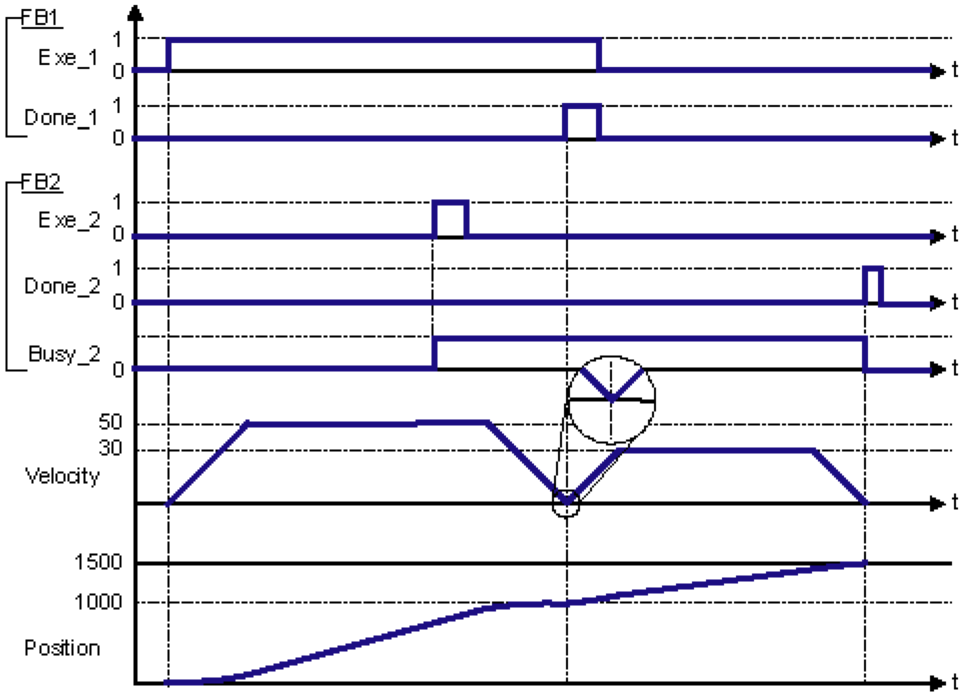
The signal profile below shows the "appending" characteristic of technology function "MC_MoveRelative" (*Mode = 1*).

Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_1*) of FB1. The axis (*Axis_1*) accelerates to its final velocity *50* at the defined *Acceleration = 10*. A further MC_MoveRelative command is started (positive edge *Exe_2* at FB2) before the axis has travelled the defined distance (*Distance = 1000*). *Busy_2* reports that the new command is active.

Instead of being canceled, the current positioning command is "appended" by setting *Mode = 1* at FB2. "MC_MoveRelative" decelerates the axis with *Deceleration = 10* and travels by the distance *1000* according to the initiated command. Completion of the command is reported with *Done_1*.

After the axis has traversed the distance *1000*, the motion of the second MC_MoveRelative command is appended immediately without time gap. Refer to the zoomed view. The axis (*Axis_1*) moves by the distance *= 500* along the set acceleration and deceleration ramps. *Done_2* changes to *TRUE* and *Busy_2* changes to *FALSE* after the axis has covered the *Distance = 500*.


FB1
MC_MoveRelative
Axis_1
Exe_1
1000
50
10
10
0
0
0
Axis
Execute
Distance
Velocity
Acceleration
Deceleration
Jerk
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_1
FB2
MC_MoveRelative
Axis_1
Exe_2
500
30
10
10
0
1
0
Axis
Execute
Distance
Velocity
Acceleration
Deceleration
Jerk
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_2
Busy_2


FB1
Exe_1
Done_1
FB2
Exe_2
Done_2
Busy_2
1
0
50
30
Velocity
1500
1000
Position
t

MC_MoveRelative - Example - "Overlay motion 1"

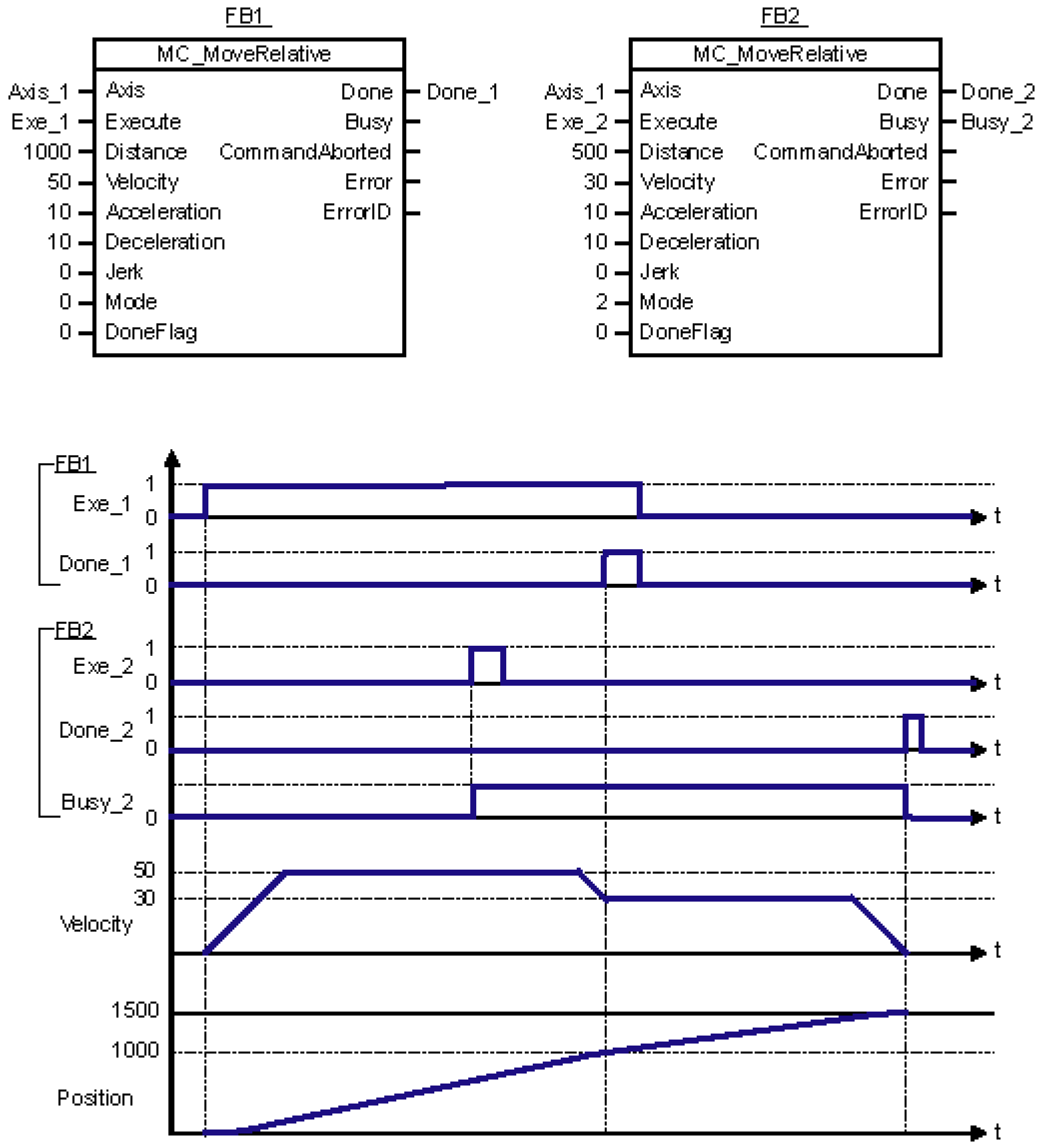
The signal profile shown below shows the "overlay" characteristic of technology function "MC_MoveRelative" (*Mode = 2)* in situations where the current velocity exceeds the new velocity.

Current velocity > new velocity

Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_1*) of FB1. The axis (*Axis_1*) accelerates to its final velocity *50* at the defined *Acceleration = 10*. A further MC_MoveRelative command is started (positive edge *Exe_2* at FB2) before the axis has travelled the defined distance (*Distance = 1000*). *Busy_2* reports that the new command is active.

Instead of being canceled, the current positioning command is "overlaid" by setting *Mode = 2* at FB2. "MC_MoveRelative" decelerates the axis with *Deceleration = 10* so that the axis has reached its final velocity *30* set by the overlaying MC_MoveRelative command after it has travelled the distance *1000*. Completion of the command is reported with at FB1 with *Done_1*.

After having covered the distance *1000*, the axis (*Axis_1*) continues to move at velocity *30*. The axis (*Axis_1*) moves by the distance *= 500* along the set speed and deceleration ramps. *Done_2* changes to *TRUE* and *Busy_2* changes to *FALSE* after the axis has covered the *Distance = 500*.


FB1
MC_MoveRelative
Axis_1 Axis Done Done_1
Exe_1 Execute Busy
1000 Distance CommandAborted
50 Velocity Error
10 Acceleration ErrorID
10 Deceleration
0 Jerk
0 Mode
0 DoneFlag
FB2
MC_MoveRelative
Axis_1 Axis Done Done_2
Exe_2 Execute Busy Busy_2
500 Distance CommandAborted
30 Velocity Error
10 Acceleration ErrorID
10 Deceleration
0 Jerk
2 Mode
0 DoneFlag
FB1
Exe_1
Done_1
FB2
Exe_2
Done_2
Busy_2
50
30
Velocity
1500
1000
Position
t

MC_MoveRelative - Example - "Overlay motion 2"

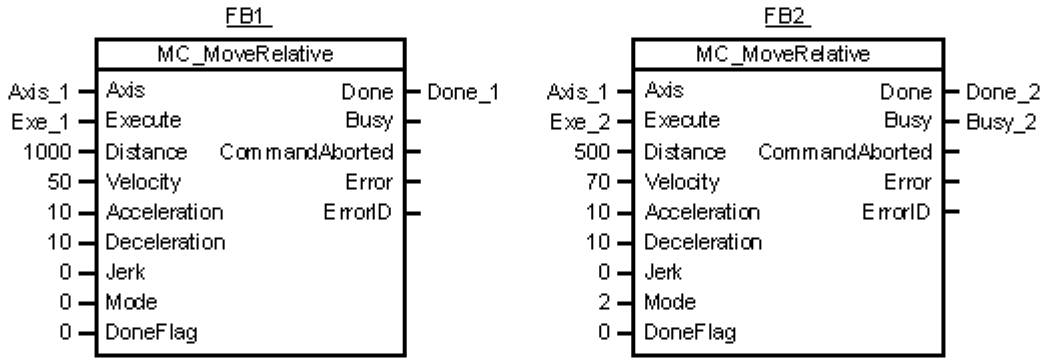
The signal chart shown below shows the "blending" characteristic of technology function "MC_MoveRelative" (*Mode = 2*) in situations where the current velocity is less than the new velocity.

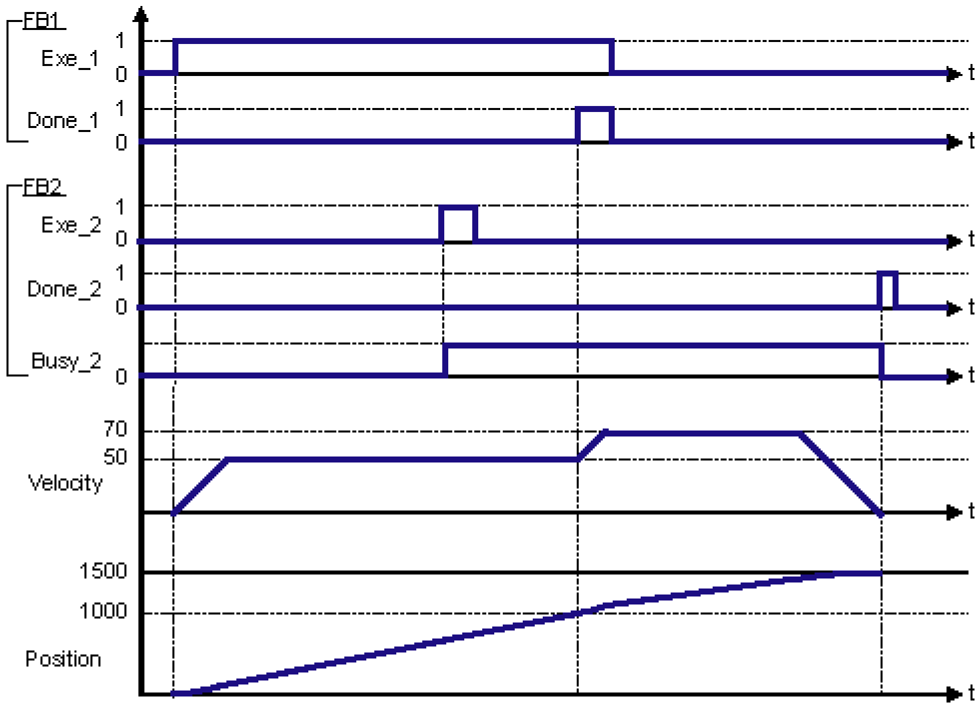
Current velocity < new velocity

Relative positioning is started on a rising edge at FB1 input parameter *Execute* (*Exe_1*). The axis (*Axis_1*) accelerates at the specified rate (*Acceleration = 10*) to its final velocity *50*. An additional MC_MoveRelative command is started (rising edge *Exe_2* at FB2) before the axis has traveled the defined distance (*Distance = 1000*). *Busy_2* signals that the new command is active.

Instead of being canceled, the current positioning command is "blended" by setting *Mode = 2* at FB2. "MC_MoveRelative" moves the axis over the remaining distance at final velocity *50*. Completion of the command at FB1 is reported with *Done_1*.

After having traversed the distance *1000*, the axis accelerates (*Axis_1*) with *Acceleration = 10* to final velocity *70* set by the second command. The axis (*Axis_1*) traverses with the set velocity and deceleration for the remaining distance. *Done_2* switches to *TRUE* and *Busy_2* switches to *FALSE,* when the distance *Distance = 500* has been covered.


FB1
MC_MoveRelative
Axis_1
Exe_1
1000
50
10
10
0
0
0
Axis
Execute
Distance
Velocity
Acceleration
Deceleration
Jerk
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_1
FB2
MC_MoveRelative
Axis_1
Exe_2
500
70
10
10
0
2
0
Axis
Execute
Distance
Velocity
Acceleration
Deceleration
Jerk
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_2
Busy_2


FB1
Exe_1
Done_1
FB2
Exe_2
Done_2
Busy_2
70
50
Velocity
1500
1000
Position
t

MC_MoveRelative - Example - "Overlay motion 3"

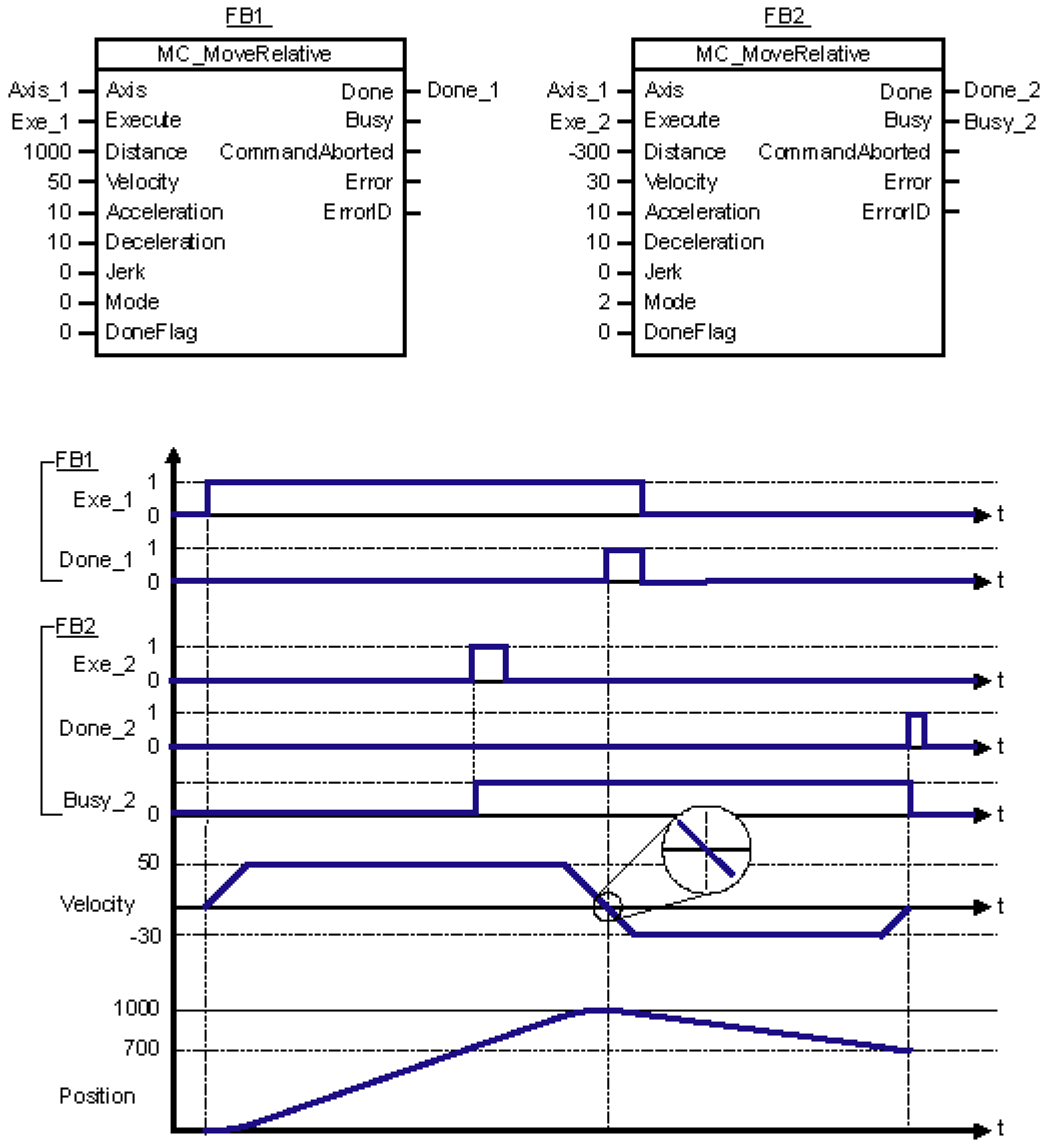
The signal profile below shows the "overlaying" characteristic of technology function "MC_MoveRelative" (*Mode = 2*) with direction reversal.

Direction reversed

Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_ 1*) of FB1. The axis (*Axis_ 1*) accelerates to its final velocity *50* at the defined *Acceleration = 10*. A further MC_MoveRelative command is started (positive edge *Exe_2* at FB2) before the axis has travelled the defined distance (*Distance = 1000*). *Busy_2* reports that the new command is active.

Instead of being canceled, the current positioning command is "overlaid" by setting *Mode = 2* at FB2. The direction is reversed, because a negative distance was set at FB2 (*Distance = -300*). "MC_MoveRelative" decelerates the axis with *Deceleration = 10* so that the axis has reached *0* velocity after having traversed the distance *1000*. Completion of the command is reported with at FB1 with *Done_1*.

After having traversed the distance *1000*, the axis accelerates (*Axis_ 1*) with *Acceleration = 10* to velocity *-30.* The zoom view shows that there are no gaps at the transition. The axis (*Axis_ 1*) covers the remaining distance at the set velocity and deceleration. *Done_2* changes to *TRUE* and *Busy_2* changes to *FALSE* after the axis has covered the *Distance = -300*.


FB1
MC_MoveRelative
Axis_1
Exe_1
1000
50
10
10
0
0
0
Axis
Execute
Distance
Velocity
Acceleration
Deceleration
Jerk
Mode
DoneFlag
Done
Busy
CommandAborted
Error
ErrorID
Done_1
FB2
MC_MoveRelative
Axis_1
Exe_2
-300
30
10
10
0
2
0
Done_2
Busy_2
FB1
Exe_1
Done_1
FB2
Exe_2
Done_2
Busy_2
50
Velocity
-30
1000
700
Position
t

MC_MoveRelative - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *DoneFlag* or *Mode* input parameters.
8044	Command not supported by the technology object	Sending a command to an output cam, for example
8045	Command not allowed in current state	The MC_Stop command is active, for example
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8055	Latching motion not allowed in current state	A latching command cannot be accepted at the current axis state. Reasons: • A basic synchronization or superimposing synchronization command is being executed at the axis • "MC_Halt" is active
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.8 FB412 MC_MoveAdditive - Relative positioning to current target position

Positioning relative to current target position with FB 412 "MC_MoveAdditive"

Purpose

- The "MC_MoveAdditive" technology function starts axis positioning with user-definable dynamic values to a position which is relative to the target position of the current positioning command. This method allows the correction of a previously commanded target position by a defined distance.
- Define the dynamic response of the motion at the input parameters *Velocity*, *Jerk*,*Acceleration*, *Deceleration*.
- The axis stops at the target position
- An MC_MoveAdditive command overrides the active command

Supported for

- Positioning axes
- Synchronization axes

Prerequisites

- The axis must be enabled for position-controlled operation.
- The axis must be homed if "Homing required" was set in the configuration
  - and if the axis is in motion
  - or if "MC_MoveAdditive" overrides an active motion command (exception "MC_MoveVelocity").
- Axis does not have to be homed
  - if the axis is at a standstill
  - if an active MC_MoveVelocity command is overridden
- An MC_Stop command is not being executed.

Note

When one of the following requirements applies, "MC_MoveAdditive" behaves like "MC_MoveRelative":

- The axis is at a standstill at the start of the command
- A **continuous function** is overridden, that is, if the target position is not defined. In this case, the target position depends on the override position and is derived from the *Distance* and the position at the start of the operation.

Restraints for modulo axes

The application of this function to modulo axes is restricted and only possible for small distances (in terms of modulo length) which may not exceed the current modulo length:

New target position = (old target position + distance) modulo length

- *Distance* must be less than one modulo length.
- The distance to go must be less than one modulo length.
- *Distance* plus the distance to go of the overriding command must be less than one modulo length.

Interaction of commands

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Distance	REAL	*0.0*	Positioning distance (negative or positive)	
Velocity	REAL	*-1.0*	Maximum velocity (not always reached):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Acceleration (increasing motor power):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Deceleration (decreasing energy in the motor):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk:	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

Output parameters (status outputs)

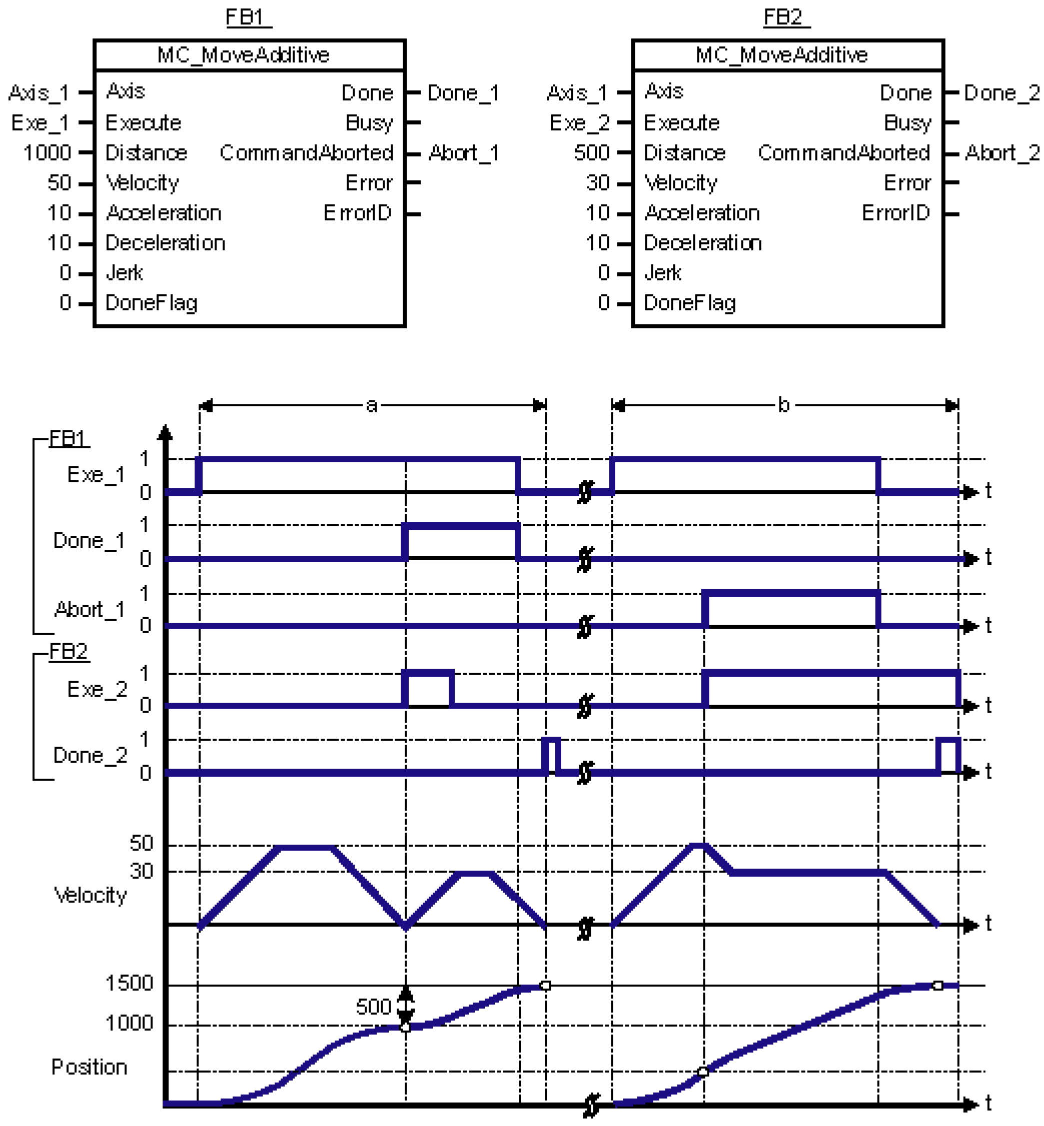
Parameter	Data type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Target position reached
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 509) of the *Error* output parameter.

MC_MoveAdditive - example

Signal profile

Case a: Two MC_MoveAdditive commands are started in succession.

Case b: The second MC_MoveAdditive (FB 2) is started before the first is completed. This action cancels the first command (FB 1). The target position is derived from the target position of the first command, corrected by the distance of the second command.



MC_MoveAdditive - ErrorIDs

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *DoneFlag* input parameter.
8044	Command not supported by the technology object	Sending a command to an output cam, for example.
8045	Command not allowed in current state	MC_Stop- command active.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.9 FB413 MC_MoveSuperImposed - Superimposed positioning

Superimposed positioning with FB413 "MC_MoveSuperImposed"

Purpose

- The "MC_MoveSuperImposed" technology function allows superimposed positioning of an axis, for example, for pressure mark adjustment.
- Define the dynamic response of the motion at the input parameters *VelocityDiff*, *Jerk*,*Acceleration*, *Deceleration*.
- The dynamic values of technology function "MC_MoveSuperImposed" add up to form the dynamic values of the basic motion If the dynamic values derived from this addition exceed the dynamic value configured in S7T Config (Dialog **Default > Dynamic response** dialog box) the actual dynamic values are limited to the value configured in S7T Config. The axis technology DB returns warning *0021* in this case.
- Active positioning or synchronization operations are not canceled.
- An active MC_MoveSuperImposed command is overridden by a new MC_MoveSuperImposed command. The distance-to-go value of the canceled MC_MoveSuperImposed is lost!

Supported for

- Positioning axes
- Synchronization axes

Prerequisites

- Axis is enabled for position-controlled operation
- No active MC_Stop command
- The axis velocity usually needs to be increased for superimposed positioning. For this reason, the basic velocity of the axis may not have reached its maximum when this positioning function is set.

Interaction of commands

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Input parameters

Parameters	Data type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Distance	REAL	*0.0*	Additional distance for superimposed positioning	
VelocityDiff	REAL	*-1.0*	Maximum velocity deviation compared to current motion:	
			Value > *0*:	Use the defined value
			Value = *0*:	Not allowed
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Acceleration (increasing motor power):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Deceleration (decreasing energy in the motor):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk:	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

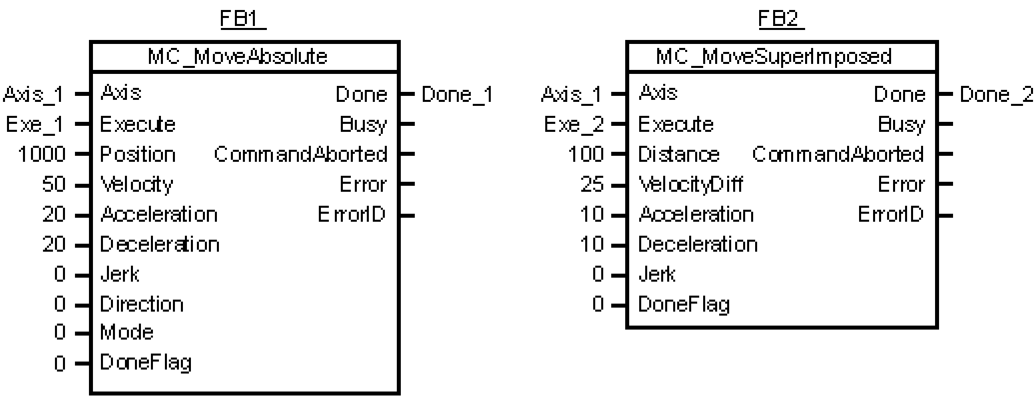
Output parameters (status outputs)

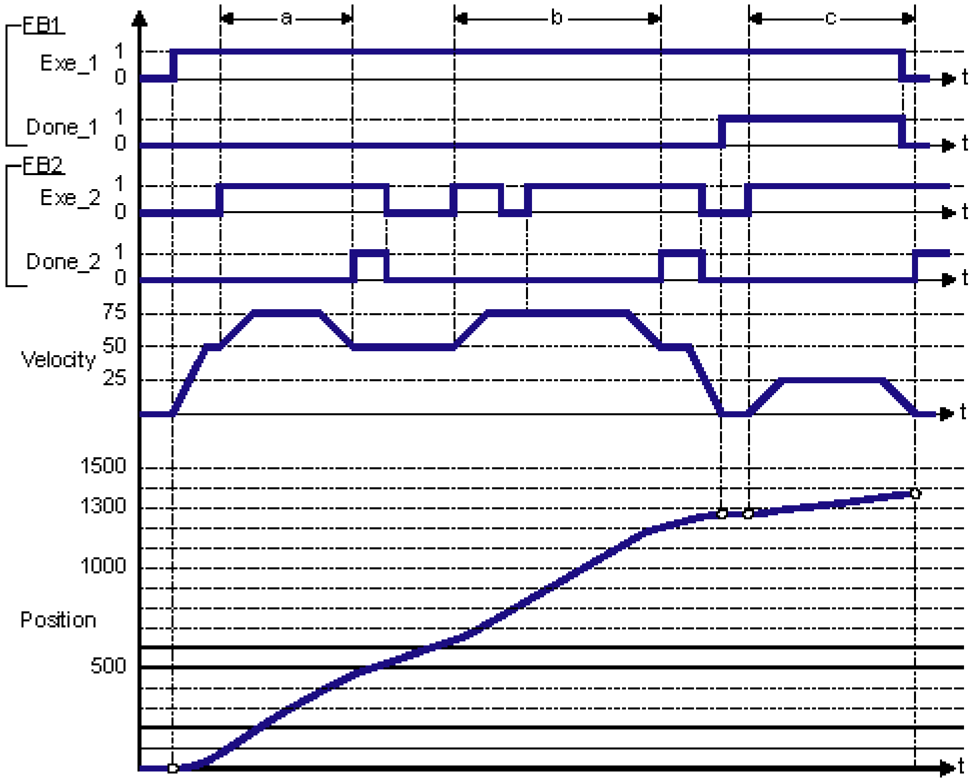
Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Superimposed positioning completed	
Busy	BOOL	*FALSE*	*TRUE*: Superimposed positioning is active	
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 515) of the *Error* output parameter.	

MC_MoveSuperImposed - Example - "Absolute positioning"

Signal profile: Effect of superimposed positioning

a) "MC_MoveSuperImposed" is started in the course of absolute positioning.
b) "MC_MoveSuperImposed" is restarted before "MC_MoveSuperImposed" is done.
c) Start "MC_MoveSuperImposed" when the axis is at a standstill.

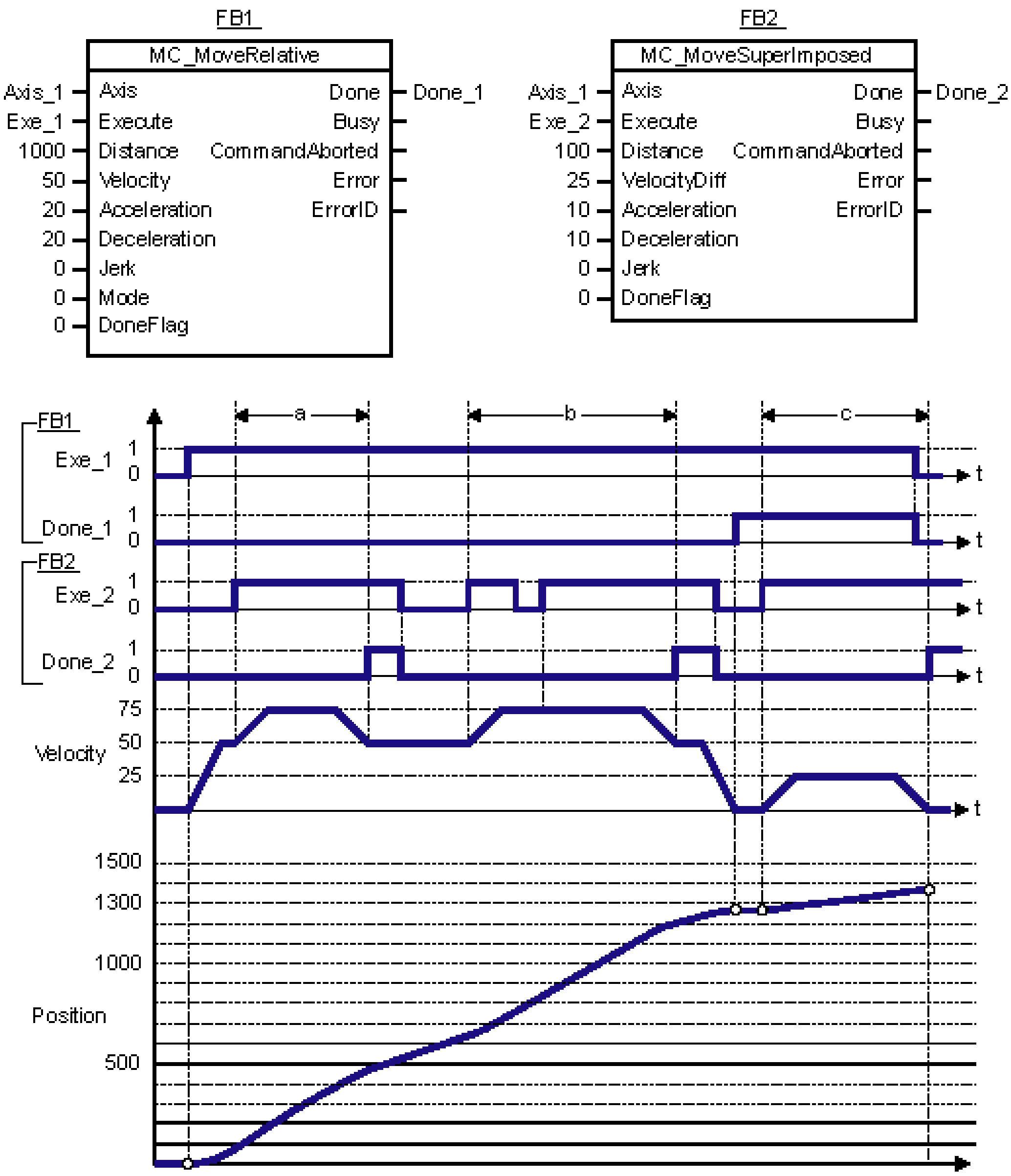




MC_MoveSuperImposed - Example - "Relative positioning"

Signal profile: Effect of superimposed positioning

a) "MC_MoveSuperImposed" is started in the course of relative positioning.
(b) "MC_MoveSuperImposed" is restarted before "MC_MoveSuperImposed" is done.
(c) Start "MC_MoveSuperImposed" when the axis is at a standstill.



MC_MoveSuperImposed - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *DoneFlag* input parameter.
8044	Command not supported by the technology object	Command request to a speed-controlled axis, for example.
8045	Command not allowed in current state	MC_Stop command active.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.10 FB 414 MC_MoveVelocity - Motion with speed preset

Moving the axis with speed preset using FB 414 "MC_MoveVelocity"

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

Purpose

- The "MC_MoveVelocity" technology function initiates acceleration of the axis to the preset velocity.
- Define motion dynamic response of the motion at the input parameters *Jerk*, *Acceleration*, *Deceleration*.
- Allowances are made for any active velocity override function in the calculation of the final velocity (*InVelocity* output parameter). Make allowances for this reaction in the user program.
- You can use input parameter *PositionControl* to set position- or speed-controlled mode at position-controlled axes.
- Use input parameter *Mode* to specify whether to trigger an active motion, to override the current motion or to append the motion.

Supported for

- Speed-controlled axes
- Positioning axes
- Synchronization axes

Prerequisites

- The axis is enabled for speed- or position-controlled operation
- No active MC_Stop command
- Rule for appending motions (*Mode = 1*):
  - Neither "MC_CamIn", nor "MC_GearIn" were started.
  - No active basic synchronism
  - "MC_MoveVelocity" cannot be saved and appended to an active "MC_MoveVelocity" command

Interaction of commands

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Velocity	REAL	*-1.0*	Final velocity:	
			Value < *0*:	Use default
			Value = *0*:	supported
			Value > *0*:	Final velocity
Acceleration	REAL	*-1.0*	Acceleration (increasing motor power):	
			Value > *0*:	Use the defined value
			Value = *0*:	Only permitted if the axis does not have to be accelerated in order to reach final velocity.
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Deceleration (decreasing energy in the motor):	
			Value > *0*:	Use the defined value
			Value = *0*:	Only permitted if the axis does not have to be decelerated in order to reach final velocity.
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk:	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
Direction	INT	*0*	Direction preset:	
			Value = *0*:	Default in S7T Config *
			Value = *1*:	Positive direction of rotation
			Value = *2*:	Not allowed
			Value = *3*:	Negative direction of rotation
			Value = *4*:	Current direction of rotation (last used direction of rotation)
Current	BOOL	*FALSE*	Maintain current velocity:	
			FALSE:	"Maintain current velocity" is disabled
			TRUE:	The current velocity and direction are maintained. Used to terminate synchronous operation, for example. Input parameter *Velocity* is ignored. When the drive resumes operation at the current velocity, the *InVelocity* output returns the value *TRUE.*

Parameters	Data Type	Initial value	Description
PositionControl	BOOL	*TRUE*	*TRUE*: Position-controlled mode *FALSE*: Speed-controlled mode Use input parameter *PositionControl* to toggle position-controlled / speed-controlled mode at the axis. Input parameter *PositionControl* is ignored at speed-controlled axes. To allow its use as leading axis, the axis must be set to operate in position-controlled mode. You can read the currently active mode from *Statusword.SpeedMode* of the technology DB. The changeover to speed-controlled mode is retentive. Speed-controlled mode can be terminated by a new MC_MoveVelocity command with *PositionControl = TRUE* and by any other motion command (with the exception of MC_Halt" and "MC_Stop".) You can stop speed-controlled mode without active motion with the MC_MoveRelative command by setting input parameter *Distance = 0.0*.
Mode	INT	*0*	Override mode: Value = *0*: Override motion: The current motion is canceled Value = *1*: Append motion: The motion command is written to the command buffer. The axis stops at the motion transition
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB The DoneFlag value is inverted when the final velocity is reached.
* It is not advisable to select the direction by setting "Determine based on sign of velocity setpoint" in the defaults of S7T Config, because input parameter *Velocity* does not support negative velocity setpoints. The axis always moves in positive direction if the configuration in S7T Config is faulty.			

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
InVelocity	BOOL	*FALSE*	*TRUE*: The velocity defined at the *Velocity* input parameter was reached or is retained.
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted. Observe the following information.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 521) of the *Error* output parameter.

Note

At speed setpoint **zero** (*Velocity = 0.0*):

InVelocity is set when the axis has reached a standstill, and remains set as long as *Execute = 1*. Command execution is completed when *InVelocity* is set, that is, *Busy = FALSE*, and the technological function can neither report *CommandAborted*, nor *Error*.

Speed-controlled operation of a position-controlled axis

The transition from "position-controlled motion" to "speed-controlled motion" and vice versa can be set both when the axis is at a standstill and when it is moving.

You can use input parameter *PositionControl* to set position- or speed-controlled mode at position-controlled axes.

In speed-controlled mode with velocity preset *0*, the axes can be ramped down immediately, regardless whether a greater following error has developed at the axis as a result of torque limiting, for example.

Software limit switch monitoring stays active.

Note

Speed-controlled operation is maintained until one of the following positioning commands is started:

- "MC_MoveAbsolute"
- "MC_MoveRelative"
- "MC_MoveAdditive"
- "MC_MoveVelocity" (when *PositionControl = TRUE*)
- "MC_MoveToEndPos"
- "MC_GearIn"
- "MC_CamIn"

Restraints when speed control is active:

- Homing not possible ("MC_Home")
- The axis must operate position-controlled to be used as a leading axis with setpoint coupling! When *PositionControl = FALSE*, the setpoint position of the axis is held constant.

Note

The effect of the software limit switch in speed-controlled mode depends on the settings in S7T Config **Axis >Limitations**, "Position and velocity" tab, "Effect of the software limit switches" drop-down list.

- **Setting "Stop at software limit switch only when moving in position-controlled mode"**
  With this setting, the user program can command an axis operating in speed-controlled mode to pass the software limit switch. Subsequent retraction is only possible if the software limit switch is deactivated.
  If the user program does not allow this deactivation you can only return the axis to position-controlled mode by cycling POWER OFF / ON. You can then retract the axis.
- **Setting "Stop at software limit switch in all modes of operation and force position-controlled mode"**
  This setting allows you to stop the axis at the software limit switch if operated in speed-controlled mode. The axis changes to position-controlled mode in the first step. Retraction is not required.

MC_MoveVelocity - Example - "Override motion"

Valid for Integrated Technology with firmware V3.1.x or higher

The signal profile below shows the overriding characteristic of (*Mode = 0*) technology function "MC_MoveVelocity".

Phase "a"

The first positive edge at input parameter *Execute* (*Exe_1)* of FB1 initiates axis motion (*Axis_1*) at a velocity *50*. The final velocity is reported at *InVel_1*.

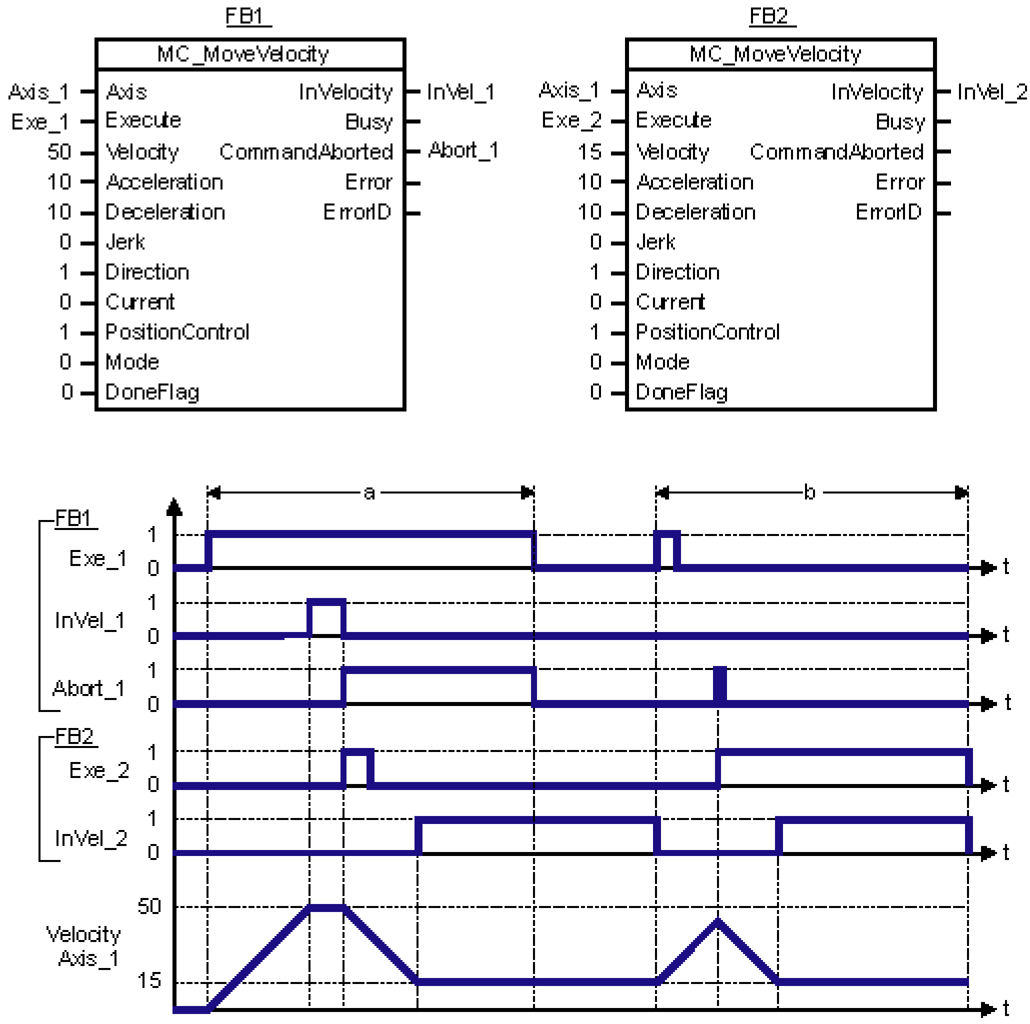
A positive edge at input parameter *Execute* (*Exe_2)* of FB 2 overrides the current motion. The override operation is reported to *Abort_1*. *Abort_1* remains set as long *Exe_1* is set at FB1. After the override, the axis moves (*Axis_1*) at velocity *15*.

Phase "b"

A further positive edge at input parameter *Execute* (*Exe_1)* overrides the current motion and resets velocity *50*.

A further command at FB2 (positive edge at *Exe_2*) overrides the motion before the axis has reached a velocity *50*. After the override, the axis moves (*Axis_1*) at velocity *15*. The final velocity is reported at *InVel_2*.

Parameter *Exe_1* is only set briefly in phase "b"; *Abort_1* is therefore returned only for the duration of one cycle.



MC_MoveVelocity - Example - "Append motion"

Valid for Integrated Technology with firmware V3.1.x or higher

The signal profile below shows the "appending" characteristic of technology function "MC_MoveVelocity" (*Mode = 1*).

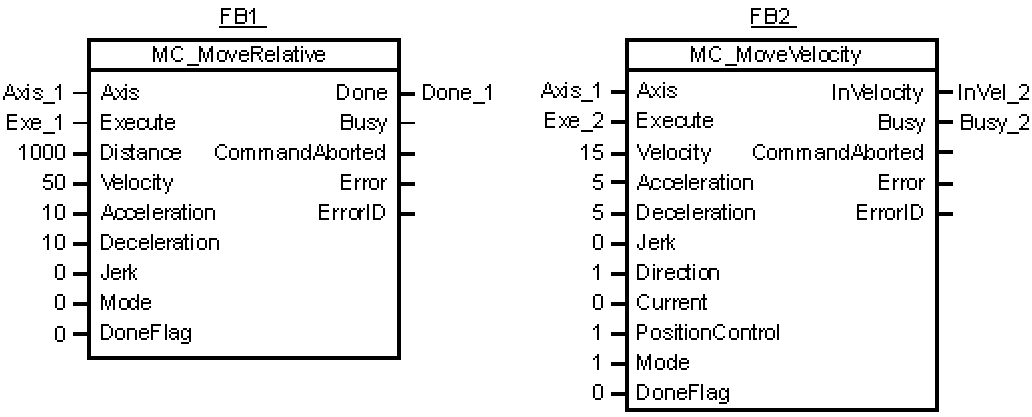
Phase - relative positioning

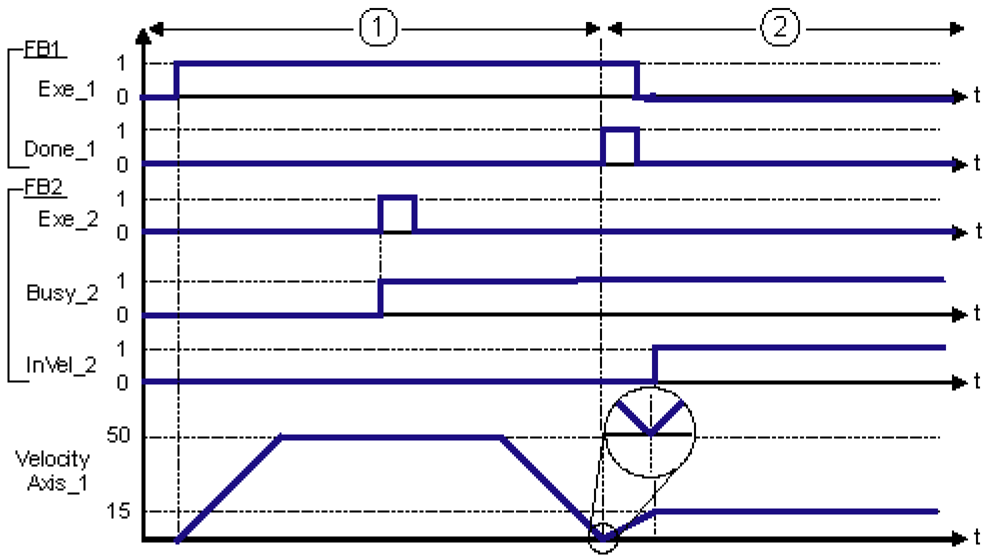
Relative positioning is started with a positive edge at input parameter *Execute* (*Exe_1*) of FB1. The axis (*Axis_1*) accelerates to its final velocity along the set acceleration ramp (*Acceleration = 10*). Before the axis has reached its target position (*Distance = 1000*), the MC_MoveVelocity command is output (positive edge *Exe_2* at FB2).

The current positioning command is not canceled if *Mode = 1* is set at "MC_MoveVelocity". "MC_MoveRelative" decelerates the axis with *Deceleration = 10* and then approaches the target position according to the initiated command. Completion of the command is reported with *Done_1*.

Phase - Motion with velocity preset

As soon as the axis has reached the target position defined by the MC_MoveRelativecommand the MC_MoveVelocity motion is appended immediately. Parameter *InVel_2* reports the final velocity with short delay; the axis (*Axis_1*) moves at the set velocity *15*.





MC_MoveVelocity - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE* . Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Concerns all input parameters of data type REAL, or the input parameters *Direction, Mode* or *DoneFlag.*
8044	Command not supported by the technology object	Sending a command to an output cam, for example
8045	Command not allowed in current state	MC_Stop command active
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8055	Latching motion not allowed in current state	A latching command cannot be accepted at the current axis state. Reasons: • A basic synchronization or superimposing synchronization command is being executed at the axis • "MC_Halt" is active
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.11 FB 415 MC_MoveToEndPos - Move to end stop/clamping

Moving to fixed end stop / clamping by calling FB 415 "MC_MoveToEndPos"

Purpose

- The "MC_MoveToEndPos" technology function moves the axis into contact with an obstruction and holds it in this position, for example, at the end of the distance traversed.
- Define the dynamic response of the motion at the input parameters *Velocity*, *Jerk*,*Acceleration*, *Deceleration*.
- The method and criteria for detecting the end position can be defined axis configuration of S7T Config, for example, following error or torque.
- If drive torque can be limited, torque limiting at the technology function also influences the end stop detection setting "use following error“ (in S7T Config)

Supported for

- Positioning axes (only electrical real axes)
- Synchronization axes (only electrical real axes)

Prerequisites

- Axis is enabled for position-controlled operation
- No active MC_Stop command
- End stop detection must be enabled for the axis.
  Set the end stop detection criteria "use following error“ or "use torque" in the **Axis > Limiting > End stop** dialog box of S7T Config.

Interaction of commands

- Other motion commands can override the current command before the fixed end stop is reached.
- If the axis is clamped at the end stop (*InClamping = TRUE*) the system only accepts motion commands which move the axis away from this position. The MC_MoveToEndPos remains active until the drive has left the clamping tolerance window. New motion commands in direction of the end stop can now be input. Any synchronization command which is executed within the clamping tolerance window to initiate reversal in direction of the end stop will be canceled.
- The MC_MoveToEndPos command can only override another active MC_MoveToEndPos command if both set the same effective direction.

New command - active single command (2) (Page 741)

New command – active commands (2) (Page 746)

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Direction	INT	*0*	Direction preset:	
			Value = *0*:	Default setting
			Value = *1*:	Positive direction of rotation
			Value = *2*:	Not permitted
			Value = *3*:	Negative direction of rotation
			Value = *4*:	Current direction of rotation (last used direction of rotation)
Torque	REAL	*1.0*	Drive torque limit in [N/m]. Condition: The drive supports and uses telegrams 101 to 106, or the maximum torque of the drive is entered.	
			Value > *0*:	Use the defined value
			Value <= *0*:	Not permitted
Velocity	REAL	*0.0*	Maximum velocity (not always reached):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Acceleration (increasing motor power):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Deceleration (decreasing energy in the motor):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk:	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
InClamping	BOOL	*FALSE*	*TRUE*: the axis has reached the end stop and is now within the "position tolerance after fixed end stop detection". Position tolerance setting in S7T Config**Limits > "Fixed end stop" tab >** "Position tolerance after fixed end stop detection" parameter
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 530) of the *Error* output parameter.

Fine resolution of the torque reduction

For firmware versions V4.1.x or higher of the integrated technology, the fine resolution of the torque reduction can be used for SINAMICS and SIMODRIVE drives. You can use the technology function "MC_WriteParameter" to set the fine resolution of torque reduction using parameter *4502*, or configuration parameter *TypeOfAxis.SetPointDriverInfo.DriveData.torqueReductionGranularity* in S7T Config. The drive parameters must be adjusted according to the following table: fine granular torque reduction

Resolution	ENUM value (technology parameter 4502)	SIMODRIVE	SINAMICS
1/100	BASIC (standard setting)	*P0881 = 4000h* (16384) (default)	*P1544 = 4000h* (16384)
1/16384	STANDARD	*P0881 = 64h* (100)	*P1544 = 64h* (100) (default)

Note

If "use following error" is set at the "Fixed end stop" tab of the " **Axis > Limits** " dialog box in the S7T Config axis configuration, following error monitoring is disabled if the technology function is used.

If the fixed end stop breaks, the following error is reduced according to limits configured at the drive. Neither dynamic values defined at the input parameters, nor limits set in the technology object influence this operation.

Always reduce the following error before you stop the axis with "MC_Stop" or "MC_Halt."

Dynamic parameter *Velocity*

Note

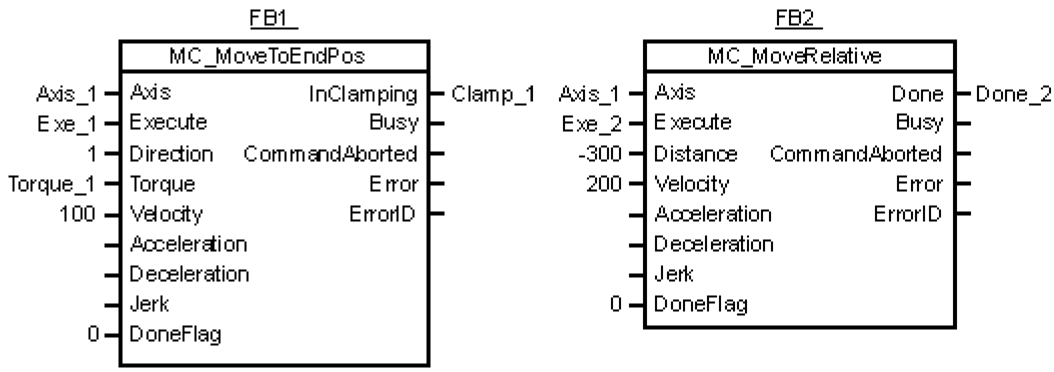
The initial velocity value is set to *0.0* for reasons of safety. This value is invalid in actual fact and will lead to an error. Define a suitable value.

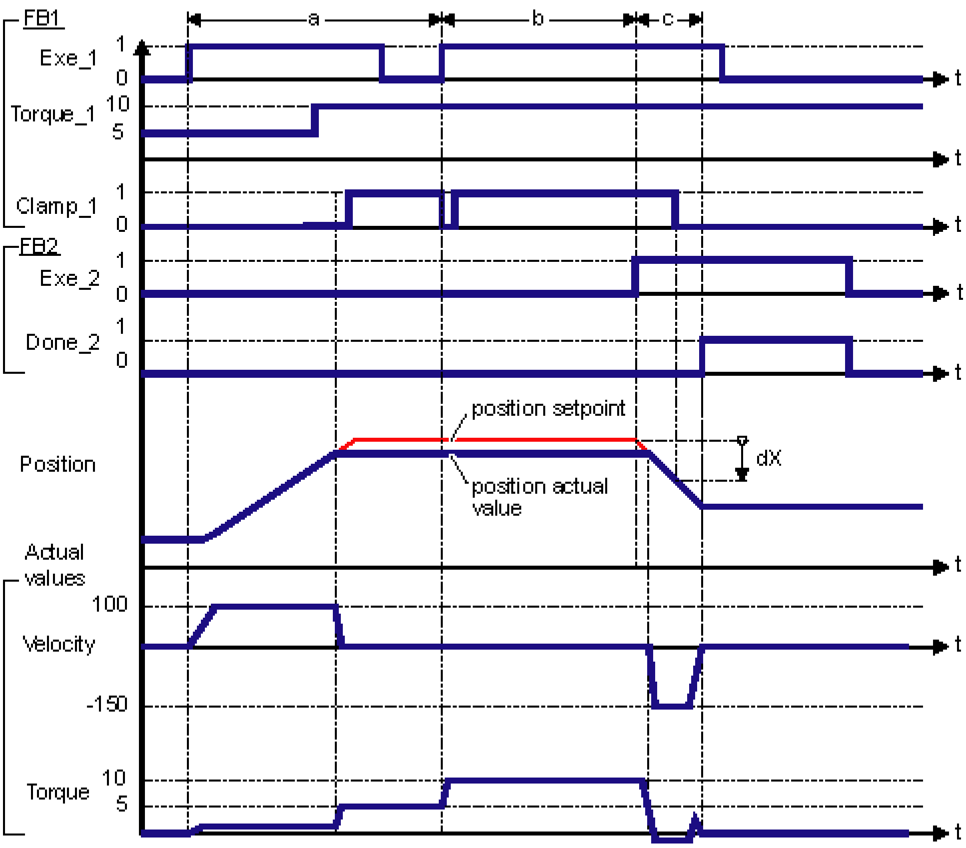
MC_MoveToEndPos - example

a) An axis is moved to the fixed end stop with reduced torque. A following error is generated based on the position setpoint > actual position value in order to maintain torque.

b) The clamping torque is doubled when a new command moves the axis towards the same direction.

c) Clamping is terminated by starting a "MC_MoveRelative" command which releases the axis from the clamping position. Clamping is terminated as soon as the position setpoint is outside of the clamping tolerance window.





MC_MoveToEndPos - ErrorIDs

ErrorID	Warning message	Description / to correct or avoid errors
0000	No error	-
0029	"MC_MoveToEndPos" ignored in reverse direction	A new MC_MoveToEndPos command for reversing the direction was started in order to override the active MC_MoveToEndPos command. This is not allowed. For information on valid commands and conditions for terminating an active MC_MoveToEndPos command, refer to the "MC_MoveToEndPos" documentation.

ErrorID	Error message	Description / to correct or avoid errors
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at these next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* is *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	• Relates to all input parameters of data type REAL, or the *Direction* or *DoneFlag* input parameters. • Fixed end stop detection is not enabled for the axis in S7T Config • Either the message frame used is unequal *101* to *106*, or the value at the *Torque* input parameter does not correspond with the maximum drive torque.
8044	Command not supported by the technology object	Sending a command to an output cam, for example.

ErrorID	Error message	Description / to correct or avoid errors
8045	Command not allowed in current state	• MC_Stop command active • The MC_MoveToEndPos command and clamping are active, and a new instance of "MC_MoveToEndPos" is started to reverse the direction of movement. This is not allowed.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
804D	Invalid axis type	The "MC_MoveToEndPos" technology function is not supported for this axis type. • The axis is a virtual axis • The axis is a hydraulic axis
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DB x, both in OB 1 as well as in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.

ErrorID	Error message	Description / to correct or avoid errors
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology *Objects* Management" and then recreate it.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.12 FB437 MC_SetTorqueLimit - Enable / disable torque limiting

Torque limiting with FB 437 "MC_SetTorqueLimit"

Purpose

- The "MC_SetTorqueLimit" technology function enables/disables torque limiting.
- From firmware of the integrated technology V4.1.x and higher, the torque limit is maintained even after removal of the enable by the "MC-Power" technology function.

Supported for

- Speed-controlled axes
- Positioning axes
- Synchronization axes

Prerequisites

- No active MC_Stop command
- The axis is enabled for speed- or position-controlled operation
- The torque reference for the axis must be set correctly in order to maintain the specified torque The default value of the torque reference (Page 918) is 3.2 N/m.
- The drive must support torque reduction, and must be operated by one of the telegrams 101 to 106.

Interaction of commands

New command - active single command (3) (Page 743)

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
EnableLimit	BOOL	*TRUE*	Enable / disable limits	
MaxTorque	REAL	*3.2*	Maximum effective drive torque. * Define the torque in [N/m].	
			Value = *0*:	Use the defined value
			Value < *0*:	Not permitted
* Always set the "Max. drive torque" parameter in your axis configuration (in the Expert list (Page 918): *TypeOfAxis.SetPointDriverInfo.DriveData.maxTorque*).				

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command successfully executed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID* .
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 537) of the *Error* output parameter.	

Fine resolution of the torque reduction

For firmware versions V4.1.x or higher of the integrated technology, the fine resolution of the torque reduction can be used for SINAMICS and SIMODRIVE drives. You can use the technology function "MC_WriteParameter" to set the fine resolution of torque reduction using parameter *4502*, or configuration parameter *TypeOfAxis.SetPointDriverInfo.DriveData.torqueReductionGranularity* in S7T Config. The drive parameters must be adjusted according to the following table: fine granular torque reduction

Resolution	ENUM value (technology parameter 4502)	SIMODRIVE	SINAMICS
1/100	BASIC (default)	*P0881 = 4000h* (16384) (default)	*P1544 = 4000h* (16384)
1/16384	STANDARD	*P0881 = 64h* (100)	*P1544 = 64h* (100) (default)

Note

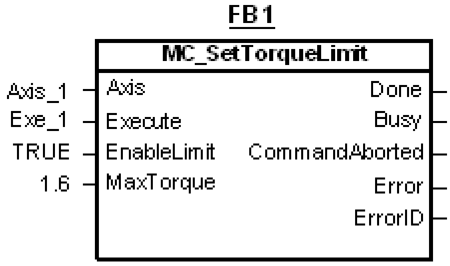
Following error monitoring is disabled when the technology function is being used. Any developed following error is eliminated based on the configured drive limits when torque is increased or countering torque is reduced. Neither dynamic values defined at the input parameters, nor limits set in the technology object influence this operation.

Always reduce the following error before you stop the axis with "MC_Stop" or "MC_Halt." You should preferably stop the axis without position control using technology function "MC_MoveVelocity" with *PositionControl = FALSE* and *Velocity = 0.0* setting.

MC_SetTorqueLimit - example

A value of 3.2 N/m is set by default for torque limiting (configuration parameter *TypeOfAxis.SetPointDriverInfo.maxTorque*).

In the example below torque limiting is reduced to 1.6 N/m.



MC_SetTorqueLimit - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at these next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Concerns *MaxTorque* input parameter (value must be greater than or equal zero)
8044	Command not supported by the technology object	Sending a command to an output cam, for example

ErrorID	Error message	Description / to correct or avoid errors
8045	Command not allowed in current state	MC_Stop command active
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
804D	Invalid axis type	The "MC_SetTorqueLimit" technology function is not supported for this axis type. • The axis is a virtual axis • The axis is a hydraulic axis
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.

ErrorID	Error message	Description / to correct or avoid errors
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.2.13 FB439 MC_SetCharacteristic - Activate valve profile

Activating a valve profile by calling FB 439 "MC_SetCharacteristic"

Supported by Integrated Technology with firmware version V3.2.x or higher

Purpose

- Activating a valve profile for a Q-valve (speed valve)
- The valve profile can also be changed while the hydraulic axis is in motion The optional control signal compensation is limited
- A repeated call of the technology function with a positive edge at input parameter *Execute* cancels the previous command.

Supported for

- Hydraulic speed-controlled axes
- Hydraulic positioning axes
- Hydraulic synchronization axes

Prerequisites

- The selected axis must be configured for operation as hydraulic axis
- The profile is mapped to a cam disk and configured in S7T Config as optional valve profile.

Interaction of commands

New command - active single command (3) (Page 743)

Input parameters

Parameters	Data type	Initial value	Description
Axis	INT	*0*	Number of the axis technology DB
CamTable	INT	*0*	Number of the cam technology disk DB
Execute	BOOL	*FALSE*	Activate profile at the positive edge
Mode	INT	*0*	Profile type value = *0*: Q-valve profile
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB

Output parameters (status outputs)

Parameters	Data Type	Initial value	Significance	
Done	BOOL	*FALSE*	*TRUE*: Profile activated	
Busy	BOOL	*FALSE*	*TRUE:* Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information on the cause, refer to the *ErrorID* .
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 542) of the *Error* output parameter.	

Limitation of control signal compensation

At changeover, the output signal compensation is limited by default to *100%*/s, meaning that an output signal jump of *100%* is compensated for in 1 second.

Control signal compensation can be configured in the expert list of the hydraulic axis:

"System variable" tag *userdefaultqfaxis.maxderivate.qoutput*

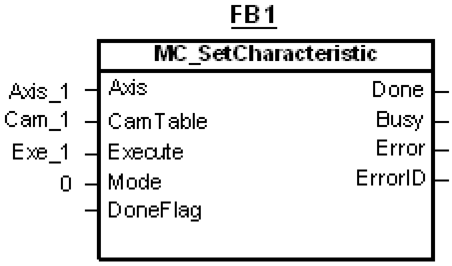
Note

Conditions to be met before you enable the hydraulic axis by calling technology function "MC_Power":

- The hydraulic axis must be assigned a cam disk as valve profile.
- The valve profile must be activated by the "MC_SetCharacteristic" technology function.

MC_SetCharacteristic - example

The characteristic valve profile is stored in cam disk (Cam_1). When called, it is used at the hydraulic axis to determine control signals for motion commands.



MC_SetCharacteristic - ErrorIDs

Supported by Integrated Technology with firmware V3.2.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project / software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Invalid parameter value	Relates to input parameters *Mode* and*DoneFlag.*
8044	Command not supported by the technology object	Relates to input parameters *Axis* and *CamTable*.. • *Axis* is not an axis or is not a hydraulic axis • *CamTable* is not a cam disk
804A	Required object connection is missing	The configured cam disk was not assigned to the axis as valve profile in S7T Config. (**Axis > Profile** dialog box)

ErrorID	Error message	Description / to correct or avoid errors
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.
804D	Invalid axis type	The technology function can only be applied to a hydraulic axis.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*.. • The user has entered invalid data at the technology DB defined at input parameter *Axis* . Delete the technology DB in "Technology Objects Management" and then create a new one.

6.3 Technology functions - Gearing/camming

6.3.1 FB420 MC_GearIn - Start gearing

Starting gearing with FB 420 "MC_GearIn"

Valid for Integrated Technology with firmware V3.2.x or higher

This section describes applications with firmware V3.0.x and V3.1.x.

Purpose

- The "MC_GearIn" technology function starts basic gearing between the leading (master) and following (slave) axes
- You define the dynamic response during synchronization of the following axis at the dynamic parameters *Velocity, Jerk, Acceleration, Deceleration* .
- The gear ratio is defined as a function of the ratio between two integer numbers at the input parameter (numerator / denominator)
- Synchronism can be relative to the start position (random position values upon reaching synchronism) or absolute
- The gear ratio can be changed while the system is in RUN by requesting a new MC_GearIn command. This operation does not require a stop of the leading or following axis Transitions are executed according to specified acceleration or deceleration values
- The function can be started when the leading axis is at a standstill, or when it is in motion.

Supported for

- Synchronization axes

Prerequisites

- The leading axis is configured for operation as external encoder, or as positioning axis, or as synchronization axis
- The following axis is configured for operation as synchronization axis
- The leading axis is set for use as master value at the base synchronization object of the following axis
- The following axis is enabled for position-controlled operation
- No MC_Stop command is being executed at the following axis.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data Type	Initial value	Description	
Master	INT	*0*	Number of the leading axis technology DB	
Slave	INT	*0*	Number of the following axis technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
RatioNumerator	DINT	*1*	Gear ratio numerator (value *0* is not allowed)	
RatioDenominator	DINT	*1*	Gear ratio denominator (value *0* is not allowed)	
Velocity	REAL	*-1.0*	Maximum velocity at the transition to gearing (valid only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Acceleration at the transition to gearing (only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Deceleration at the transition to gearing (only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk at the transition to gearing (only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
PhaseShift	REAL	*0.0*	Phase offset after reaching synchronism The phase offset effect is absolute when synchronous operation is reached if *Absolute = TRUE*. The specified phase offset is added to the phase offset determined by the relative relationship if *Absolute = FALSE*.	
			Value > *0*:	Use the defined value
			Value = *0*:	No phase offset
			Value < *0*:	Not permitted
Absolute	BOOL	*TRUE*	*TRUE*:	Absolute gearing Valid for integrated technology with firmware up to and including V3.2.x: • Parameter preset at *Jerk* is ignored; the trapezoidal motion profile is always used (unlimited jerk)
			FALSE:	Relative gearing

Parameters	Data Type	Initial value	Description
Mode	INT	*0*	Synchronization mode / coupling mode:
			Value = *0*: Use the default values of the corresponding synchronization object
			Value = *1*: Time-based synchronization: Immediate synchronization according to the *Velocity*, *Jerk*, *Acceleration, Deceleration* input parameters (only use for possible compatibility reasons - do not use for new projects).
			Value = *2*: Time-based synchronization: Immediate synchronization according to the *Velocity, Jerk, Acceleration, Deceleration* input parameters. The program also evaluates the *userdefault.gearingsetting.synchronizingdirection* system variable. You can set this system variable in the expert list of S7T Config or by means of the "MC_WriteParameter" technology function.
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB The DoneFlag value is inverted when gearing is reached.

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
InGear	BOOL	*FALSE*	*TRUE*: Basic gearing achieved
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 546) of the *Error* output parameter.

Input parameter *Mode = 1*

Synchronization (compensating) starts immediately when operating with **relative** synchronism. When synchronism is reached, a random offset is set between the leading and following axes.

Synchronization may start with delay when operating with **absolute** synchronism. When synchronism is reached, the actual position value of the leading axis is equivalent to that of the following axis (offset = 0).
At a gear ratio of 1:1 and equal modulo length, or with infinite axes, the offset value stays at zero during synchronous operation.

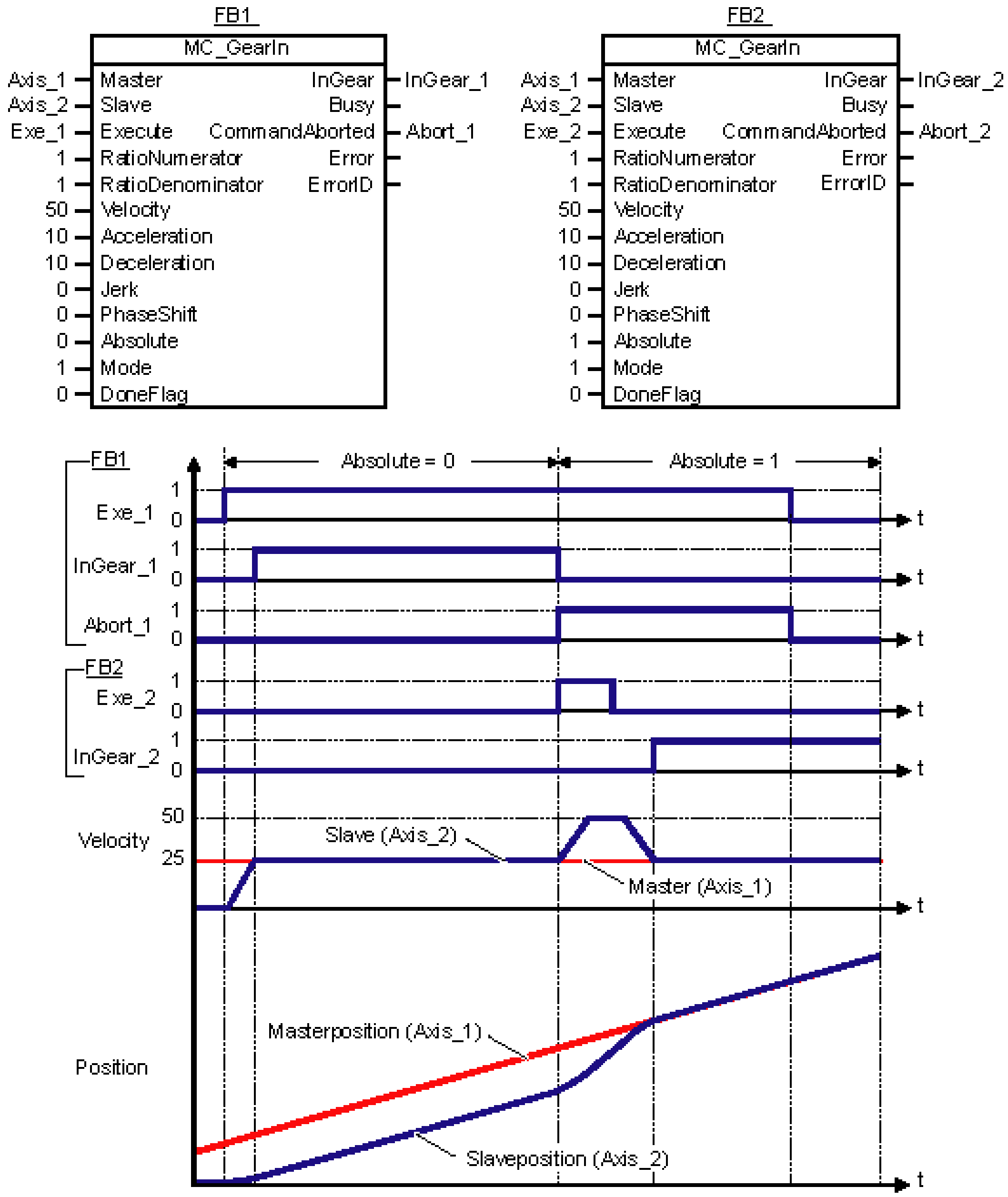
Synchronization direction at modulo axes: "Compatibility mode" is active if *Mode = 1*, regardless of the settings in S7T Config:

- Gearing:
  The following axis is synchronized in positive direction

MC_GearIn - Example - "Relative/absolute synchronism"

Absolute / relative base synchronism after synchronization based on preset dynamic parameters

In the first step, Axis_2 is synchronized to Axis_1 when relative synchronism is active (*Absolute = FALSE*). A random offset develops between both axes. Synchronization is restarted with *Absolute = TRUE*. This eliminates any offset between the axes.


FB1
MC_GearIn
Axis_1 Master InGear InGear_1
Axis_2 Slave Busy
Exe_1 Execute CommandAborted Abort_1
1 RatioNumerator Error
1 RatioDenominator ErrorID
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 PhaseShift
0 Absolute
1 Mode
0 DoneFlag
FB2
MC_GearIn
Axis_1 Master InGear InGear_2
Axis_2 Slave Busy
Exe_2 Execute CommandAborted Abort_2
1 RatioNumerator Error
1 RatioDenominator ErrorID
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 PhaseShift
1 Absolute
1 Mode
0 DoneFlag
Absolute = 0
Absolute = 1
FB1
Exe_1
InGear_1
Abort_1
FB2
Exe_2
InGear_2
Velocity
50
25
Slave (Axis_2)
Master (Axis_1)
Position
Masterposition (Axis_1)
Slaveposition (Axis_2)
t

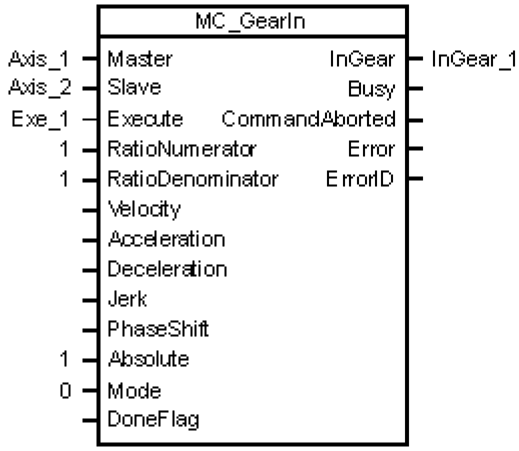
MC_GearIn - Example - "Synchronization based on configuration data"

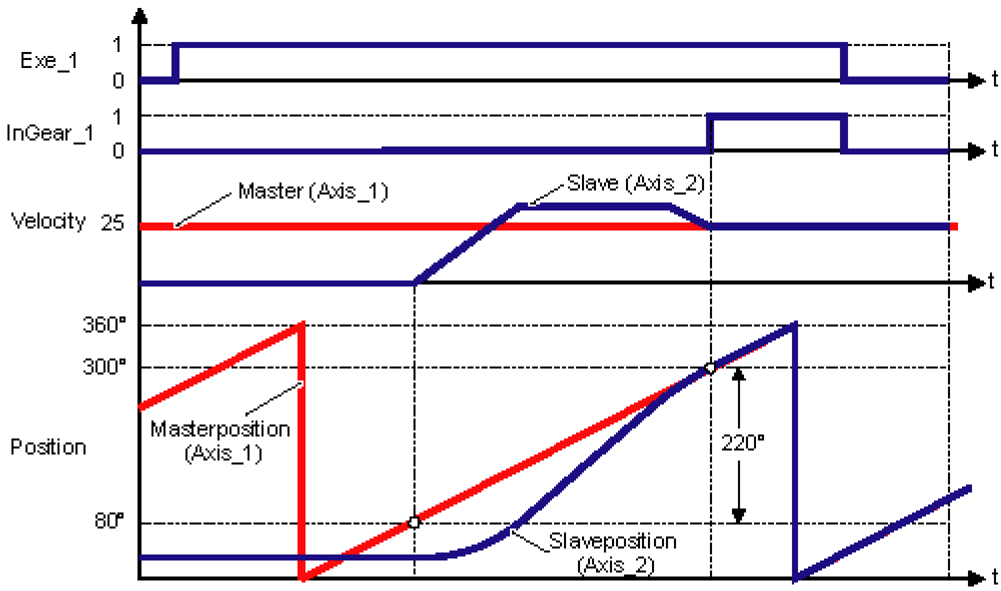
If *Mode* = *0*, the axes are synchronized based on axis configuration data. In this example, we have configured:

Synchronization:	"*Default synchronization position of the leading axis*"
Position reference:	"*Synchronize from synchronization position*"
Sync. pos. master setpoint:	*80.0*
Profile setting:	"*Synchronization profile specific to the leading axis*"
Sync. Length:	*220.0*

Synchronization starts when the leading axis has passed the synchronization position at *80 °*. The operation ends when the leading axis reaches *220 °*.

The dynamic parameters *Velocity, Acceleration, Deceleration, Jerk* are irrelevant.





MC_GearIn - Example - "Phase shift"

The example below shows the differences based on the signal profile between absolute synchronism with and without phase shift.

Absolute synchronism without phase shift

Start positions of the signal profile:

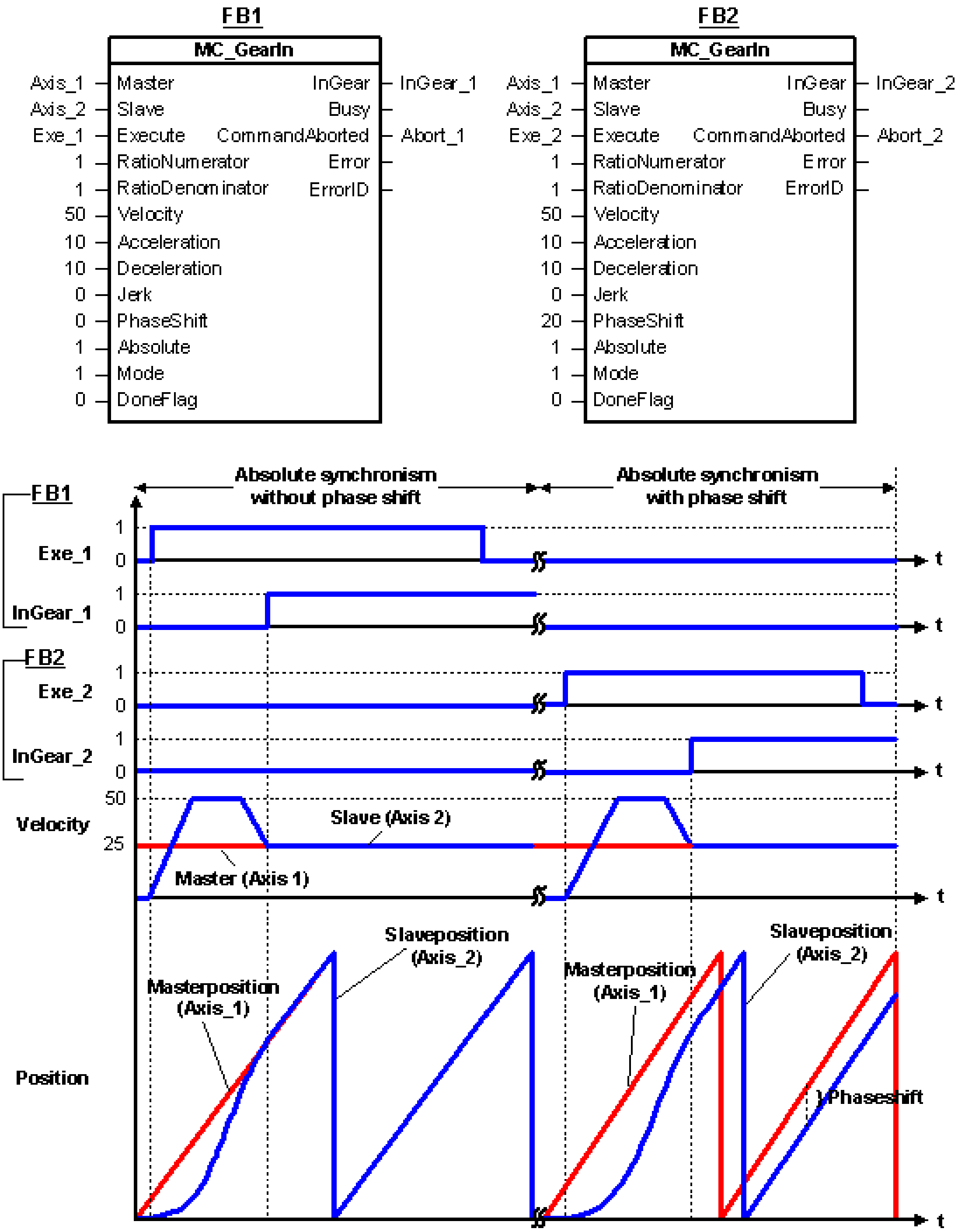
- Master position (*Axis_1*) *= 0*
- Slave position (*Axis_2*) *= 0*

A positive edge at *Exe_1* starts absolute synchronism without phase shift. After a short delay, *InGear_1* reports that absolute synchronism is reached. Absolute synchronism is reached between the leading axis position (*Axis_1)* and the following axis position (*Axis_2*), that is, their positions are identical.

Absolute synchronism with phase shift

The signal profile shown applies to the start positions same as to the signal profile without phase shift.

A positive edge at *Exe_2* starts absolute synchronism with phase shift (*PhaseShift = 20*). After a short delay, *InGear_2* reports that absolute synchronism is reached. The specified phase shift between the leading axis position and the following axis position is applied.


FB1
MC_GearIn
Axis_1 Master InGear InGear_1
Axis_2 Slave Busy
Exe_1 Execute CommandAborted Abort_1
1 RatioNumerator Error
1 RatioDenominator ErrorID
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 PhaseShift
1 Absolute
1 Mode
0 DoneFlag
FB2
MC_GearIn
Axis_1 Master InGear InGear_2
Axis_2 Slave Busy
Exe_2 Execute CommandAborted Abort_2
1 RatioNumerator Error
1 RatioDenominator ErrorID
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
20 PhaseShift
1 Absolute
1 Mode
0 DoneFlag
Absolute synchronism without phase shift
Absolute synchronism with phase shift
FB1
Exe_1
InGear_1
FB2
Exe_2
InGear_2
Velocity
Slave (Axis 2)
Master (Axis 1)
Position
Slaveposition (Axis_2)
Masterposition (Axis_1)
Phaseshift
t

MC_GearIn - ErrorIDs

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware V3.0.x and V3.1.x.

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8022	No actual value at the axis / external encoder	Encoder or data bus not ready, for example
8040	The axis / external encoder are disabled, or the wrong mode is set	The signal required for a command to enable the following axis is missing.
8043	Invalid parameter value or default value	Concerns all input parameters of data type REAL, or the input parameters *RatioNumerator, RatioDenominator, Mode* or *DoneFlag*. There may be a configuration error in S7T Config, typical for a configuration with synchronization length *0*.
8044	Command not supported by the technology object	The technology DB defined at input parameter *Master* must be of the type "positioning axis", "synchronization axis" or external encoder". The technology DB defined at input parameter *Slave* must be of the type "synchronization axis."
8045	Command not allowed in current state	MC_Stop command active
804A	Required object connection is missing	The leading axis defined at input parameter *Master* is not selected in the configuration of the basic synchronization object.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.

ErrorID	Error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
807A	Invalid master setpoint	Invalid actual value of the leading axis.
807B	Recursive interconnection of technology objects	The leading axis defined at the input parameter is already active as following axis in synchronous operation, and the master setpoint is derived directly or indirectly from the following axis defined at the input parameter.
8083	DB is not a technology DB	The DB specified at input parameter *Master* or *Slave* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Master* or *Slave*. Download the current technology to the target system or change the DB number at input parameter *Master* or *Slave*. • The user has entered invalid data at the technology DB defined at input parameter *Master* or *Slave*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.3.2 FB422 MC_GearOut - End gearing

Stopping gearing with FB 422 "MC_GearOut"

Purpose

- The "MC_GearOut" technology function ends basic gearing. Superimposing motions or superimposed synchronism are not affected by this operation.
- Synchronous operation is stopped according to the default setting of the synchronization object in S7T Config .
- The technology function is terminated after basic gearing of the following axis was stopped.

Note

Use the technology function when the shutdown process should depend on the position of the leading axis and/or of the following axis.
The "MC_Halt", "MC_Stop", "MC_MoveRelative", "MC_MoveAdditive", "MC_MoveAbsolute", "MC_MoveVelocity" or "MC_MoveToEndPos" technology functions can also be used to end synchronous operation of the following axis.

Supported for

- Synchronization axes

Prerequisites

- The axis must be operated as following axis (slave) for basic gearing. The command is ignored if no basic gearing is active and the technology function reports *Done = TRUE*.
- No MC_Stop command in process at the following axis.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data Type	Initial value	Description
Slave	INT	*0*	Number of the following axis technology DB
Execute	BOOL	*FALSE*	Start of the command at the positive edge

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Basic gearing has been ended Output parameter *Done* also reports *TRUE* if no basic gearing command was active.
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 556) of the *Error* output parameter.

Note

Simultaneous output of MC_GearIn and MC_GearOut to the same technology object may cancel the MC_GearIn command (*CommandAborted = TRUE*). The "MC_GearOut" technology function therefore reports *Done = TRUE.*

Do not start the MC_GearOut command unless the MC_GearIn has reported the synchronized state (*InGear = TRUE)*.

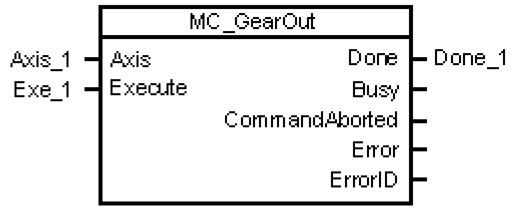
MC_GearOut - example

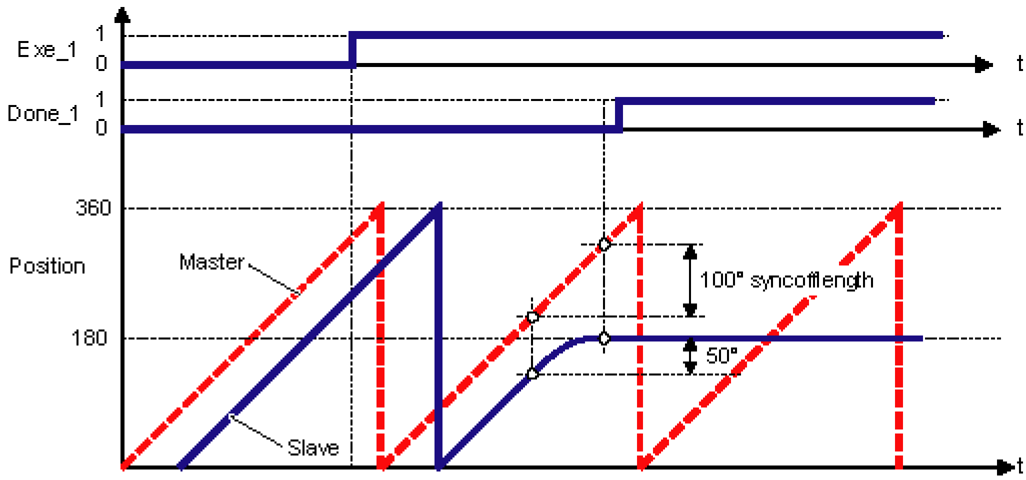
The "MC_GearOut" technology function is called to stop basic gearing and to stop the axis according to the configured mode. In this example we have a relative basic gearing with an offset between the leading and following axes which has developed during synchronization. The gear ratio is 1:1. The following axis should stop at position 180 °. Corresponding settings:

Desynchronization: "Specification of desynchronization position for following axis"
Position reference: "Stop before desynchronization position"
Desynchronization following axis: 180 °

The axis is desynchronized based on the leading axis position. The following axis is desynchronized within the desynchronization length of 100 ° covered by the leading axis. The following axis covers half the distance of the leading axis (50 °).

Profile setting: "Leading axis-related synchronization profile"
Desynchronization Length: 100 °





MC_GearOut - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The signal required for a command to enable the following axis is missing.
8043	Invalid parameter value or default value	There may be a configuration error in S7T Config, typically a desynchronization length of *0* is configured.
8044	Command not supported by the technology object	The technology DB defined at input parameter *Slave* must be at least of the type "synchronization axis."
8045	Command not allowed in current state	"MC_Stop" is still active, for example
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (positive or negative edge at input parameter *Execute / Enable)* is not transferred to the integrated technology • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Slave* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Slave*. Download the current technology to the target system, or change the DB number at input parameter *Slave*. • The user has entered invalid data at the technology DB defined at input parameter *Slave*. Delete the technology DB in "Technology Objects Management" and then create a new one.

6.3.3 FB440 MC_GearInSuperImposed - Start superimposed gearing

Starting superimposed gearing with FB440 "MC_GearInSuperImposed"

Supported by Integrated Technology with firmware version V3.2.x or higher

This section describes applications with firmware V3.1.x

Purpose

- The "MC_GearInSuperImposed" technology function starts superimposed gearing between a leading axis and a following axis.
- Define the dynamic response during synchronization of the following axis at the dynamic parameters *Velocity, Jerk, Acceleration, Deceleration* .
- The gear ratio is defined as a function of the ratio between two integer numbers at the input parameter (numerator / denominator)
- Synchronism can be relative to the start position (random position values upon reaching synchronism) or absolute
- The gear ratio can be changed while the system is in RUN by requesting a new MC_GearInSuperImposed command. This operation does not require a stop of the leading or following axis Transitions are executed according to specified acceleration or deceleration values
- The function can be started when the leading axis is at a standstill, or when it is in motion.

Supported for

- Synchronization axes with superimposed synchronization object

Prerequisites

- The leading axis is configured for operation as external encoder, or as positioning axis, or as synchronization axis
- The following axis is configured as synchronization axis with superimposed synchronization object
- The leading axis is set as possible superimposing synchronization object of the following axis
- The following axis is enabled for position-controlled operation
- No MC_Stop command is being executed at the following axis.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data Type	Initial value	Description	
Master	INT	*0*	Number of the leading axis technology DB	
Slave	INT	*0*	Number of the following axis technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
RatioNumerator	DINT	*1*	Gear ratio numerator	
RatioDenominator	DINT	*1*	Gear ratio denominator	
Velocity	REAL	*-1.0*	Maximum velocity at the transition to gearing (valid only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Acceleration at the transition to gearing (only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Deceleration at the transition to gearing (only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk at the transition to gearing (only if *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
PhaseShift	REAL	*0.0*	Phase offset after reaching synchronism. The phase offset effect is absolute when synchronous operation is reached if *Absolute = TRUE*. The specified phase offset is added to the phase offset determined by the relative relationship if *Absolute = FALSE*.	
			Value > *0*:	Use the defined value
			Value = *0*:	No phase offset
			Value < *0*:	Not permitted
Absolute	BOOL	*TRUE*	*TRUE*:	Absolute superimposed gearing Valid for integrated technology with firmware up to and including V3.2.x: • Parameter preset at *Jerk* is ignored. The trapezoidal motion profile is always used (unlimited jerk).
			FALSE:	Relative superimposed gearing

Parameters	Data Type	Initial value	Description	
Mode	INT	*0*	Synchronization mode / coupling mode:	
			Value = *0*:	Use the default values of the corresponding synchronization object
			Value = *1*:	Time-based synchronization: Immediate synchronization according to the *Velocity, Jerk, Acceleration, Deceleration* input parameters (corresponds with "Compatibility mode" for the synchronization direction set in S7T Config)
			Value = *2*:	Time-based synchronization: Immediate synchronization according to the *Velocity, Jerk, Acceleration, Deceleration*. input parameters. The program also evaluates the *userdefault.gearingsetting.synchronizingdirection* system variable. You can set this system variable in the expert list of S7T Config or by means of the "MC_WriteParameter" technology function.
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB The DoneFlag value is inverted when gearing state is reached.	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
InGear	BOOL	*FALSE*	*TRUE*: Superimposed gearing achieved	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
CommandAborted	BOOL	*FALSE*	*TRUE:* The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 561) of the *Error* output parameter.	

Input parameter *Mode = 1*

Synchronization direction at modulo axes: "Compatibility mode" is active if *Mode = 1*, regardless of the settings in S7T Config:

- Gearing:
  The following axis is synchronized in positive direction

MC_GearInSuperImposed - Example - "Superimposition"

The signal profile in the example below demonstrates the differences between absolute base synchronism and absolute superimposing synchronism.

Phase one - Base synchronism

Exe_1 starts absolute base synchronism. The following axis (*Axis_3*) changes to synchronous operation after a short delay. The technology function reports *InGear_1*.

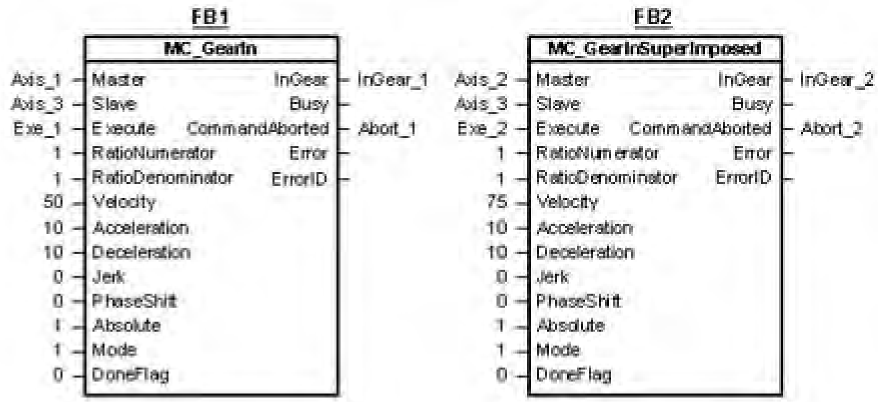
The position of the basic motion corresponds with the position of the additive motion within the phase of base synchronism, because the position of the superimposing motion starts at *0* by default.

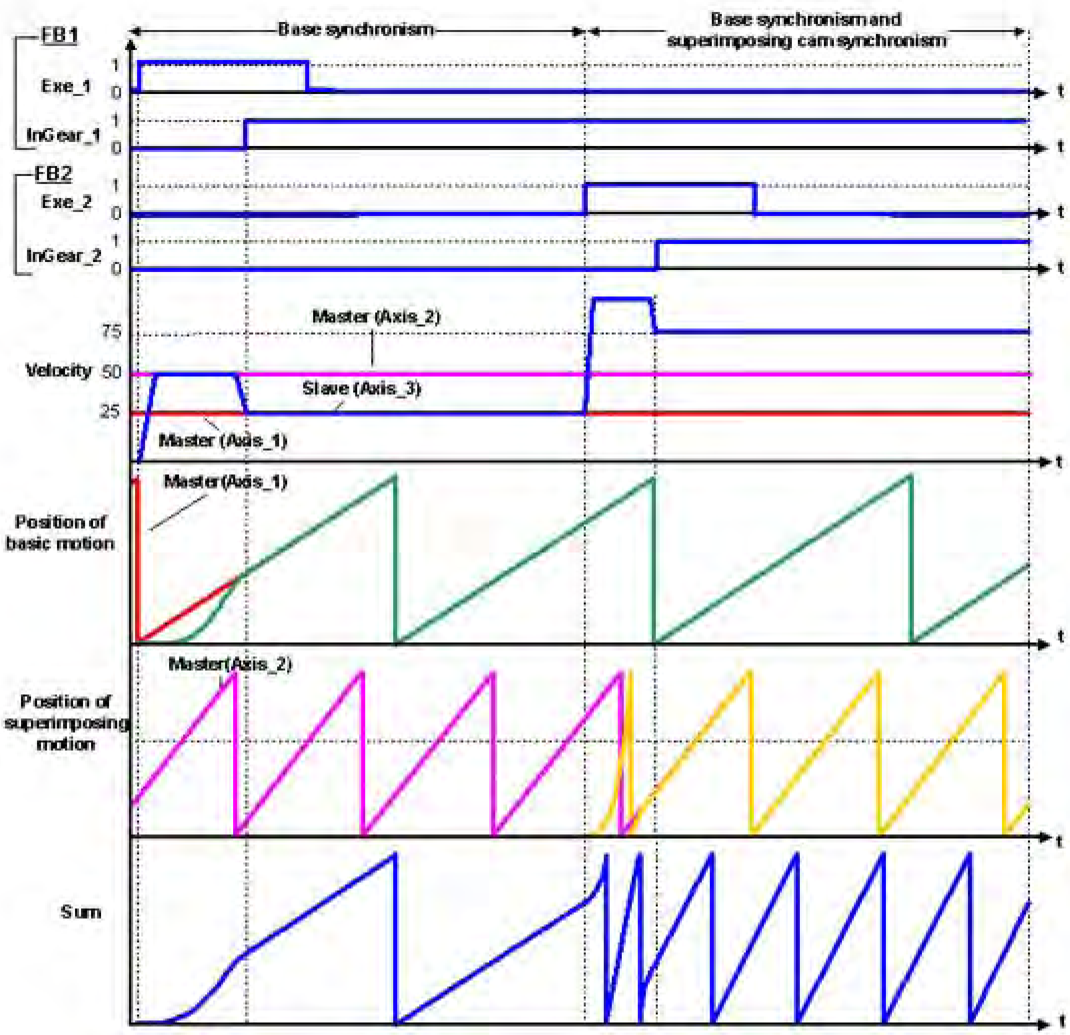
The absolute reference is given as a function of the leading axis position (*Axis_1*) to the position of the basic motion (green signal profile.)

Phase two - Base synchronism and superimposing synchronism

Exe_2 tarts absolute superimposing synchronism (orange-colored signal profile.) The absolute reference is given as a function of the leading axis position (*Axis_2*) to the superimposing motion of the following axis. The absolute reference between the leading axis position (*Axis_1*) and the position of the basic motion (green signal profile) is maintained.

The following axis position (*Axis_3*) is derived from the summation of the "position of basic motion" plus the "position of superimposing motion" values. The technology function reports *InGear_2* when superimposed synchronism is reached.


FB1
MC_GearIn
Axis_1
Axis_3
Exe_1
1
1
50
10
10
0
0
1
1
0
Master
Slave
Execute
RatioNumerator
RatioDenominator
Velocity
Acceleration
Deceleration
Jerk
PhaseShift
Absolute
Mode
DoneFlag
InGear
Busy
CommandAborted
Error
ErrorID
InGear_1
Abort_1
FB2
MC_GearInSuperImposed
Axis_2
Axis_3
Exe_2
1
1
75
10
10
0
0
1
1
0
Master
Slave
Execute
RatioNumerator
RatioDenominator
Velocity
Acceleration
Deceleration
Jerk
PhaseShift
Absolute
Mode
DoneFlag
InGear
Busy
CommandAborted
Error
ErrorID
InGear_2
Abort_2


Base synchronism
Base synchronism and superimposing cam synchronism
FB1
Exe_1
InGear_1
FB2
Exe_2
InGear_2
Velocity
75
50
25
Master (Axis_2)
Slave (Axis_3)
Master (Axis_1)
Position of basic motion
Master(Axis_1)
Position of superimposing motion
Master(Axis_2)
Sum
t

MC_GearInSuperImposed - Example - "Relative/absolute superimposing synchronism"

The signal profile in the example below demonstrates the differences between relative and absolute superimposing synchronism. In order to obtain a better overview, basic synchronism was not carried out in the displayed signal profile.

Phase one - relative superimposing synchronism

The leading axis (*Axis_1*) and following axis (*Axis_2*) start at the same position in the signal profile shown. Relative superimposing synchronism starts with a positive edge at *Exe_1*. After a short delay, *InGear_1* reports that relative superimposing synchronism is reached.

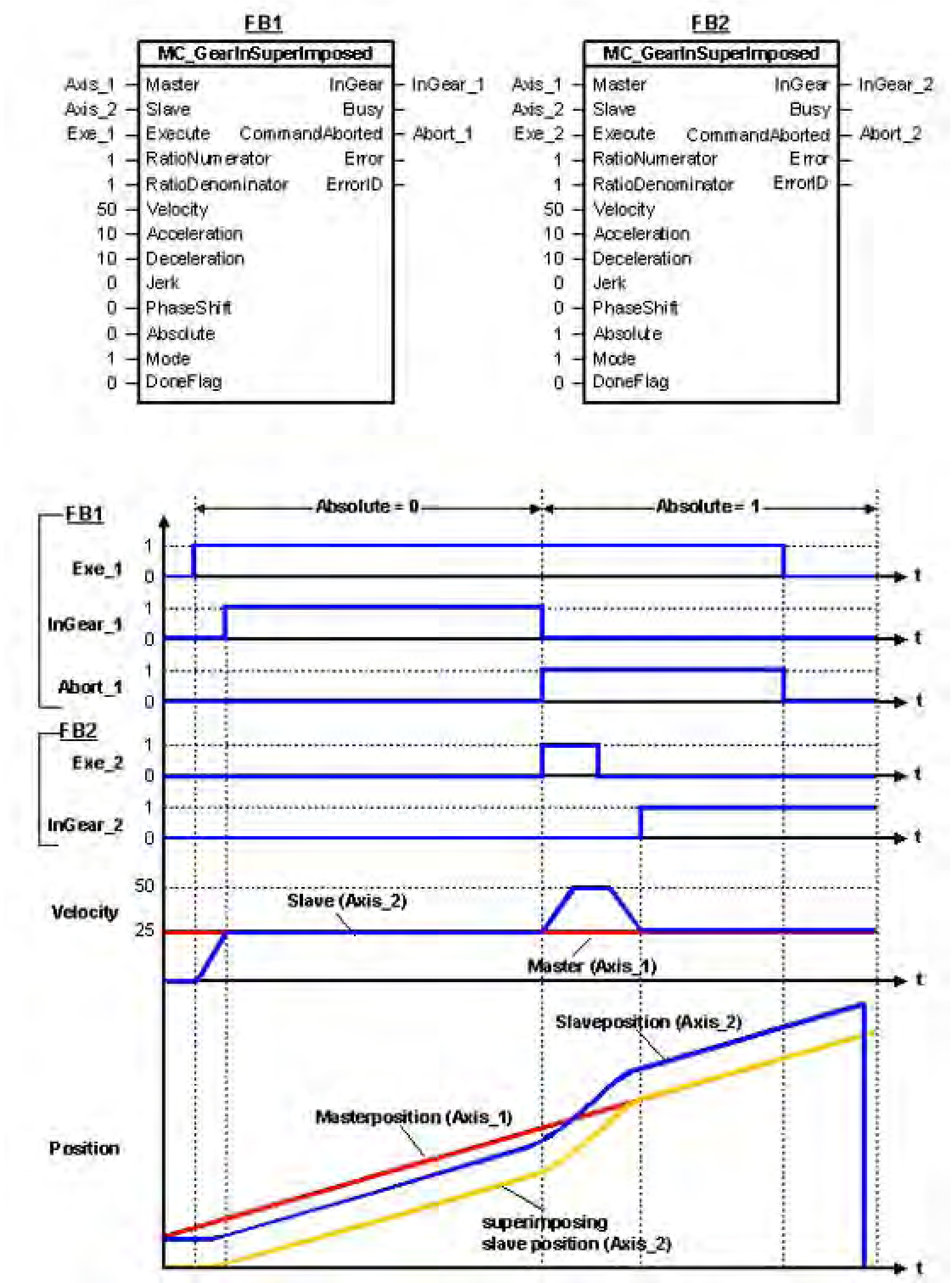
The signal profile in orange color (phase 1) shows the superimposing following axis position (*Axis_2*.) The superimposing following axis position starts at *0* by default.

Relative synchronism is established between the leading axis position (*Axis_1)* and the superimposing following axis position (*Axis_2*.)

Phase two - absolute superimposing synchronism

Exe_2 cancels current relative superimposing synchronism and starts absolute superimposing synchronism. After a short delay, *InGear_2* reports that absolute superimposing synchronism is reached.

The signal profile in orange color (phase 2) shows the superimposing following axis position (*Axis_2*.) Absolute synchronism is established between the leading axis position (*Axis_1*) and the superimposing following axis position (*Axis_2*.) The reference to the original superimposing following axis position is retained, including the offset between the superimposing following axis position and the following axis position.


FB1
MC_GearInSuperImposed
Axis_1 Master
Axis_2 Slave
Exe_1 Execute
1 RatioNumerator
1 RatioDenominator
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 PhaseShift
0 Absolute
1 Mode
0 DoneFlag
InGear InGear_1
Busy
CommandAborted Abort_1
Error
ErrorID
FB2
MC_GearInSuperImposed
Axis_1 Master
Axis_2 Slave
Exe_2 Execute
1 RatioNumerator
1 RatioDenominator
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 PhaseShift
1 Absolute
1 Mode
0 DoneFlag
InGear InGear_2
Busy
CommandAborted Abort_2
Error
ErrorID
Absolute = 0
Absolute = 1
FB1
Exe_1
InGear_1
Abort_1
FB2
Exe_2
InGear_2
Velocity
50
25
Slave (Axis_2)
Master (Axis_1)
Position
Slaveposition (Axis_2)
Masterposition (Axis_1)
superimposing
slave position (Axis_2)
t

MC_GearInSuperImposed - Example - "Phase shift"

The signal profile in the example below shows the differences between absolute superimposing synchronism with and without phase shift. In order to obtain a better overview, basic synchronism was not carried out in the displayed signal profile.

Absolute superimposing synchronism without phase shift

Start positions of the signal profile:

- Leading axis position (*Axis_1*) *= 0*
- Following axis position (*Axis_2*) *= X*
- Superimposing following axis position *= 0*

A positive edge at *Exe_1* starts absolute superimposing synchronism without phase shift. After a short delay, *InGear_1* reports that absolute superimposing synchronism is reached.

The signal profile in orange color indicates the superimposing slave position. The superimposing slave position starts at *0* by default.

Absolute synchronism is established between the leading axis position (*Axis_1)* and the superimposing following axis position (*Axis_2*.)

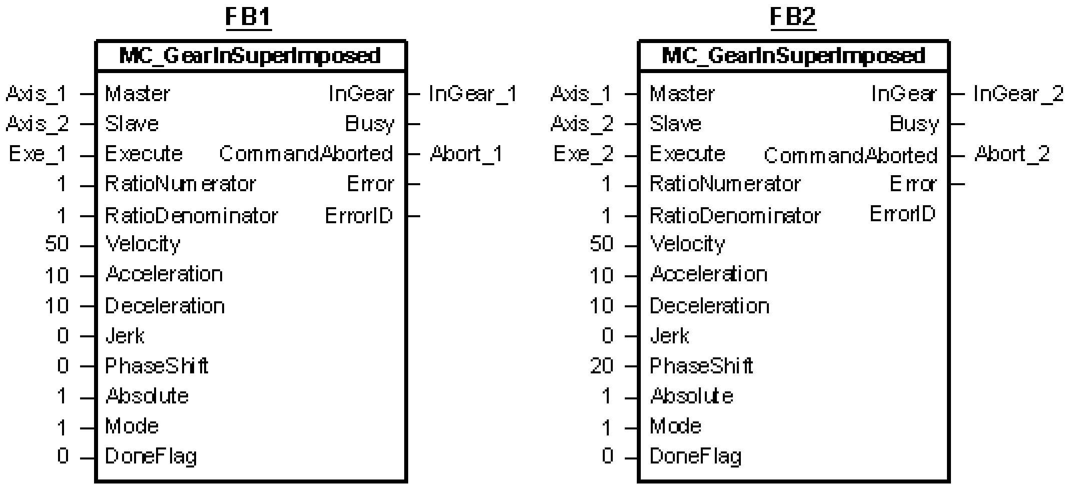
Absolute superimposing synchronism with phase shift

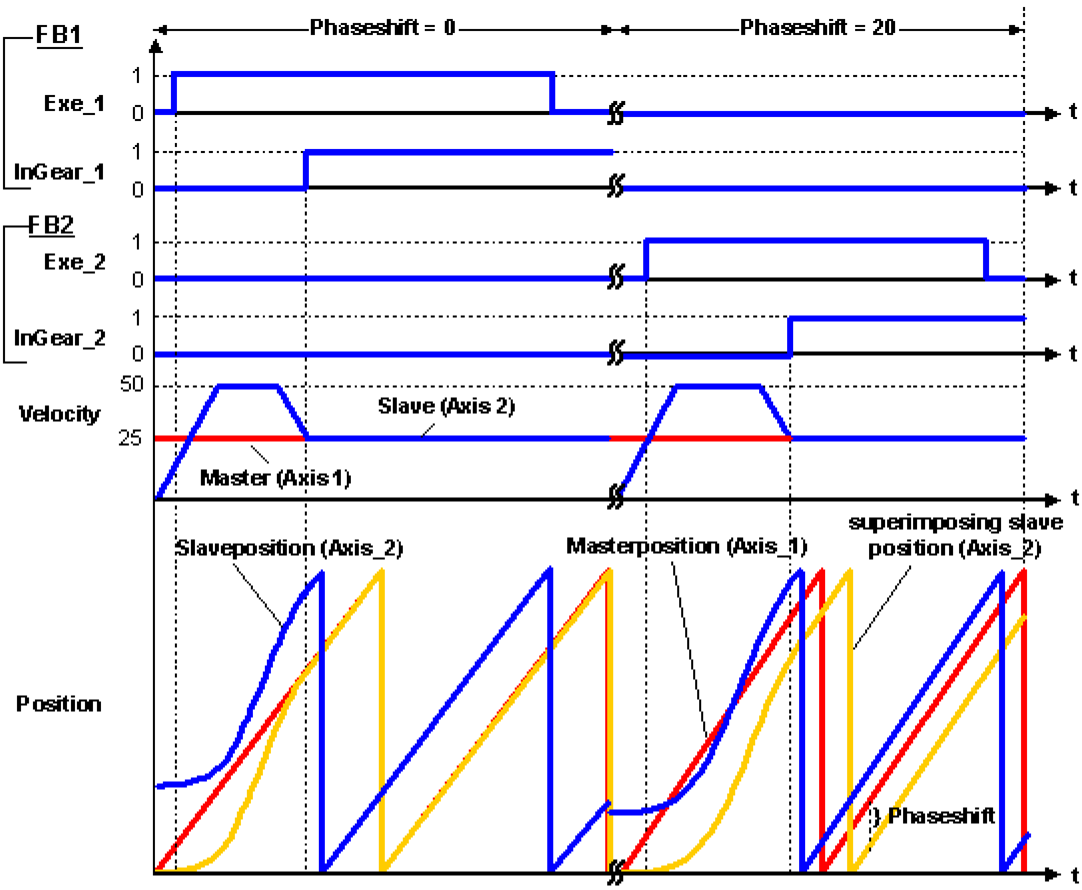
The signal profile applies to the start positions similar to the signal profile without phase shift.

- Leading axis position (*Axis_1*) *= 0*
- Following axis position (*Axis_2*) *= Y*
- Superimposing following axis position *= 0*

A positive edge at *Exe_2* starts absolute superimposing synchronism with phase shift. After a short delay, *InGear_2* reports that absolute superimposing synchronism is reached.

This orange signal profile shows the superimposing following axis position (*Axis_2*), however, with the specified phase shift. Again, the superimposing following axis position starts at *0* by default.


FB1
MC_GearInSuperImposed
Axis_1 Master
Axis_2 Slave
Exe_1 Execute
1 RatioNumerator
1 RatioDenominator
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 PhaseShift
1 Absolute
1 Mode
0 DoneFlag
InGear InGear_1
Busy
CommandAborted Abort_1
Error
ErrorID
FB2
MC_GearInSuperImposed
Axis_1 Master
Axis_2 Slave
Exe_2 Execute
1 RatioNumerator
1 RatioDenominator
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
20 PhaseShift
1 Absolute
1 Mode
0 DoneFlag
InGear InGear_2
Busy
CommandAborted Abort_2
Error
ErrorID


Phaseshift = 0
Phaseshift = 20
FB1
Exe_1
InGear_1
FB2
Exe_2
InGear_2
Velocity
50
25
Slave (Axis 2)
Master (Axis 1)
Position
Slaveposition (Axis_2)
Masterposition (Axis_1)
superimposing slave position (Axis_2)
Phaseshift
t

MC_GearInSuperImposed - ErrorIDs

Supported by Integrated Technology with firmware V3.2.x

This section describes applications with firmware V3.1.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8022	No actual value at the axis / external encoder	Encoder or data bus not ready, for example
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *RatioNumerator, RatioDenominator, Mode* or *DoneFlag* input parameters. This may be caused by the configuration, as this error also appears when a faulty command parameter is found within the block that forms the default based on the configuration, for example, the synchronization length = *0*.
8044	Command not supported by the technology object	The technology DB defined at input parameter *Master* must be at least of the type "positioning axis." The technology DB defined at input parameter *Slave* must be at least of the type "synchronization axis."
8045	Command not allowed in current state	Examples: • MC_Stop command active • The motion of the slave axis is speed-controlled by the technology function "MC_MoveVelocity" if input parameter *PositionControl = TRUE*. The slave axis must be in position-controlled mode for a superimposed synchronous operation.
804A	Required object connection is missing	The leading axis defined at input parameter *Master* is not selected in the configuration of the superimposing synchronization object.

ErrorID	Error message	Description / to correct or avoid errors
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
807A	Invalid master setpoint	Invalid actual value of the leading axis.
807B	Recursive interconnection of technology objects	The leading axis defined at the input parameter is already active as following axis in synchronous operation, and the master setpoint is derived directly or indirectly from the following axis defined at the input parameter.
8083	DB is not a technology DB	One of the DBs defined at the *Master* or *Slave* input parameters was not found or is not a technology DB.

ErrorID	Error message	Description / to correct or avoid errors
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Master or Slave*. Download the current technology to the target system, or change the DB number at input parameter *Master* or *Slave*. • The user has entered invalid data at the technology DB defined at input parameter *Master* or *Slave*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.3.4 FB442 MC_GearOutSuperImposed - Stop superimposed gearing

Stopping superimposed gearing with FB442 "MC_GearOutSuperImposed"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_GearOutSuperImposed" technology function stops superimposed gearing. This does not affect basic motions or basic synchronous operation.
- Superimposed gearing can be stopped immediately in *Mode = 1* using the *Deceleration* and *Jerk* input parameters. You can also desynchronize the axes if *Mode = 0* according to the default settings of the superimposing synchronization object in S7T Config .
- The technology function is terminated after superimposing synchronism of the following axis is terminated.

Supported for

- Synchronization axes with superimposing synchronization object

Prerequisites

- The axis must be configured as following axis (slave) for operation on a system with superimposed synchronization. The command is ignored and the technology function reports *Done = TRUE* if no superimposing gearing is active
- An MC_Stop command is not being executed.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data Type	Initial value	Description	
Slave	INT	*0*	Number of the following axis technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Deceleration	REAL	*-1.0*	Deceleration at the end of superimposed gearing (only if *Mode = 1*)	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk at the end of superimposed gearing (only if *Mode = 1*)	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
Mode	INT	*0*	Desynchronization mode:	
			Value = *0*:	Use the default values of the corresponding synchronization object
			Value = *1*:	Desynchronize immediately according to the input parameters *Deceleration* and *Jerk*
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Superimposed gearing stopped.	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 572) of the *Error* output parameter.	

Note

Simultaneous output of MC_GearInSuperImposed and MC_GearOutSuperImpsed to the same technology object may cancel the MC_GearInSuperImposed command (*CommandAborted = TRUE*). The "MC_GearOutSuperImposed" technology function therefore reports *Done = TRUE.*

Do not start the MC_GearOutSuperImposed command unless the MC_GearInSuperImposed has reported the synchronized state (*InGear = TRUE)*.

MC_GearOutSuperImposed - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *DoneFlag* input parameters.
8044	Command not supported by the technology object	Sending a command to a measuring input or to an output cam, for example
8045	Command not allowed in current state	The MC_Stop command is active, for example
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.

<table>
<tr><th>ErrorID</th><th>error message</th><th>Description / to correct or avoid errors</th></tr>
<tr><td>8050</td><td>Technology not ready</td><td>During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected.
The command was output in a restart OB.</td></tr>
<tr><td>8052</td><td>Block call at different run levels</td><td>This technology function was called at different run levels using the same instance DB.
Example:
Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter ErrorID of both calls.
Error responses to be expected:
The new command (rising or falling edge at input parameter Execute / Enable) is not transferred to the integrated technology.
The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology
Notice:
Use different instance DBs at different run levels, or interlock the call of the technology function.</td></tr>
<tr><td>8053</td><td>Invalid instance DB</td><td>Faulty instance DB of the technology function (wrong length, for example).</td></tr>
<tr><td>8083</td><td>DB is not a technology DB</td><td>The DB specified at input parameter Slave was not found or is not a technology DB.</td></tr>
<tr><td>8084</td><td>Invalid technology DB</td><td>A technology object does not exist in the controller for the technology DB defined at input parameter Slave.
Download the current technology to the target system, or change the DB number at input parameter Slave.
The user has entered invalid data at the technology DB defined at input parameter Slave.
Delete the technology DB in "Technology Objects Management" and then create a new one.</td></tr>
<tr><td>808B</td><td>Parameter value of invalid REAL format</td><td>The value at an input parameter of the data type REAL does not correspond to the valid floating formats.
Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).</td></tr>
</table>

6.3.5 FB421 "MC_CamIn" - Starting camming

Starting camming with FB421 "MC_CamIn"

Valid for Integrated Technology with firmware V3.2.x or higher

This section describes applications with firmware up to V3.1.x

Purpose

- The "MC_CamIn" technology function starts camming between the leading and following axes (master and slave.)
- You can define the dynamic response during synchronization of the following axis at the dynamic parameters *Velocity*, *Jerk, Acceleration* .
- You can scale the specified cam, and/or shift its position.
- The specified cam can optionally be run through once or periodically.
- The synchronism can be absolute or relative.

Supported for

- Synchronization axes

Prerequisites

- The leading axis is configured for operation as external encoder, or as positioning axis, or as synchronization axis
- The following axis is configured for operation as synchronization axis
- In your configuration of the synchronization object of the following axis, you must have selected the required cam disk and leading axis.
- The following axis is enabled for position-controlled operation
- No MC_Stop command is being executed at the following axis.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data Type	Initial value	Description	
Master	INT	*0*	Number of the leading axis technology DB	
Slave	INT	*0*	Number of the following axis technology DB	
CamTable	INT	*0*	Number of the cam technology disk DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
MasterOffset	REAL	*0.0*	Shifting the cam in the leading axis coordinates (only takes effect if *MasterAbsolute = TRUE*)	
SlaveOffset	REAL	*0.0*	Shifting the cam in the following axis coordinates (only takes effect if *SlaveAbsolute = TRUE*)	
MasterScaling	REAL	*1.0*	Cam scaling factor in leading axis coordinates (Value *0* not allowed)	
SlaveScaling	REAL	*1.0*	Cam scaling factor in following axis coordinates (Value *0* not allowed)	
MasterAbsolute	BOOL	*TRUE*	Interpret cam disk coordinates absolute or relative to the leading axis:	
			TRUE:	Absolute coordinates
			FALSE:	Relative coordinates
SlaveAbsolute	BOOL	*TRUE*	Interpret cam disk absolute or relative to the following axis	
			TRUE:	Absolute coordinates
			FALSE:	Relative coordinates
CyclicMode	BOOL	*TRUE*	Processing the cam disk	
			TRUE:	Cyclic execution
			FALSE:	No cyclic execution
Velocity	REAL	*-1.0*	Maximum coupling velocity (only if *Mode* = *1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Acceleration when coupling: (only if *Mode* = *1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk when coupling the following axis (only if *Mode* = *1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">Mode</td><td rowspan="4">INT</td><td rowspan="4">0</td><td colspan="2">Synchronization mode / coupling mode:</td></tr>
<tr><td>Value = 0:</td><td>Use the default values of the corresponding synchronization object</td></tr>
<tr><td>Value = 1:</td><td>Time-based synchronization:
Immediate synchronization according to the Velocity, Jerk, Acceleration
input parameters (corresponds with "Compatibility mode" for the synchronization direction set in S7T Config)</td></tr>
<tr><td>Value = 2:</td><td>Time-based synchronization:
Immediate synchronization according to the input parameters Velocity, Jerk, Acceleration, Deceleration.
The program also makes allowances for settings in the "Gearing" tab of the Axis_SYNCHRONISM > Default dialog box of the synchronization object. It also evaluates system variable userdefault.cammingsetting. synchronizingdirection .</td></tr>
<tr><td>DoneFlag</td><td>INT</td><td>0</td><td colspan="2">DoneFlag (Page 750) generation in the MCDevice DB
The value of DoneFlag is inverted when the command (InSync) is completed</td></tr>
</table>

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
InSync	BOOL	*FALSE*	*TRUE*: Basic camming achieved
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 576) of the *Error* output parameter.

Input parameter *Mode = 1*

Synchronization direction at modulo axes: "Compatibility mode" is active if *Mode = 1*, regardless of the settings in S7T Config:

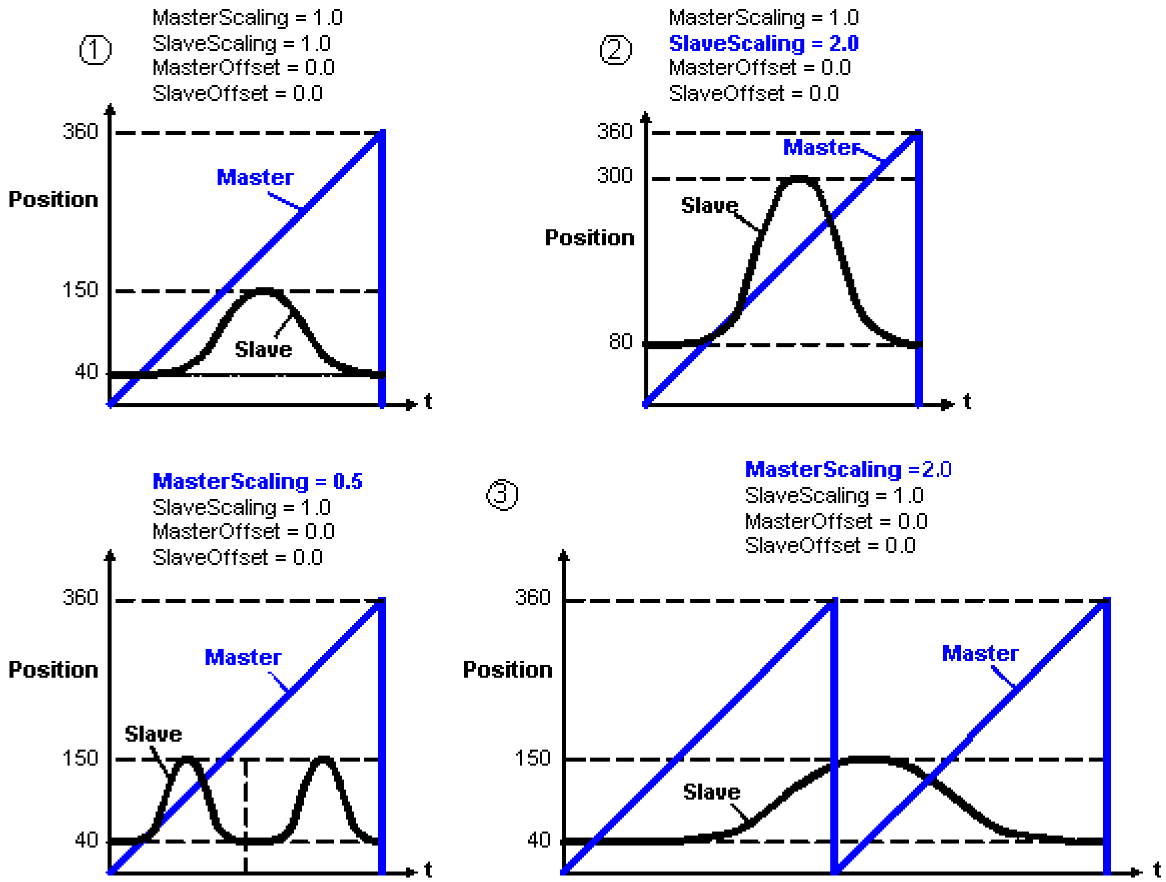
- Camming:
  The following axis is synchronized within the shortest distance.

Recommendation for the start of camming

1. Move the following axis (*Slave*) to position X_S.
2. Move the leading axis (*Master*) to position X_m, whereby X_S = cam value (X_m)
3. Start the "MC_CamIn" technology function.

MC_CamIn - Example - "Effect of scaling factors and offset values"

Effect of the scaling factors

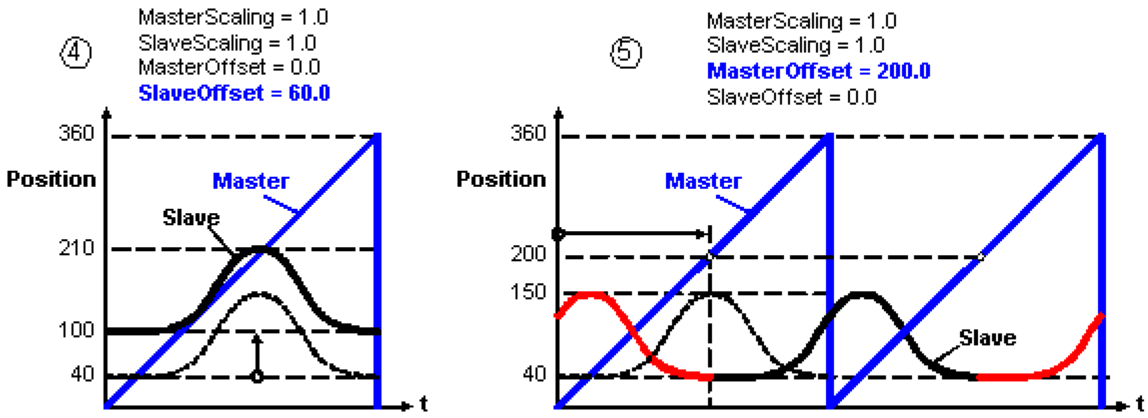


① Non-scaled output cams (scaling factors 1.0.)

② Parameter *SlaveScaling* can be used to expand or compress the cam along the Y coordinate. The following axis is therefore displaced more or less within a leading axis cycle.

③ Parameter *MasterScaling* can be used to expand or compress the cam along the X coordinate.

Effect of offset values



④ The *SlaveOffset* parameter shifts the entire cam along the Y coordinate.

⑤ The *MasterOffset* parameter shifts the cam along the X coordinate to generate a cam with phase shift.

MC_CamIn - Example - "Synchronization with the leading axis position (cyclic relative synchronism)"

Synchronization mode:

Relative camming, cyclic mode

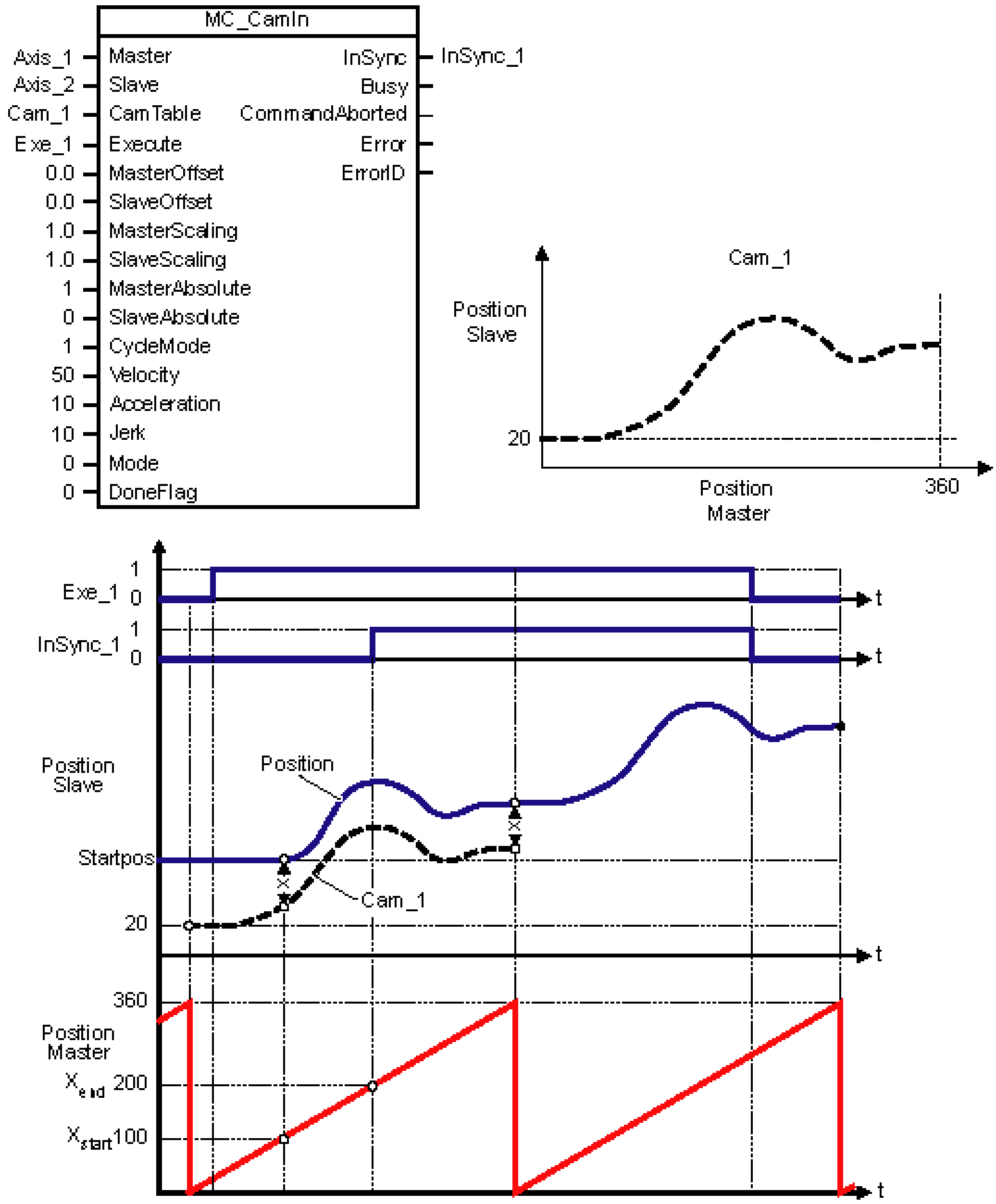
In relative camming mode, the cam start and end values may assume any value along the Y coordinate. At the end of a current cam cycle, the cam is added to the current Y value.

Synchronization condition:

The synchronization condition defined in the axis configuration or by input at the corresponding parameters is active if *Mode* = *0*:

Synchronization:	"*Default synchronization position of the leading axis*"
Position reference:	"*Synchronize from synchronization position*"
Sync. pos. master setpoint:	*100.0*
Profile setting:	"*Synchronization profile specific to the leading axis*"
Sync. Length:	*100.0*

Synchronization starts when the leading axis has passed position 100 °. The following axis is synchronized while the leading axis is moving within the angular range from 100 ° to 200 ° (synchronization length = 100 °).


MC_CamIn
Axis_1
Axis_2
Cam_1
Exe_1
0.0
0.0
1.0
1.0
1
0
1
50
10
10
0
0
Master
Slave
CamTable
Execute
MasterOffset
SlaveOffset
MasterScaling
SlaveScaling
MasterAbsolute
SlaveAbsolute
CycleMode
Velocity
Acceleration
Jerk
Mode
DoneFlag
InSync
Busy
CommandAborted
Error
ErrorID
InSync_1
Cam_1
Position Slave
20
Position Master
360
Exe_1
InSync_1
1
0
t
Position Slave
Position
Startpos
Cam_1
20
360
Position Master
X_{end} 200
X_{start} 100

MC_CamIn - Example - "Synchronization with the leading axis position (cyclic absolute synchronism)"

Synchronization mode:

Absolute camming, cyclic mode

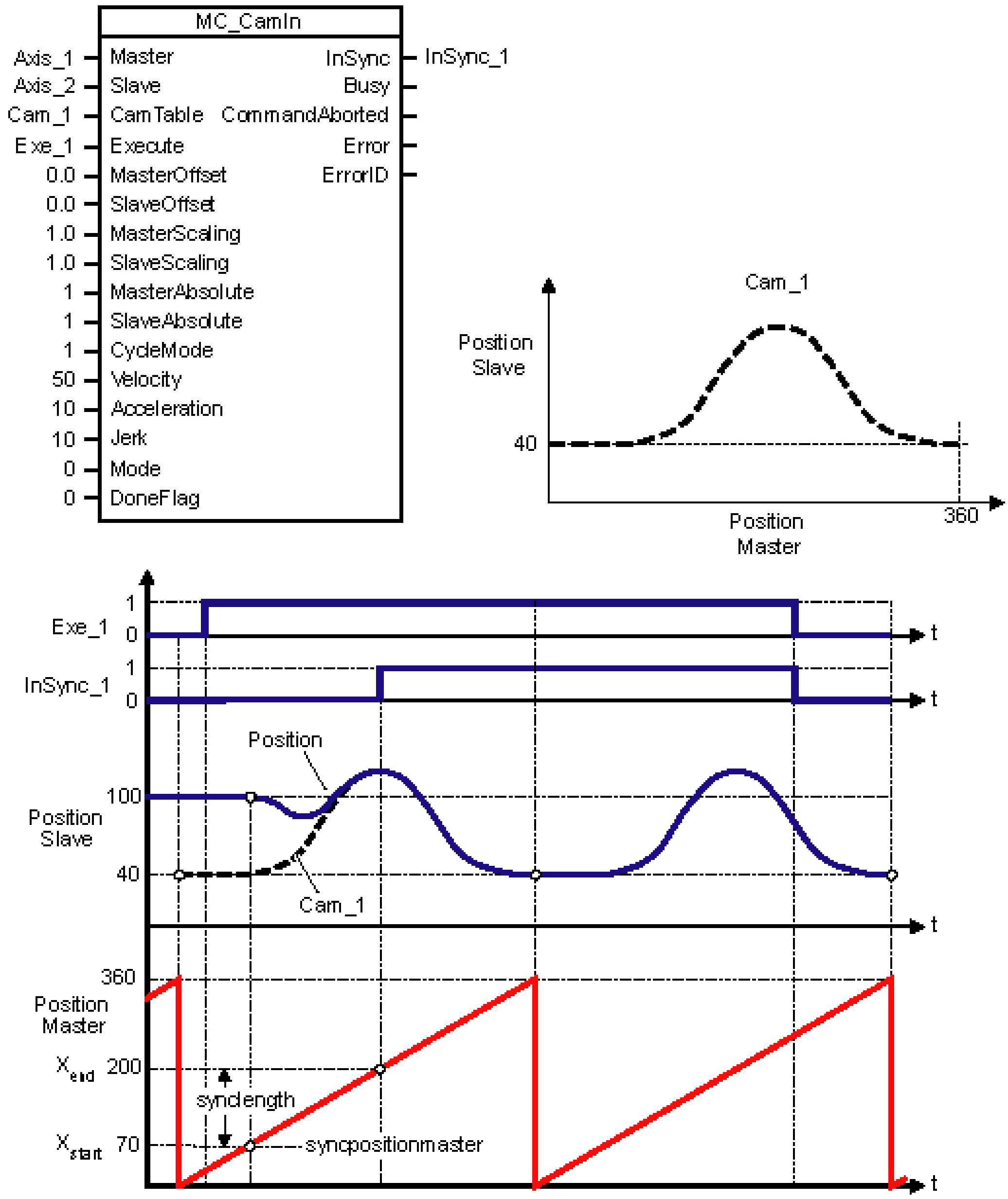
The Y values and gradients at the cam start and end position are identical. This allows you to join the cams in cyclic mode without causing discontinuity.

Synchronization condition:

The synchronization condition defined in the axis configuration or by input at the corresponding parameters is active if *Mode* = *0*:

Synchronization:	"*Default synchronization position of the leading axis*"
Position reference:	"*Synchronize from synchronization position*"
Sync. pos. master setpoint:	*70.0*
Profile setting:	"*Synchronization profile specific to the leading axis*"
Sync. Length:	*130.0*

Synchronization starts when the leading axis has passed position 70 °. The following axis is synchronized while the leading axis is moving within the angular range from 70 ° to 200 ° (synchronization length = 130 °).


MC_CamIn
Axis_1
Axis_2
Cam_1
Exe_1
0.0
0.0
1.0
1.0
1
1
1
50
10
10
0
0
Master
Slave
CamTable
Execute
MasterOffset
SlaveOffset
MasterScaling
SlaveScaling
MasterAbsolute
SlaveAbsolute
CycleMode
Velocity
Acceleration
Jerk
Mode
DoneFlag
InSync
Busy
CommandAborted
Error
ErrorID
InSync_1
Cam_1
Position Slave
40
Position Master
360
Exe_1
InSync_1
1
0
t
Position
Position Slave
100
40
Cam_1
360
Position Master
X_{end} 200
X_{start} 70
synclength
syncpositionmaster

MC_CamIn - "Signal profile with immediate synchronization" example

Synchronization mode:

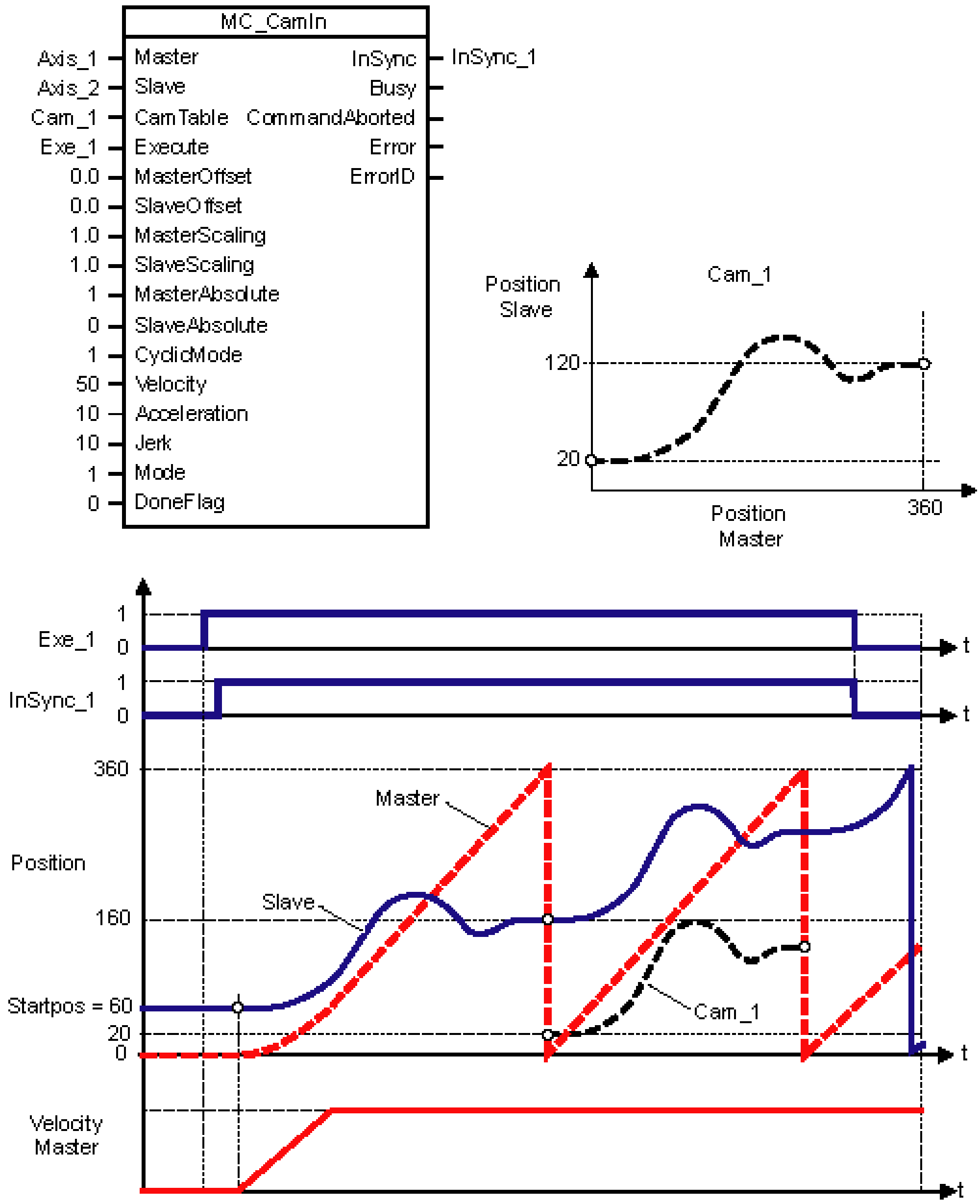
Relative camming, cyclic mode

In relative camming mode, the cam start and end values may assume any value along the Y coordinate. At the end of a current cam cycle, the cam is added to the current Y value.

Synchronization condition:

If *Mode* = *1*, the cam is synchronized immediately based on the dynamic parameters set at the "MC_CamIn" technology function.

In this example, the leading and following axes are initially moved to their initial positions by means of positioning commands. The "MC_CamIn" technology function is started in the next step. Synchronism comes into effect instantaneously, because both axes are at standstill and relative synchronism is requested. For absolute synchronism, the slave would first have to be moved to the absolute cam interpolation point that is assigned to the leading axis.


MC_CamIn
Axis_1
Axis_2
Cam_1
Exe_1
0.0
0.0
1.0
1.0
1
0
1
50
10
10
1
0
Master
Slave
CamTable
Execute
MasterOffset
SlaveOffset
MasterScaling
SlaveScaling
MasterAbsolute
SlaveAbsolute
CyclicMode
Velocity
Acceleration
Jerk
Mode
DoneFlag
InSync
Busy
CommandAborted
Error
ErrorID
InSync_1
Position Slave
Cam_1
120
20
Position Master
360
Exe_1
InSync_1
1
0
1
0
t
t
360
Master
Position
Slave
160
Startpos = 60
20
0
Cam_1
t
Velocity Master
t

MC_CamIn - Example - "Cam changeover at the end of a cycle"

Synchronization mode:

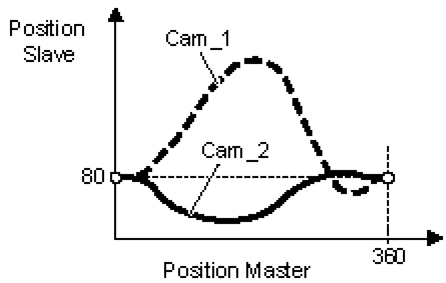
The following axis operates in absolute camming mode. The leading axis must be operated in relative synchronization mode for synchronization "*at the end of active cycle*."

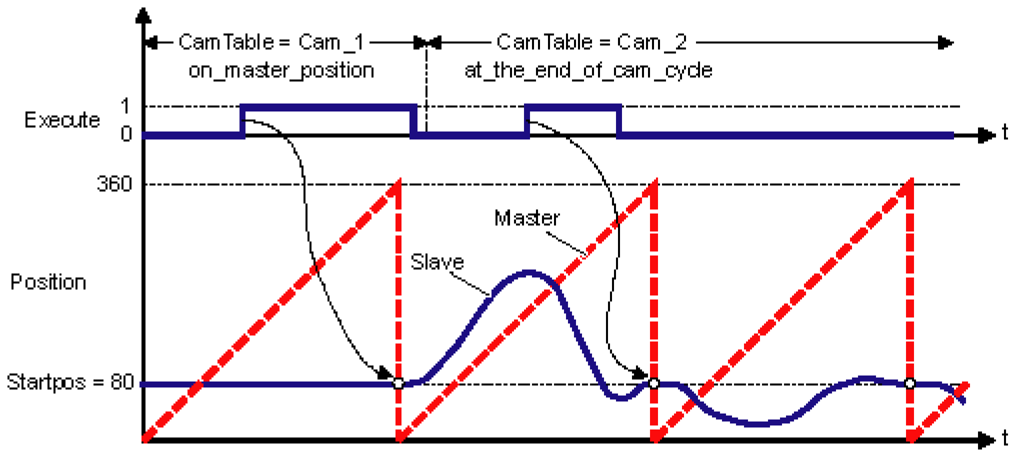
Synchronization condition:

Input parameter *Mode = 0* activates the synchronization condition which is defined in the axis configuration or by entries at the corresponding parameters. In this example, the synchronization condition is changed while the system is in RUN.

Sequence

1. "*Preset synchronization position of the leading axis*" was set in S7T Config as the default synchronization mode of the synchronization axis. The position reference was set to "*Synchronize from synchronization position*", with Sync. Pos. master value *0.0*. *Execute = 1* initiates synchronization with "Cam_1", starting at leading axis value *0.0*.
2. The synchronization position is changed by using the "MC_WriteParameter" technology function. To this purpose Parameter *4266* of the following axis is set to the value "AT_THE_END_OF_CAM_CYCLE".
3. The cam instance of "Cam_2" is set at the *CamTable* input parameter.
4. Synchronization with "Cam_2" starts at the end of the active cams the next time *Execute = 1*.





MC_CamIn - Example - "Synchronization condition AT_MASTER_AND_SLAVE_POSITION"

Synchronization mode:

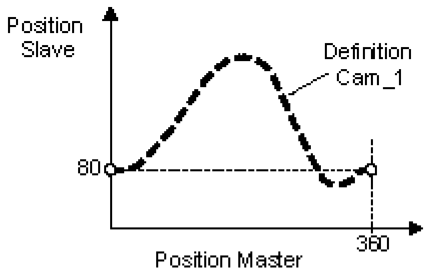
The leading and following axes are operating in absolute camming mode.

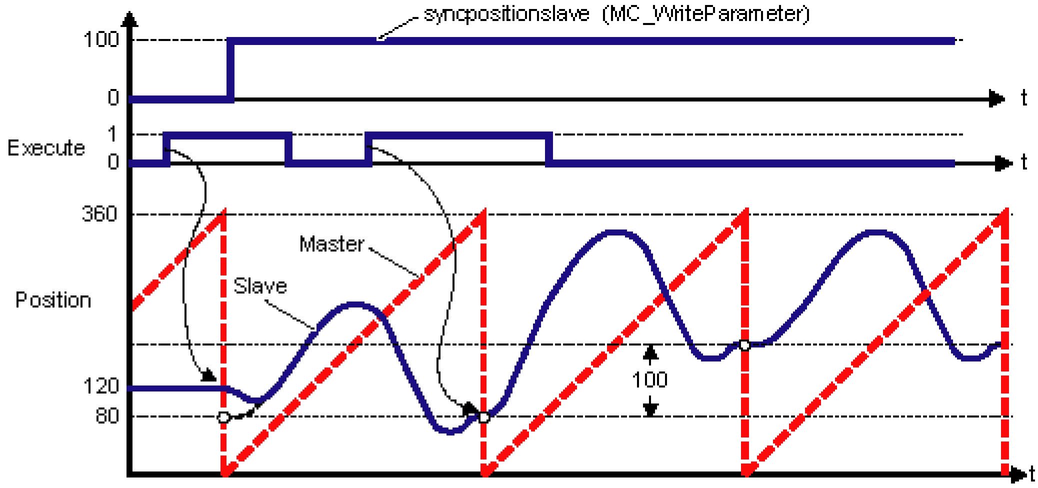
Synchronization condition:

Input parameter *Mode* = *0* activates the synchronization condition which is defined in the axis configuration or by entries at the corresponding parameters.
In this example, the synchronization position (parameter 4269) of the following axis is changed while the system is in RUN.

Sequence

1. Synchronization condition: "*Default synchronization position of the leading axis and following axis*"
   Position reference: "*Synchronize from synchronization position*"
   Sync. Pos. master axis: *0.0*
   Sync. pos. following axis: *0.0*
2. Changing the synchronization position by calling the "MC_WriteParameter" technology function (Parameter *4269*)
3. Synchronization condition: "*Default synchronization position of the leading axis and following axis*"
   Position reference: "*Synchronize from synchronization position*"
   Sync. Pos. master axis: *0.0*
   Sync. pos. following axis: *100.0*





MC_CamIn - ErrorIDs

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware V3.1.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8022	No actual value at the axis / external encoder	Encoder or data bus not ready, for example
8040	The axis / external encoder are disabled, or the wrong mode is set	The signal required for a command to enable the following axis is missing.
8043	Invalid parameter value or default value	Relates to all input parameters of data type REAL, or the *Mode* or *DoneFlag* input parameters. This could also be a faulty configuration in S7T Config. Typically a synchronization length of *0* has been configured.
8044	Command not supported by the technology object	The technology DB defined at input parameter *Master* must be at least of the type "positioning axis." The technology DB defined at input parameter *Slave* must be at least of the type "synchronization axis."
8045	Command not allowed in current state	MC_ Stop command active
804A	Required object connection is missing	Neither the leading axis defined at input parameter *Master* nor the cam disk defined at input parameter *CamTable* are selected in the configuration of the basic synchronization object
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.

ErrorID	Error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
807A	Invalid master setpoint	Invalid actual value of the leading axis.
807B	Recursive interconnection of technology objects	The leading axis defined at the input parameter is already active as following axis in synchronous operation, and the master setpoint is derived directly or indirectly from the following axis defined at the input parameter.
8083	DB is not a technology DB	The DB specified at input parameter *Master, Slave* or *CamTable* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Master, Slave* or *CamTable*. Download the current technology to the target system or change the DB number at input parameter *Master, Slave* or *CamTable*. • The user has entered invalid data for the technology DB defined at input parameter *Master, Slave* or *CamTable*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.3.6 FB423 MC_CamOut - Stopping camming

Stopping camming with FB 423 "MC_CamOut"

Purpose

- "MC_CamOut" stops camming and the following axis. You can define the corresponding conditions in the axis configuration

Note

Use the technology function when the shutdown process should depend on the position of the leading axis and/or of the following axis.
The "MC_Halt", "MC_Stop", "MC_MoveRelative", "MC_MoveAdditive", "MC_MoveAbsolute", "MC_MoveVelocity" or "MC_MoveToEndPos" technology functions can also be used to end synchronous operation of the following axis.

Supported for

- Synchronization axes

Prerequisites

- The axis must be configured as following axis (Slave) for camming operations. The command is ignored if no camming is active and the technology function reports *Done = TRUE*.
- No active MC_Stop command

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data type	Initial value	Description
Slave	INT	*0*	Number of the following axis technology DB
Execute	BOOL	*FALSE*	Start of the command at the positive edge

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Camming is stopped Output parameter *Done* also reports *TRUE* if no basic camming command was active.	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
CommandAborted	BOOL	*FALSE*	*TRUE:* The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 591) of the *Error* output parameter.	

Note

Simultaneous output of MC_CamIn and MC_CamOut to the same technology object may cancel the MC_CamIn command (*CommandAborted = TRUE*). The "MC_CamOut" technology function therefore reports *Done = TRUE.*

Do not start the MC_CamOut command unless the MC_CamIn has reported the camming state (*InSync = TRUE)*.

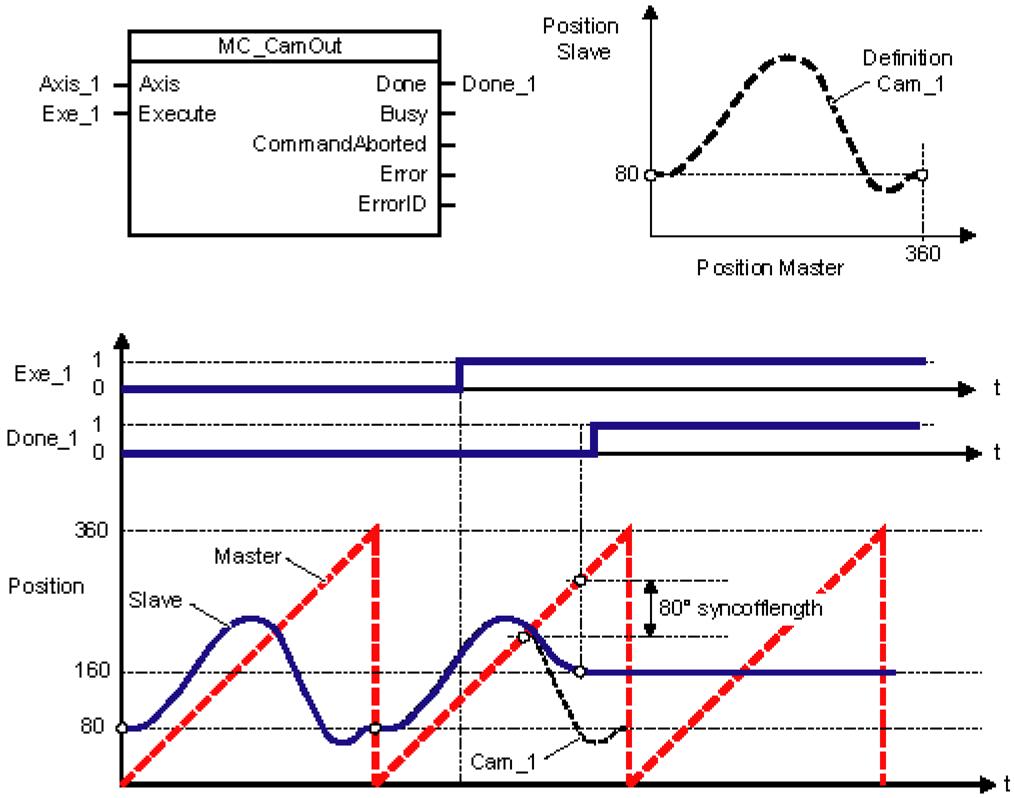
MC_CamOut - example

The "MC_CamOut" technology function is used to stop camming and to stop the axis according to the configured mode. In this example, the following axis should stop at position *160 °*. Corresponding settings:

Desynchronization: "at the following axis position"
Position reference: "Stop before desynchronization position"
Desynchronization following axis: *160 °*

The axis is desynchronized based on the leading axis position. The following axis is desynchronized within the desynchronization length of *80 °* covered by the leading axis. The time from which desynchronization starts is calculated internally.

Profile setting: "Leading axis-related synchronization profile"
Desynchronization Length: *80 °*


MC_CamOut
Axis_1
Exe_1
Axis
Execute
Done
Busy
CommandAborted
Error
ErrorID
Done_1
Position Slave
Definition Cam_1
80
360
Position Master
Exe_1
Done_1
1
0
t
360
160
80
Position
Master
Slave
80° syncofflength
Cam_1

MC_CamOut - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE* . Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The signal required for a command to enable the following axis is missing.
8043	Invalid parameter value or default value	There could also be an error in the configuration in S7T Config. Typically a desynchronization length of *0* has been configured.
8044	Command not supported by the technology object	The technology DB defined at input parameter *Slave* must be at least of the type "synchronization axis."
8045	Command not allowed in current state	"MC_Stop" is still active, for example
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Slave* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Slave*. Download the current technology to the target system, or change the DB number at input parameter *Slave*. • The user has entered invalid data at the technology DB defined at input parameter *Slave*. Delete the technology DB in "Technology Objects Management" and then create a new one.

6.3.7 FB 441 MC_CamInSuperImposed - Start superimposed camming

Starting superimposed camming with FB 441 "MC_CamInSuperImposed"

Valid for Integrated Technology with firmware V3.2.x or higher

This section describes applications with firmware V3.1.x

Purpose

- The "MC_CamInSuperImposed" technology function starts a superimposing camming between the leading and following axes (master and slave.)
- Define the dynamic response during synchronization of the following axis at the dynamic parameters *Velocity, Jerk, Acceleration*.
- You can scale the specified cam, and/or shift its position.
- The synchronism can be absolute or relative.
- The specified cam disk can be executed once or cyclically.

Supported for

- Synchronization axes with a superimposed synchronization object

Prerequisites

- The leading axis is configured for operation as external encoder, or as positioning axis, or as synchronization axis
- The following axis is configured as synchronization axis with superimposing synchronization object
- The leading axis is set as possible superimposing synchronization object of the following axis
- The cam disk is marked available at the superimposing synchronization object.
- The following axis is enabled for position-controlled operation
- No active MC_Stop command at the following axis.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data Type	Initial value	Description	
Master	INT	*0*	Number of the leading axis technology DB	
Slave	INT	*0*	Number of the following axis technology DB	
CamTable	INT	*0*	Number of the cam technology disk DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
MasterOffset	REAL	*0.0*	Shift of the cam along the leading axis coordinates	
SlaveOffset	REAL	*0.0*	Shift of the cam along the following axis coordinates	
MasterScaling	REAL	*1.0*	Cam scaling factor in leading axis coordinates (Value *0* not allowed)	
SlaveScaling	REAL	*1.0*	Cam scaling factor in following axis coordinates (Value *0* not allowed)	
MasterAbsolute	BOOL	*TRUE*	Interpret cam disk coordinates absolute or relative to the leading axis:	
			TRUE:	Absolute coordinates
			FALSE:	Relative coordinates
SlaveAbsolute	BOOL	*TRUE*	Interpret cam disk absolute or relative to the following axis	
			TRUE:	Absolute coordinates
			FALSE:	Relative coordinates
CyclicMode	BOOL	*TRUE*	Processing the cam disk	
			TRUE:	Cyclic execution
			FALSE:	No cyclic execution
Velocity	REAL	*-1.0*	Maximum velocity when coupling the slave (only if *Mode = 1*)	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Maximum acceleration when coupling the slave (only if *Mode = 1*)	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk when coupling the slave (only if *Mode = 1*)	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default

Parameters	Data Type	Initial value	Description
Mode	INT	*0*	Synchronization mode / coupling mode: Value = *0*: Use the default values of the corresponding synchronization object Value = *1*: Time-related synchronization: Immediate synchronization according to the *Velocity, Jerk, Acceleration* input parameters (corresponds with "Compatibility mode" for the synchronization direction set in S7T Config) Value = *2*: Time-based synchronization: Immediate synchronization according to the input parameters *Velocity, Jerk, Acceleration, Deceleration.* .The program also makes allowances for settings in the "Gearing" tab of the **Axis_SYNCHRONISM > Default** dialog box of the synchronization object. It also evaluates system variable *userdefault.cammingsetting.synchronizingdirection* .
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB The value of DoneFlag is inverted when the command (*InSync*) is completed.

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
InSync	BOOL	*FALSE*	*TRUE*: Superimposed camming reached
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 596) of the *Error* output parameter.

Input parameter *Mode = 1*

Synchronization direction at modulo axes: "Compatibility mode" is active if *Mode = 1*, regardless of the settings in S7T Config:

- Camming:
  The following axis is synchronized within the shortest distance.

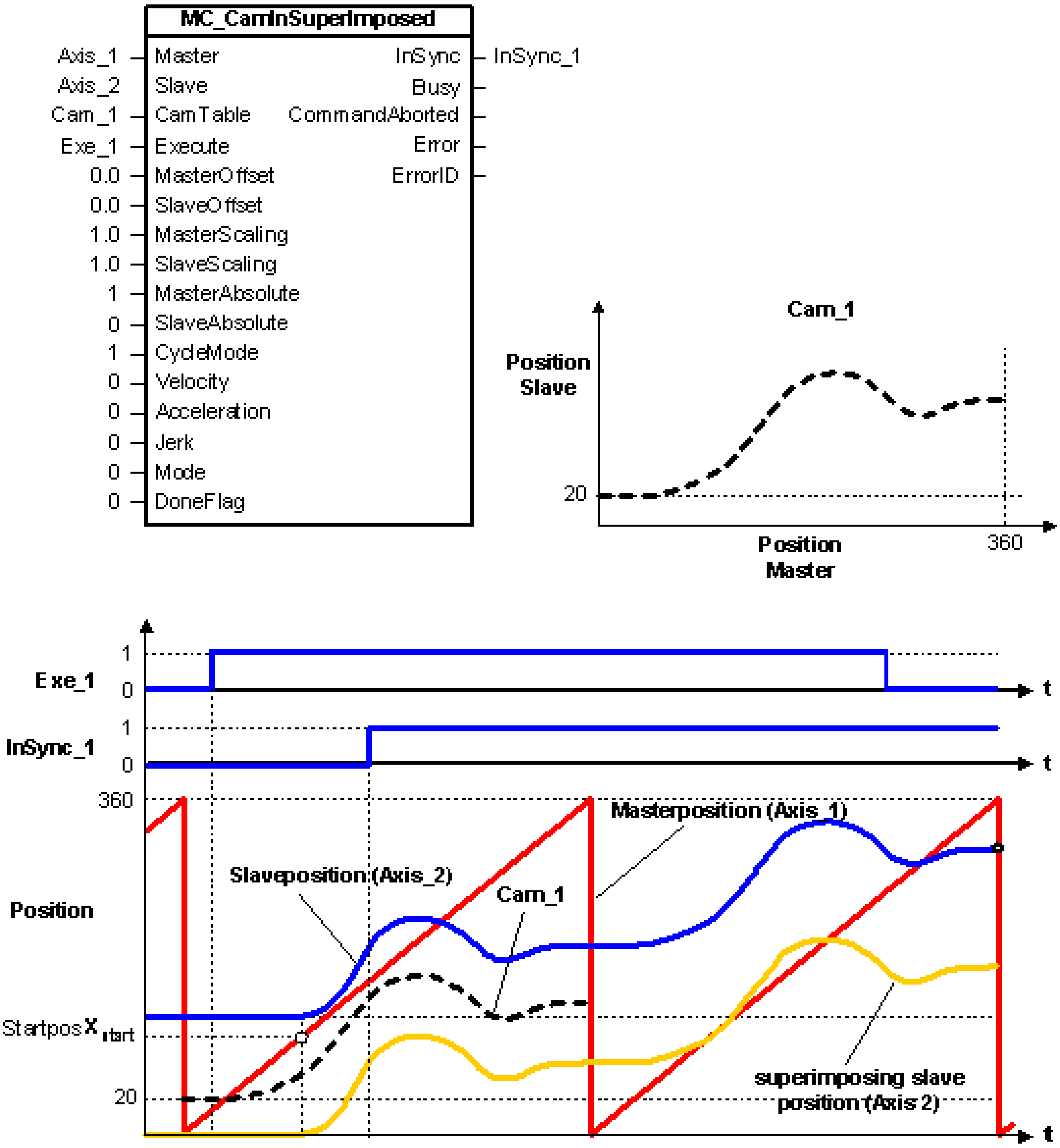
MC_CamInSuperImposed - Example: "superimposition"

The example below shows the operating principle of a superimposed cam, based on its signal chart. In order to obtain a better overview, basic synchronism was not carried out in the displayed signal profile.

At the start of the signal chart shown, the leading axis (*Axis_1*) and the following axis (*Axis_2*) each have a starting position greater than (>) *0*. The superimposed following axis position starts at *0* by default.

Superimposed camming starts with *Exe_1*. On the basis of *Mode = 0*, synchronization starts in accordance with the default settings in S7T Config (synchronization starting at leading axis position *100* with dynamic response specifications). Synchronization starts when the leading axis (*Axis_1*) reaches position *100*. After a short delay, *InSync_1* signals that camming has been achieved.

The orange signal chart profile indicates the superimposed following axis position. The following axis position follows according to its start position and with constant offset to the superimposed following axis position.


MC_CamInSuperImposed
Axis_1
Axis_2
Cam_1
Exe_1
0.0
0.0
1.0
1.0
1
0
1
0
0
0
0
0
Master
Slave
CamTable
Execute
MasterOffset
SlaveOffset
MasterScaling
SlaveScaling
MasterAbsolute
SlaveAbsolute
CycleMode
Velocity
Acceleration
Jerk
Mode
DoneFlag
InSync
Busy
CommandAborted
Error
ErrorID
InSync_1
Cam_1
Position
Slave
20
Position
Master
360
Exe_1
1
0
t
InSync_1
1
0
t
360
Position
Startpos X start
20
Masterposition (Axis_1)
Slaveposition (Axis_2)
Cam_1
superimposing slave
position (Axis 2)
t

MC_CamInSuperImposed - Example: "Basic synchronous operation with superimposed camming"

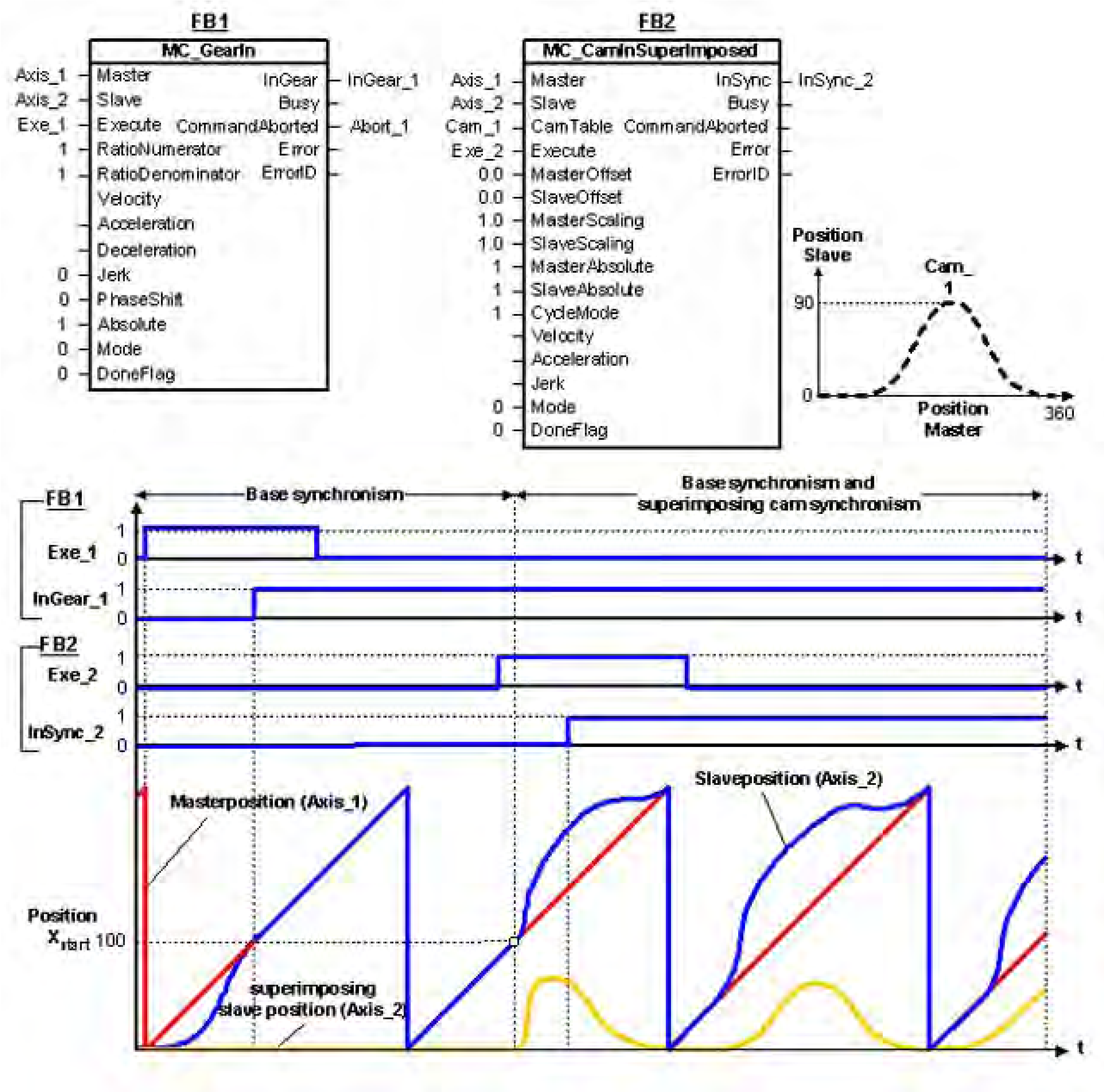
The signal chart in the example below demonstrates the effect of the absolute basic synchronous operation and superimposed camming.

Phase one - basic synchronous operation

The absolute basic synchronous operation starts with *Exe_1*. On the basis of *Mode = 0* at "MC_GearIn", synchronization starts in accordance with the default settings in S7T Config (synchronization starting at leading axis position *100* with dynamic response specifications). Synchronization starts when the leading axis (*Axis_1*) reaches position *100*. After a short delay, *InGear_1* signals that the absolute basic synchronous operation has been achieved.

Phase two - Basic synchronous operation and superimposed camming

Superimposed camming starts with *Exe_2*. The superimposed slave position starts at *0* by default. After a short delay, *InSync_2* signals that superimposed camming has been achieved. The slave position results from the addition of the "basic slave position" (identical to the master position (*Axis_1*)) and the "superimposed slave position."


FB1
MC_GearIn
Axis_1 Master
Axis_2 Slave
Exe_1 Execute
1 RatioNumerator
1 RatioDenominator
Velocity
Acceleration
Deceleration
0 Jerk
0 PhaseShift
1 Absolute
0 Mode
0 DoneFlag
InGear InGear_1
Busy
CommandAborted Abort_1
Error
ErrorID
FB2
MC_CamInSuperImposed
Axis_1 Master
Axis_2 Slave
Cam_1 CamTable
Exe_2 Execute
0.0 MasterOffset
0.0 SlaveOffset
1.0 MasterScaling
1.0 SlaveScaling
1 MasterAbsolute
1 SlaveAbsolute
1 CycleMode
Velocity
Acceleration
Jerk
0 Mode
0 DoneFlag
InSync InSync_2
Busy
CommandAborted
Error
ErrorID
Position Slave
Cam_1
90
0
Position Master
360
Base synchronism
Base synchronism and superimposing cam synchronism
FB1
Exe_1
InGear_1
FB2
Exe_2
InSync_2
t
Masterposition (Axis_1)
Slaveposition (Axis_2)
Position
X_{start} 100
superimposing slave position (Axis_2)

MC_CamInSuperImposed - ErrorIDs

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware V3.1.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8022	No actual value at the axis / external encoder	Encoder or data bus not ready, for example
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *DoneFlag* or *Mode* input parameters. The source of the error may also be present in the configuration, as this error also appears when the block contains an incorrect command parameter that was accepted as a default setting from configuration. Example: synchronization length = *0*. The configuration error could also be generated if, for example, the specified cam is not selected in the configuration of the synchronous object.
8044	Command not supported by the technology object	Sending a command to a measuring input or to an output cam, for example.
8045	Command not allowed in current state	Examples: • MC_Stop command is active • The specified cam is currently performing a restart. Wait until "MC_Reset" is finished (*Restart = TRUE*). Then, call "MC_CamInSuperImposed" again. • The motion of the slave axis is speed-controlled by the technology function "MC_MoveVelocity"" if input parameter *PositionControl = TRUE*. The slave axis must be in position-controlled mode for a superimposed synchronous operation.

ErrorID	Error message	Description / to correct or avoid errors
804A	Required object connection is missing	Neither the leading axis defined at input parameter *Master* nor the cam defined at input parameter *CamTable* is selected in the configuration of the superimposed synchronous object.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
807A	Invalid master setpoint	Invalid actual value of the leading axis.
807B	Recursive interconnection of technology objects	The leading axis defined at the input parameter is already active as following axis in synchronous operation, and the master setpoint is derived directly or indirectly from the following axis defined at the input parameter.

ErrorID	Error message	Description / to correct or avoid errors
8083	DB is not a technology DB	The DB specified at input parameter *Master*, *Slave* or *CamTable* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Master*, *Slave* or *CamTable* . Download the current technology to the target system or change the DB number at input parameter *Master*, *Slave* or *CamTable*. • The user has entered invalid data at the technology DB defined at input parameter *Master*, *Slave* or *CamTable* . Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.3.8 FB 443 MC_CamOutSuperImposed - End superimposed camming

Stopping superimposed camming with FB 443 "MC_CamOutSuperImposed"

Valid for Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_CamOutSuperImposed" technology function stops superimposed camming. This does not affect basic motions or basic synchronous operation.
- The command is completed when the slave axis is desynchronized according to the axis configuration or to the dynamic response parameters of the technology function.

Supported for

- Synchronization axes with a superimposed synchronous object

Prerequisites

- The axis must be operated as a following axis (slave) for a superimposed camming process. The command is ignored if no synchronous operation is active, and the technology function reports *Done = TRUE*..
- An MC_Stop command is not being executed.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameters	Data Type	Initial value	Description	
Slave	INT	*0*	Number of the following axis technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Deceleration	REAL	*-1.0*	Deceleration at the end of camming (only when *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Jerk at the end of camming (only when *Mode = 1*):	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
Mode	INT	*0*	Decoupling mode	
			Value = *0*:	Default setting
			Value = *1*:	according to dynamic response specifications
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice-DB. The DoneFlag value is inverted when final velocity is achieved.	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Superimposed camming is terminated	
Busy	BOOL	*FALSE*	*TRUE:* The command is being executed	
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 605) of the *Error* output parameter.	

Note

Simultaneous output of MC_CamInSuperImposed and MC_CamOutSuperImposed to the same technology object may cancel the MC_CamInSuperImposed command (*CommandAborted = TRUE*). The "MC_CamOutSuperImposed" technology function therefore reports *Done = TRUE.*

Do not start the MC_CamOutSuperImposed command unless the MC_CamInSuperImposed has reported the synchronous state (*InGear = TRUE)*.

MC_CamOutSuperImposed - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *DoneFlag* input parameters. This could also be a faulty configuration in S7T Config.
8044	Command not supported by the technology object	Sending a command to a measuring input or to an output cam, for example
8045	Command not allowed in current state	The MC_Stop command is active, for example
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Slave* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Slave*. Download the current technology to the target system, or change the DB number at input parameter *Slave*. • The user has entered invalid data at the technology DB defined at input parameter *Slave*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.3.9 FB 424 MC_Phasing - Changing the phase shift between the leading axis and the following axis

Changing the phase shift between the leading and following axes by calling FB424 "MC_Phasing"

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

Purpose

- The "MC_Phasing" technology function causes a phase shift of the leading axis relative to the following axis. The actual position of the leading axis is not influenced.
- The effect on a following axis operating in camming mode is the same as a horizontal cam shift.
- The phase reference can be either relative or absolute.
- The function can be started when the leading axis is at a standstill or in motion.
- Define the dynamic response of the motion with the input parameters *Velocity, Jerk, Acceleration, Deceleration*.

Supported for

- Synchronous axes operating in camming or gearing mode

Prerequisites

- No active MC_Stop command at the following axis.
- The axis must be operated as a following axis in camming or gearing modes.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

Input parameters

Parameter	Data type	Initial value	Description
Master	INT	*0*	Number of the leading axis technology DB
Slave	INT	*0*	Number of the following axis technology DB
Execute	BOOL	*FALSE*	Start of the command at a positive edge
PhaseShift	REAL	*0.0*	Phase shift
Velocity	REAL	*-1.0*	Maximum velocity for setting the phase position:
			Value > *0*: Use the defined value
			Value = *0*: Not permitted
			Value < *0*: Use default
Acceleration	REAL	*-1.0*	Acceleration for setting the phase position:
			Value > *0*: Use the defined value
			Value = *0*: Not permitted
			Value < *0*: Use default
Deceleration	REAL	*-1.0*	Deceleration for setting the phase position:
			Value > *0*: Use the defined value
			Value = *0*: Not permitted
			Value < *0*: Use default
Jerk	REAL	*-1.0*	Jerk for setting the phase position:
			Value > *0*: Use the defined value
			Value = *0*: Use trapezoidal motion profile
			Value < *0*: Use default
Absolute	BOOL	*TRUE*	Phase shift:
			Value = *0*: Relative to the current phase:
			Value = *1*: absolute to the current phase (see MC_Phasing - example)
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB

Output parameters (status outputs)

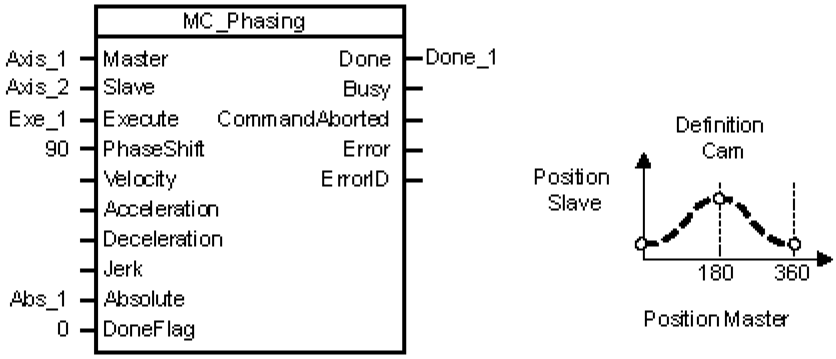
Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Required phase position reached
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE: Command initiation **without error**.
ErrorID	WORD	0	ErrorID (Page 609) of the *Error* output parameter.

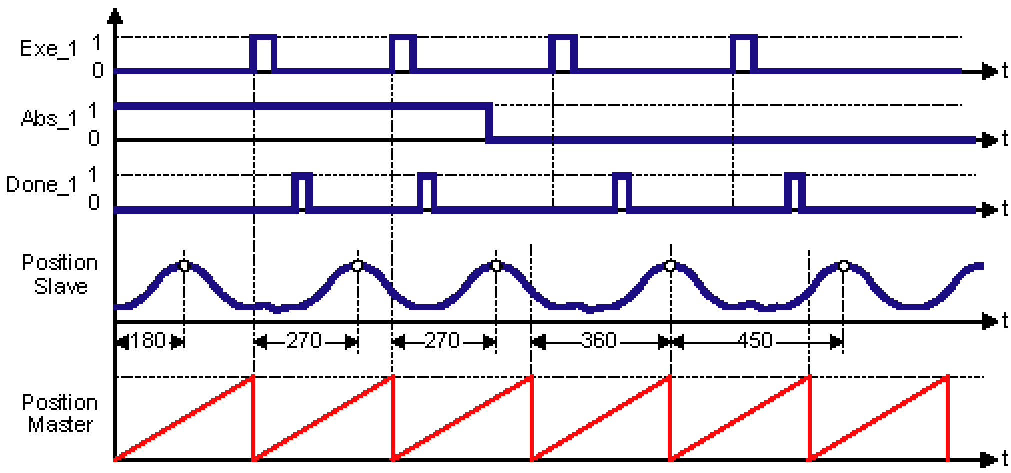
MC_Phasing - Example

Use the "MC_Phasing" technology function to shift the phase position of a cam during camming.

The *Absolute = 1* setting defines the cam shift as an absolute value compared to the cam definition. Repeated shifting with the same "PhaseShift" " value does not have any effect.

When *Absolute = 0* , the shift change is relative, meaning that each command shifts the phase position by the value defined at *PhaseShift.*





MC_Phasing - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done, CommandAborted* or *Error* is *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8040	The axis / external encoder are disabled, or the wrong mode is set	The signal required for a command to enable the following axis is missing.
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *DoneFlag* input parameter.
8044	Command not supported by the technology object	Sending a command to an output cam, for example
8045	Command not allowed in current state	No synchronous operation active
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx in both OB 1 and OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function. However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8079	No synchronous operation with specified master active	The specified master and slave are not in synchronous operation. Synchronize the relevant axes before you call the function.
8083	DB is not a technology DB	The DB specified at input parameter *Master* or *Slave* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Master* or *Slave*. Download the current technology to the target system or change the DB number at input parameter *Master* or *Slave*. • The user has entered invalid data for the technology DB defined at input parameter *Master* or *Slave*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.3.10 FB 444 MC_PhasingSuperImposed - Changing superimposed phase shift

Changing superimposed phase shift with FB 444 "MC_PhasingSuperImposed"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The MC_PhasingSuperImposed technology function sets a superimposed phase shift between the leading axis and following axis positions. The phase shift value is defined based on the following axis position. The superimposed phase shift affects the following axis. It does not influence the position of the leading axis.
- The effect on a following axis operating in camming mode is the same as a horizontal cam shift.
- The phase reference can be either relative or absolute.
- The function can be started when the leading axis is at a standstill or in motion.
- Define the dynamic response of the motion at the input parameters *Velocity*, *Jerk*,*Acceleration*, *Deceleration*.

Supported for

- Synchronous axes with a superimposed synchronous object operating in camming and gearing mode

Prerequisites

- No active MC_Stop- command at the following axis
- The axis must be operated as a following axis (slave) in superimposed camming or gearing modes.

Interaction of commands

New command - active single command (1) (Page 739)

New command – active commands (1) (Page 744)

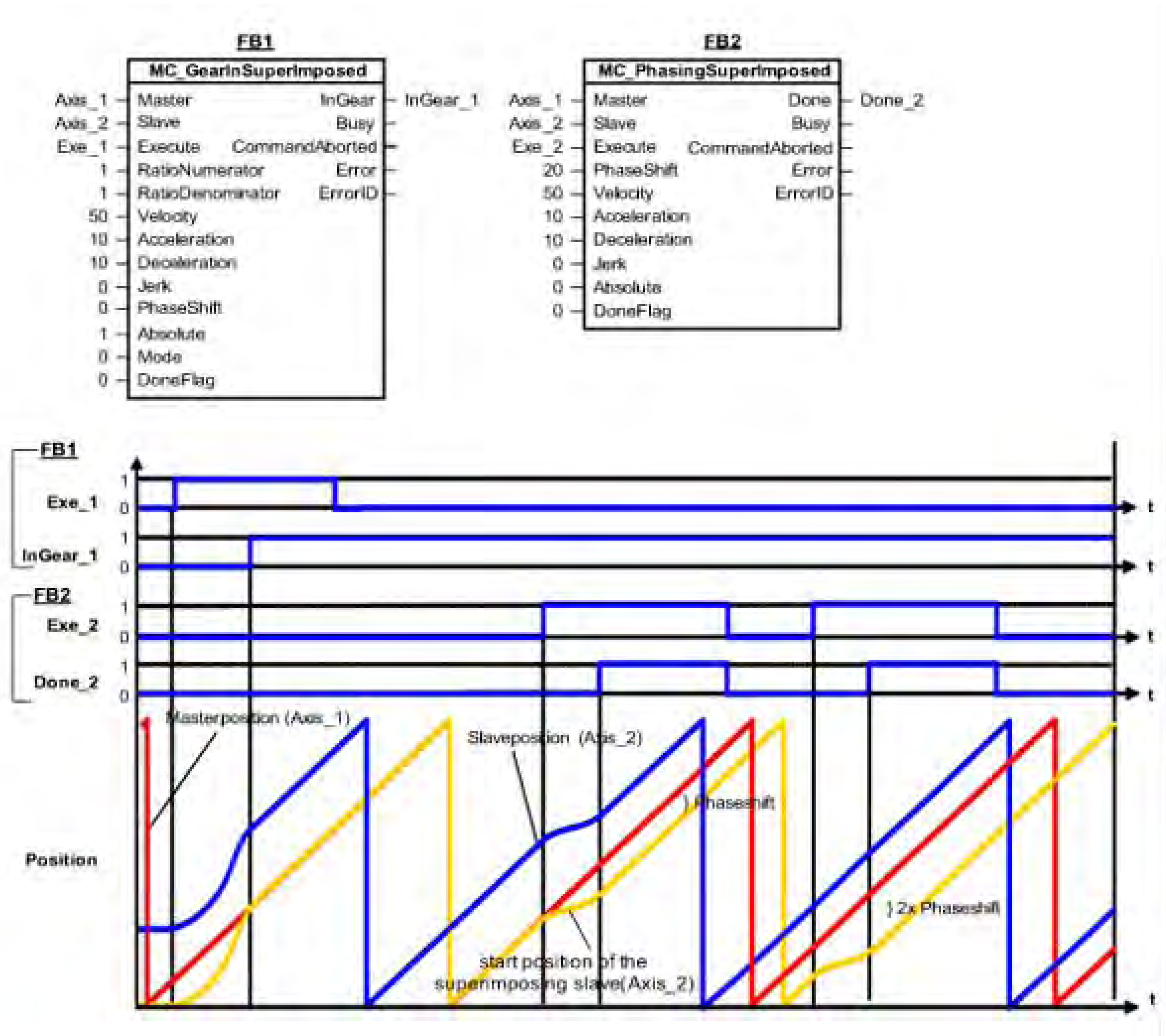
Input parameters

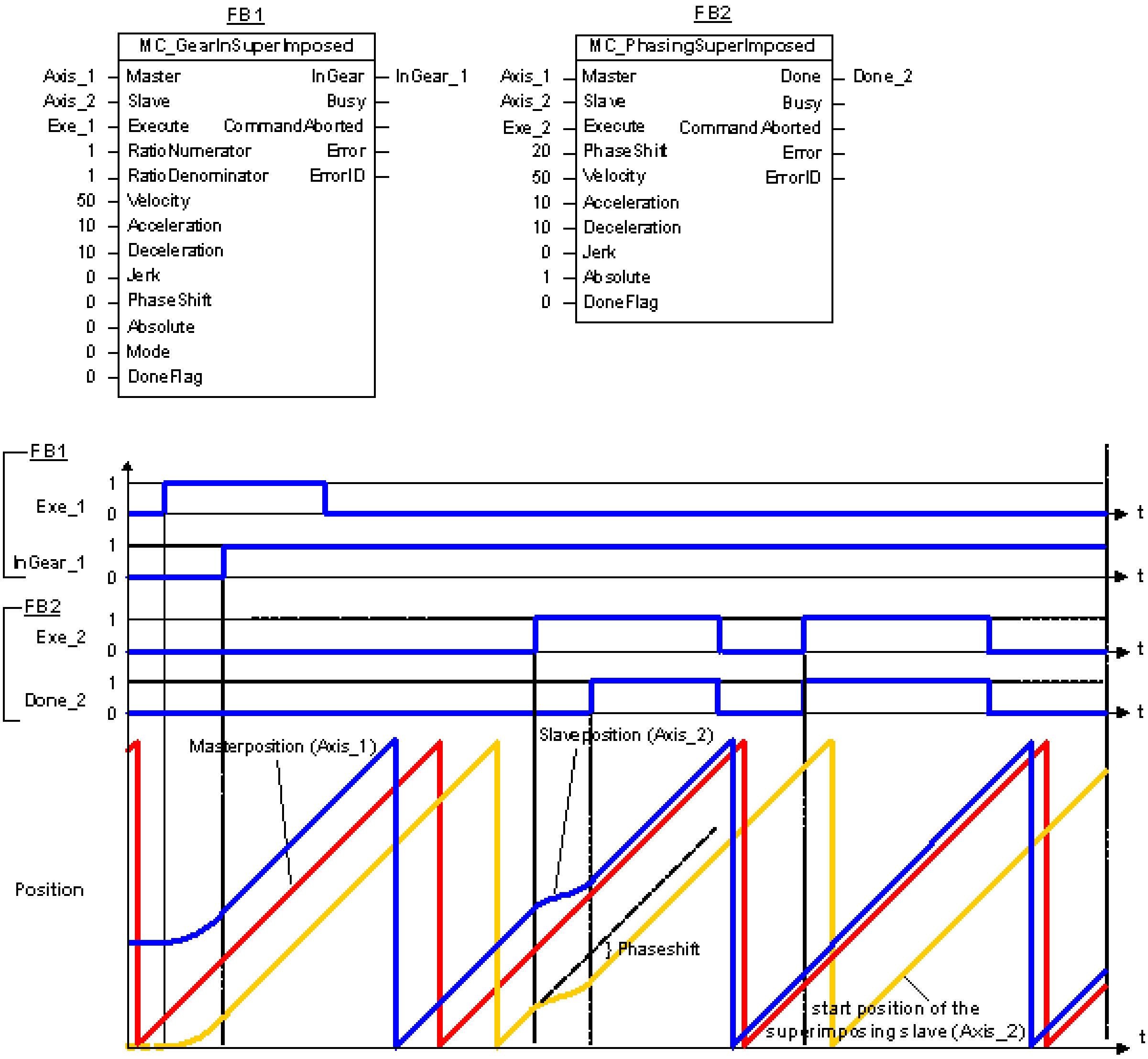
Parameters	Data Type	Initial value	Description	
Master	INT	*0*	Number of the leading axis technology DB	
Slave	INT	*0*	Number of the following axis technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
PhaseShift	REAL	*0.0*	Phase shift	
Velocity	REAL	*-1.0*	Maximum velocity for setting the phase position	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Acceleration	REAL	*-1.0*	Max. acceleration for setting the phase position	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Deceleration	REAL	*-1.0*	Max. deceleration for setting the phase position	
			Value > *0*:	Use the defined value
			Value = *0*:	Not permitted
			Value < *0*:	Use default
Jerk	REAL	*-1.0*	Max. jerk for setting the phase position	
			Value > *0*:	Use the defined value
			Value = *0*:	Use trapezoidal motion profile
			Value < *0*:	Use default
Absolute	BOOL	*TRUE*	Phase shift	
			Value = *0*:	Relative
			Value = *1*:	absolute to the current phase
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

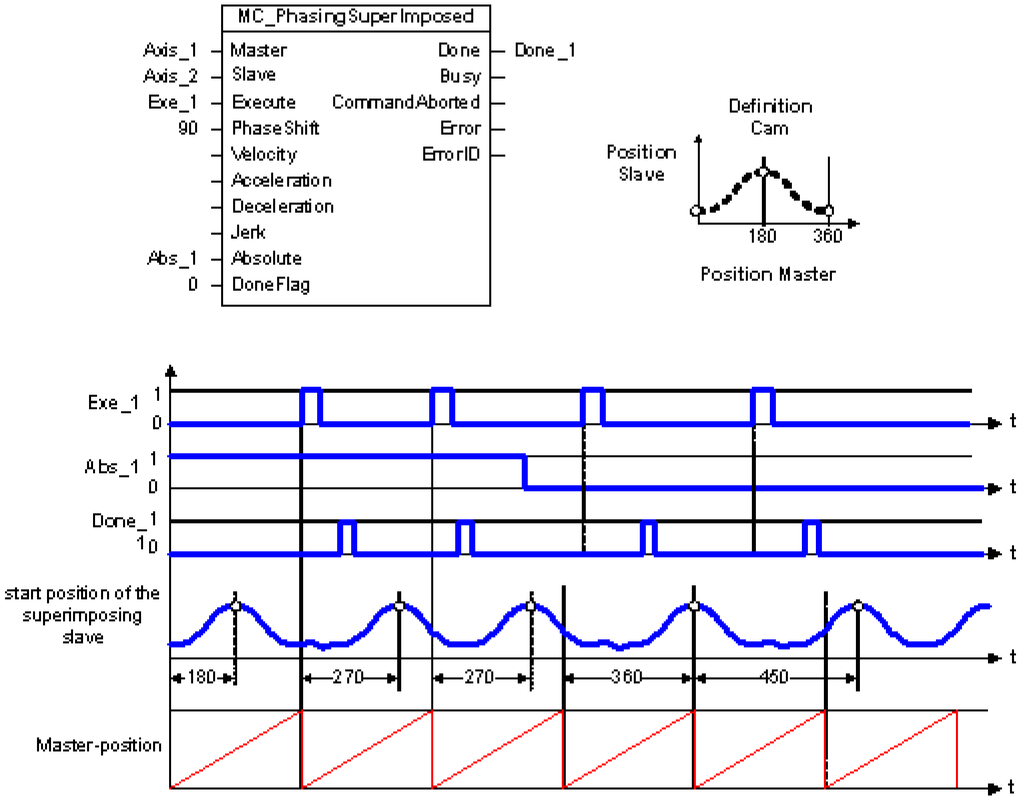
Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Required phase position reached
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	0	ErrorID (Page 614) of the *Error* output parameter.

MC_PhasingSuperImposed - example




FB1
MC_GearInSuperImposed
Axis_1 Master
Axis_2 Slave
Exe_1 Execute
1 RatioNumerator
1 RatioDenominator
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
0 PhaseShift
0 Absolute
0 Mode
0 DoneFlag
InGear InGear_1
Busy
CommandAborted
Error
ErrorID
FB2
MC_PhasingSuperImposed
Axis_1 Master
Axis_2 Slave
Exe_2 Execute
20 PhaseShift
50 Velocity
10 Acceleration
10 Deceleration
0 Jerk
1 Absolute
0 DoneFlag
Done Done_2
Busy
Command Aborted
Error
ErrorID
FB1
Exe_1
InGear_1
FB2
Exe_2
Done_2
Position
Masterposition (Axis_1)
Slaveposition (Axis_2)
} Phaseshift
start position of the superimposing slave (Axis_2)
t



MC_PhasingSuperImposed - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.

ErrorID	Error message	Description / to correct or avoid errors
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable required for a motion command is missing. Eliminate and acknowledge all queued errors and then enable the relevant mode axis (for example position-controlled mode).
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *DoneFlag* input parameter.
8044	Command not supported by the technology object	Sending a command to a measuring sensor or cam, for example
8045	Command not allowed in current state	The MC_Stop command is active, for example
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8056	Cancellation due to active travel to fixed stop	The axis has moved to the fixed stop (*InClamping = TRUE*). New commands are only accepted if these release the axis from the fixed stop. The command initiated does not fulfill this condition.
8079	No synchronous operation with specified master active	The specified master and slave are not in synchronous operation. Synchronize the relevant axes before you call the function.

ErrorID	Error message	Description / to correct or avoid errors
8083	DB is not a technology DB	The DB specified at input parameter *Master bzw. Slave* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Master* or *Slave*. Download the current technology to the target system or change the DB number at input parameter *Master* or *Slave*. • The user has entered invalid data for the technology DB defined at input parameter *Master* or *Slave*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.4 Technology functions - Cam disks

6.4.1 FB 434 MC_CamClear - Deleting cams

Deleting cams with FB 434 "MC_CamClear"

Purpose

- The "MC_CamClear" technology function deletes all defined interpolation points or segments of a cam, and sets the cam to edit mode.

Supported for

- Cams

Prerequisites

- The cam must not be in use when you delete it.
- No MC_CamSectorAdd commands must be active (restriction only applies to firmware V3.0.x of integrated technology).

Overriding commands

MC_CamClear commands can not be canceled by any other command.

MC_CamClear commands do not cancel any other commands.

Input parameters

Parameter	Data type	Initial value	Description
CamTable	INT	*0*	Number of the technology DB
Execute	BOOL	*FALSE*	Starts delete operation at a positive edge

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Cam sectors have been deleted	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 621) of the *Error* output parameter.	

Note

It may take some time to delete cams with a large number of interpolation points or polynomials. Other technology objects can not accept any new commands, not even MC_Stop, within this period. Note that you can not cancel commands without a defined termination ("MC_MoveVelocity", for example) within this time.

If the setting of the command monitoring time is too short in S7T Config , the deletion may lead to a timeout and, thus, to a STOP of the Technology CPU.

MC_CamClear - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8044	Command not supported by the technology object	Command request to an axis, for example
8045	Command not allowed in current state	• The cam is in use • An "MC_CamSectorAdd" command is busy inserting interpolation points into the cam. (This reason only applies to integrated technology with firmware V3.0.x.)
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8051	Read-/write-access denied.	Read-/write-access to the relevant technology object failed. This error may occur, for example, if "MC_CamClear" is started while a cam is actively being tracked in synchronous operation.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.

6.4.2 FB 435 MC_CamSectorAdd - Add cam sectors

Adding a cam sector with FB 435 "MC_CamSectorAdd"

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware up to V3.2.x

Purpose

- The "MC_CamSectorAdd" technology function adds new interpolation points or polynomials to a cam (unlimited number per command).

Supported for

- Cams

Prerequisites

- The cam is in edit mode

  Edit mode is started by calling the "MC_CamClear" technology function and ends with cam interpolation by the calling the "MC_CamInterpolate" technology function. The cam can then be reused for camming.

Overriding commands

MC_CamSectorAdd commands can not be canceled by any other command.

MC_CamSectorAdd commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description	
CamTable	INT	*0*	Number of the cam disk technology DB.	
Execute	BOOL	*FALSE*	Start at the positive edge	
Data	ANY	-	Data area that contains the cam segments to be inserted (interpolation points or polynomials).	
Table	INT	*0*	Specification of the data area:	
			Value = *0*:	Polynomial (mathematical function)
			Value = *1:*	Interpolation point table

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Change of cam sector complete	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 625) of the *Error* output parameter.	

Input parameter *Data*: Definition of polynomials (mathematical function)

> **Note**
>
> Note the following:
> - Only use mathematical functions up to the 3rd order when you add segments. Any coefficients of a higher order may lead to inaccuracies due to the resolution of floating-point values.
> - Segments defined as polynomials are recalculated in the Technology CPU. When you subsequently read out the values to S7T Config, deviating coefficients may be returned. The y-coordinates of the start and end points may similarly also be swapped.
>   The cam shape, however, is retained.

Mathematical functions are derived from the following formula:

$$Y = A_0 + A_1 \cdot X + A_2 \cdot X^2 + A_3 \cdot X^3 + A_4 \cdot X^4 + A_5 \cdot X^5 + A_6 \cdot X^6 + B_0 \cdot (\sin(B_1 \cdot X) + B_2)$$

Relative byte address	Data type	Variable	Meaning
0	INT	Type	Cam type (0 = polynomial)
2	INT	Spare	-
4	REAL	X - Min	x-coordinate (minimum value)
8	REAL	Y - Min	y-coordinate (minimum value)
12	REAL	X - Max	x-coordinate (maximum value)
16	REAL	Y - Max	y-coordinate (maximum value)
20	REAL	A0	Order 0 coefficient
24	REAL	A1	Order 1 coefficient
28	REAL	A2	Order 2 coefficient
32	REAL	A3	Order 3 coefficient
36	REAL	A4	Order 4 coefficient
40	REAL	A5	Order 5 coefficient
44	REAL	A6	Order 6 coefficient
48	REAL	B0	Coefficient 0 of the trigonometric element
52	REAL	B1	Coefficient 1 of the trigonometric element
56	REAL	B2	Coefficient 2 of the trigonometric element

Example:

```
CALL  "MC_CamSectorAdd" , DB435
 CamTable:=
 Execute :=
 Data    :=P#DB20.DBX 16.0 BYTE 60
 Table   :=0
 Done    :=
 Error   :=
 ErrorID :=
```

In the example, the ANY pointer refers to a range starting at address 16 in DB20. The data structure of the polynomial is fixed and has a set length of 60 bytes.

Note

You can save yourself editing work by using UDT 20 from the "S7-Tech" library. UDT 20 contains the data structure of the polynomial. You can add up to 4 polynomials in one call by repeating this structure.

Input parameter *Data*: Interpolation point table

When cams are created using the interpolation point tables, the x and y values are fetched from a data area in the user program. The interpolation point table contains at least one interpolation point. Each interpolation point consist of two REAL values. You can transfer up to 30 interpolation points per call.

Example:

```
CALL  "MC_CamSectorAdd" , DB435
 CamTable:=
 Execute :=
 Data    :=P#DB10.DBX 6.0 BYTE 64
 Table   :=1
 Done    :=
 Busy    :=
 Error   :=
 ErrorID :=
```

In our example, the ANY pointer refers to a range of interpolation points that begins at address 6 in DB10 and consists of eight interpolation points (eight value pairs).

Relative byte address	Data type	Variable	Meaning
0	REAL	X1	x-coordinate of interpolation point 1
4	REAL	Y1	y-coordinate of interpolation point 1
8	REAL	X2	x-coordinate of interpolation point 2
12	REAL	Y2	y-coordinate of interpolation point 2
...	...	...	...
	REAL	Xn	x-coordinate of the last interpolation point
	REAL	Yn	y-coordinate of the last interpolation point

Disadvantage: The creation of cams based on interpolation point tables takes longer, because the data volume required for the cam description is higher compared to polynomial definitions.

Note

Integrated technology with firmware V3.1.x or higher

This firmware version allows you to add all interpolation points and polynomials of the MC_CamSectorAdd command in a single command execution cycle. You can add more than 30 interpolation points or more than 4 polynomials to the cam by running several MC_CamSectorAdd commands.

Integrated Technology with firmware V3.0.x

With this firmware version, the insertion of interpolation points and polynomials is distributed to several command execution cycles.

While one MC_CamSectorAdd command is running, no other MC_CamSectorAdd, MC_CamClear, or MC_CamInterpolate commands are permitted to start.

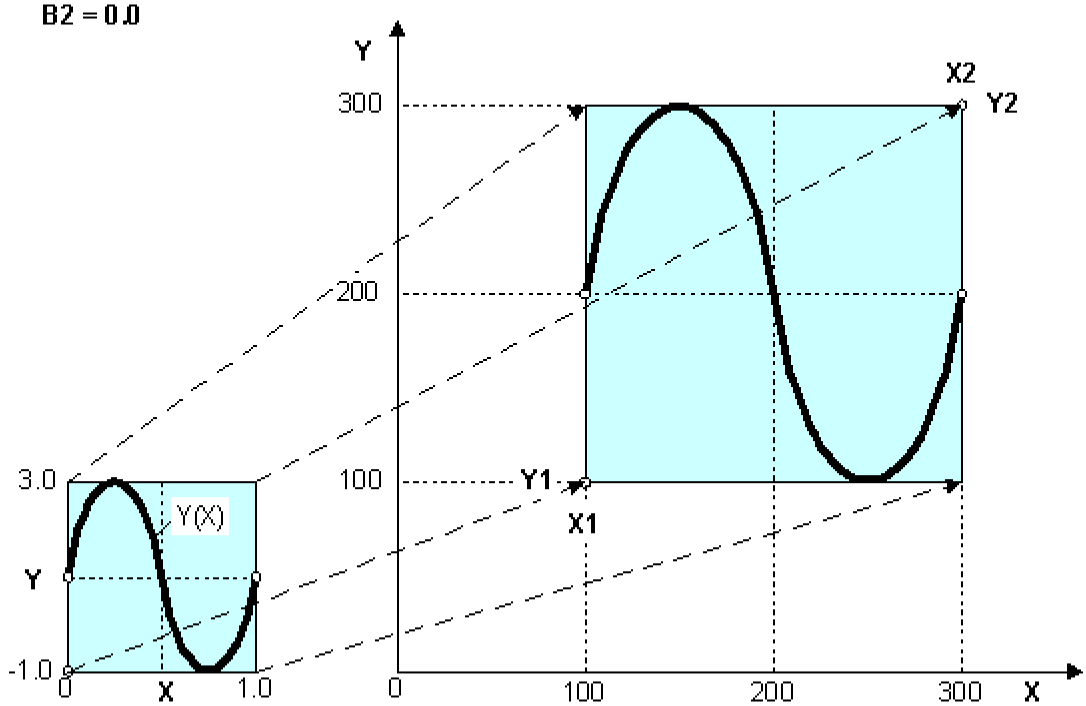
Effect of the coefficients of a cam segment

The coefficients *A0* to *A6* and *B0* to *B2* define the normalized curve characteristic of a cam segment. The range of the cam within interval 0.0 ≤ X ≤ 1.0 is transformed by interpolation into the range defined by *{X1,Y1}* and *{X2,Y2}*.

Y(X) = 1.0 + 2.0·sin(2π·X)

A0 = 1.0
A1 = A2 = A3 = A4 = A5 = A6 = 0.0
B0 = 2.0
B1 = 6.2831
B2 = 0.0

X1 = 100 Y1 = 100
X2 = 300 Y2 = 300



Defining cam disks

In the **Insert Cam** dialog box, you can choose to define the cam based on interpolation points or on polynomials.

- **Definition based on interpolation points**

  Interpolation points are represented in an interpolation point table in the form **P = P(x,y)**. The order in which the value pairs are entered is irrelevant. They are automatically sorted in ascending order in the domain.
  S7T Config interpolates the cams according to the configured interpolation type.

- **Definition based on polynomials/segments**

  The various polynomials are described in accordance with "VDI Directive 2143, Motion Laws for Cam Mechanisms." The maximum degree of the polynomial is 6. A polynomial can also contain a trigonometric function.

Methods for defining cams

Both definition methods, i.e., based on interpolation points or on polynomials, have their advantages and disadvantages. Your application determines which one of these methods will suit the requirements.
A combination of both methods in one cam is only possible in runtime using the MC_CamSectorAdd technology function.

	Definition based on interpolation points	Definition based on polynomials
Advantage	• Simple definition • Any algorithms can be mapped by interpolation points • Plotting by Teach-In • Simple interface to HMI	• Low data volume for the definition • Standard transitions in accordance with VDI... • Extremely precise contour, continuous transitions
Disadvantages	• Large number of interpolation points required for smooth contour	• Requires complex calculation of coefficients

MC_CamSectorAdd - ErrorIDs

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware up to V3.2.x

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process.	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.

ErrorID	error message	Description / to correct or avoid errors
8006	Insufficient memory	Integrated Technology is out of memory. Reduce the number of interpolation points or polynomials to be inserted.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the number of interpolation points / polynomials.
8043	Illegal parameter value	*Table = 167* or illegal polynomial value (*Xmin* and *Xmax* are equal), for example.
8044	Command not supported by the technology object	Sending a command to an output cam, for example.
8045	Command not allowed in current state	An "MC_CamSectorAdd" is active and is busy inserting interpolation points into the cam. Wait for the Done bit of the other "MC_CamSectorAdd," and then retrigger this command. (This error can only occur in integrated technology with firmware V3.0.x.)
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8051	Read-/write-access denied.	Read-/write-access to the relevant technology object failed. An error may occur if the cam is actively tracked in synchronous operation, for example.

ErrorID	error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8062	Access to interpolated cam not permitted	The cam is already interpolated. For this reason, you can not insert any new interpolation points or polynomials at this cam. Remedy: Delete the content of the cam with "MC_CamClear" ," and then recreate it.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data record length (ANY pointer)	For mathematical functions 60 bytes must be transferred, and for interpolation point tables a multiple of 8 bytes must be transferred. An interpolation point consists of x and y values of the data type REAL (4 bytes).

ErrorID	error message	Description / to correct or avoid errors
808A	Invalid REAL values in DB	The DB transferred to the technology contains at least one value that is incompatible with a permissible REAL format specification Check all DB values and adjust these as required.
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL

6.4.3 FB 436 MC_CamInterpolate - Interpolating cams

Interpolating cams with FB 436 "MC_CamInterpolate"

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware up and including V3.1.x

Purpose

- The "MC_CamInterpolate" technology function interpolates a cam. After its interpolation, the cam can be used for camming.
- Interpolation closes the gaps between the cam interpolation points and polynomials, according to the specified type of interpolation.

Supported for

- Cams

Prerequisites

- The cam is in edit mode

  Edit mode is started by calling the "MC_CamClear" technology function and ends with cam interpolation by the calling the "MC_CamInterpolate" technology function. The cam can then be reused for camming.

Conditions

- Rule for interpolation points:
  - The last superimposed interpolation point created takes priority.
  - The value *1E-4* is used in the system for the continuity check.
- Rule for segments:
  - Gaps between segments are filled with an interpolated cam.
  - The segment start point is used if segments overlap; the previous segment is truncated as of this position.
- Rule for mixed mode cams (segments and interpolation points):
  - The start point takes priority as with pure segments.
- The cam is calculated in accordance with settings at input parameter *CamMode*. This system may change the cam boundaries accordingly.

Overriding commands

MC_CamInterpolate commands can not be canceled by any other command.

MC_CamInterpolate commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description	
CamTable	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Starts at a positive edge	
Mode	INT	*0*	Interpolation type:	
			Value = *0*:	Linear interpolation
			Value = *1*:	Cubic splines (curve characteristic through the interpolation points or cam segments)
			Value = *2*:	Bezier splines (curve characteristic along the interpolation points or cam segments)
CamMode	INT	*0*	Cam type: You define the way interpolation handles irregularities at the cam edges at input parameter *CamMode*.	
			Value = *0*:	cyclic relative - continuous velocity: The start and end points of the cam are interpolated such that any velocity jumps are prevented at the following axis when operating the cam in cyclic mode. The cam operates at constant velocity in its edge areas.
			Value = *1*:	Cyclic absolute - continuous position and velocity: The start and end points of the cam are equated. At the same time, the cam is interpolated such that any velocity jumps are prevented when operating the cam in cyclic mode.
			Value = *2*:	Non-cyclic - not constant in the edge points The start and end points of the cam are not changed. The traversing range of the cam is not limited by the *StartPoint* and *EndPoint* input parameters. The runtime system uses the cam as specified, including all discontinuities at the edges, even if the cam is operated in cyclic mode. However, the acceleration limits and inertia of the mechanical system and drive play a decisive role.
			Set *CamMode = 0* or *1* to operate the cam in cyclic mode. Set *CamMode = 2* if the cam is not to be operated in cyclic mode.	
StartPoint	REAL	*0.0*	Start point (minimum x-value of the cam) (maximum value: 3.402 823E+38)	
EndPoint	REAL	*0.0*	End point (maximum x-value of the cam) (maximum value: 3.402 823E+38)	

With a "cyclic absolute" or "relative" setting, the cam can only be calculated in the runtime system if there is a sufficient interval between the last defined position of the cam and the end of the domain, or between the beginning of the domain and the first defined value.

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: The cam can only be used for synchronous operation. *TRUE* is also indicated if a blank cam was interpolated	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 633) of the *Error* output parameter.	

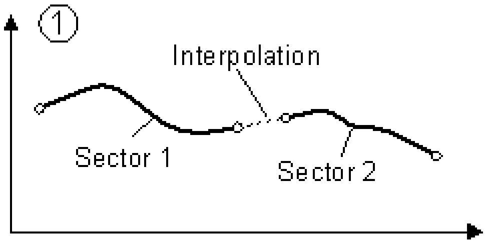
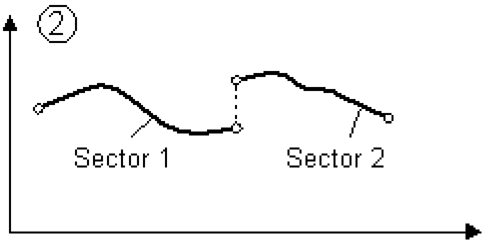
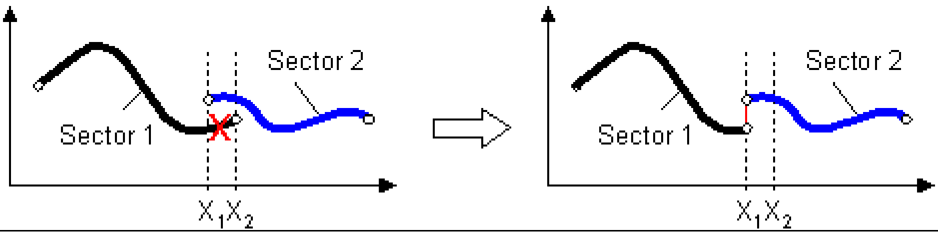
Note

It may take some time to interpolate cams with a large number of interpolation points or polynomials. Other technology objects can not accept any new commands, not even MC_Stop, within this period. Note that you can not cancel commands without a defined termination ("MC_MoveVelocity", for example) within this time.

If the command monitoring time is set too short in S7T Config, the interpolation may lead to timeout and subsequently to a STOP of the Technology CPU.

Interpolation between two cam segments

If a cam comprises several segments, various kinds of discontinuity can develop at the segment boundaries:

1.	A gap between two segments is closed automatically during interpolation. A gap smaller than a specific limit is closed by joining the interpolation points. If a cam is created in S7T Config, you can define the gap limit in expert mode (go to **CamEdit > "Interpolation"** tab, and then select the "Expert mode" check box. If the cam disk is created with the "MC_CamSectorAdd" technology function, the limit of the gap lies fixed at *1E-04*. Above this limit the gap is closed by interpolation. Below this limit the gap is closed by joining the points. 
2.	Where the end point of one segment and the start point of the next have the same x value, but different y values, this discontinuity persists irrespective of the interpolation. 
3.	The start position of the successive segment (sector 2) is applied where two successive segments overlap. The overlapping end position of the previous segment (sector 1) is truncated and therefore lost. Overlapping segments are particularly prone to discontinuity at the segment limits, because as a result of the truncation, sector 1 is assigned a random y value at the end of the segment (y(x1). 

MC_CamInterpolate - ErrorIDs

Valid for Integrated Technology with firmware V3.2.x or higher

This section describes applications with firmware up to and including V3.1.x

ErrorID	Warning	Description / to correct or avoid errors
0000	No warning	-
0027	Interpolation property can not be maintained	An interpolation property of the cam could not be maintained, even though the cam was interpolated. Example: A cam contains only one polynomial with a different gradient at its start and end point.

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8006	Insufficient memory	Integrated Technology is out of memory. Cam interpolation failed. Reduce the number of interpolation points or polynomials to be inserted, and then interpolate the cam again.
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *CamMode.* input parameters. The values at input parameters *StartPoint* and *EndPoint* are identical (the value 0.0 is allowed at both input parameters).
8044	Command not supported by the technology object	Sending a command to an output cam, for example.

ErrorID	Error message	Description / to correct or avoid errors
8045	Command not allowed in current state	• The cam is already interpolated • An "MC_CamSectorAdd" command is busy inserting interpolation points into the cam. This reason only applies to integrated technology with firmware V3.0.x.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8051	Read-/write-access denied.	Read-/write-access to the relevant technology object failed. This error may occur, for example, if MC_CamInterpolate is started while a cam is actively being tracked in synchronous operation.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8062	Cam interpolation failed	The cam is in use or is already interpolated.

ErrorID	Error message	Description / to correct or avoid errors
8066	The cam disk is empty	The cam disk does not contain any interpolation points or cam segments.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.4.4 FB 438 MC_GetCamPoint - Read points from cam

Reading points from the cam with FB 438 "MC_GetCamPoint"

Supported by Integrated Technology with firmware V4.1.x

This section describes applications with firmware V3.1.x

Purpose

- The "MC_GetCamPoint" technology function is used to determine the position of the following axis relative to a leading axis position as well as the first and second derivation of the cam point from an existing cam.
- The "MC_GetCamPoint" technology function is used to determine the position of a leading axis relative to a following axis position from an existing cam.
  - Because the same slave axis positions can be entered for various master axis positions when the cam is defined, the master axis position must be defined more precisely. This can be done at input parameter *ApproachPosition*. Enter an approximation of the presumed leading axis position at this input parameter.
- If the cam was configured in S7T Config using CamEdit or CamTool, the scaling and shift defined there are included when you calculate the positions. Cams created dynamically in runtime are always unscaled and unshifted.
  Any scaling or shift operations by the input parameters of the MC_CamIn technology function are ignored when the positions are determined.

Supported for

- Cams

Prerequisites

- The cam was interpolated without error (status at technology data block *2* = interpolated).

Overriding commands

MC_GetCamPoint commands can not be canceled by any other command.

MC_GetCamPoint commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description
CamTable	INT	*0*	Number of the cam technology disk DB
Execute	BOOL	*FALSE*	Start of the command at the positive edge
Mode	INT	*0*	Value = *0*: The position of the following axis is determined relative to the position of the leading axis Value = *1*: The position of the leading axis is determined relative to the position of the following axis Value = *2*: The first derivation of the cam point is determined relative to the position of the leading axis Value = *3*: The second derivation of the cam point is determined relative to the position of the leading axis Any scaling and offset configured in S7T Config are taken into account in both modes.
Position	REAL	*0.0*	Leading or following axis position for which the corresponding value is to be determined: • *Mode = 0, 2, 3*: Position of the leading axis • *Mode = 1*: Position of the following axis The valid range of parameter values is derived from the domain or from the cam's range of values.
ApproachPosition	REAL	*0.0*	Assumed leading axis position Approximate value within the domain of the cam, used to determine the leading axis position (effective only when *Mode = 1*). • The function determines only the leading axis position nearest to the *ApproachPosition*. • When two leading axes are positioned at the same distance to the *ApproachPosition*, the leading axis position with the lower position value is determined.
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB

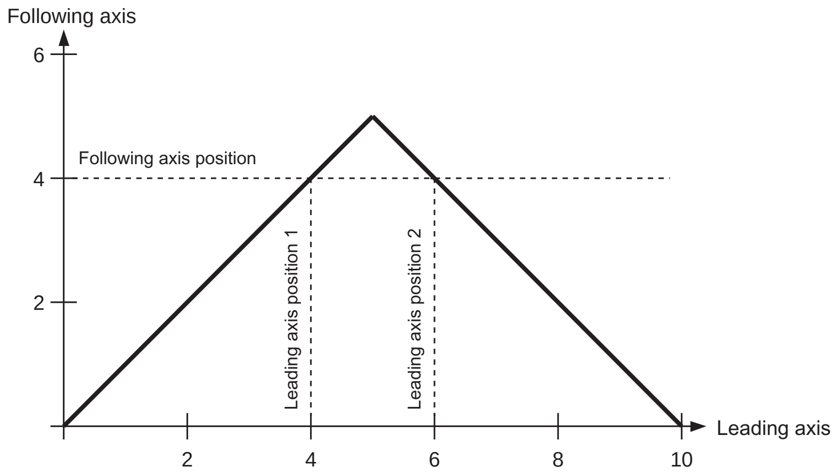
Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command successfully completed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 640) of the *Error* output parameter.	
Value	REAL	*0.0*	Read position value *Mode = 0*: Position of the following axis *Mode = 1*: Position of the leading axis *Mode = 2* : 1. derivation of the cam point *Mode = 3*: 2. derivation of the cam point	

Mode 1

Any number of leading axis positions may exist for a following axis position. You can specify an assumed leading axis position via the input parameter *ApproachPosition*.

Example:



The corresponding leading axis position for following axis position = 4 is to be determined. In the example shown, there are two leading axis positions for the following axis position. Define which leading axis position is to be determined with the input parameter *ApproachPosition*:

- *ApproachPosition* < 5
  The position value of the leading axis position 1 is determined.
- *ApproachPosition* > 5
  The position value of the leading axis position 2 is determined.
- *ApproachPosition* = 5
  The value for *ApproachPosition* lies exactly between leading axis position 1 and leading axis position 2. By definition, in this case the leading axis position with the lower position value is determined. To avoid rounding effects, definite values for *ApproachPosition* should be used.

The value for *ApproachPosition* may also lie outside the cam definition. For *ApproachPosition* = *-10* , in this example the value for the leading axis position 1 is determined.

Mode 2 and 3

If a cam disk is generated and interpolated in the Technology CPU, the result of the interpolation cannot be checked in S7T Config. The following formulae apply for velocity and acceleration:

Velocity of the following axis

*vfollowing axis = vleading axis * (1st derivation of the cam point)*

Acceleration of the following axis

*afollowing axis = aleading axis * (1st derivation of the cam point) + vleading axis2 * (2nd derivation of the cam point)*

Use the "MC_GetCamPoint" technology function is you want to check, for example, the continuity of the cam. Read the derivations of the cam points at the critical points at suitable intervals and apply the formulae mentioned above.

MC_GetCamPoint - ErrorIDs

Valid for integrated technology with firmware V4.1.x or higher

This section describes applications with firmware V3.1.x

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *DoneFlag* input parameters.
8044	Command not supported by the technology object	Command request to an axis, for example
8045	Command not allowed in current state.	The cam is not interpolated

ErrorID	error message	Description / to correct or avoid errors
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx in both OB 1 and OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function. However, the command may still be active in the integrated technology. **Notice:** Use different instance DBs for different run levels or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8061	Cam point out of the domain/range	The leading/following axis position defined at "MC_GetCamPoint" does not exist in the domain/range of the cam. Set a valid position at "MC_GetCamPoint".
8083	DB is not a technology DB	The DB specified at input parameter *CamTable* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *CamTable*. Download the current technology to the target system, or change the DB number at input parameter *CamTable*. • The user has entered invalid data at the technology DB defined at input parameter *CamTable*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.5 Technology functions - output cams, cam tracks

6.5.1 FB 430 MC_CamSwitch - Position-based cam

Position-based cams or uni-directional output cams with FB 430 "MC_CamSwitch"

Purpose

- The “MC_CamSwitch" technology function is used to enable/disable position-based cams or uni-directional output cams, and to define their switching performance.
- In your cam configuration, you can define whether the switching positions are based on setpoints or on actual values.
- Switching points can be advanced or retarded (rate time and deactivation time). The time unit set in S7T Config applies to the axis.
- The current switching state of the output cam is saved to the technology DB.

Supported for

- Position-based cam
- Uni-directional output cam

Requirements

- The Output cam technology object must have been configured in S7T Config and the current configuration must have been loaded into the Technology CPU.
- The technology DB of the output cam must have been generated and loaded to the control system.
- An output cam configured as a setpoint output cam is only activated when the axis is in position-controlled mode.

Overriding commands

MC_CamSwitch commands can only be canceled by another MC_CamSwitch command output to the same TO. MC_CamSwitch commands do not cancel any other commands.

Input parameters

Parameter	Data type	Initial value	Description
CamSwitch	INT	*0*	Number of the technology DB
Execute	BOOL	*FALSE*	Operating state transition on a rising edge
OnPosition	REAL	*0.0*	Starting position
OffPosition	REAL	*0.0*	End position (applies to position-based cam)
Hysteresis	REAL	*0.0*	Hysteresis
Delay	REAL	*0.0*	Time-based offset of output cam switching points Value < *0*: advanced activation Value > *0*: retarded activation The time unit set for the output cam in S7T Config applies. To verify or set the time unit in S7T Config: Select the Output Cam TO in the Navigator, and then select shortcut menu **Expert > Configure Units**.
Mode	INT	*2*	Operating mode: Value = *1*: Deactivate output cam Output cam deactivated permanently, technology DB: *state* = *0* (OFF) Value = *2*: Output cam enabled (output is not inverted) The output cam output is activated within the activation range Value = *3*: Output cam enabled (output is inverted) The output cam output is activated outside the activation range Value = *4*: Output cam permanently activated (output-cam output permanently activated, Technology DB: *state= 1* (ON))
Direction	INT	*1*	Effective direction of the output cam: Value = *1*: Positive effective direction Value = *2*: Positive and negative effective direction (i.e., the effective direction is irrelevant) Value = *3*: Negative effective direction Value = *4*: Use last active effective output cam direction
DoneFlagPos	INT	*0*	DoneFlag (Page 750) generation in MCDevice-DB when the output cam is activated.
DoneFlagNeg	INT	*0*	DoneFlag (Page 750) generation in MCDevice-DB when the output cam is deactivated.

Note

Response when the drive of an actual value output cam fails:
An output cam configured as an actual value output cam in S7T Config requires a valid encoder value. The associated output cam is therefore disabled when the drive fails. The command is canceled with *ErrorID = 8021* at the technology DB of the output cam.

Response when the drive of a setpoint output cam fails:
An output cam configured as a setpoint output cam in S7T Config does not require an associated encoder value. The output cam remains active when the drive fails. The output cam resumes switching operations at the configured positions after the error is acknowledged and the drive is enabled.

Response to axis restart
The output cam is deactivated when its associated axis performs a restart ("MC_Reset", *Restart = TRUE*). This always applies, regardless of whether the output cam was configured as a setpoint or as an actual value cam in S7T Config.

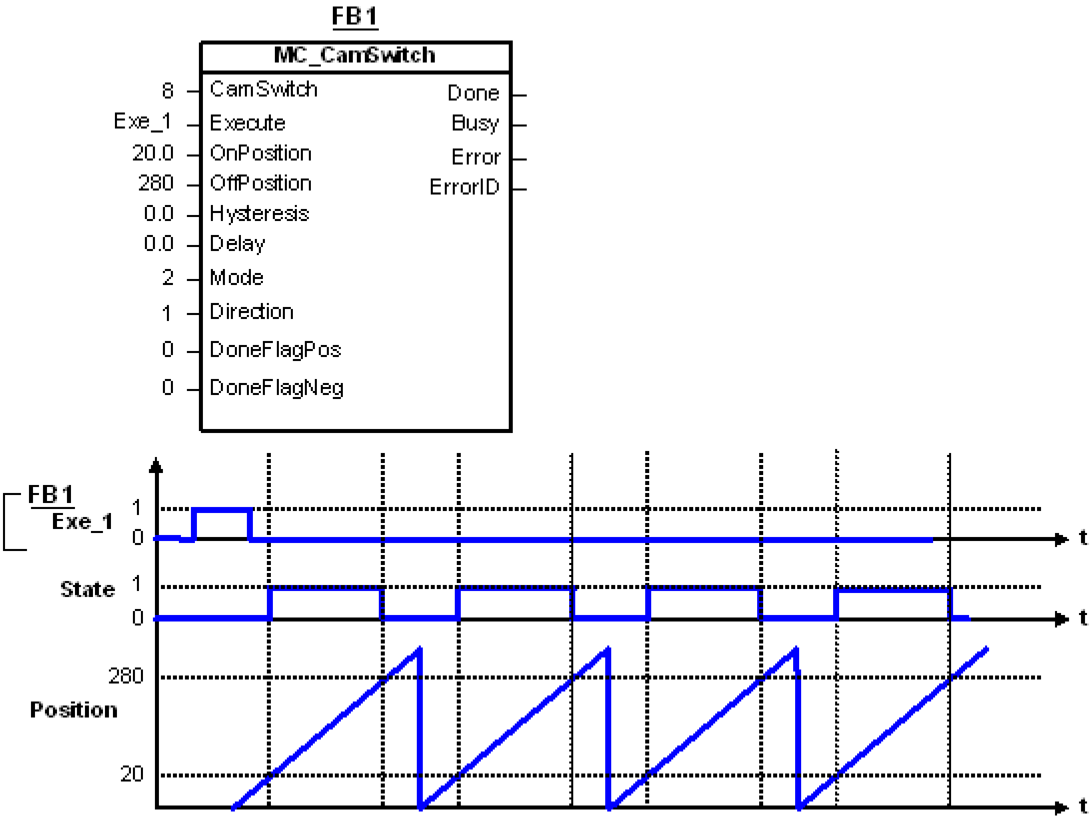
Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Change accepted	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 646) of the *Error* output parameter.	

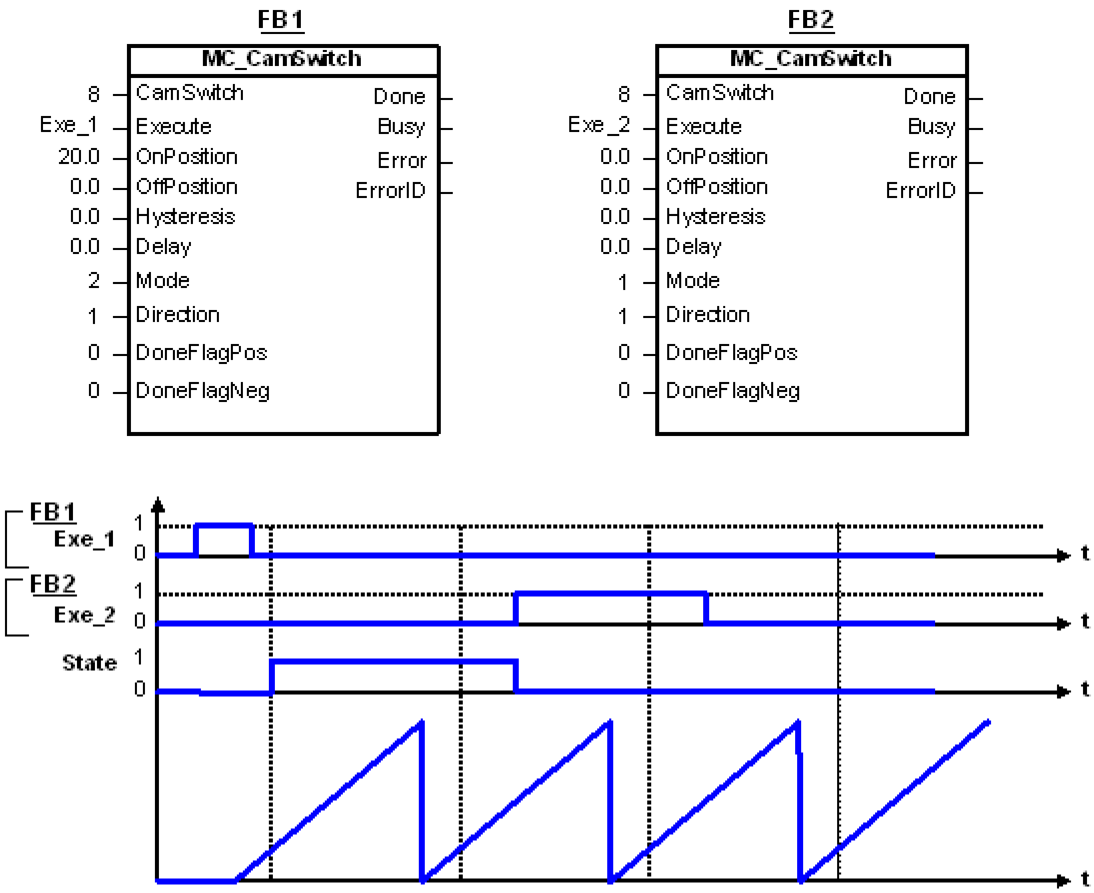
Input parameter *OffPosition*

An end position is not defined for the uni-directional output cam. The *OffPosition* input does not have any effect. The uni-directional output cam can be reset by the user program, for example, by repeating the call of the "MC_CamSwitch" technology function.

MC_CamSwitch - Example - "Position-based cam"



MC_CamSwitch - Example - "Uni-directional output cam"



Call of FB1:

The "Enable output cam" operating mode is selected by setting input parameter *Mode = 2*. The uni-directional output cam is enabled by a rising edge at input parameter *Execute*.

The direction of motion corresponds to the "positive effective direction" (input parameter *Direction = 1*). The uni-directional output cam is activated when the starting position is passed (input parameter *OnPosition = 20*) and remains in this state.

Call of FB 2:

The "Disable output cam" operating mode is selected by setting input parameter *Mode = 1*. The uni-directional output cam is disabled by a rising edge at input parameter *Execute*.

MC_CamSwitch - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode, Direction, DoneFlagPos* or *DoneFlagNeg* input parameters. Valid for input parameter *Hysteresis*: • Infinite axis - Value > ¼ of the operating range The operating range is defined by the position of the software limit switches. • Modulo axes - value > ¼ of the modulo length
8044	Command not supported by the technology object	• The command was sent to a measuring input, for example. • The command was output to a cam configured as time-based cam, for example.
8045	Command not allowed in current state	Possible causes: • Output cam output failed at hardware end, for example at an ET 200M • The technology object was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804A	Required object connection is missing	Possible causes: • The axis belonging to the output cam was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.

ErrorID	Error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.5.2 FB 431 MC_CamSwitchTime - Time-based cam

Time-based cams with FB 431 "MC_CamSwitchTime"

Purpose

- The "MC_CamSwitchTime" technology function is used to activate/deactivate time-based cams and to define their switching performance. Time-based cams are activated when the starting position is reached, and are reset when the pulse period has expired.
- In your configuration, you can define whether the switching positions are based on setpoints or actual values.
- All time definitions refer to the time base configured in S7T Config .
- The current activation state of the output cam is stored in the technology DB.

Supported for

- Time-based cam

Requirements

- The Output cam technology object must have been configured in S7T Config and the current configuration must have been loaded into the Technology CPU.
- The technology DB of the output cam must have been generated and loaded to the control system.
- An output cam configured as a setpoint output cam is only activated when the axis is in position-controlled mode.

Overriding commands

MC_CamSwitchTime commands can only be canceled by another MC_CamSwitchTime command output to the same TO. MC_CamSwitchTime commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description	
CamSwitch	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Operating state transition on a rising edge	
OnPosition	REAL	*0.0*	Starting position	
Duration	REAL	*0.0*	Pulse width (physical unit according to the "time" unit set at the corresponding axis)	
Hysteresis	REAL	*0.0*	Hysteresis	
Delay	REAL	*0.0*	Time-based offset of output cam switching points	
			Value = < *0*:	advanced activation
			Value = > *0*:	retarded activation
			The time unit set for the output cam in S7T Config applies. To verify or set the time unit in S7T Config: Select the Output Cam TO in the Navigator, and then select shortcut menu **Expert > Configure Units**.	
Mode	INT	*2*	Operating mode:	
			Value = *1*:	Output cam blocked
			Value = *2*:	Output cam enabled (output is not inverted)
			Value = *3*:	Output cam enabled (output is inverted)
			Value = *4*:	Output cam permanently activated
Direction	INT	*1*	Effective direction of the output cam:	
			Value = *1*:	Positive effective direction
			Value = *2*:	Positive and negative effective direction (meaning that the effective direction is irrelevant)
			Value = *3*:	Negative effective direction
			Value = *4*:	Use last active effective output cam direction
DoneFlagPos	INT	*0*	DoneFlag (Page 750) generation in MCDevice DB when the output cam is activated.	
DoneFlagNeg	INT	*0*	DoneFlag (Page 750) generation in MCDevice DB when the output cam is deactivated.	

Note

Response when the drive of an actual value output cam fails:
An output cam configured as an actual value output cam in S7T Config requires a valid encoder value. The associated output cam is therefore disabled when the drive fails. The command is canceled with *ErrorID = 8021* at the technology DB of the output cam.

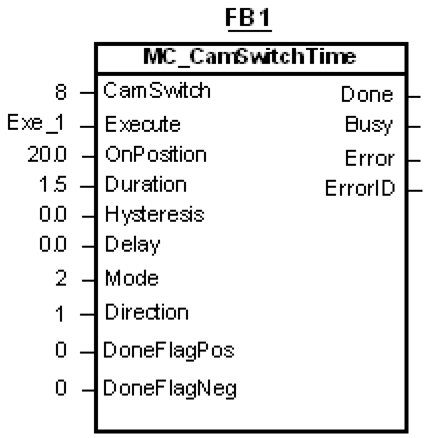
Response when the drive of a setpoint output cam fails:
An output cam configured as a setpoint output cam in S7T Config does not require an associated encoder value. The output cam remains active when the drive fails. The output cam resumes switching operations at the configured positions after the error is acknowledged and the drive is enabled.

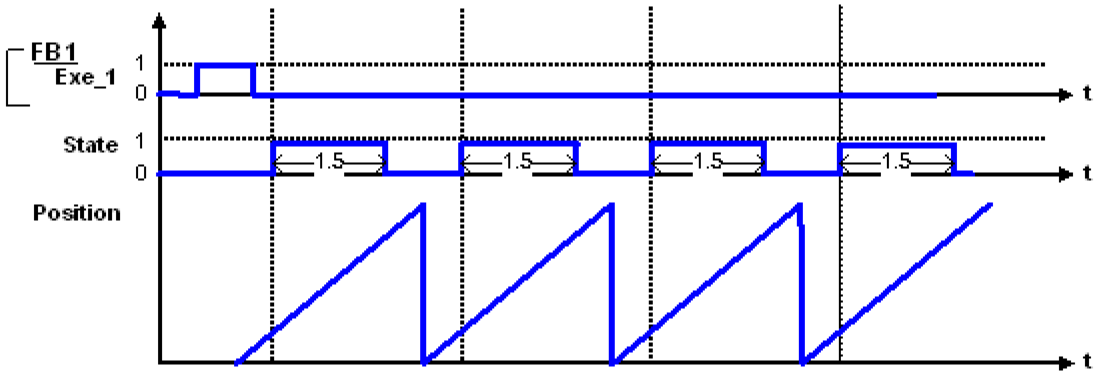
Response to axis restart
The output cam is deactivated when its associated axis performs a restart ("MC_Reset", *Restart = TRUE*). This always applies, regardless of whether the output cam was configured as a setpoint or as an actual value cam in S7T Config.

Output parameters (status outputs)

Parameter	Data type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Change accepted	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation with error. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation without error.
ErrorID	WORD	*0*	ErrorID (Page 653) of the *Error* output parameter.	

MC_CamSwitchTime - example





MC_CamSwitchTime - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode, Direction, DoneFlagPos* or *DoneFlagNeg* input parameters. Valid for input parameter *Hysteresis*: • Infinite axis - Value > ¼ of the operating range The operating range is defined by the position of the software limit switches. • Modulo axes - value > ¼ of the modulo length
8044	Command not supported by the technology object	• The command was sent to a measuring input, for example. • The command was sent to a position-based cam or switching cam, for example.
8045	Command not allowed in current state	Possible causes: • Output cam output failed at hardware end, for example at an ET 200M • The technology object was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804A	Required object connection is missing	Possible causes: • The axis belonging to the output cam was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active you may not be able to monitor it at the status outputs.

ErrorID	Error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.5.3 FB 461 MC_CamTrack - cam track

Activating the cam track with FB 461 "MC_CamTrack"

Valid for Integrated Technology with firmware version V4.1.x

Purpose

- With the technology function "MC_CamTrack" you define the current operating mode of a cam track.
- The switching state of the individual output cams of the cam tracks can be evaluated as software cams in the software cam in the variable *SingleCamState[0 ... 31]*.
- Depending on the configuration in S7T Config a common hardware output cam output is switched for all the activated individual output cams.

Supported for

- Cam tracks

Prerequisites

- The cam track technology object must have been configured in S7T Config and the current configuration must have been loaded into the technology CPU.
- The technology DB of the cam track must have been generated and loaded to the control system.
- A cam track configured as a setpoint output cam is only activated when the axis is in position-controlled mode.

Overriding commands

A command can only be canceled by another command output to the same technology object, does not cancel any further commands.

Input parameters

Parameters	Data Type	Initial value	Description	
CamTrack	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Execution of the command at the positive edge	
CyclicMode	BOOL	*TRUE*	Creation of the cam track at the axis reference position.	
			TRUE	Cyclic creation of the cam track. The cam track is created at the axis reference position (input parameter *ReferencePosition)*. The created length of the cam track corresponds to the track length (input parameter *CamTrackLength).*The cam track is repeatedly created at the axis reference position according to its track length. The cam track is active in all axis ranges.
			FALSE	Non-cyclic creation of the cam track. The cam track is created at the axis reference position (input parameter *ReferencePosition)*. The created length of the cam track corresponds to the track length (input parameter *CamTrackLength).*
Mode	INT	*2*	Operating mode of the cam track	
			Value = *1*	Disable cam track
			Value = *2*	Enable cam track
			Value = *3*	Value illegal
			Value = *4*	Individual output cam and output cam output are activated permanently

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">*CommandMode*</td><td rowspan="4">INT</td><td rowspan="4">*1*</td><td colspan="2">The input parameter only acts at *Mode* = *1* and *2*.</td></tr>
<tr><td>Value = *1*</td><td>• *Mode = 1* (lock)
Cam track is deactivated immediately.
• *Mode = 2* (release)
If the axis position is within the cam track length, the cam track is immediately activated. If the axis position is outside the cam track length, the cam track is activated when the axis reference position is crossed (input parameter *ReferencePosition*) The axis reference position must be crossed in a positive direction.</td></tr>
<tr><td>Value = *2*</td><td>• Non-cyclic creation of the cam track:
The cam track is activated when the axis reference position (input parameter *ReferencePosition*) is crossed. The axis reference position must be crossed in a positive direction.
The deactivation of the cam track is dependent on the configuration in S7T Config.
• Cyclic creation of the cam track:
The cam track is activated when the axis reference position or its cyclic repetition is crossed. The axis reference position must be crossed in a positive direction.
The cam track remains active until the cam track is locked.</td></tr>
<tr><td>Value = *3*</td><td>An active cam track command continues to be executed until the cam track output, or all the individual output cams, have been deactivated.
Afterwards the cam track is activated / deactivated with the new command.</td></tr>
<tr><td rowspan="4">*CamTrackLength*</td><td rowspan="4">REAL</td><td rowspan="4">*0.0*</td><td colspan="2">Cam track length
The length of the cam track to be activated is defined by using the cam track length. In the case of a cyclic use of the cam track, the cam track length determines the cam track cycle.</td></tr>
<tr><td>Value < *0*:</td><td>Not permitted</td></tr>
<tr><td>Value = *0*:</td><td>• In the case of modulo axes the track length is set to the modulo length
• Not permitted for non-modulo axes</td></tr>
<tr><td>Value > *0*:</td><td>Track length of the cam track</td></tr>
<tr><td>*ReferencePosition*</td><td>REAL</td><td>*0.0*</td><td colspan="2">Axis reference position
Axis position from which the cam track is to be created.</td></tr>
<tr><td>*Hysteresis*</td><td>REAL</td><td>*0.0*</td><td colspan="2">Hysteresis of the individual output cams of the cam track
Range around the switching position in which the output cam does not change its switching change in spite of minimum position fluctuations.</td></tr>
</table>

Parameters	Data Type	Initial value	Description
ActivationDelay	REAL	*0.0*	Time-based offset for activating (to compensate switching times of digital outputs and switching elements)
			Value < *0*: Advanced activation
			Value = *0:* No delayed activation
			Value > *0*: Delayed activation
DeactivationDelay	REAL	*0.0*	Time-based offset for deactivating (to compensate switching times of digital outputs and switching elements)
			Value < *0*: Advanced deactivation
			Value = *0:* No delayed deactivation
			Value > *0*: Delayed deactivation

NOTICE

When selecting the cam track length, ensure that no valid individual output cams are located partially or completely outside the cam track length. If necessary, extend the cam track length or remove the validity of the corresponding individual output cam.

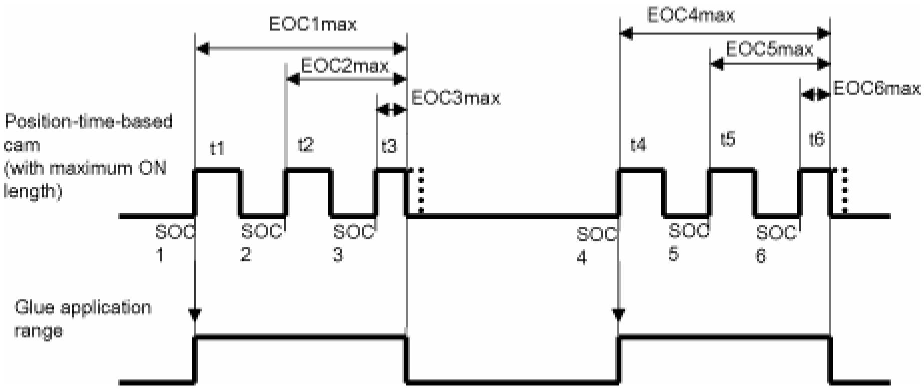
Valid individual cams that are located partially or completely outside the cam track length are projected into the range of the cam track length and cause unwanted switching states.

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Command was executed
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE:* The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE: Command initiation **without error**.
ErrorID	BOOL	*0*	ErrorID (Page 658) of the *Error* output parameter.

Example of a cam track used to control a glue application

In the following example, a cam track with three output cams is used to control the application of glue onto a workpiece. No glue may be applied outside of a predefined area.



The amount of glue is controlled by means of the time-based cams *t1 to t3* and *t4 to t6*. The beginning of the glue application is defined by means of the cam beginning *SOC1* or *SOC4* respectively.

To ensure that glue is not applied outside the workpiece lengths if the axis moves rapidly, the maximum ON lengths are limited.

In the example the ON lengths *EOC1max*, *EOC2max*, and *EOC3max* ensure that glue is not applied outside the glue application range. The same applies for the second glue application range, irrespective of the axis speed.

MC_CamSwitch - ErrorIDs

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.

ErrorID	error message	Description / to correct or avoid errors
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *CommandMode* input parameters. Valid for input parameter *Hysteresis*: • Infinite axis - Value > ¼ of the operating range The operating range is defined by the position of the software limit switches. • Modulo axes - value > ¼ of the modulo length
8044	Command not supported by the technology object	• For example, command is applied to the technology object of an axis.
8045	Command not allowed in current state	Examples: • Output cam output failed at hardware end at an ET 200M • The axis assigned to the cam track is in speed-controlled mode.
804A	Required object connection is missing	Possible causes: • The axis belonging to the cam track was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *CamTrack* was not found or is not a technology DB.

ErrorID	error message	Description / to correct or avoid errors
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *CamTrack*. Download the current technology to the target system, or change the DB number at input parameter *CamTrack*. • The user has entered invalid data at the technology DB defined at input parameter *CamTrack*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.5.4 FB 462 MC_ReadCamTrackData - read out cam track

Reading out the cam track with FB 462 "MC_ReadCamTrackData"

Valid for Integrated Technology with firmware version V4.1.x

Purpose

- The technology function is used to read out the individual output cam data of a cam track from the integrated technology.
- The *Mode* input parameter is used to determine whether the "actual values" of the individual output cam data of a cam track or the currently effective individual cam data are read out.

Supported for

- Cam tracks

Overriding commands

A command can only be canceled by another command output on the same technology object. A command does not cancel any further commands.

Input parameters

Parameters	Data Type	Initial value	Description	
CamTrack	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Execution of the command at the positive edge	
Mode	INT	*0*	Determines the range to be read.	
			Value = *0:*	Reads the "actual values" of the individual output cam data of a cam track. The values can be found in online mode of S7T Config: Expert List of the cam track > System variables tab > "Actual value" column > System variable *userdefault. singlecamsettings. cam. cam[x]*.
			Value = *1*:	Reads the effectively effective individual output cam data of a cam track. The values can be found in online mode of S7T Config: Expert List of the cam track > System variables tab > "Actual value" column > System variable *effectivedata. singlecamsettings. cam. cam[x]*.
StartCam	INT	*0*	Number of the individual output cam (0 to 31) from which the individual cam data are to be read out. The individual output cam data are always read out up to individual output cam 31.	
Data	ANY	-	Destination for read data. The maximum length of data to be read is defined by the data type and by the repetition coefficient at the ANY pointer.	

Note

You can save yourself editing work by using UDT 100 from the "S7-Tech" library. UDT 100 contains the data structure for reading the cam track data.

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command was executed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	BOOL	*0*	ErrorID (Page 664) of the *Error* output parameter.	

MC_ReadCamTrackData - ErrorIDs

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the number of cam data.
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *StartCam* input parameter. Valid for input parameter *Hysteresis*: • Infinite axis - Value > ¼ of the operating range The operating range is defined by the position of the software limit switches. • Modulo axes - value > ¼ of the modulo length

ErrorID	error message	Description / to correct or avoid errors
8044	Command not supported by the technology object	• For example, command is applied to the technology object of an axis
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *CamTrack* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *CamTrack* . Download the current technology to the target system, or change the DB number at input parameter *CamTrack* . • The user has entered invalid data at the technology DB defined at input parameter *CamTrack* . Delete the technology DB in "Technology Objects Management" and then create a new one.
8086	Parameter not available	Currently unable to execute the read / write command; repeat the command.

ErrorID	error message	Description / to correct or avoid errors
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data record length (ANY pointer)	Specified data length is invalid. Possible causes: • The length exceeds 576 bytes. • The specified length cannot be divided by 18.
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL

6.5.5 FB 463 MC_WriteCamTrackData - write cam track

Writing a cam track with FB463 "MC_WriteCamTrackData"

Valid for Integrated Technology with firmware version V4.1.x

Purpose

- The technology function is used to write the individual output cam data of a cam track into the "actual values" of the integrated technology. The changed individual output cam data can be found in online mode of S7T Config: Expert List of the cam track > System variables tab > "Actual value" column > System variable *userdefault. singlecamsettings. cam. cam[x]*.

Supported for

- Cam tracks

Overriding commands

A command can only be canceled by another command output on the same technology object. A command does not cancel any further commands.

Input parameters

Parameters	Data Type	Initial value	Description
CamTrack	INT	*0*	Number of the technology DB
Execute	BOOL	*FALSE*	Execution of the command at the positive edge
StartCam	INT	*0*	Number of the individual output cam (0 to 31) from which the individual cam data are to be written. The individual output cam data are always written up to individual output cam 31.
Data	ANY	-	Source area of data to be written. The length of data to be written is defined by the data type and the repetition factor in the ANY pointer.

Note

You can save yourself editing work by using UDT 100 from the "S7-Tech" library. UDT 100 contains the data structure for writing the cam track data.

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command was executed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 668) of the *Error* output parameter.	

MC_WriteCamTrackData - ErrorIDs

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the number of cam data.
8043	Illegal parameter value	Concerns all input parameters of the data type REAL, or the *StartCam* input parameter. Valid for input parameter *Hysteresis*: • Infinite axis - Value > ¼ of the operating range The operating range is defined by the position of the software limit switches. • Modulo axes - value > ¼ of the modulo length

ErrorID	error message	Description / to correct or avoid errors
8044	Command not supported by the technology object	Possible causes: • The command was sent to a measuring input.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *CamTrack*. Download the current technology to the target system, or change the DB number at input parameter *CamTrack*. • The user has entered invalid data at the technology DB defined at input parameter *CamTrack*. Delete the technology DB in "Technology Objects Management" and then create a new one.
8086	Parameter not available	Currently unable to execute the read / write command; repeat the command.
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short

ErrorID	error message	Description / to correct or avoid errors
8089	Invalid data record length (ANY pointer)	Specified data length is invalid. Possible causes: • The length exceeds 576 bytes. • The specified length cannot be divided by 18.
808A	Invalid REAL values in DB	The DB transferred to the technology contains at least one value that is incompatible with a permissible REAL format specification Check all DB values and adjust these as required.
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL

6.6 Technology functions - measuring inputs, external encoders

6.6.1 FB433 MC_MeasuringInput - Measuring input

Measuring input with FB 433 "MC_MeasuringInput"

Purpose

- The "MC_MeasuringInput" technology function is used to enable the acquisition of **a single** measurement by means of a measuring input.
- A measuring input records the position of a configured axis or of an external encoder based on the occurrence of an external event, such as a pressure mark or reference cam.
- The measurement can be limited to a specific measuring range. The first measured value within the range defined by the *StartPosition* and *EndPosition* is accepted
- When *StartPosition = EndPosition*, the measurement range is ignored.
- The function is terminated after the measured value has been recorded and output, or when the measurement was canceled, for example, by another command.
- Several operating modes can be selected at input *Mode*, based on the drive and measuring element used.

Supported for

- Measuring input

Prerequisites

- The axis is enabled for position-controlled operation.
- The connected drive or measuring element supports the selected mode. Refer to the relevant drive documentation for further information!
- The drive interface IM 174/ADI4 does not support the simultaneous execution of MC_MeasuringInput and MC_Home commands.

Note

Make allowances for the drive response times. For further information, refer to the relevant drive documentation (on the "S7-Technology" CD, for example).

Overriding commands

MC_MeasuringInput commands can only be canceled by another MC_MeasuringInput command output to the same TO. MC_MeasuringInput commands do not cancel any other commands.

Input parameters

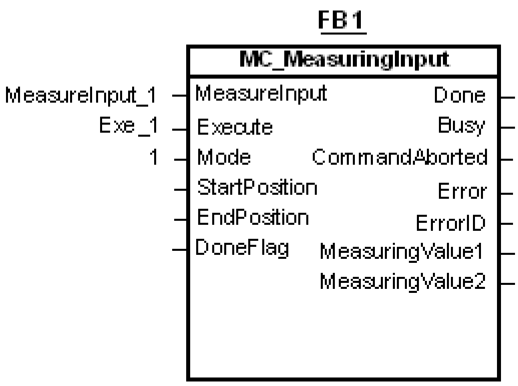
Parameters	Data Type	Initial value	Description	
MeasureInput	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the operating mode at the rising edge	
Mode	INT	*1*	Operating mode:	
			Value = *1*:	Measurement on rising edge
			Value = *2*:	Measurement on falling edge
			Value = *3*:	Measurement on both edges, starting with the rising edge (two separate measured values)
			Value = *4*:	Measurement on both edges, starting with the falling edge (two separate measured values)
			Value = *5*:	Cancel current measurement
StartPosition	REAL	*0.0*	Start of the measuring range	
EndPosition	REAL	*0.0*	End of the measuring range	
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB The DoneFlags value is inverted when a new measured value has been returned	

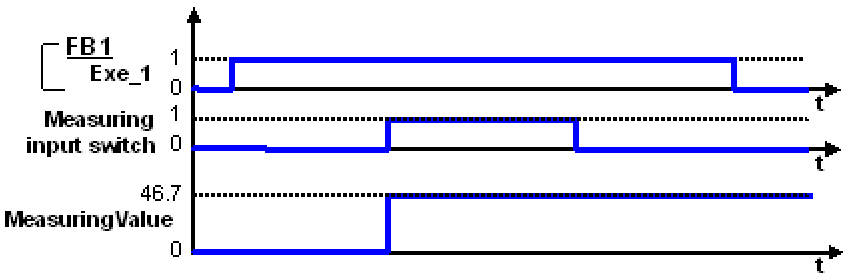
Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Measurement completed; measured value is valid
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The measurement command was canceled. (no edge detected within the measuring range)
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	0	ErrorID (Page 672) of the *Error* output parameter.
MeasuringValue1	REAL	*0.0*	*Messwert1*
MeasuringValue2	REAL	*0.0*	*Messwert2* (only valid for measurement at both edges)

MC_MeasuringInput - example

The measuring input is activated. The actual position value (46.7, for example) will be saved to *MeasuringValue1* when the measuring input is detected.





MC_MeasuringInput - ErrorIDs

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands at the next technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8034	Could not record a measured value in the measuring range	Possible causes: • No measurement event within the specified measuring range • Measuring input on the drive not programmed
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *DoneFlag*. input parameters.
8044	Command not supported by the technology object	Sending a command to an output cam, for example.
8045	Command not allowed in current state	Possible causes: • The technology object was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804A	Required object connection is missing	Possible causes: • The axis belonging to the measuring input was deactivated in S7T Config or by using the "MC_ActivateTO" technology function.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value with invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.6.2 FB432 MC_ExternalEncoder - External encoder

External encoders with FB 432 "MC_ExternalEncoder"

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware up and including V3.1.x

Purpose

- The "MC_ExternalEncoder" technology function controls an external encoder that can be used, for example, as a real leading axis (master) for synchronous operation.
- The external encoder can be enabled / disabled
- Absolute encoder adjustment is supported
- Synchronization: The actual position value of an incremental encoder can be influenced by corresponding settings or by passive homing.

Supported for

- External encoders

Prerequisites

- The encoder value must be valid. Check the variables *Statusword.CyclicInterface* and *Statusword.EncoderValid* in the technology data block.
- You can only modify the actual position value of an enabled external encoder.

Overriding commands

MC_ExternalEncoder commands can only be canceled by another MC_ExternalEncoder command output to the same TO. MC_ExternalEncoder commands do not cancel any other commands.

Input parameters

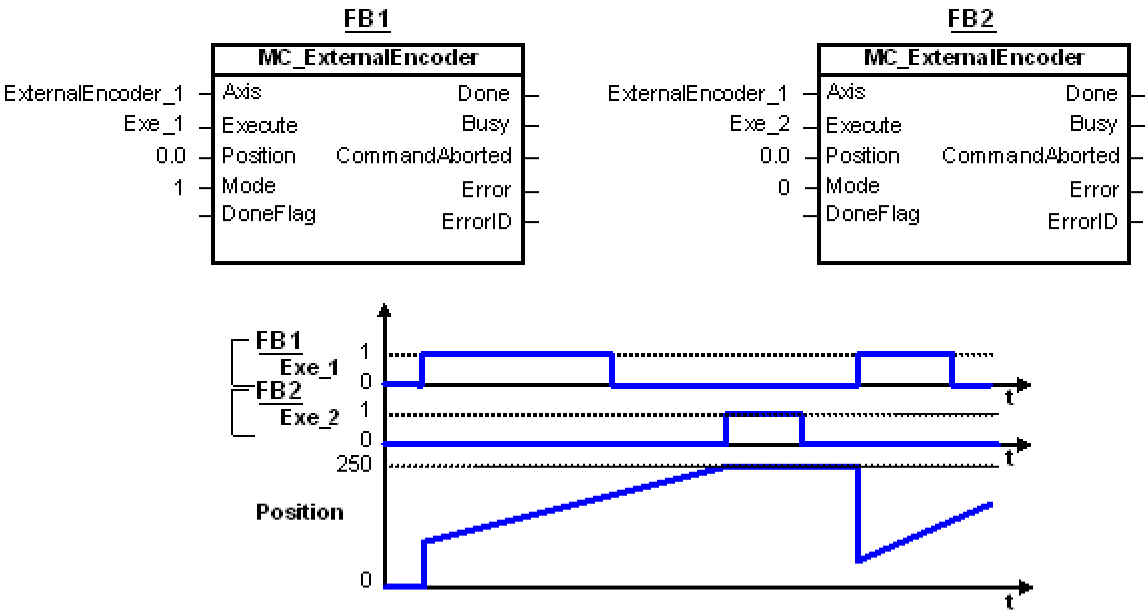
Parameters	Data Type	Initial value	Description
Axis	INT	*0*	Number of the technology DB
Execute	BOOL	*FALSE*	Start at the positive edge
Position	REAL	*0.0*	Position setpoint
Mode	INT	*1*	Operating mode:
			Value = *0*: Deactivating an external encoder The positional actual value is no longer updated.
			Value = *1*: Enable external encoder (positional actual value is updated)
			Value = *2*: Direct homing: The current position is assigned the value of the *Position* input parameter.
			Value = *3*: Passive homing with "encoder zero mark only", or "external zero mark only", or "reference cam and encoder zero mark" (default: "encoder zero mark only"): The current position is assigned the value of input parameter *Position* at the reference point.
			Value = *4*: Direct homing: The default reference position preconfigured in S7T Config is assigned to the current position.
			Value = *5*: Passive homing with "encoder zero mark only", or "external zero mark only", or "reference cam and encoder zero mark" (default: "encoder zero mark only"): The default reference position programmed in S7T Config is assigned at the reference point to the current position.
			Value = *6*: Absolute encoder adjustment The current position is assigned the value of input parameter *Position*. The position offset derived from absolute encoder adjustment is retained after a power failure. The axis must be at a standstill in order to allow precise adjustment.
			Value = *7:* Actual value correction Position value = Actual position - *Position* input parameter
			Note that in addition to the absolute value encoder adjustment, any position adjustments (*Mode = 2, 4* or *7*) will also affect the axis. Only the absolute encoder adjustment is activated after POWER OFF or restart ("MC_Reset", *Restart* = *TRUE*).
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Command completed
Busy	BOOL	*FALSE*	*TRUE*: Command in process
CommandAborted	BOOL	*FALSE*	*TRUE*: The command was canceled by another command or as a result of error during its execution. If no error is displayed in the *ErrorStatus* of the technology DB, the command was canceled by a subsequent command. If an error is indicated in the *ErrorStatus* tag of the technology DB, an error affecting the technology object during command execution caused the command to be aborted.
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 677) of the *Error* output parameter.

MC_ExternalEncoder - example

An external encoder is enabled. The position value is zero prior to this enable. The position value not updated until activation is completed. The position value is no longer updated if the encoder is subsequently disabled, regardless whether the encoder can be moved. The position changes to the actual value when the encoder is enabled again.



MC_ExternalEncoder - ErrorIDs

Valid for Integrated Technology with firmware V3.2.x

This section describes applications with firmware up to V3.1.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8022	No actual value at the axis / external encoder	Encoder or data bus not ready, for example
8040	The axis / external encoder are disabled, or the wrong mode is set	The external encoder is disabled. The external encoder must be enabled in order to call "MC_ExternalEncoder" with *Mode > 1* . Remedy: • Enable the external encoder at the technology function "MC_ExternalEncoder" with *Mode = 1* . • Eliminate and acknowledge any error that may have disabled the external encoder, and then enable the external encoder by setting *Mode = 1* at the "MC_ExternalEncoder" technology function.
8043	Illegal parameter value	Relates to all input parameters of data type REAL, or the *Mode* or *DoneFlag.* input parameters. For example *Mode* = *167*
8044	Command not supported by the technology object	Command request to an axis, for example
8045	Command not allowed in current state	Example: The encoder value is invalid at the time of command start (for example, hardware switched off).
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.

ErrorID	Error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.7 Technology functions - Basic functions

6.7.1 FB402 MC_Reset - Acknowledging errors

Acknowledging errors by calling FB 402 "MC_Reset"

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

Purpose

- Call the "MC_Reset" technology function to acknowledge all errors accepting acknowledgment in the user program. Acknowledgment of fatal errors is only possible either by cycling power, or by downloading the project data to the module again. The acknowledgment also clears the entries at the *ErrorID* and *ErrorBuffer* parameters, and resets the *ErrorStatus* bits in the technology DB.
- Call the technology function "MC_Reset" *Restart = TRUE* to start the initialization (restart) of axes, output cams, cam tracks, cam disks, external encoders, and measuring inputs. Axes with incremental encoder are in "non-homed" state after their initialization. The cam originally configured in S7T Config is restored at the cam disks.
- If the technology function "MC_Reset" (*Restart = TRUE* or *FALSE*) is applied to the technology data block "MCDevice"(*Axis* = DB No. of the MCDevice DB), the maximum runtime of the command execution is reset to *0* (*MaxLoopDuration* at the MCDevice DB). When using integrated technology with firmware V3.2x or higher, the system also sets the *TODBTaskOverflows* (number of overflows in technology DB updates) variable to *0*.
- The system continues command execution during a restart. However, the technology object performing the restart cannot accept any commands for the duration of the restart.

Supported for

- Speed-controlled axes
- Positioning axes
- Synchronization axes
- Output cams
- Cam tracks
- Measuring inputs
- Cams
- External encoders
- MCDevice-DB

Restart conditions

- "Axis" TO
  Restart is only possible when the axis is disabled ("MC_Power" output parameter *Status = FALSE* and *Busy = FALSE*)
- "External encoder" technology object
  Restart is only possible when the external encoder is disabled (technology DB, variable *Statusword.DriveEnabled = FALSE*)
- "Output cam" technology object
  Restart is only possible if the output cam is disabled (technology DB variable *Mode = 1*)
- "Cam tracks" technology object
  Restart is only possible if the output cam is disabled (technology DB variable *Control = 2 / INACTIVE*)
- A restart of technology object "Measuring input"
  is only possible if the measuring input is inactive (technology DB variable *Status = 0, 2* or *3*)
- "Cam" technology object
  Restart is only possible if the cam is not in use (technology DB, variable *UserCount = 0*)

Note

Set *Restart = FALSE* if you only want to clear the errors of a technology object.

Reinitialization of a technology object (*Restart = TRUE*) may take several hundred milliseconds. The technology object is not available for the duration of this process.

Interaction of commands

New command - active single command (3) (Page 743)

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Restart	BOOL	*FALSE*	*FALSE*:	Acknowledge error
			TRUE:	Restart - initialization and activation of modified configuration parameters

Output parameters (status outputs)

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td>Done</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: Error was acknowledged or restart was executed</td></tr>
<tr><td>Busy</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: Command in process</td></tr>
<tr><td>CommandAborted</td><td>BOOL</td><td>FALSE</td><td colspan="2">TRUE: The command was canceled by another command or as a result of error during its execution.
If ErrorStatus of the technology DB does not report an error, the command was canceled by a subsequent command.
If an error is indicated in ErrorStatus, an error affecting the technology object during command execution caused the command to be aborted.</td></tr>
<tr><td rowspan="2">Error</td><td rowspan="2">BOOL</td><td rowspan="2">FALSE</td><td>TRUE:</td><td>Command initiation with error. The command is not executed. For information about the cause, refer to the ErrorID.</td></tr>
<tr><td>FALSE:</td><td>Command initiation without error.</td></tr>
<tr><td>ErrorID</td><td>WORD</td><td>0</td><td colspan="2">ErrorID (Page 682) of the Error output parameter.</td></tr>
</table>

MC_Reset - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.

ErrorID	Error message	Description / to correct or avoid errors
8045	Not permitted at current status	"MC_*Reset*" (*Restart = TRUE*)is already active. The command is canceled
8047	Cannot acknowledge this error	It is not possible to acknowledge the error by calling technology function "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in the restart phase. • Superimposing commands limited dynamic values (see warning *0021* at the Synchronization technology DB) Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.
8048	Unable to restart	• Restart is only permitted when the axis is disabled. • The cam disk is in use by another application and can thus not be reset.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example.)
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.

ErrorID	Error message	Description / to correct or avoid errors
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.

6.7.2 FB 460 MC_ActivateTO - Deactivating / activating a technology object

Deactivating / activating a technology object with FB 460 "MC_ActivateTO"

Valid for integrated technology with firmware V4.1.x or higher

Purpose

- Use the "MC_ActivateTO" technology function if you want to remove configured technology functions temporarily or permanently out of cyclic processing of the integrated technology. For standard operation it is not necessary to activate a technology objects that was not deactivated beforehand.
- Avail the computing performance of the integrated technology to a better purpose. Remove the technology objects that are not required from the cyclic processing of the integrated technology by the "MC_ActivateTO" technology function.
- When a technology object is deactivated, it is no longer processed in the integrated technology. A deactivated technology object cannot accept any commands and not update the corresponding technology DB are longer until it has been activated again.
- The activation of a technology object cancels a previously valid deactivation.
- The activation status of a technology object can be read out by using *Mode = 0*. Call the "MC_ActivateTO" technology function in a second instance if you want to check the current activation or deactivation.

Note

You can also deactivate technology objects permanently in S7T Config.

Select the "Technology" object in the Navigator and then select the **Edit > Object states** command.

The deactivation of the technology objects is stored in the project and also remains after POWER OFF, POWER ON and restart.

Be sure to observe the relevant notes of the corresponding online help.

Supported for

- Axes (real and virtual axis)

  In the case of synchronized axes the corresponding synchronous objects are also deactivated / activated.
- External encoders
- Output cam
- Cam tracks
- Measuring inputs
- Cams

Requirement

- The technology CPU must be in cyclic operation ("RUN" status.)
- The technology object may not be active or in use.

Overriding commands

MC_ActivateTO commands can not be canceled by any other command.

A MC_ActivateTO command does not cancel any other command.

Input parameters

Parameters	Data Type	Initial value	Description	
Axis	INT	*0*	Number of the technology DB (of the corresponding technology object)	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Mode	INT	*0*	Value = *0*:	Supplies the current activation status of the technology object at the *Status* output parameter.
			Value = *1*	Activate a technology object
			Value = *2*	Deactivate a technology object

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	Command completed: • Technology object was activated • Technology object was deactivated • Activation status was read out	
Busy	BOOL	*FALSE*	*TRUE*	The command is being executed
Error	BOOL	*FALSE*	*FALSE*	Command initiation without error.
			TRUE	Command initiation with error. The command is not executed. For information about the cause, refer to the *ErrorID*.
ErrorID	WORD	*0*	ErrorID (Page 686) of *Error* output parameter.	
Status	INT	*0*	Shows the current activation status of the technology object at *Done = TRUE* . The status is only displayed at *Mode = 0*.	
			Value = *0*	Activation status was not implemented, or is not yet updated.
			Value = *1*	Technology object was activated.
			Value = *2*	Technology object was deactivated.
			Value = *3*	Technology object is being activated. Activation has not been completed yet.
			Value = *4*	Technology object is being deactivated. Deactivation has not been completed yet.

When deactivating technology objects take the following conditions into account in the program:

- Ensure that the subordinate technology objects (output cams, cam tracks and measuring inputs) are deactivated first.
- Ensure that these are not used when cam disks are activated.
- Disable the assigned hydraulic axis if you want to deactivate a cam disk used as a valve profile.

Take the following steps when deactivating technology objects in the user program:

1. Terminate all the commands at the technology object to be deactivated. Wait until all the motion commands have been terminated.
2. Deactivate or disable the technology object.
3. Deactivate the technology object by using the "MC_ActivateTO" technology function.

When all the technology objects belonging to a drive or DP slave have been deactivated, the corresponding DP slave can be deactivated by using the "MC_ActivateDPSlave" technology function. Follow the instructions in the "Deactivating / Activating a DP slave with FB 457 MC_ActivateDPSlave" chapter to this purpose.

Note

Configuration data after deactivation and renewed activation

Configuration data that were changed during operation remain changed even after renewed activation. If a required restart was not carried out before deactivation, the restart has to be carried out after renewed activation.

Configuration data of the technology object cannot be changed while the technology object is deactivated.

Note

System variables after deactivation and renewed activation

System variables are reinitialized after renewed activation, as if the technology object has been restarted. The reinitialization has the following effects:

- A referenced axis with an incremental encoder loses its homing.
- As a measuring input the previous measured values are deleted.
- Position offsets in the basic or higher-level coordinate system are deleted.

Note

If the technology object of an axis is deactivated by using the "MC_ActivateTO" technology function or in S7T Config , the corresponding drive can indicate an error in S7T Config , since cyclic communication with the drive was stopped.

MC_ActivateTO - ErrorIDs

Valid for integrated technology with firmware V4.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Invalid value at the *Mode* input parameter.
8045	Command not allowed in current state	Possible causes: • During deactivation the axis is enabled with the "MC_Power" technology function. • A command to the technology object is not yet completed, e.g. restart still active.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute* / *Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8054	Deactivation of a technology object not permitted	Deactivation of the technology object is not possible, since the technology object is still being used.
8057	A deactivation / activation command is already active	Wait until the deactivation / activation command has been completed before starting a new command.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the PLC for the technology DB specified at the *Axis* input parameter. • Download the current technology to the target system, or change the DB number at the *Axis* input parameter. • The user has written invalid data to the technology DB specified at the *Axis* input parameter. • In "Technology Objects Management", delete and then recreate the technology DB

6.7.3 FB 457 MC_ActivateDPSlave - Deactivating / activating a DP slave

Deactivating / activating a DP slave with FB 457 "MC_ActivateDPSlave"

Valid for integrated technology with firmware V4.1.x or higher

Purpose

- Use the "MC_ActivateDPSlave" technology function if you want to deactivate configured DP slaves at the DP(DRIVE) temporarily or permanently. This makes sense, for example, if plant units are to be separated temporarily from the DP(DRIVE) . For standard operation it is not necessary to activate a DP slave that was not deactivated beforehand.
- A deactivated DP slave does not indicate a bus fault at the technology CPU.
- A deactivated DP slave no longer participates in cyclic data exchange at the DP(DRIVE) . The minimum period for the DP cycle that can be set in HW Config continues to depend on the number of configured DP slaves.
- If a deactivated DP slave remains at the DP(DRIVE), , it can continue to be configured by means of the STARTER functionality of S7T Config .
- The activation status of a DP slave be read out by using *Mode = 0*. Call the "MC_ActivateDPSlave" technology function in a second instance if you want to check the current activation or deactivation.

Supported for

- Any DP slaves at DP(DRIVE)

Requirement

- The technology CPU must be in cyclic operation ("RUN" status.)
- Technology objects that reference addresses of the DP slave have to be deactivated.

Overriding commands

MC_ActivateDPSlave commands can not be canceled by any other command.

A MC_ActivateDPSlave command does not cancel any other command.

Input parameters

Parameters	Data Type	Initial value	Description	
Diagnostic address	INT	*0*	Diagnostic address of the DP slave (see the entry in HW Config)	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
Mode	INT	*0*	Value = *0*:	Supplies the current activation status of the DP slave at the *Status* output parameter.
			Value = *1*	Activate a DP slave The cyclic communication to the DP slave is established.
			Value = *2*	Deactivate a DP slave The cyclic communication to the DP slave is disconnected.

Output parameters (status outputs)

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Initial value</th><th colspan="2">Description</th></tr>
<tr><td>Done</td><td>BOOL</td><td>FALSE</td><td colspan="2">Command initiated:
• The activation command was activated
If the DP slave to be activated does not exist at the DP(DRIVE), the DP slave with the connection at the DP(DRIVE) is activated.
• The deactivation command was initiated
If the DP slave to be deactivated was removed from the DP(DRIVE), the bus fault LED additionally extinguishes. The entry "Station return" is entered in the diagnostics buffer of the CPU and integrated technology.
• Activation status was read out
Call the "MC_ActivateDPSlave" technology function with Mode = 0 to check the activation / deactivation status of the DP slave.</td></tr>
<tr><td>Busy</td><td>BOOL</td><td>FALSE</td><td>TRUE</td><td>The command is being executed</td></tr>
<tr><td rowspan="2">Error</td><td rowspan="2">BOOL</td><td rowspan="2">FALSE</td><td>FALSE</td><td>Command initiation without error.</td></tr>
<tr><td>TRUE</td><td>Command initiation with error. The command is not executed. For information about the cause, refer to the ErrorID.</td></tr>
<tr><td>ErrorID</td><td>WORD</td><td>0</td><td colspan="2">ErrorID (Page 692) of Error output parameter.</td></tr>
<tr><td rowspan="7">Status</td><td rowspan="7">INT</td><td rowspan="7">0</td><td colspan="2">Shows the current activation status of the DP slave at Done = TRUE.
The status is only displayed at Mode = 0.</td></tr>
<tr><td>Value = 0</td><td>Activation status was not implemented, or is not yet updated</td></tr>
<tr><td>Value = 1</td><td>DP slave was activated</td></tr>
<tr><td>Value = 2</td><td>DP slave was deactivated or is not located at the DP(DRIVE)</td></tr>
<tr><td>Value = 3</td><td>DP slave is being activated. Activation has not been completed yet.
If the status value 3 is constantly displayed, this indicates an error at the DP slave.</td></tr>
<tr><td>Value = 4</td><td>Status value 4 is not displayed.</td></tr>
<tr><td>Value = 5</td><td>DP slave is not connected to the DP(DRIVE)</td></tr>
</table>

Before deactivating the DP slave, ensure that all the technology objects that are connected have been deactivated. Follow the instructions in the "MC_ActivateTO - Deactivating / activating a technology object" chapter to this purpose.

After all the technology objects that are connected have been deactivated, the DP slave can be deactivated and then removed from the bus.

MC_ActivateDPSlave - ErrorIDs

Valid for integrated technology with firmware V4.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits • Too many active commands at the following technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted*or *Error* is *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Invalid value at the *Mode*. input parameter
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute* / *Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Caution:** Use different instance DBs at different run levels, or interlock the call of the technology function.

ErrorID	error message	Description / to correct or avoid errors
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8057	A deactivation / activation command is already active	Wait until the deactivation / activation command has been completed before starting a new command.
8058	DP slave cannot be configured	The configuration of the DP slave could not be loaded from the DP master during activation of the DP slave. Check the slave's hardware.
8090	Logical address invalid	The specified diagnostics address is not assigned to any DP slave.

6.7.4 FB406 MC_ReadSysParameter - Read parameters

Reading parameters with FB 406 "MC_ReadSysParameter"

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

Purpose

- The MC_ReadSysParameter technology function returns the value of the most important system variables and configuration data of the TOs, for example, axis or output cam data. The system variables and configuration data are addressed by setting a parameter number
- The result is transferred to the corresponding output parameter, depending on the data type of parameters read. The other outputs return a *0* value.
- Input parameter *Index* can be used to access multiple instances of the parameters Multiple parameters exist in these areas:
  - Data sets (*Dataset_1* to *Dataset_16*)
  - Encoders (*Encoder_1* to *Encoder_8*)
  - Synchronous objects (*1* for synchronous object, *2* for superimposed synchronous object)

Supported for

- System variable and configuration data

Requirement

- The parameter defined by the parameter number and index must be configured in S7T Config
- Check the validity of the encoder values before you read the encoder parameters. You can validate the values of the currently active encoder by reading the *Statusword.EncoderValid* and *Statusword.CyclicInterface* variables at the technology DB of the axis or external encoder. In cases where axes are assigned several data records, the status of the other encoder can be requested by calling the "MC_ReadSysParameter" technology function with parameter number *4050*.

Overriding commands

MC_ReadSysParameter commands can not be canceled by any other command.

MC_ReadSysParameter commands do not cancel any other commands.

Input parameters

Parameter	Data type	Initial value	Description
Axis	INT	*0*	Number of the technology DB
Execute	BOOL	*FALSE*	Request of a parameter value at the positive edge
ParameterNumber	INT	*0*	Number of the parameter to read
Index	INT	*1*	Multiple parameter number (*2* for *Datensatz_2*, for example)

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Depending on the parameter data type, the requested parameter value is available at the outputs *Value*, *ValueDInt,ValueDword1, ValueDword2* or *ValueBool.*
Busy	BOOL	*FALSE*	*TRUE*: Command in process
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is rejected. *FALSE*: Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 695) of the *Error* output parameter.
Value	REAL	*0.0*	Value of the specified parameter, if this is of the type REAL (otherwise *0.0*).
ValueDInt	DINT	*0*	Value of the specified parameter, if of the type DINT (otherwise *0*). The ENUM values are also output at this parameter.
ValueDword1	DWORD	*0*	Value of the specified parameter, if this is of the type DWORD and it is has a length of *1* DWORD (otherwise *0*).
ValueDword2	DWORD	*0*	Value of the specified parameter, if this is of the type DWORD and it is has a length of *2* DWORD (otherwise *0*).
ValueBool	BOOL	*FALSE*	Value of the specified parameter, if it is of the BOOL type (otherwise *FALSE*).

MC_ReadSysParameter - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
8043	Illegal parameter value	Concerns input parameter *ParameterNumber* or *Index*..
8044	Command not supported by the technology object	Sending a command to MCDevice DB, for example
8045	Command not allowed in current state	Possible causes: Synchronization axis was deactivated with the "MC_ActivateTO" technology function
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	Error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8080	Invalid parameter number or parameter index	The technology object does not support this parameter or parameter index.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
8086	Parameter not available	Possible causes: • Access to incremental encoder parameters in a system with absolute value encoders • Access to parameters of an encoder of the type: "Absolute encoder cyclic absolute" • The data record defined in the Index is not active. The data record must be changed by calling "MC_ChangeDataset" • The technology object is currently performing a restart. It is therefore not possible to read the parameter.

6.7.5 FB407 MC_WriteParameter - Changing parameters

Changing parameters with FB 407 "MC_WriteParameter"

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

Purpose

- You can use the "MC_WriteParameter" technology function to edit the most important system variable and configuration data of the TOs while the technology CPU is in RUN. The system variables and configuration data are addressed by setting a parameter number The change is active until the next POWER OFF / POWER ON cycle, or until the CPU performs a memory reset. Exception: the value set during absolute encoder adjustment is retained even under those conditions.
- Of the input parameters *Value*, *ValueDInt*, *ValueDword1*, *ValueDword2* and *ValueBool*, the system evaluates only the one that conforms to the relevant technology parameter (for details, refer to the List of technology parameters).
- The technology function can only modify parameters identified in the list of technology parameters by access mode "W" ("Write".)
- Input parameter *Index* can be used to access multiple instances of the parameters Multiple parameters exist in these areas:
  - Data sets (Dataset_1 to Dataset_16)
  - Encoders (Encoder_1 to Encoder_8)
  - Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object)

> **CAUTION**
>
> The use of this technology function is intended for experienced users. Wrong input may result in uncontrolled response of the drive!
>
> The system does not check the plausibility of your entries!
> Incorrect entries may lead to inconsistency of the technology object.

Supported for

- Parameters of the technology objects you configured in S7T Config .
  Note that certain parameter changes require a system restart (for information, refer to the list of technology parameters in the "Activation" column.)

Prerequisites

- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."
- Validate the encoder values before you write the encoder parameters (absolute encoder adjustment, for example). You can query the encoder values of the currently active encoder and check their validity by means of the *Statusword.EncoderValid* variable at the technology DB of the axis or external encoder. In cases where axes are assigned several data records, the status of the other encoder can be requested by calling the "MC_ReadSysParameter" technology function with parameter number *4050*.

Overriding commands

MC_WriteParameter commands can not be canceled by any other command.

MC_WriteParameter commands do not cancel any other commands.

Input parameters

Parameter	Data type	Initial value	Description
Axis	INT	*0*	Number of the technology DB
Execute	BOOL	*FALSE*	Write parameters at a positive edge e at *Execute*
ParameterNumber	INT	*0*	Number of the technology parameter
Index	INT	*1*	Parameter field number, for example, *2* for dataset_2 (the value is ignored if only one dataset number exists).
Value	REAL	*0.0*	New value of the specified parameter if it is of the type REAL (ignored otherwise).
ValueDInt	DINT	*0*	New value of the specified parameter, if it is of the type DINT (the value is ignored otherwise.)
ValueDword1	DWORD	*0*	New value of the defined parameter, if it is of the type DWORD and if it has a length of *1* DWORD (the value is ignored otherwise.)
ValueDword2	DWORD	*0*	New value of the defined parameter, if it is of the type DWORD and if it has a length of *2* DWORD (the value is ignored otherwise.)
ValueBool	BOOL	*0*	New value of the specified parameter, if it is of the type BOOL (the value is ignored otherwise.)

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Technology parameter successfully changed
Busy	BOOL	*FALSE*	*TRUE*: Command in process
Error	BOOL	*FALSE*	*TRUE*: Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*. *FALSE*: Command initiation **without error**.
ErrorID	WORD	0	ErrorID (Page 699) of the *Error* output parameter.

NOTICE

Observe the valid range limits of the parameters. The valid range limits are listed in the list of technology parameters.

When editing data record parameters, note that certain parameters must be identical in the data records:

- Parameters changing the structure (different controller types such as PV or PID controllers, for example)
- Important activating parameters (following monitoring on/off, DSC on/off, for example)

The system reports a configuration error if you assign invalid values to data record parameters in runtime by means of the technology function "MC_WriteParameter". You should therefore edit only the data record parameters you can compile as consistent project when you configure these in S7T Config.

Note

To change several parameters which do not require a restart and come into effect immediately in one pass, you should collect all parameter changes and then activate these.

To do this, proceed as follows:

1. Use "MC_WriteParameter" to set the technology parameter *ParameterNumber = 4001* (*activationmodechangedconfigdata*) to *COLLECT_CHANGED_CONFIG_DATA (82).*.

   This forms a collection of all subsequent changes at the relevant technology object.
2. Use "MC_WriteParameter" to change the relevant technology parameters.
3. Use "MC_WriteParameter" to set *activationmodechangedconfigdata* to *ACTIVATE_CHANGED_CONFIG_DATA (81)* to apply all changes.

This procedure is called for when editing technology parameters requiring identical values in all data records.

MC_WriteParameter - Example - "Override"

Example of changing parameters with "Override"

Factors can be superimposed online on the current traversing velocity or the acceleration / deceleration. The velocity override is applied to the velocity, and the acceleration override is applied to the acceleration and deceleration.

Set the following input parameter values, in order to set **velocity** override:

ParameterNumber = 4142

Value = 0.0 to *200.0 (*as a percent value)

Set the following input parameter values for **acceleration / deceleration** override:

ParameterNumber = 4141

Value = 1.0 to *1000.0 (*as a percent value)

MC_WriteParameter - Example - "Drive control word"

Drives coupled via PROFIBUS communicate by means of standard message frames. These message frames contain up to two drive control words. The content of these words is specified in PROFIdrive standards.

Structure of control word CW1			Structure of control word CW2		
Bit	Reserved for the technology object	PROFIdrive V3.1 definition	Bit	Reserved for the technology object	PROFIdrive V3.1 definition
0	X	Drive ON (not OFF1)	*0*		free
1	X	Operating condition (not OFF2)	*1*		Free
2	X	Operating condition (not OFF3)	*2*		Free
3	X	Rectifier enable	*3*		Free
4	X	Ramp-up generator enabled	*4*		Free
5	X	Start ramp-function generator	*5*		Free
6	X	Setpoint enable	*6*		Free
7	X	Reset error memory	*7*		Free
8		Step distance 1 (optional)	*8*		Free
9		Step distance 2 (optional)	*9*		Free
10	X	Control required	*10*		Free
11		Free	*11*		Free
12		Free	*12*	X	Master sign of life
13		Free	*13*	X	Master sign of life
14		Free	*14*	X	Master sign of life
15		Free	*15*	X	Master sign of life

Drive control words can be written by setting parameter number *4500* at "MC_WriteParameter", or be read by calling "MC_ReadSysParameter." Both control words are available for reading at output parameter *ValueDword1* (unmasked).

The control word is written using the corresponding value and mask. The mask determines which bits within the control word can be accessed by the write operation. This mask can be used to manipulate single or several bits. Unmasked bits (value *0*) retain their current state. Criterion for selecting the mask is that the bits reserved for the technology object remain unchanged.

The values for control words *CW1* and *CW2* are defined at the input parameter *ValueDword1* . The corresponding mask values for control words *CW1* and *CW2* are set at the input parameter *ValueDword2* .

Example of *ValueDword1*

DW#16#	*0001*	*0300*
	CW2	*CW1*

Example of *ValueDword2*

DW#16#	*0FFF*	*FB00*
	Mask *CW2*	Mask *CW1*

Set mask value *FB00* for control word *CW1* if you want to overwrite all free bits of the control word. Meaning of the hexadecimal coding:

F = *1111* / Bit 12 - 15 are written
B = *1011* / Bit 8, 9 and 11 are written; Bit 10 is protected
0 = *0000* / Bit 4 - 7 are protected
0 = 0000 / Bit 0 - 3 are protected

Set mask value *0FFF* for control word *CW1* if you want to overwrite all free bits of the control word. Meaning of the hexadecimal coding:

0 = *0000* / Bit 12 - 15 are protected
F = *1111* / Bit 8 - 11 are written
F = *1111* / Bit 4 - 7 are written
F = 1111 / Bit 0 - 3 are written

MC_WriteParameter - Example - "Backup of absolute encoder adjustment data"

Editing parameters, based on the example "Backup of absolute encoder adjustment data"

The data of absolute encoder adjustment are saved to nonvolatile memory of the Technology CPU. These data will therefore be lost when you replace the CPU.

The "MC_ReadSysParameter" and "MC_WriteParameter" technology functions can be used to back up the absolute encoder adjustment data to a Micro Memory Card, and to restore these to the nonvolatile memory of your Technology CPU following a CPU replacement. Prerequisite is that you have set the "absolute encoder" type in S7T Config.

Once you have completed absolute encoder adjustment using the "MC_Home" technology function, proceed as described below:

Backing up absolute encoder adjustment data

Step	Description
1.	Read the current value of absolute encoder adjustment by calling the "MC_ReadSysParameter" technology function and reading the input parameters described below, for example: • *Axis* = Technology DB of the relevant axis • *Execute* = rising edge to start the read operation • *ParameterNumber* = *4010* parameter number of absolute encoder adjustment • *ValueDword1= DBx.DBD0* (temporary storage location) • *ValueDword2 = DBx.DBD4* (temporary storage location)
2.	Call SFC82 "CREA_DBL" to generate a DBy in load memory of the Technology CPU. The DB is also generated in work memory when you reset *0* at input parameter *ATTRIB* .
3.	Call SFC84 "WRIT_DBL" to write the previously backed up data of the absolute encoder adjustment to the new DB.

Replacing the CPU and restoring absolute encoder adjustment data

Step	Description
1.	Replace the Technology CPU, and then insert the MMC which contains the configuration data. Note that the backup DB for absolute encoder adjustment is available in the RAM after startup of the technology CPU.
2.	Call the "MC_WriteParameter" technology function to restore the data of absolute encoder adjustment to the nonvolatile memory of your technology CPU. Use these input parameters: • *Axis* = Technology DB of the relevant axis • *Execute* = rising edge to start the write operation • *ParameterNumber= 4010* = parameter number of absolute encoder adjustment • *ValueDword1* = DBx.DBD0 (x = number of the DB generated by SFC82) • *ValueDword2* = DBx.DBD4 (x = number of the DB generated by SFC82)

MC_WriteParameter - ErrorIDs

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

ErrorID	Warning	Description / to correct or avoid errors
0000	No error	-
0020	Restart to activate the parameter changes	A parameter was changed, and normally requires a restart of the technology object. The restart is not required if the new parameter value corresponds to the old parameter value (for example, the parameter value "5" is modified to "5"). The information whether a restart is required is contained in the technology data block. • For axes and external encoders, the *RequestRestart* bit is set in the status word. • Restarts required of output cams and measuring inputs are indicated by ErrorID *0020* of the technology data block.

ErrorID	Error message	Description / to correct or avoid errors
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.

ErrorID	Error message	Description / to correct or avoid errors
8043	Illegal parameter value	Concerns all input parameters of data type REAL, or the input parameters *Value*, *ValueDInt, ValueDword1* or *ValueDword2.* The error can also occur if there has been any interaction with other parameters. Examples: • Parameter *1100* The axis can no longer reach maximum velocity if you reduce the leadscrew pitch (parameter *1100*.) Adjust the maximum velocity as well (parameter *8*.) • Parameters *2020* and *2021* The axis may no longer be able to reach maximum velocity if you change the gear factor (parameter *2020* or *2021*.) Adjust the maximum velocity as well (parameter *8*.) • Parameter *8* If the maximum velocity (parameter *8)* is less than the minimum velocity for dynamic following error monitoring, increase the minimum velocity for dynamic following error monitoring (parameter *2072*.) • Invalid value for the selected *ENUM*
8044	Command not supported by the technology object	Sending a command to MCDevice DB, for example
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx in both OB 1 and OB 35. Execution of the technology function started in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.

ErrorID	Error message	Description / to correct or avoid errors
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8080	Invalid parameter number or parameter index	The technology object does not support this parameter or parameter index.
8081	Parameter cannot be changed	The selected parameter is read-only. Check the selected parameter number and the ID of the technology object.
8083	DB is not a technology DB	The DB specified at input parameter *Axis* was not found or is not a technology DB.
8084	Invalid technology DB	• A technology object does not exist in the controller for the technology DB defined at input parameter *Axis*. Download the current technology to the target system, or change the DB number at input parameter *Axis*. • The user has entered invalid data at the technology DB defined at input parameter *Axis*. Delete the technology DB in "Technology Objects Management" and then create a new one.
8086	Parameter not available	Possible causes: • Access to incremental encoder parameters in a system with absolute value encoders • Access to parameters of an encoder of the type: "Absolute encoder cyclic absolute" • The data record defined in the Index is not active. The data record must be changed by calling "MC_ChangeDataset." • The technology object is currently performing a restart. It is therefore not possible to read the parameter.
808B	Parameter value of invalid REAL format	The value at an input parameter of the data type REAL does not correspond to the valid floating formats. Check the input parameter values or the instance DB data. Values in an invalid format cannot be represented in floating-point format. They are shown in hexadecimal format (DW16# ...).

6.7.6 FB450 MC_ReadPeriphery - Reading technology I/O

Reading technology I/O with FB 450 "MC_ReadPeriphery"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_ReadPeriphery" technology function is used to read the I/O image of integrated technology. The time-based dependency of the update is shown in the example.

Supported for

- DP I/O on DP(DRIVE) at address area *0* to *63*
- Integrated I/O of the technology CPU at address area *0* to *63*

Prerequisites

- The addressed I/O must exist in the I/O image DP(DRIVE) of the integrated technology.

Overriding commands

MC_ReadPeriphery commands can not be canceled by any other command.

MC_ReadPeriphery commands do not cancel any other commands.

Input parameters

Parameter	Data type	Initial value	Description
Enable	BOOL	*FALSE*	An area of the I/O image is read as long as *Enable = TRUE*. Changes at the input parameters *InputOutput*, *ByteAddress* and *Data* are only activated with a rising edge at *Enable*.
InputOutput	BOOL	*FALSE*	Definition of the I/O area to be read: *FALSE*: I/O image of inputs *TRUE*: I/O image of outputs
ByteAddress	INT	*0*	Byte start address from where reading is to start (possible values: *0 ... 63*).
Data	ANY	--	Destination for read data. The length of data to be read is determined by the data type and the repetition factor in the ANY pointer. The maximum length is *64* bytes.

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
DataValid	BOOL	*FALSE*	*TRUE*: Valid read data available (set with the first successful read operation)	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 708) of the *Error* output parameter.	

Note

Due to the runtime of command execution at the integrated technology and the runtime of the OBx cycle in which "MC_ReadPeriphery" is called, the response times can be quite long. You should therefore connect the I/O requiring quick reactions to the MPI/DP interface of your technology CPU.

MC_ReadPeriphery - example

The example in the diagram below shows the time-based coherency when reading or loading the I/O image DP(DRIVE).

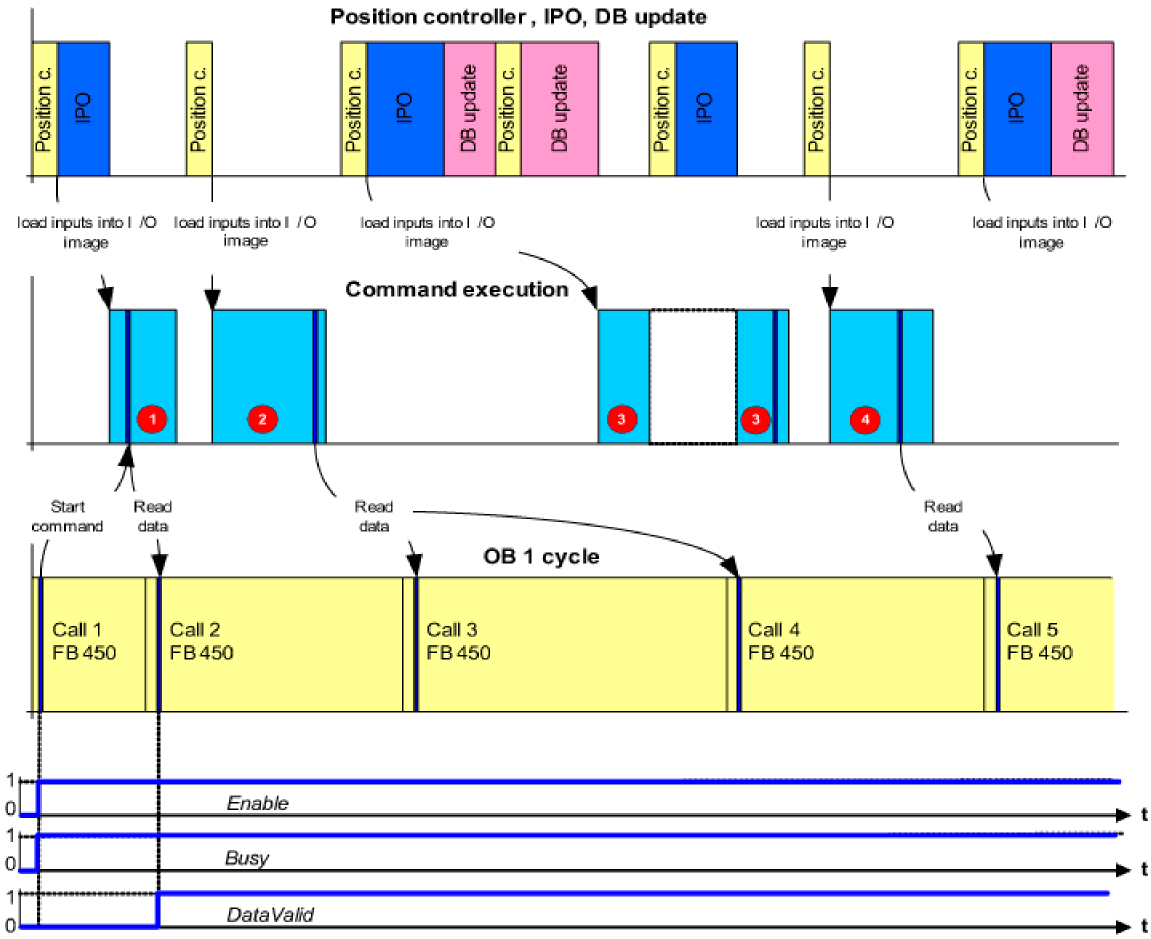
At the start of each command execution cycle, the signal status of inputs is loaded from the position controller and written to the I/O image of inputs. This load operation is independent of the "MC_ReadPeriphery" technology function. The "MC_ReadPeriphery" technology function accesses this I/O image and provides the data to the user program.

The I/O image is transferred to the controller, starting at the first call of MC_ReadPeriphery with *Enable = TRUE*. Output parameter *Busy* is set to *TRUE* simultaneously. The MC_ReadPeriphery command is executed as a new command to the time slice of the first command execution cycle. New commands are executed at the start of the command execution. The I/O input image is read into the controller inputs at the next call of MC_ReadPeriphery. Output parameter *DataValid* reports *TRUE* status.

The current MC_ReadPeriphery command is monitored starting at the second command execution cycle. Current commands are processed at the end of command execution, whereby the most recent command is executed last. The I/O input image of the second command execution cycle is transferred to the controller at the third call of MC_ReadPeriphery.

The DB update is triggered by the subsequent IPO cycle. This is not concluded until the next position control cycle is completed, and is interrupted by the position control. The time slice for the third command execution is available after the DB update. Due to the delayed start of command execution, this is interrupted by the next position control and by the subsequent IPO. An updated I/O image is, therefore, not available at the time of the fourth call of MC_ReadPeriphery. Hence, the same I/O input image as in Call 3 is transferred to the controller.

The I/O image of the fourth call is available at the fifth call of MC_ReadPeriphery and is transferred to the controller. The controller does not read the I/O image of the third command execution cycle.


Position controller , IPO, DB update
Position c.
IPO
Position c.
Position c.
IPO
DB update
Position c.
DB update
Position c.
IPO
Position c.
Position c.
IPO
DB update
load inputs into I /O image
load inputs into I /O image
load inputs into I /O image
load inputs into I /O image
load inputs into I /O image
Command execution
1
2
3
3
4
Start command
Read data
Read data
Read data
OB 1 cycle
Call 1 FB 450
Call 2 FB 450
Call 3 FB 450
Call 4 FB 450
Call 5 FB 450
1
0
Enable
t
1
0
Busy
t
1
0
DataValid
t

MC_ReadPeriphery - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the size of the I/O range to be read.
8043	Illegal parameter value	Relates to input parameter *ByteAddress.*
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.

ErrorID	error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute* / *Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB.	Faulty instance DB of the technology function (wrong length, for example).
8086	Parameter not available	Currently unable to execute the read / write command; repeat the command.
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data length (ANY pointer)	Specified data length is invalid. Possible causes: • The length exceeds 64 bytes. • The specified length plus the start address defines an address outside of the I/O image DP(DRIVE).
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL
808D	Data record length out of range	An attempt was made to transfer a data record of a length exceeding 240 bytes. Reduce the length definition in the ANY pointer.

6.7.7 FB451 MC_WritePeriphery - Writing technology I/O

Writing technology I/O with FB 451 "MC_WritePeriphery"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_WritePeriphery" technology function is used to write to the I/O image of integrated technology. The time-based dependency of the update is shown in the example.

Supported for

- DP I/O on DP(DRIVE) at address area *0* to *63*
- Integrated I/O of the technology CPU at address area *0* to *63*

Prerequisites

- The addressed I/O must exist in the I/O image DP(DRIVE) of the integrated technology.

Overriding commands

MC_WritePeriphery commands can not be canceled by any other command.

MC_WritePeriphery commands do not cancel any other commands.

Input parameters

Parameters	Data type	Initial value	Description
Execute	BOOL	*FALSE*	Start of a single write command at the positive edge
ByteAddress	INT	*0*	Byte start address from where reading is to start (possible values: *0 ... 63*)
Data	ANY	--	Source area of data to be written. The length of data to be written is defined by the data type and the repetition factor in the ANY pointer. The maximum length is 64 bytes.
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command successfully completed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command initiation **with error**. The command is not executed. For information about the cause, refer to the *ErrorID*.
			FALSE:	Command initiation **without error**.
ErrorID	WORD	*0*	ErrorID (Page 714) of the *Error* output parameter.	

Note

Due to the runtime of command execution at the integrated technology and the runtime of the OBx cycle in which "MC_WritePeriphery" is called, the response times can be quite long. You should therefore connect any I/O requiring quick write access to the MPI/DP interface of your technology CPU.

To write to the I/O image DP(DRIVE) cyclically, always retrigger input parameter *Execute* after each write operation.

To do so, link input parameter *Execute* with the inverted signals of the output parameters *Busy* and *Error*.

MC_WritePeriphery - example

The example shown in the diagram shows the time-based coherency when writing the I/O image DP(DRIVE).

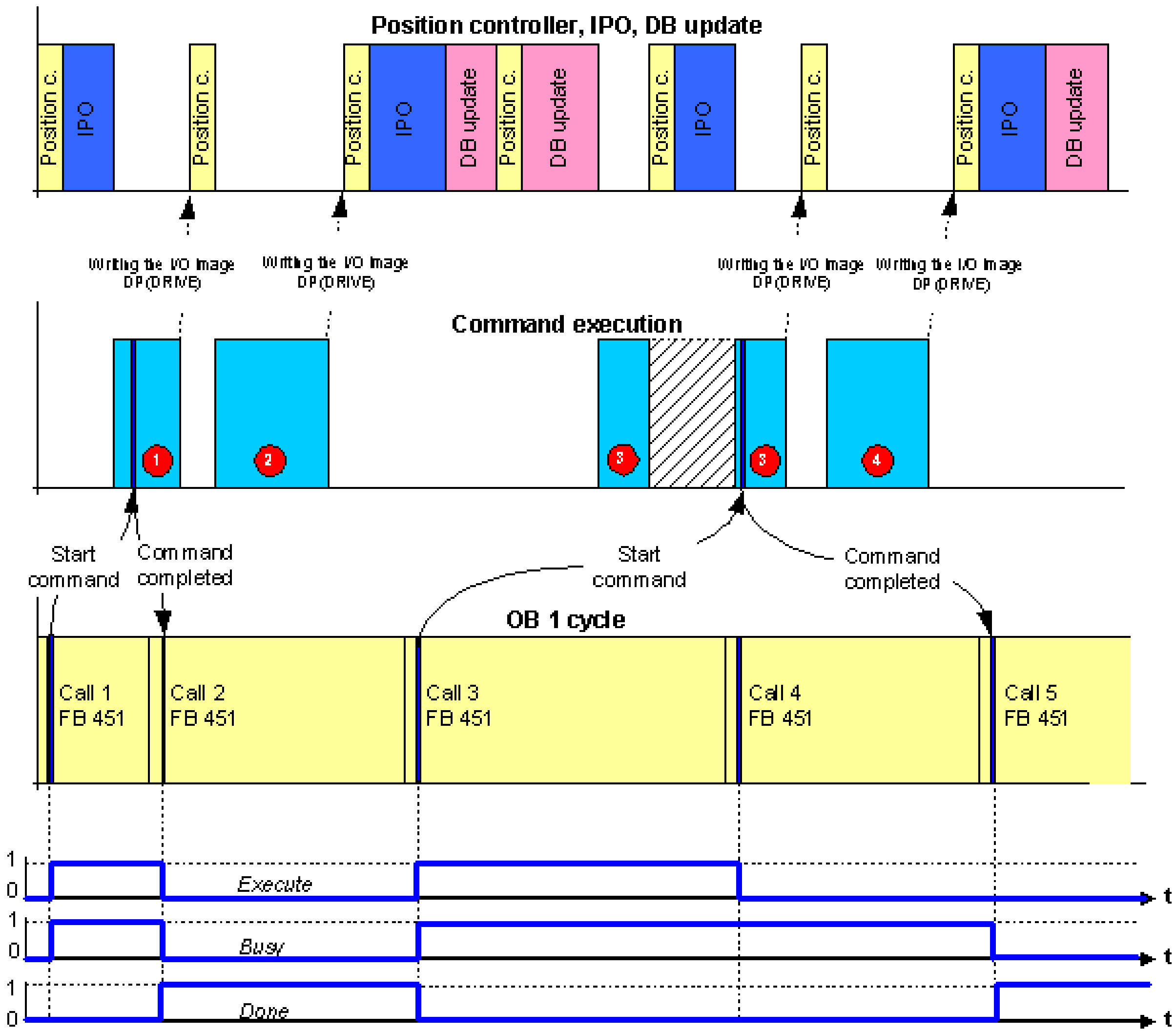
The I/O image of outputs is written to the I/O via the position controller at the end of each time slice for command execution. This write operation is carried out regardless of the run state of the "MC_WritePeriphery" technology function. "MC_WritePeriphery" is used to write the controller data specified at input parameter *Data* to the I/O output image.

A rising edge at input parameter *Execute;* triggers write access to the I/O output image, and at the same time, the *Busy* output parameter is set to *TRUE*. The MC_WritePeriphery command is added as a new command to the time slice of the first command execution cycle. Command execution is started within the first command execution time slice. The second MC_WritePeripherycall reports completion of this command (sets *Busy* to *FALSE* and *Done* to *TRUE*).

At the end of the second command execution time slice, the most recently written I/O image of outputs is once again written to the I/O using the position controller.

The DB update is triggered in the next position controller cycle. The DB update is interrupted within the time slice of the position controller, but is resumed on expiration of this time slice. The start of command execution is delayed due to the missing time gaps. The delayed command execution cycle is interrupted by the next position controller time slice and by the subsequent IPO time slice.

A third call of MC_WritePeriphery plus a rising edge at input parameter *Execute* starts a new write command. Because of the fact that further commands were started, the new write command is queued in the second half of the command execution time slice. For this reason, the completion of the command is not immediately reported at the fourth call of MC_WritePeriphery, but rather at the fifth call of MC_WritePeriphery (now, *Busy* changes to *FALSE* and *Done* changes to *TRUE*).


Position controller, IPO, DB update
Position c.
IPO
DB update
Writing the I/O image DP(DRIVE)
Command execution
Start command
Command completed
OB 1 cycle
Call 1 FB 451
Call 2 FB 451
Call 3 FB 451
Call 4 FB 451
Call 5 FB 451
Execute
Busy
Done
t

MC_WritePeriphery - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the size of the I/O range to be written.
8043	Illegal parameter value	Concerns input parameter *ByteAddress* or *DoneFlag*.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8086	Parameter not available	Currently unable to execute the read / write command; repeat the command.
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data length (ANY pointer)	Specified data length is invalid. Possible causes: • The length exceeds 64 bytes • The specified length plus the start address defines an address outside of the I/O image DP(DRIVE).
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL
808D	Data record length out of range	An attempt was made to transfer a data record of a length exceeding 240 bytes. Reduce the length definition in the ANY pointer.

6.7.8 FB453 MC_ReadRecord - Reading data records

Reading data records with FB 453 "MC_ReadRecord"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_ReadRecord" technology function can be used to read a data record from an I/O module on DP(DRIVE)..
- The technology function can be compared with the system function SFC59 "RD_REC" which is used to read data records at the DP interface of an S7 CPU.

Supported for

- DP I/O on DP(DRIVE)

Prerequisites

- The I/O must support data record communication.
- The I/O module must support the relevant data record.

Overriding commands

MC_ReadRecord commands can not be canceled by any other command.

MC_ReadRecord commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description	
Execute	BOOL	*FALSE*	Start of the command at the positive edge	
InOut	BOOL	*FALSE*	I/O assignment of the logical base address of the I/O module	
			FALSE:	Input address
			TRUE:	Output address
Address	INT	*0*	Logical base address of the I/O module.	
RecordNumber	INT	*0*	Number of the data record of the I/O module (possible values *0* to *255*)	
Data	ANY	-	Destination for read data. The maximum length of data to be read is defined by the data type and by the repetition coefficient at the ANY pointer.	
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Command successfully completed
Busy	BOOL	*FALSE*	*TRUE*: Command in process
Error	BOOL	*FALSE*	*TRUE*: Command execution **with error**. Command execution failed; refer to the *ErrorID* for information about the cause.
			FALSE: Command execution **without error**.
ErrorID	WORD	*0*	ErrorID (Page 719) of the *Error* output parameter.
DataLength	INT	*0*	Length of read data record information in bytes

MC_ReadRecord - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process.	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the size of the data area to be read.

ErrorID	error message	Description / to correct or avoid errors
8043	Illegal parameter value	Concerns input parameter *RecordNumber* or *DoneFlag*.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data length (ANY pointer)	The data length specified at the ANY pointer is shorter than the data record length.
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL
808D	Data record length out of range	An attempt was made to transfer a data record of a length exceeding 240 bytes. Reduce the length definition in the ANY pointer.

ErrorID	error message	Description / to correct or avoid errors
8090	Invalid logical base address	• Error during data set transfer • Job aborted • The specified logical base address is invalid: No assignment in SDB1/SDB2x exists, or it is not a base address.
8091	Logical base address is not available	Error in data record transfer, command canceled. The function cannot access the specified logical base address. The error occurs when an attempt is made to establish data record communication with the integrated I/O of the Technology CPU.
8092	Faulty response ID	Error in data record transfer, command canceled. Error in response identifier.
80A0	Error when reading module data	Error in data record transfer, command canceled. Negative acknowledgment when reading module data: • Module was removed during the read operation • Defective module
80A2	DP protocol error in layer 2	Error in data record transfer, retry possible in next program cycle. PROFIBUS DP error at Layer 2: • Station failure • Timeout • Protocol error • Bus error
80A3	DP protocol error in user interface	Error in data record transfer, command canceled. PROFIBUS DP protocol error in user interface/user.
80B0	Data record not supported	Error in data record transfer, command canceled. • System function not supported for this module type. • Module does not recognize the data record • Data record number out of the range from 0 to 255
80B1	Faulty length definition	Error in data record transfer, command was canceled. Wrong length definition at input parameter Data. Examples: • The data record read 240 bytes, but the ANY pointer only points to a field with a length of 80 bytes. • The length definition in the ANY pointer is not supported, for example, 260 bytes.
80C0	No read data	Error in data record transfer, retry possible in next program cycle. The module maintains the data record, but read data do not exist yet.
80C2	Maximum number of commands reached	Error in data record transfer. Immediate command repetition possible. The module is currently executing the maximum possible number of commands for a CPU.
80C3	Module at capacity limits	Error in data record transfer. Immediate command repetition possible. Required resources are currently in use by another application: • In the module

ErrorID	error message	Description / to correct or avoid errors
80C4	Communication error	Error in data record transfer. Immediate command repetition possible. Communication error: • Parity error • SW ready not set • Error in block length management • Checksum error on CPU side • Checksum error on module side
80C5	Access to distributed I/O failed	Error in data record transfer, retry possible in next program cycle. Distributed I/O currently unavailable.
80C6	Priority class error	Error in data record transfer, retry possible in next program cycle. Data record transfer canceled because of priority class cancellation (restart or background).

Note

A DP slave can report the errors *A0* to *CF* in accordance with PROFIBUS DP V1. The errors *A0* to *CF* are mapped to the *ErrorID* of the technology function as errors *80A0* to *80CF*.

Consult the DP slave documentation if output parameter *ErrorID* returns errors within the range *80A0* to *80CF* that are not defined in the list shown above.

6.7.9 FB454 MC_WriteRecord - Writing data records

Writing data records with FB 454 "MC_WriteRecord"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_WriteRecord" technology function can be used to write a data record to an I/O module on DP(DRIVE).
- The technology function can be compared with system function SFC58 "WR_REC“ which is used to write data records at the DP interface of an S7 CPU.

Supported for

- DP I/O on DP(DRIVE)

Prerequisites

- The I/O must support data record communication.
- The I/O module must support the relevant data record.

Overriding commands

MC_WriteRecord commands can not be canceled by any other command.

MC_WriteRecord commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description
Execute	BOOL	*FALSE*	Start of the command at the positive edge
InOut	BOOL	*FALSE*	I/O assignment of the logical base address of the I/O module *FALSE*: Input address *TRUE*: Output address
Address	INT	*0*	Logical base address of the I/O module.
RecordNumber	INT	*0*	Number of the data record of the I/O module (possible values *0* to *255*)
Data	ANY	-	Source area of data to be written. The length of data to be written is defined by the data type and the repetition factor in the ANY pointer.
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command successfully completed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command execution **with error**. Command execution failed; refer to the *ErrorID* for information about the cause.
			FALSE:	Command execution **without error**.
ErrorID	WORD	*0*	ErrorID (Page 724) of the *Error* output parameter.	

MC_WriteRecord - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the size of the data area to be written.
8043	Illegal parameter value	Concerns input parameter *RecordNumber* or *DoneFlag*.

ErrorID	Error message	Description / to correct or avoid errors
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and was interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data length (ANY pointer)	The data length specified at the ANY pointer is shorter than the data record length.
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL

ErrorID	Error message	Description / to correct or avoid errors
808D	Data record length out of range	An attempt was made to transfer a data record of a length exceeding 240 bytes. Correct the length definition in the ANY pointer.
8090	Invalid logical base address	Error in data record transfer, command canceled. Specified logical base address invalid: No assignment in SDB1/SDB2x exists, or it is not a base address.
8091	Logical base address is not available	Error in data record transfer, command canceled. The function cannot access the specified logical base address. The error occurs when an attempt is made to establish data record communication with the integrated I/O of the technology CPU.
8092	Faulty response ID	Error in data record transfer, command canceled. Error in response identifier.
80A1	Error when writing to module	Error in data record transfer, command canceled. Negative acknowledgment when writing to module: • Module removed during write operation • Defective module
80A2	DP protocol error in layer 2	Error in data record transfer, retry possible in next program cycle. PROFIBUS DP error at Layer 2: • Station failure • Timeout • Protocol error • Bus error
80A3	DP protocol error in user interface	Error in data record transfer, command canceled. PROFIBUS DP protocol error in user interface/user.
80B0	Data record not supported	Error in data record transfer, command canceled. • System function not supported for this module type. • Module does not recognize the data record • Data record number out of the range from 0 to 255
80B1	Faulty length definition	Error in data record transfer, command canceled. Wrong length definition at input parameter *Data*. • The data record read 240 bytes, but the ANY pointer only points to a field with a length of 80 bytes. • The length definition in the ANY pointer is not supported, for example, 260 bytes.
80C0	Previous write command is still busy	Error in data record transfer. Immediate command repetition possible. The module has not yet processed the data of the previous write command to the same data record.
80C2	Maximum number of commands reached	Error in data record transfer. Immediate command repetition possible. The module is currently executing the maximum possible number of commands for a CPU.

ErrorID	Error message	Description / to correct or avoid errors
80C3	Module at capacity limits	Error in data record transfer. Immediate command repetition possible. Required resources are currently in use by another application: • In the module
80C4	Communication error	Error in data record transfer. Immediate command repetition possible. Communication error: • Parity error • SW ready not set • Error in block length management • Checksum error on CPU side • Checksum error on module side
80C5	Access to distributed I/O failed	Error in data record transfer, retry possible in next program cycle. Distributed I/O currently unavailable.
80C6	Priority class error	Error in data record transfer, retry possible in next program cycle. Data record transfer canceled because of priority class cancellation (restart or background).

Note

A DP slave can report the errors *A0* to *CF* in accordance with PROFIBUS DP V1. The errors *A0* to *CF* are mapped to the *ErrorID* of the technology function as errors *80A0* to *80CF*.

Consult the DP slave documentation if output parameter *ErrorID* returns errors within the range *80A0* to *80CF* that are not defined in the list shown above.

6.7.10 FB455 MC_ReadDriveParameter - Reading drive parameters

Reading drive parameters with FB 455 "MC_ReadDriveParameter"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_ReadDriveParameter" technology function allows the user program to read the parameters of a drive connected to DP(DRIVE).

Supported for

- PROFIdrive-compliant drives on DP(DRIVE)

Prerequisites

- The drive is configured for operation on DP(DRIVE).
- The drive must support data record communication.
- The drive must support the parameter number and the index.

Overriding commands

MC_ReadDriveParameter commands can not be canceled by any other command.

MC_ReadDriveParameter commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description
Execute	BOOL	*FALSE*	Start of the command at the positive edge
InOut	BOOL	*FALSE*	I/O assignment of the logical base address of the drive *FALSE*: Input address *TRUE*: Output address
Address	INT	*0*	Definition of the logical base address of the module
ParameterNumber	DINT	*0*	Specifies the number of the parameter from which the data is to be read - Range of values: *0 ... 65535*
SubIndex	DINT	*0*	Specifies the first parameter index to be read (with *NumberOfElements= 0* the value is internally set to zero) - Range: *0 ... 65535*
NumberOfElements	DINT	*0*	Number of sub-parameters to be read: Value > *0*: Parameter with subindex, range *0 to 234* (according to the maximum length of the data area at input parameter *Data*) Value = *0*: Parameter without subindex Value < *0*: Not permitted
Data	ANY	-	Destination data area to which the parameter values should be saved. The maximum length of data to be read is defined by the data type and the repetition coefficient at the ANY pointer. The data area has a maximum size of 240 bytes.
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice DB

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description
Done	BOOL	*FALSE*	*TRUE*: Command successfully completed
Busy	BOOL	*FALSE*	*TRUE*: Command in process
Error	BOOL	*FALSE*	*TRUE*: Command execution **with error**. Command execution failed; refer to the *ErrorID* for information about the cause. *FALSE*: Command execution **without error**.
ErrorID	WORD	*0*	ErrorID (Page 728) of the *Error* output parameter.
DataLength	INT	*0*	Length of read data record information in bytes

MC_ReadDriveParameter - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the number of the sub-parameters to be read.
8043	Illegal parameter value	Relates to input parameter *ParameterNumber, SubIndex, NumberOfElements* or *DoneFlag.*.
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.

ErrorID	error message	Description / to correct or avoid errors
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data length (ANY pointer)	The data length specified at the ANY pointer is shorter than the data record length. The length of the data record is determined by the number and type of parameter values to be read.
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL
808D	Data record length out of range	An attempt was made to transfer a data record of a length exceeding 240 bytes. Reduce the length definition in the ANY pointer.
8090	Invalid logical base address	Error in data record transfer, command canceled. Specified logical base address invalid: No assignment in SDB1/SDB2x exists, or it is not a base address.
8091	Logical base address is not available	Error in data record transfer, command canceled. The function cannot access the specified logical base address. The error occurs when an attempt is made to establish data record communication with the integrated I/O of the Technology CPU.
8092	Faulty response ID	Error in data record transfer, command canceled. Error in response identifier.
80A0	Error when reading module data	Error in data record transfer, command canceled. Negative acknowledgment when reading module data: • Module was removed during the read operation • Defective module

ErrorID	error message	Description / to correct or avoid errors
80A2	DP protocol error in layer 2	Error in data record transfer, retry possible in next program cycle. PROFIBUS DP error at Layer 2: • Station failure • Timeout • Protocol error • Bus error
80B0	Data record not supported	Error in data record transfer, command canceled. • System function not supported for this module type. • Module does not recognize the data record • Data record number out of the range from 0 to 255
80B5	System function cannot be executed	Error in data record transfer, retry possible in next program cycle. The system function cannot be executed due to the internal processing state of the drive.
80B7	Faulty command	Error in data record transfer, PROFIBUS DP error, command canceled. The faulty command could not be output.
80C3	Module at capacity limits	Error in data record transfer, retry possible in next program cycle. Required resources currently in use by other applications: • In technology function "MC_ReadDriveParameter" • In the module
80C5	Access to distributed I/O failed	Error in data record transfer, retry possible in next program cycle. Distributed I/O currently unavailable.
80C7	A command has already been output to the drive	Error in data record transfer, retry possible in next program cycle. Another command has already been output to the drive
8100	Parameter not found	Parameter error, command canceled. Access to a non-existent parameter.
8103	Subindex not found	Parameter error, command canceled. Access to a non-existent subindex.
8104	Parameter not indexed	Parameter error, command canceled. Access with subindex to non-indexed parameter.
8111	Access denied in current state	Parameter error, command canceled. Request cannot be executed due to operating status.
8115	Max. transferable length exceeded	Parameter error, command canceled. The requested number of sub parameters cannot be transferred with a single command. Please note that the number of bytes of the parameter data type multiplied by *NumberOfElements* must not exceed 234.
8116	Value is invalid or not supported	Parameter error, command canceled. Invalid or non-supported value for attribute, number of elements, parameter number, subindex or a combination of these.
8119	Axis does not exist	Parameter error, command canceled. Access to a non-existent axis.
8120 ... 8164	Spare	Reserved error codes to PROFIdrive specification
8165 ... 81FF	Manufacturer-specific error	Manufacturer-specific error codes to PROFIdrive specification. Errors are formed by the drive.

Note

A DP slave can report the errors *A0* to *CF* in accordance with PROFIBUS DP V1. The errors *A0* to *CF* are mapped to the *ErrorID* of the technology function as errors *80A0* to *80CF*.

Consult the DP slave documentation if output parameter *ErrorID* returns errors within the range *80A0* to *80CF* that are not defined in the list shown above.

6.7.11 FB456 MC_WriteDriveParameter - Writing drive parameters

Writing drive parameters with FB 456 "MC_WriteDriveParameter"

Supported by Integrated Technology with firmware V3.1.x or higher

Purpose

- The "MC_WriteDriveParameter" technology function allows the user program to write the parameters of a drive connected to DP(DRIVE).

Supported for

- PROFIdrive-compliant drives on DP(DRIVE)

Prerequisites

- The drive is configured for operation on DP(DRIVE).
- The drive must support data record communication.
- The drive supports the parameter number and the index.

Overriding commands

MC_WriteDriveParameter- commands can not be canceled by any other command.

MC_WriteDriveParameter- commands do not cancel any other commands.

Input parameters

Parameters	Data Type	Initial value	Description	
Execute	BOOL	*FALSE*	Start of the command at a positive edge	
InOut	BOOL	*0*	I/O assignment of the logical base address of the drive	
			FALSE:	Input address
			TRUE:	Output address
Address	INT	*0*	Definition of the logical base address of the module	
ParameterNumber	DINT	*0*	Definition of the parameter number to which data should be written. Range of values: *0 ... 65535*	
SubIndex	DINT	*0*	Specifies the first parameter index to be written (with *NumberOfElements = 0* the value is internally set to zero) - Range: *0 ... 65535*	
NumberOfElements	DINT	*0*	Number of sub-parameters to be written:	
			Value > *0*:	Parameter with subindex, range *1 to 228* (according to the maximum length of the data area at input parameter *Data*)
			Value = *0*:	Parameter without subindex
			Value < *0*:	Not permitted
Data	ANY	--	Data area containing the parameter values. The data length is determined by the parameter data type and the number of parameters to be written. The combination consisting of the data type and repetition coefficient of the ANY pointer must result in the same data length. The data area has a maximum size of 228 bytes.	
DoneFlag	INT	*0*	DoneFlag (Page 750) generation in the MCDevice-DB	

Output parameters (status outputs)

Parameters	Data Type	Initial value	Description	
Done	BOOL	*FALSE*	*TRUE*: Command successfully completed	
Busy	BOOL	*FALSE*	*TRUE*: Command in process	
Error	BOOL	*FALSE*	*TRUE*:	Command execution **with error**. Command execution failed; refer to the *ErrorID* for information about the cause.
			FALSE:	Command execution **without error**.
ErrorID	WORD	*0*	ErrorID (Page 733) of the *Error* output parameter.	

MC_WriteDriveParameter - ErrorIDs

Supported by Integrated Technology with firmware V3.1.x or higher

ErrorID	Error message	Description / to correct or avoid errors
0000	No error	-
8001	Internal error	Faulty or inconsistent project/software.
8005	Command canceled because command memory is in use by another process	The command cannot be executed due to insufficient command capacity. Possible causes: • The number of active commands has exceeded limits. • Too many active commands for the technology functions: "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" "MC_ReadCamTrackData" "MC_WriteCamTrackData" Call the technology functions in the same cycle until one of the output parameters *Done*, *CommandAborted* or *Error* changes to *TRUE*. Verify that the program does not contain unnecessary (redundant) commands.
800B	Transfer buffer cannot be reserved	The transfer buffer for the command is not sufficient. Possible remedies: • The transfer buffer is already occupied by other active technology functions. Prevent the simultaneous execution of the following technology functions: "MC_ReadCamTrackData" "MC_WriteCamTrackData" "MC_CamSectorAdd" "MC_ReadPeriphery" "MC_WritePeriphery" "MC_ReadRecord" "MC_WriteRecord" "MC_ReadDriveParameter" "MC_WriteDriveParameter" • Reduce the number of the sub-parameters to be written.
8043	Illegal parameter value	Relates to input parameter *ParameterNumber*, *SubIndex, NumberOfElements* or *DoneFlag.*
804C	Command output rate too high	The rate at which commands with the same instance DB were output exceeded the capacity of the command interface. The second command is rejected in order not to violate the consistency of the first command. For high command output rates, always use a separate instance DB, or request the command again. Note that although the first accepted command may be active, you may not be able to monitor it at the status outputs.

ErrorID	Error message	Description / to correct or avoid errors
8050	Technology not ready	• During testing with breakpoints (CPU in HOLD), the technology function has output a new command to the technology. The command is rejected. • The command was output in a restart OB.
8052	Block call at different run levels	This technology function was called at different run levels using the same instance DB. Example: Technology function x is called with the instance DBx, both in OB 1 and in OB 35. Execution of the technology function starts in OB 1, and is interrupted by its call in OB 35. Based on the shared use of the instance DB, the error is indicated at output parameter *ErrorID* of both calls. Error responses to be expected: • The new command (rising or falling edge at input parameter *Execute / Enable*) is not transferred to the integrated technology. • The initially started command can not be monitored at the output parameters of the technology function However, the command may still be active in the integrated technology **Notice:** Use different instance DBs at different run levels, or interlock the call of the technology function.
8053	Invalid instance DB	Faulty instance DB of the technology function (wrong length, for example).
8088	Invalid DB (ANY pointer)	The ANY pointer refers to an invalid data area. • Data block not found • Specified start address not found • Data area too short
8089	Invalid data length (ANY pointer)	The data length specified at the ANY pointer is shorter than the data record length. The length of the data record is determined by the number and type of parameter values to be written.
808C	Pointer format is invalid or not supported	The ANY pointer contains invalid definitions. Accepted areas: I, O, M, DB, DI Accepted data types: BYTE, CHAR, WORD, INT, DWORD, DINT, REAL
808D	Data record length out of range	An attempt was made to transfer a data record of a length exceeding 228 bytes. Reduce the length definition in the ANY pointer.
808E	Invalid data format	Error in data record transfer, command canceled. No valid data format found.
8090	Invalid logical base address	Error in data record transfer, command canceled. Specified logical base address invalid: No assignment in SDB1/SDB2x exists, or it is not a base address.
8091	Logical base address is not available	Error in data record transfer, command canceled. The function cannot access the specified logical base address. The error occurs when an attempt is made to establish data record communication with the integrated I/O of the Technology CPU.

ErrorID	Error message	Description / to correct or avoid errors
8092	Faulty response ID	Error in data record transfer, command canceled. Error in response identifier.
80A1	Error when writing to module	Error in data record transfer, command canceled. Negative acknowledgment when writing to module: • Module removed during write operation • Defective module
80A2	DP protocol error in layer 2	Error in data record transfer, retry possible in next program cycle. PROFIBUS DP error at Layer 2: • Station failure • Timeout • Protocol error • Bus error
80B0	Data record not supported	Error in data record transfer, command canceled. • System function not supported for this module type. • Module does not recognize the data record • Data record number out of the range from 0 to 255
80B5	System function cannot be executed	Error in data record transfer, retry possible in next program cycle. The system function cannot be executed due to the internal processing state of the drive.
80B7	Faulty command	PROFIBUS DP error, command canceled. The faulty command could not be output.
80C3	Module at capacity limits	Error in data record transfer, retry possible in next program cycle. Required resources currently in use by other applications: • In technology function "MC_WriteDriveParameter" • In the module
80C5	Access to distributed I/O failed	Error in data record transfer, retry possible in next program cycle. Distributed I/O currently unavailable.
80C7	A command has already been output to the drive	Error in data record transfer, retry possible in next program cycle. Another command has already been output to the drive
8100	Parameter not found	Parameter error, command canceled. Access to a non-existent parameter.
8101	Parameter cannot be changed	Parameter error, command canceled. Attempt to modify a read only parameter.
8102	Parameter limits violated	Parameter error, command canceled. Modification access with value outside value limits.
8103	Subindex not found	Parameter error, command canceled. Access to a non-existent subindex.
8104	Parameter not indexed	Parameter error, command canceled. Access with subindex to non-indexed parameter.
8105	Invalid data type	Parameter error, command canceled. Attempt to modify a value to a value not supported by the data type of the parameter.
8106	Parameter value unequal *0*	Parameter error, command canceled. Attempt to modify with value unequal *0*.
810B	No exclusive access	Parameter error, command canceled. Attempt to modify without exclusive access privileges.
8111	Access denied in current state	Parameter error, command canceled Request cannot be executed due to operating status

ErrorID	Error message	Description / to correct or avoid errors
8114	Illegal parameter value	Parameter error, command canceled. Attempt to modify with value which may not violate value limits, but is still not permissible for reasons of permanent compatibility (existing parameters with defined single values.)
8116	Value is invalid or not supported	Parameter error, command canceled. Invalid or non-supported value for attribute, number of elements, parameter number, subindex or a combination of these.
8117	Invalid format	Parameter error, command canceled. Invalid format
8118	Invalid number of parameter data	Parameter error, command canceled. The number of values in parameter data does not match the number of addresses in the parameter address.
8119	Axis does not exist	Parameter error, command canceled. Access to a non-existent axis.
8120 ... 8164	Spare	Reserved error codes to PROFIdrive specification
8165 ... 81FF	Manufacturer-specific error	Manufacturer-specific error codes to PROFIdrive specification Errors are formed by the drive.

Note

A DP slave can report the errors *A0* to *CF* in accordance with PROFIBUS DP V1. The errors *A0* to *CF* are mapped to the *ErrorID* of the technology function as errors *80A0* to *80CF*.

Consult the DP slave documentation if output parameter *ErrorID* returns errors within the range *80A0* to *80CF* that are not defined in the list shown above.

6.8 Interaction of commands

6.8.1 New command - active single command (1)

The table below shows the reciprocal actions of a new command on active single commands.

new command ↓ / busy single command →		MC_CamIn d	MC_CamIn w	MC_GearIn d	MC_GearIn w	MC_CamOut d	MC_GearOut d	MC_CamInSuperImposed d	MC_CamInSuperImposed w	MC_GearInSuperImposed d	MC_GearInSuperImposed w	MC_CamOutSuperImposed d	MC_GearOutSuperImposed d	MC_Home (Mode=0, 1)	MC_Home (Mode=2-7), Achse freigegeben	MC_Halt	MC_Stop	MC_MoveAbsolute	MC_MoveAdditive	MC_MoveRelative	MC_MoveVelocity	MC_MoveToEndPos (bClamping=True)	MC_MoveToEndPos (bClamping=False)	MC_MoveSuperImposed
MC_CamIn	w	Wa	A1	Wa	A1	Wa	Wa	Wa	A1	Wa	A1	Wa	Wa	Wa	S	Wa	E	Wa	Wa	Wa	Wa	Ad	Wa	Wa
MC_CamIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_GearIn	w	Wa	A1	Wa	A1	Wa	Wa	Wa	A1	Wa	A1	Wa	Wa	Wa	S	Wa	E	Wa	Wa	Wa	Wa	Ad	Wa	Wa
MC_GearIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_CamOut	w	Wa	Wa	Ae	Ae	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_CamOut	d	A1	A1	Ae	Ae	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_GearOut	w	Ae	Ae	Wa	Wa	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_GearOut	d	Ae	Ae	A1	A1	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_Phasing		S	E	S	E	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Ae	E	E	E	E	E	E	E	E	E	E	E
MC_CamInSuperImposed	w	Sw	Sa	Sw	Sa	S	S	Wa	A1	Wa	A1	Wa	Wa	E	E	Sw	E	Sw	Sw	Sw	Sw	Ad	Sw	Sw
MC_CamInSuperImposed	d	S	Sa	S	Sa	S	S	A1	A1	A1	A1	A1	A1	E	E	S	E	S	S	S	S	Ad	S	A1
MC_GearInSuperImposed	w	Sw	Sa	Sw	Sa	S	S	Wa	A1	Wa	A1	Wa	Wa	E	E	W	E	Sw	Sw	Sw	Sw	Ad	Sw	Sw
MC_GearInSuperImposed	d	S	Sa	S	Sa	S	S	A1	A1	A1	A1	A1	A1	E	E	S	E	S	S	S	S	Ad	S	A1
MC_CamOutSuperImposed	w	N	N	N	N	N	N	Wa	Wa	Ae	Ae	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_CamOutSuperImposed	d	N	N	N	N	N	N	A1	A1	Ae	Ae	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_GearOutSuperImposed	w	N	N	N	N	N	N	Ae	Ae	Wa	Wa	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_GearOutSuperImposed	d	N	N	N	N	N	N	Ae	Ae	A1	A1	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_PhasingSuperImposed		Ae	Ae	Ae	Ae	Ae	Ae	S	Ae	S	Ae	Ae	Ae	E	E	E	E	E	E	E	E	E	E	E

d	driving The axis is moving, that is, it is operating in synchronous mode, or is currently being synchronized or desynchronized. • The output parameters of the *InGear* or *InSync* technology function are set in synchronous operation. (System variable of the synchronous object *enablecommand=active).* • The command is active when a synchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *enablecommand=active).* • The output parameters *InGear* and *InSync* of the technology function are *FALSE* when a desynchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is still set (system variable of synchronous object *disablecommand=active)..*	w	waiting The axis is waiting for the (de)synchronization condition. • The command is active in synchronization wait state. The *Statusword.SynchrCommand* variable of the technology DB is not yet set (system variable of synchronous object *enablecommand=waiting_to_start*). • In desynchronization wait state, the *InSync* and *InGear* output parameters of the technology function are still set. In addition, the *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *disablecommand=waiting_to_start*).
A1	Abort 1 The new command is executed. The active command is canceled immediately.	Ad	Abort direction Either the MC_MoveToEndPos- command is aborted or the new command is canceled with an error, depending on the direction of the new command.
Ae	Abort error The new command is canceled, an ErrorID is entered at the technology DB, and execution of the active command continues.	E	Error A new command reports an error at the technology function and is not executed. Execution of the active command is continued.
N	No influence Does not have any direct influence on the active function. Command execution depends on the technology object status.	S	Superimposed The new motion command is superimposed on the current motion command. The active command is not canceled.
Sa	Superimposed abort The new motion command is superimposed on the current motion command. The active command is not canceled. If neither the synchronization nor desynchronization condition of the active command is met, the new command will be canceled when a condition is satisfied.	Sw	Superimposed waiting A new command is executed and waits for the desynchronization condition. The motion initiated by this new command is superimposed on the motion of the active command without canceling the active command.
W	Waiting The new command is queued in memory and is appended in waiting state to the active command. The waiting command is executed after the active command was completed.	Wa	Waiting abort A new command is executed and waits for the synchronization or desynchronization condition. The active command is aborted when either the synchronization or the desynchronization condition of the new command is satisfied.

For axes, the content of the above table is valid for the standard axis configuration (configuration data element *TypeOfAxis.DecodingConfig.transferSuperimposedPosition = TRANSFER_STANDSTILL (0)).*

6.8.2 New command - active single command (2)

The table below shows the reciprocal actions of a new command on active single commands.

new command ↓ / busy single command →	MC_CamIn d	MC_CamIn w	MC_GearIn d	MC_GearIn w	MC_CamOut d	MC_GearOut d	MC_CamInSuperImposed d	MC_CamInSuperImposed w	MC_GearInSuperImposed d	MC_GearInSuperImposed w	MC_CamOutSuperImposed d	MC_GearOutSuperImposed d	MC_Home (*Mode=0, 1*)	MC_Home (*Mode=2-7*), Achse freigegeben	MC_Halt	MC_Stop	MC_MoveAbsolute	MC_MoveAdditive	MC_MoveRelative	MC_MoveVelocity	MC_MoveToEndPos (*nClamping=True*)	MC_MoveToEndPos (*nClamping=False*)	MC_MoveSuperImposed
MC_Home (*Mode=0, 1*)	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	E	E	A1
MC_Home (*Mode=2-7*)	S	S	S	S	S	S	S	S	S	S	S	S	A1	S	S	S	S	S	S	S	S	S	S
MC_Halt	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	E	A1	A1
MC_Stop	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_MoveAbsolute (*Mode=0*)	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_MoveAdditive	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_MoveRelative (*Mode=0*)	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_MoveVelocity (*Mode=0*)	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_MoveAbsolute (*Mode=1, 2*)	E	E	E	E	E	E	E	E	E	E	E	E	W	S	E	E	W	W	W	W	Ad	W	S
MC_MoveRelative (*Mode=1, 2*)	E	E	E	E	E	E	E	E	E	E	E	E	W	S	E	E	W	W	W	W	Ad	W	S
MC_MoveVelocity (*Mode=1*)	E	E	E	E	E	E	E	E	E	E	E	E	W	S	E	E	W	W	W	W	Ad	W	S
MC_MoveSuperImposed	S	Sa	S	Sa	S	S	A1	A1	A1	A1	A1	A1	A1	S	S	E	S	S	S	S	Ad	S	A1
MC_MoveToEndPos	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	E	S	A1	E	A1	A1	A1	A1	Ad	Ad	A1

d	driving The axis is moving, that is, it is operating in synchronous mode, or is currently being synchronized or desynchronized. • The output parameters of the *InGear* or *InSync* technology function are set in synchronous operation. (System variable of the synchronous object *enablecommand=active)*. • The command is active when a synchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *enablecommand=active)*. The output parameters *InGear* and *InSync* of the technology function are *FALSE* when a desynchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is still set (system variable of synchronous object *disablecommand=active)*..	w	waiting The axis is waiting for the (de)synchronization condition. • The command is active in synchronization wait state. The *Statusword.SynchrCommand* variable of the technology DB is not yet set (system variable of synchronous object *enablecommand=waiting_to_start*). In desynchronization wait state, the *InSync* and *InGear* output parameters of the technology function are still set. In addition, the *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *disablecommand=waiting_to_start*).
A1	Abort 1 The new command is executed. The active command is canceled immediately.	Ad	Abort direction Either the MC_MoveToEndPos- command is aborted or the new command is canceled with an error, depending on the direction of the new command.
Ae	Abort error The new command is canceled, an ErrorID is entered at the technology DB, and execution of the active command continues.	E	Error A new command reports an error at the technology function and is not executed. Execution of the active command continues.
N	No influence Does not have any direct influence on the active function. Command execution depends on the technology object status.	S	Superimposed The new motion command is superimposed on the current motion command. The active command is not canceled.
Sa	Superimposed abort The new motion command is superimposed on the active motion command. The active command is not canceled. If neither the synchronization nor desynchronization condition of the active command is met, the new command will be canceled when a condition is satisfied.	W	Waiting The new command is queued in memory and is appended in waiting state to the active command. The waiting command is executed after the active command was completed.

For axes, the content of the above table is valid for the standard axis configuration (configuration data element *TypeOfAxis.DecodingConfig.transferSuperimposedPosition = TRANSFER_STANDSTILL (0)*).

6.8.3 New command - active single command (3)

The table below shows the reciprocal actions of a new command on active single commands.

new command ↓ / busy single command →	MC_Power (*Enable=False*)	MC_Power (*Enable=True*)	MC_Reset (*Restart=False*)	MC_Reset (*Restart=True*)	MC_SetCharacteristic	MC_SetTorqueLimit	MC_ChangeDataset
MC_Power (*Enable=False*)		A1	E			S	S
MC_Power (*Enable=True*)	S	E	S	S		S	S
MC_Reset (*Restart=False*)	S	S	A1	E	S	S	S
MC_Reset (*Restart=True*)	S	E	A1	E	A1	A1	A1
MC_SetCharacteristic					A1		
MC_SetTorqueLimit	S	S	S	A1		A1	S
MC_ChangeDataset				E			A1

	No reciprocal effect, or status not possible.	A1	Abort 1 The new command is executed. The active command is canceled immediately.
E	Error A new command reports an error at the technology function and is not executed. Execution of the active command is continued.	S	Superimposed The new motion command is superimposed on the active motion command. The active command is not canceled.
W	Waiting The new command is queued in memory and is appended in waiting state to the active command. The waiting command is executed after the active command was completed.		

For axes, the content of the above table is valid for the standard axis configuration (configuration data element *TypeOfAxis.DecodingConfig.transferSuperimposedPosition = TRANSFER_STANDSTILL (0)*).

6.8.4 New command - active commands (1)

The table below shows the reciprocal actions of a new command on active commands.

busy single command →

new command ↓

new command		MC_CamIn w + MC_CamOut w	MC_CamIn d + MC_CamIn w	MC_CamIn d + MC_CamOut w	MC_CamIn d + MC_GearIn w	MC_CamOut d + MC_CamIn w	MC_CamOut d + MC_GearIn w	MC_GearIn w + MC_GearOut w	MC_GearIn d + MC_CamIn w	MC_GearIn d + MC_GearIn w	MC_GearIn d + MC_GearOut w	MC_GearOut d + MC_CamIn w	MC_GearOut d + MC_GearIn w	MC_CamIn d + MC_Phasing d	MC_GearIn d + MC_Phasing d
MC_CamIn	w	A1	A2	A2	A2	A2	A2	A1	A2	A2	A2	A2	A2	Wa	Wa
MC_CamIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_GearIn	w	A1	A2	A2	A2	A2	A2	A1	A2	A2	A2	A2	A2	Wa	Wa
MC_GearIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_CamOut	w	A2	A3	A2	A2	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Wa	Ae
MC_CamOut	d	A1	A1	A1	A1	A3	Ae	Ae	A3	Ae	Ae	A3	Ae	A1	Ae
MC_GearOut	w	Ae	Ae	Ae	Ae	Ae	Ae	A2	A2	A3	A2	Ae	Ae	Ae	Wa
MC_GearOut	d	Ae	Ae	Ae	A3	Ae	A3	A1	A1	A1	A1	Ae	A3	Ae	A1
MC_Phasing		Ae	S	S	S	Ae	Ae	Ae	S	S	S	Ae	Ae	A3	A3
MC_CamInSuperImposed	w	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S
MC_CamInSuperImposed	d	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S
MC_GearInSuperImposed	w	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S
MC_GearInSuperImposed	d	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S
MC_CamOutSuperImposed	w	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_CamOutSuperImposed	d	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_GearOutSuperImposed	w	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_GearOutSuperImposed	d	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_PhasingSuperImposed		N	N	N	N	N	N	N	N	N	N	N	N	N	N

new command		MC_CamInSuperImposed w + MC_CamOutSuperImposed w	MC_CamInSuperImposed d + MC_CamInSuperImposed w	MC_CamInSuperImposed d + MC_CamOutSuperImposed w	MC_CamInSuperImposed d + MC_GearInSuperImposed w	MC_CamOutSuperImposed d + MC_CamInSuperImposed w	MC_CamOutSuperImposed d + MC_GearInSuperImposed w	MC_GearInSuperImposed w + MC_GearOutSuperImposed w	MC_GearInSuperImposed d + MC_CamInSuperImposed w	MC_GearInSuperImposed d + MC_GearInSuperImposed w	MC_GearInSuperImposed d + MC_GearOutSuperImposed w	MC_GearOutSuperImposed d + MC_CamInSuperImposed w	MC_GearOutSuperImposed d + MC_GearInSuperImposed w	MC_CamInSuperImposed d + MC_PhAsingSuperImposed d	MC_GearInSuperImposed d + MC_PhAsingSuperImposed d
MC_CamIn	w	A1	Wa	Wa	Wa	Wa	Wa	A1	Wa	Wa	Wa	Wa	Wa	Wa	Wa
MC_CamIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_GearIn	w	A1	Wa	Wa	Wa	Wa	Wa	A1	Wa	Wa	Wa	Wa	Wa	Wa	Wa
MC_GearIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_CamOut	w	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_CamOut	d	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_GearOut	w	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_GearOut	d	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_Phasing		N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_CamInSuperImposed	w	A1	A2	A2	A2	A2	A2	A	A2	A2	A2	A2	A2	Wa	Wa
MC_CamInSuperImposed	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_GearInSuperImposed	w	A1	A2	A2	A2	A2	A2	A	A2	A2	A2	A2	A2	Wa	Wa
MC_GearInSuperImposed	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_CamOutSuperImposed	w	A2	A3	A2	A2	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Wa	Ae
MC_CamOutSuperImposed	d	A1	A1	A1	A1	A3	Ae	Ae	A3	Ae	Ae	A3	Ae	A1	Ae
MC_GearOutSuperImposed	w	Ae	Ae	Ae	Ae	Ae	Ae	A2	A2	A3	A2	Ae	Ae	Ae	Wa
MC_GearOutSuperImposed	d	Ae	Ae	Ae	A3	Ae	A3	A1	A1	A1	A1	Ae	A3	Ae	A1
MC_PhasingSuperImposed		Ae	S	S	S	Ae	Ae	Ae	S	S	S	Ae	Ae	A3	A3

d	driving The axis is moving, that is, it is operating in synchronous mode, or is currently being synchronized or desynchronized. • The output parameters of the *InGear* or *InSync* technology function are set in synchronous operation. (System variable of the synchronous object *enablecommand=active)*. • The command is active when a synchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *enablecommand=active)*. • The output parameters *InGear* and *InSync* of the technology function are *FALSE* when a desynchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is still set (system variable of synchronous object *disablecommand=active)*..	w	waiting The axis is waiting for the (de)synchronization condition. • The command is active in synchronization wait state. The *Statusword.SynchrCommand* variable of the technology DB is not yet set (system variable of synchronous object *enablecommand=waiting_to_start*). • In desynchronization wait state, the *InSync* and *InGear* output parameters of the technology function are still set. In addition, the *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *disablecommand=waiting_to_start*).
A1	Abort 1 The new command is executed. Active commands are canceled immediately.	A2	Abort 2 The new command is executed. A waiting command is canceled immediately. An active command is canceled if either the synchronization or the desynchronization condition of the new command is satisfied.
A3	Abort 3 The new command is executed and immediately cancels the second active command. Execution of the first active command continues and is not canceled.	Ad	Abort direction Either the MC_MoveToEndPos command is aborted or the new command is canceled with an error, depending on the direction of the new command.
Ae	Abort error The new command is canceled, an ErrorID is entered at the technology DB, and execution of the active commands continues.	E	Error A new command reports an error at the technology function and is not executed. Execution of active commands is continued.
N	No influence Does not have any direct influence on the active function. Command execution is determined by the technology object status.	S	Superimposed The new motion command is superimposed on the active motion commands. The active commands are not canceled.
Sa	Superimposed abort The new motion command is superimposed on the active motion commands. Active commands are not canceled. If neither the synchronization nor desynchronization condition of the active command is met, the new command will be canceled when a condition is satisfied.	Wa	Waiting abort A new command is executed and waits for the synchronization or desynchronization condition. The active commands are aborted when either the synchronization or the desynchronization condition of the new command is satisfied.

For axes, the content of the above table is valid for the standard axis configuration (configuration data element *TypeOfAxis.DecodingConfig.transferSuperimposedPosition = TRANSFER_STANDSTILL (0)*).

6.8.5 New command - active commands (2)

The table below shows the reciprocal actions of a new command on active commands.

new command ↓ / busy single command →	w MC_CamOut + w MC_CamIn	w MC_CamIn + d MC_CamIn	w MC_CamOut + d MC_CamIn	w MC_GearIn + d MC_CamIn	w MC_CamIn + d MC_CamOut	w MC_GearIn + d MC_CamOut	w MC_GearOut + w MC_GearIn	w MC_CamIn + d MC_GearIn	w MC_GearIn + d MC_GearIn	w MC_GearOut + d MC_GearIn	w MC_CamIn + d MC_GearOut	w MC_GearIn + d MC_GearOut	d MC_Phasing + d MC_CamIn	d MC_Phasing + d MC_GearIn	w MC_CamOutSuperImposed + w MC_CamInSuperImposed	w MC_CamInSuperImposed + d MC_CamInSuperImposed	w MC_CamOutSuperImposed + d MC_CamInSuperImposed	w MC_GearInSuperImposed + d MC_CamInSuperImposed	w MC_CamInSuperImposed + d MC_CamOutSuperImposed	w MC_GearInSuperImposed + d MC_CamOutSuperImposed	w MC_GearOutSuperImposed + w MC_GearInSuperImposed	w MC_CamInSuperImposed + d MC_GearInSuperImposed	w MC_GearInSuperImposed + d MC_GearInSuperImposed	w MC_GearOutSuperImposed + d MC_GearInSuperImposed	w MC_CamInSuperImposed + d MC_GearOutSuperImposed	w MC_GearInSuperImposed + d MC_GearOutSuperImposed	d MC_PhAsingSuperImposed + d MC_CamInSuperImposed	d MC_PhAsingSuperImposed + d MC_GearInSuperImposed
MC_Halt	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_Home (*Mode 0,1*)	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_MoveAbsolute	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_MoveAdditive	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_MoveRelative	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_MoveSuperImposed	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_MoveToEndPos	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_MoveVelocity	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_Stop	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1

d	driving The axis is moving, that is, it is operating in synchronous mode, or is currently being synchronized or desynchronized. • The output parameters of the *InGear* or *InSync* technology function are set in synchronous operation. (System variable of the synchronous object *enablecommand=active)*. • The command is active when a synchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *enablecommand=active)*. • The output parameters *InGear* and *InSync* of the technology function are *FALSE* when a desynchronization motion is active. The *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is still set (system variable of synchronous object *disablecommand=active)*..	w	waiting The axis is waiting for the (de)synchronization condition. • The command is active in synchronization wait state. The *Statusword.SynchrCommand* variable of the technology DB is not yet set (system variable of synchronous object *enablecommand=waiting_to_start*). • In desynchronization wait state, the *InSync* and *InGear* output parameters of the technology function are still set. In addition, the *Statusword.SynchrCommand* variable of the technology DB of the synchronous axis is set (system variable of synchronous object *disablecommand=waiting_to_start*).
A1	Abort 1 The new command is executed. Active commands are canceled immediately.	S	Superimposed The new motion command is superimposed on the active motion commands. Active commands are not canceled.
Sa	Superimposed abort The new motion command is superimposed on the active motion commands. Active commands are not canceled. If neither the synchronization nor desynchronization condition of the active command is met, the new command will be canceled when a condition is satisfied.		

For axes, the content of the above table is valid for the standard axis configuration (configuration data element *TypeOfAxis.DecodingConfig.transferSuperimposedPosition = TRANSFER_STANDSTILL (0)*).

6.8.6 Examples of using the tables

Example 1

busy single command →

new command ↓

new command		MC_CamIn	MC_CamIn	MC_GearIn	MC_GearIn	MC_CamOut	MC_GearOut	MC_CamInSuperImposed	MC_CamInSuperImposed	MC_GearInSuperImposed	MC_GearInSuperImposed	MC_CamOutSuperImposed	MC_GearOutSuperImposed	MC_Home (*Mode=0, 1*)	MC_Home (*Mode=2-7*), Achse freigegeben	MC_Halt	MC_Stop	MC_MoveAbsolute	MC_MoveAdditive	MC_MoveRelative	MC_MoveVelocity	MC_MoveToEndPos (*nClamping=True*)	MC_MoveToEndPos (*nClamping=False*)	MC_MoveSuperImposed
		d	w	d	w	d	d	d	w	d	w	d	d											
MC_CamIn	w	Wa	A1	Wa	[illegible]	Wa	Wa	Wa	A1	Wa	A1	Wa	Wa	Wa	S	Wa	E	Wa	Wa	Wa	Wa	Ad	Wa	Wa
MC_CamIn	d	A1	A1	A1	[illegible]	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_GearIn	w	Wa	A1	Wa	[illegible]	Wa	Wa	Wa	A1	Wa	A1	Wa	Wa	Wa	S	Wa	E	Wa	Wa	Wa	Wa	Ad	Wa	Wa
MC_GearIn	d	A1	A1	A1	[illegible]	A1	A1	A1	A1	A1	A1	A1	A1	A1	S	A1	E	A1	A1	A1	A1	Ad	A1	A1
MC_CamOut	w	Wa	Wa	Ae	Ae	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_CamOut	d	A1	A1	Ae	Ae	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_GearOut	w	[illegible]	[illegible]	[illegible]	Wa	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_GearOut	d	Ae	Ae	A1	A1	Ae	Ae	N	N	N	N	N	N	N	N	N	E	N	N	N	N	N	N	N
MC_Phasing		S	E	S	E	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Ae	E	E	E	E	E	E	E	E	E	E	E
MC_CamInSuperImposed	w	Sw	Sa	Sw	Sa	S	S	Wa	A1	Wa	A1	Wa	Wa	E	E	Sw	E	Sw	Sw	Sw	Sw	Ad	Sw	Sw
MC_CamInSuperImposed	d	S	Sa	S	Sa	S	S	A1	A1	A1	A1	A1	A1	E	E	S	E	S	S	S	S	Ad	S	A1
MC_GearInSuperImposed	w	Sw	Sa	Sw	Sa	S	S	Wa	A1	Wa	A1	Wa	Wa	E	E	W	E	Sw	Sw	Sw	Sw	Ad	Sw	Sw
MC_GearInSuperImposed	d	S	Sa	S	Sa	S	S	A1	A1	A1	A1	A1	A1	E	E	S	E	S	S	S	S	Ad	S	A1
MC_CamOutSuperImposed	w	N	N	N	N	N	N	Wa	Wa	Ae	Ae	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_CamOutSuperImposed	d	N	N	N	N	N	N	A1	A1	Ae	Ae	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_GearOutSuperImposed	w	N	N	N	N	N	N	Ae	Ae	Wa	Wa	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_GearOutSuperImposed	d	N	N	N	N	N	N	Ae	Ae	A1	A1	Ae	Ae	N	N	N	E	N	N	N	N	N	N	N
MC_PhasingSuperImposed		Ae	Ae	Ae	Ae	Ae	Ae	S	Ae	S	Ae	Ae	Ae	E	E	E	E	E	E	E	E	E	E	E

An MC_GearIn command is waiting for its synchronization condition ("w").

A new MC_GearOut command is started, and its desynchronization condition is not yet met at startup ("w"). Both commands refer to the same technology object.

The Technology CPU reacts by setting "Wa". The meaning of reaction "Wa" is defined in the corresponding legend.

Example 2

busy single command

new command

new command \ busy single command		MC_CamOut w + MC_CamIn w	MC_CamIn w + MC_CamIn d	MC_CamOut w + MC_CamIn d	MC_GearIn w + MC_CamIn d	MC_CamIn w + MC_CamOut d	MC_GearIn w + MC_CamOut d	MC_GearOut w + MC_GearIn w	MC_CamIn w + MC_GearIn d	MC_GearIn w + MC_GearIn d	MC_GearOut w + MC_GearIn d	MC_CamIn w + MC_GearOut d	MC_GearIn w + MC_GearOut d	MC_Phasing d + MC_CamIn d	MC_Phasing d + MC_GearIn d	MC_CamOutSuperImposed w + MC_CamInSuperImposed w	MC_CamInSuperImposed w + MC_CamInSuperImposed d	MC_CamOutSuperImposed w + MC_CamInSuperImposed d	MC_GearInSuperImposed w + MC_CamInSuperImposed d	MC_CamInSuperImposed w + MC_CamOutSuperImposed d	MC_GearInSuperImposed w + MC_CamOutSuperImposed d	MC_GearOutSuperImposed w + MC_GearInSuperImposed w	MC_CamInSuperImposed w + MC_GearInSuperImposed d	MC_GearInSuperImposed w + MC_GearInSuperImposed d	MC_GearOutSuperImposed w + MC_GearInSuperImposed d	MC_CamInSuperImposed w + MC_GearOutSuperImposed d	MC_GearInSuperImposed w + MC_GearOutSuperImposed d	MC_PhAsingSuperImposed d + MC_CamInSuperImposed d	MC_PhAsingSuperImposed d + MC_GearInSuperImposed d
MC_CamIn	w	A1	A2	A2	A2	A2	A2	A1	A2	A2	[illegible]	A2	A2	Wa	Wa	A1	Wa	Wa	Wa	Wa	Wa	A1	Wa	Wa	Wa	Wa	Wa	Wa	Wa
MC_CamIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	[illegible]	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_GearIn	w	A1	A2	A2	A2	A2	A2	A1	A2	A2	[illegible]	A2	A2	Wa	Wa	A1	Wa	Wa	Wa	Wa	Wa	A1	Wa	Wa	Wa	Wa	Wa	Wa	Wa
MC_GearIn	d	A1	A1	A1	A1	A1	A1	A1	A1	A1	[illegible]	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_CamOut	w	A2	A3	A2	A2	Ae	Ae	Ae	Ae	Ae	[illegible]	Ae	Ae	Wa	Ae	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_CamOut	d	A1	A1	A1	A1	A3	Ae	Ae	A3	Ae	[illegible]	A3	Ae	A1	Ae	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_GearOut	w	Ae	Ae	Ae	Ae	Ae	Ae	A2	A2	A3	[illegible]	Ae	Ae	Ae	Wa	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_GearOut	d	Ae	Ae	Ae	A3	Ae	A3	A1	A1	A1	A1	Ae	A3	Ae	A1	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_Phasing		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	S	Ae	Ae	A3	A3	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MC_CamInSuperImposed	w	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S	A1	A2	A2	A2	A2	A2	A	A2	A2	A2	A2	A2	Wa	Wa
MC_CamInSuperImposed	d	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_GearInSuperImposed	w	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S	A1	A2	A2	A2	A2	A2	A	A2	A2	A2	A2	A2	Wa	Wa
MC_GearInSuperImposed	d	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	Sa	S	S	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
MC_CamOutSuperImposed	w	N	N	N	N	N	N	N	N	N	N	N	N	N	N	A2	A3	A2	A2	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Ae	Wa	Ae
MC_CamOutSuperImposed	d	N	N	N	N	N	N	N	N	N	N	N	N	N	N	A1	A1	A1	A1	A3	Ae	Ae	A3	Ae	Ae	A3	Ae	A1	Ae
MC_GearOutSuperImposed	w	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Ae	Ae	Ae	Ae	Ae	Ae	A2	A2	A3	A2	Ae	Ae	Ae	Wa
MC_GearOutSuperImposed	d	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Ae	Ae	Ae	A3	Ae	A3	A1	A1	A1	A1	Ae	A3	Ae	A1
MC_PhasingSuperImposed		N	N	N	N	N	N	N	N	N	N	N	N	N	N	Ae	S	S	S	Ae	Ae	Ae	S	S	S	Ae	Ae	A3	A3

A technology object is in synchronism. An MC_GearIn command is active and output parameter *InGear = TRUE* ("d"). An MC_GearOut command is waiting for its desynchronization condition ("w"). Both commands refer to the same technology object.

A new MC_Phasing command is initiated at the same technology object.

The Technology CPU reacts by setting "S". The meaning of reaction "S" is defined in the corresponding legend.

6.9 Information on parameters

6.9.1 Reaction of the technology function after POWER OFF and restart

The technology function is initialized after each POWER OFF (POWER OFF -> POWER ON) and restart (RUN-STOP -> STOP-RUN).

Note

If the value *TRUE* is shown at the input parameter *Execute* after POWER OFF or restart, the system detects an edge and starts the command.

In order to prevent any unintentional start of the command, open SIMATIC Manager, and then set the "non-retain" check box in the block properties of the instance DB.

The relevant block is initialized again after POWER On or restart if "non-retain" is set.

6.9.2 Generating a DoneFlag

The *DoneFlag* can be used to display and evaluate the completion of technology object commands in the local MCDevice DB.

At the input parameter *DoneFlag* of the technology function, you define whether a *DoneFlag* is to be generated in the MCDevice DB, and if so which *DoneFlag* .

Parameter value	Meaning
0	A *DoneFlag* is not generated in the MCDevice DB
1	A *DoneFlag*[1] is generated in the MCDevice DB
...	...
32	A *DoneFlag*[32] is generated in the MCDevice DB

Each time a command is completed, the value in *DoneFlag* is inverted at the MCDevice DB (*FALSE* becomes *TRUE* and *TRUE* becomes *FALSE*) . The *DoneFlag* is only set as long as the initiated command can be monitored. The command can no longer be monitored if a subsequent command uses the same instance.

6.9.3 Ranges of values

Range of REAL values

The following applies to all REAL values at the input parameters of the technology functions:

-1E+12 < REAL value < *+1E+12*

The restrictions specified in the descriptions of the input parameters also apply.

These values in integrated technology are verified at the start of motion control commands. If these values ranges are exceeded, an error message is output at the technology DB and at the technology function.

Range of values for dynamic parameters at the technology functions

You define the dynamic response limits for axis motions with the input parameters *Velocity*, *Acceleration*, *Deceleration*, and *Jerk*

If the dynamic value of a motion lies within default limits of the axis configuration, the axis responds as follows:

Dynamic parameters	Response when < 0	Response when = 0	Response when > 0
Velocity	Use the default velocity set in the axis configuration	Error: The command is ignored Exception: "MC_MoveVelocity"	Use Velocity
Acceleration	Use the default acceleration set in the axis configuration	Error: Command is ignored (not permitted)	Use Acceleration
Deceleration	Use the default deceleration set in the axis configuration	Error: Command is ignored	Use Deceleration
Jerk	Use the default jerk values set in the axis configuration	Use a trapezoidal profile	Use Jerk

A warning is output at the technology DB if the dynamic value of a motion is outside the default limits set in the axis configuration.

Note

You can edit the default limits of the axis configuration by calling the "MC_WriteParameter" technology function.

Note

Initially, the dynamic values in the technology function are not verified at the start of motion control commands. The command is output to the system. The system verifies the dynamic values and reports any errors at the corresponding technology DB. It also sets output parameter *CommandAborted* at the technology function.

Technology DBs - Range of values

The range of values of technology DBs corresponds with the range of numbers implemented in the technology CPU used for the data blocks. For information on the range of values, refer to the technology CPU manual.

The input parameters at the technology functions *Axis, Master* , etc. are of data type INT. This data type is used to address technology DBs with positive values from *0* to *+ 32767* It is advisable to store the technology data blocks in the address area *0* to *+32767*.

If this is not possible, you can address technology DBs within the range from *32768* to *65535* as shown below:

- Solution 1
  Use the function FC400 "DB2INT".
- Solution 2
  Enter the following value for the technology DB number at the input parameters *Axis*, etc.:
  Value = number of the technology DB - *65536*
- Solution 3
  Use a variable, for example:
  – Define the variable in the declaration section of the block:
    ```
    VAR_TEMP
    TechDB: INT;
    END_VAR
    ```
  – Use the variable at the call of the technology function:
    ```
    L L#56000 //number of the technology DB
    T #TechDBNR
    CALL "MC_Power" , DB401
    Axis :=#TechDBNR'
    T #TechDBNR
    ```

6.9.4 Absolute positions of modulo axes

The technology converts all absolute positioning parameters to the modulo axis cycle.

Example of an axis with modulo start value *0°* and modulo length *360°*:

- Absolute positioning to *400°* positions the axis to *40°* (*400°* Modulo *360°* = *40°*).
- A homing position setting of *-20°* is equivalent to homing position *340°*.
- The starting position of an output cam is set to *730°*. The output cam ON position is *10°* (*730°* modulo *360°* = *10°*)

7 Technology data blocks

7.1 "Speed-controlled axis" technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

UpdateFlag / *UpdateCounter* variable

Please take note of the following points: Updating technology DBs (Page 791)

ErrorID variable

Returns the *ErrorID* of the error or warning most recently detected at the speed-controlled axis.

An error entry can be acknowledged by calling the technology function "MC_Reset" (with *Axis* = number of the technology DB).

Please take note of the following points: Possible error messages and warnings (Page 797)

ErrorBuffer[0..2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis* = number of the technology DB).

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

ErrorStatus.xxx variable

The bits in the *ErrorStatus.xxx* variable return information on axis errors.

Bit No.	Variable	Significance for the *TRUE* state
0	*ErrorStatus.SystemFault*	Internal system error
1	*ErrorStatus.ConfigFault*	Faulty technology object configuration
2	*ErrorStatus.UserFault*	The user program has caused an error by outputting an invalid command
3	*ErrorStatus.FaultDrive*	A drive or technology object reports an error
4	*ErrorStatus.Reserve4*	Not used
5	*ErrorStatus.FollowingWarning*	Not used
6	*ErrorStatus.FollowingError*	Not used
7	*ErrorStatus.StandstillFault*	The axis has moved out of the standstill window or could not reach the standstill window within the time specified.
8	*ErrorStatus.PositioningError*	Not used
9	*ErrorStatus.SynchronOpError*	Not used
10	*ErrorStatus.DynamicError*	Dynamic response limits were exceeded
11	*ErrorStatus.ClampingError*	Not used
12	*ErrorStatus.SoftwareLimitPos*	Not used
13	*ErrorStatus.SoftwareLimitNeg*	Not used
14	*ErrorStatus.LimitSwitchActive*	Hardware limit switch is active
15	*ErrorStatus.SensorFreqViolation*	Encoder limit frequency exceeded.
16	*ErrorStatus.ReferenceNotFound*	Not used
17	*ErrorStatus.ZeroMonitoring*	Zero pulse monitoring has detected an error (not a homing error)
18	*ErrorStatus.Overspeed*	Not used
19	*ErrorStatus.FollowObjectError*	Not used
20	*ErrorStatus.SupImpFollowObjectError*	Not used

Statusword.xxx variable

The variable *Statusword* indicates the current status of the speed-controlled axis.

Bit No.	Variable (system variable in S7T Config	Significance for the *TRUE* state
0	*Statusword.DriveEnabled* *(actormonitoring.power + actormonitoring.drivestate)*	The pulse enable and the drive enable are active. (Pulse enable and drive enable are not identical with enabling of the axes). For virtual axes, this value is always *TRUE*. Pulse enable is not active in follow-up mode ("MC_Power" with *Mode = 3*).
1	*Statusword.HomingDone* *(positioningstate.homed)*	Irrelevant to the speed-controlled axis.
2	*Statusword.Done*	No motion command is being executed ("MC_Power" does not represent a motion command in this case).
3	*Statusword.SuperImposedCommand*	Irrelevant to the speed-controlled axis.
4	*Statusword.Error*	At least one error has occurred (variable *ErrorID = 8xxx*). In the event of warnings, the *Statusword.Error = FALSE* status is retained.
5	*Statusword.Errorstop* *(errorreaction)*	The axis is/was stopped due to an error event; the technology object may be disabled. Eliminate the cause and acknowledge the error.
6	*Statusword.Stopping*	Active MC_Stop command at the axis. New motion commands are rejected. The status of the motion command is shown in the variables *Statusword.Standstill, Statusword.ConstantVelocity,Statusword.Accelerating* or *Statusword.Decelerating*. The response may occur with delay.
7	*Statusword.Standstill* *(motionstatedata.stillstandvelocity)*	The axis speed (absolute value) is lower than the configured standstill limit.
8	*Statusword.PositioningCommand* *(poscommand.state)*	Irrelevant to the speed-controlled axis.
9	*Statusword.SpeedCommand* *(movecommand.state)*	Active speed command at the axis. Examples: • "MC_MoveVelocity" • "MC_MoveToEndPos" (before the fixed stop is detected)
10	*Statusword.SynchrCommand*	Irrelevant to the speed-controlled axis.
11	*Statusword.Homing* *(homimgcommand.state)*	Irrelevant to the speed-controlled axis.
12	*Statusword.FollowUpControl* *(control* (inverted)	Set as long as following mode is active. This is the case if the axis is disabled with "MC_Power" or enabled with *Mode = 3*.
13	*Statusword.ConstantVelocity* *(motionstatedata.motionstate)*	Set as long as the speed setpoint remains constant.
14	*Statusword.Accelerating* *(motionstatedata.motionstate)*	Axis is accelerating (increasing drive power).
15	*Statusword.Decelerating* *(motionstatedata.motionstate)*	Axis is decelerating (declining drive power).

Bit No.	Variable (system variable in S7T Config	Significance for the *TRUE* state
16	*Statusword.RequestRestart*	Axis parameters have been changed and will not be activated until the next restart is completed ("MC_Reset" with Restart = TRUE).
17	*Statusword.Simulation* *(simulation)*	Axis in simulation mode. You achieve this with "MC_Power" by setting *Mode = 4*.
18	*Statusword.CyclicInterface* *(actormonitoring.cyclicinterface)*	**At virtual axes:** The value of this variable is always *TRUE.*. **At electrical axes:** Cyclic communication between the controller and the drive is active. No active drive errors. **At hydraulic axes:** The I/O for control signal output and encoder signals are detected on DP(DRIVE) and can be used by the axis technology object. If several axis technology objects are configured at the same I/O address, *Statusword.CyclicInterface* returns *TRUE* at all relevant technology DBs. The status indicated is independent of any technology object access to the I/O.
19	*Statusword.EncoderValid* *(sensordata.sensordata[current encoder].state* = VALID)	Irrelevant to the speed-controlled axis.
20	*Statusword.SpeedMode* *(speedmode)*	At the speed-controlled axis this is always TRUE.
21	*Statusword.TorqueLimiting* *(torquelimitingcommand. torquelimitingstate)*	The torque of the axis was limited by an "MC_SetTorqueLimit" or "MC_MoveToEndPos" command, and the drive is operating at this limit. The drive must support torque limiting (message frame 10x.)
22	*Statusword.SupImpSynchrCommand*	Irrelevant to the speed-controlled axis.
23	*Statusword.TorqueLimitingCommand* *(torquelimitingcommand.state)*	A torque-reducing command ("MC_SetTorqueLimit" or "MC_MoveToEndPos") is active at the axis.
24	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
25	*Statusword.ToDeactivated*	Technology object was deactivated
26	*Statusword.HWLimitSwitchMinus*	The HW limit switch has approached in the negative direction of motion
27	*Statusword.HWLimitSwitchPlus*	The HW limit switch has approached in the positive direction of motion

CommandVelocity variable

(S7T Config system variable *motionstatedata.commandvelocity*)

Shows the current setpoint value specification for the axis velocity (allowance is made for the current *SpeedOverride* value).

CommandAcceleration variable

(S7T Config system variable *motionstatedata.commandacceleration*)

Shows the current setpoint value specification for axis acceleration (allowance is made for the current *AccelerationOverride* value).

ActualVelocity variable

(S7T Config system variable *motionstatedata.actualvelocity*)

The variable returns the current velocity of the axis.

If the speed-controlled axis does not have an encoder, this value is always *0*.

ActualAcceleration variable

(S7T Config system variable *motionstatedata.actualacceleration*)

Returns the current acceleration of the axis.

If the speed-controlled axis does not have an encoder, this value is always *0*.

SpeedOverride variable

(S7T Config system variable *override.velocity*)

This variable returns the percentage of a currently effective *SpeedOverride* value. The percentage is reciprocal to the velocity setpoint of the axis. The default is 100%.

The *SpeedOverride* value can be set within the range from 0 % *to* 200 % using the MC_WriteParameter technology function.

AccelerationOverride variable

(S7T Config system variable *override.acceleration*)

This variable returns the percentage of a currently effective *AccelerationOverride* value. The percentage is reciprocal to the acceleration setpoint of the axis. Default value = 100 %.

The *AccelerationOverride* value can be set within the range from 1 % *to* 1000 % using the MC_WriteParameter technology function.

7.2 "Positioning axis" technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

UpdateFlag / UpdateCounter variable

Please take note of the following points: Updating technology DBs (Page 791)

ErrorID variable

The variable returns the *ErrorID* of the last error or warning detected at the positioning axis.

An error entry can be acknowledged by calling the technology function "MC_Reset" (with *Axis =* number of the technology DB).

Please take note of the following points: Possible error messages and warnings (Page 797)

ErrorBuffer[0..2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

ErrorStatus.xxx variable

The bits in the *ErrorStatus.xxx* variable return information about positioning axis errors.

Bit No.	Variable	Significance for the *TRUE* state
0	*ErrorStatus.SystemFault*	Internal system error
1	*ErrorStatus.ConfigFault*	Faulty technology object configuration
2	*ErrorStatus.UserFault*	The user program has caused an error by outputting an invalid command
3	*ErrorStatus.FaultDrive*	A drive or technology object reports an error
4	*ErrorStatus.Reserve4*	Not used
5	*ErrorStatus.FollowingWarning*	Dynamic following error monitoring function reports violation of warning limits
6	*ErrorStatus.FollowingError*	Dynamic following error monitoring function reports violation of window limits
7	*ErrorStatus.StandstillFault*	The axis has moved out of the standstill window or could not reach the standstill window within the time specified.
8	*ErrorStatus.PositioningError*	The axis failed to reach the positioning window within the specified time.
9	*ErrorStatus.SynchronOpError*	Synchronization tolerances exceeded
10	*ErrorStatus.DynamicError*	Dynamic response limits were exceeded
11	*ErrorStatus.ClampingError*	Clamping error The axis has exceeded the "Position tolerance after fixed stop detection" without receiving a new motion command. Position tolerance setting in S7T Config **Limits > "Fixed end stop" tab > "Position tolerance after fixed end stop detection" parameter**.
12	*ErrorStatus.SoftwareLimitPos*	Upper software limit switch reached or passed
13	*ErrorStatus.SoftwareLimitNeg*	Lower software limit switch reached or passed
14	*ErrorStatus.LimitSwitchActive*	Hardware limit switch is active
15	*ErrorStatus.SensorFreqViolation*	Encoder limit frequency exceeded.
16	*ErrorStatus.ReferenceNotFound*	Reference cam or zero pulse not found when homing
17	*ErrorStatus.ZeroMonitoring*	Zero pulse monitoring has detected an error (not a homing error)
18	*ErrorStatus.Overspeed*	Not used
19	*ErrorStatus.FollowObjectError*	Not used
20	*ErrorStatus.SupImpFollowObjectError*	Not used

Statusword.xxx variable

The variable *Statusword* indicates the current status of the positioning axis.

Bit No.	Variable (system variable in S7T Config	Significance for the *TRUE* state
0	*Statusword.DriveEnabled* *(actormonitoring.power + actormonitoring.drivestate)*	The pulse enable and the drive enable are active. (Pulse enable and drive enable are not identical with enabling of the axes). For virtual axes, this value is always *TRUE*. Pulse enable is not active in following mode ("MC_Power" with *Mode Mode = 3*.)
1	*Statusword.HomingDone* *(positioningstate.homed)*	The axis was homed. The homing condition was satisfied. During active homing, the *Statusword.HomingDone* in the technology DB will be set when the encoder zero mark is detected. However, the axis is still decelerating, or moving by the value of the reference point offset. Technology function "MC_Home" only reports *Done=TRUE* after the axis has reached a standstill. The axis has now reached the position of the reference point coordinate.
2	*Statusword.Done*	No motion command is being executed ("MC_Power" does not represent a motion command in this case).
3	*Statusword.SuperImposedCommand*	A superimposed motion is active (for example ""MC_MoveSuperImposed"
4	*Statusword.Error*	At least one error has occurred (variable *ErrorID = 8xxx*). In the event of warnings, the Statusword.Error = FALSE status is retained.
5	*Statusword.Errorstop* *(errorreaction <> NONE)*	The axis is/was stopped due to an error event; the technology object may be disabled. Eliminate the cause and acknowledge the error.
6	*Statusword.Stopping*	Active MC_Stop command at the axis. New motion commands are rejected. The status of the motion command is shown in the variables *Statusword.Standstill, Statusword.ConstantVelocity, Statusword.Accelerating* or *Statusword.Decelerating*. The response may occur with delay.
7	*Statusword.Standstill* *(motionstatedata.stillstandvelocity)*	The axis velocity (absolute value) is lower than the configured standstill limit.
8	*Statusword.PositioningCommand* *(poscommand.state)*	A positioning command is active at the axis (may also be superimposed.)
9	*Statusword.SpeedCommand* *(movecommand.state)*	Active speed command at the axis. Examples: • "MC_MoveVelocity" • "MC_MoveToEndPos" (before the fixed stop is detected)
10	*Statusword.SynchrCommand*	Irrelevant to the positioning axis.
11	*Statusword.Homing* *(homimgcommand.state)*	The bit is set at the start, and reset at the end of "MC_Home."
12	*Statusword.FollowUpControl* *(control ((inverted)*	Set as long as follow-up mode is active. This is the case if the axis is disabled with "MC_Power" or enabled with *Mode = 3*.

Bit No.	Variable (system variable in S7T Config	Significance for the *TRUE* state
13	*Statusword.ConstantVelocity*	Set as long as the speed setpoint remains constant.
	(motionstatedata.motionstate)	
14	*Statusword.Accelerating*	Axis is accelerating (increasing drive power).
	(motionstatedata.motionstate)	
15	*Statusword.Decelerating*	Axis is decelerating (declining drive power).
	(motionstatedata.motionstate)	
16	*Statusword.RequestRestart*	Axis parameters have been changed and will not be activated until the next restart is completed ("MC_Reset" with *Restart* = *TRUE*).
17	*Statusword.Simulation*	Axis in simulation mode.
	(simulation)	You achieve this with "MC_Power" by setting *Mode = 4*.
18	*Statusword.CyclicInterface*	**At virtual axes:**
	(actormonitoring.cyclicinterface)	The value of this variable is always *TRUE.* **At electrical axes:** Cyclic communication between the controller and the drive is active. No active drive errors. **At hydraulic axes:** The I/O for control signal output and encoder signals are detected on DP(DRIVE) and can be used by the axis technology object. If several axis technology objects are configured at the same I/O address, *Statusword.CyclicInterface* returns *TRUE* at all relevant technology DBs. The status indicated is independent of any technology object access to the I/O.
19	*Statusword.EncoderValid*	The actual position value of the encoder that is relevant to the axis is valid (of particular importance in the startup phase of absolute encoders).
	(sensordata.sensordata[current encoder].state = VALID)	
20	*Statusword.SpeedMode*	*TRUE* = speed-controlled mode
	(speedmode)	*FALSE* = position-controlled mode Speed-controlled mode is activated at the technology function "MC_MoveVelocity" by setting input parameter *PositionControl = FALSE .*
21	*Statusword.TorqueLimiting*	The torque of the axis was limited by an "MC_SetTorqueLimit" or "MC_MoveToEndPos" command, and the drive is operating at this limit. The drive must support torque limiting (message frame *10x*.)
	(torquelimitingcommand. torquelimitingstate)	
22	*Statusword.SupImpSynchrCommand*	Irrelevant to the positioning axis.
23	*Statusword.TorqueLimitingCommand*	A torque-reducing command ("MC_SetTorqueLimit" or "MC_MoveToEndPos") is active at the axis.
	(torquelimitingcommand.state)	
24	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
25	*Statusword.ToDeactivated*	Technology object was deactivated
26	*Statusword.HWLimitSwitchMinus*	The HW limit switch has approached in the negative direction of motion
27	*Statusword.HWLimitSwitchPlus*	The HW limit switch has approached in the positive direction of motion

CommandVelocity variable

(S7T Config system variable *motionstatedata.commandvelocity*)

Shows the current setpoint value specification for the axis velocity (allowance is made for the current *SpeedOverride* value).

CommandAcceleration variable

(S7T Config system variable *motionstatedata.commandacceleration*)

Shows the current setpoint value specification for axis acceleration (allowance is made for the current *AccelerationOverride* value).

ActualVelocity variable

(S7T Config system variable *motionstatedata.actualvelocity*)

The variable returns the current velocity of the axis.

ActualAcceleration variable

(S7T Config system variable *motionstatedata.actualacceleration*)

Returns the current acceleration of the axis.

SpeedOverride variable

(S7T Config system variable *override.velocity*)

This variable returns the percentage of a currently effective *SpeedOverride* value. The percentage is reciprocal to the velocity setpoint of the axis. The default is 100%.

The *SpeedOverride* value can be set from 0% to 200% at the "MC_WriteParameter" technology function.

AccelerationOverride variable

(S7T Config system variable *override.acceleration*)

Returns a percentage of the currently active AccelerationOverride. The percentage is reciprocal to the acceleration setpoint of the axis. Default value = 100%.

The *AccelerationOverride* value can be set within the range from 1% to 1000% using the *MC_WriteParameter* technology function.

ActualPosition variable

(S7T Config system variable *positioningstate.actualposition*)

Shows the current axis position.

CommandPosition variable

(S7T Config system variable *positioningstate.commandposition*)

Indicates the current position setpoint for positioning operations.

TargetPosition variable

Indicates the target position for the current command. This indicated value only applies to active positioning commands.

Distance variable

(S7T Config system variable *poscommand.distancetogo*)

Shows the current distance for relative / absolute positioning. This indicated value only applies to active positioning commands.

DecelerationDistance variable

(S7T Config system variable *poscommand.decelerationdistance*)

The variable returns the current stop distance of the axis. This indicated value only applies to active positioning commands.

FollowingError variable

(S7T Config system variable *servodata.followingerror*)

Indicates the current following error value of the axis. This indicated value only applies to active positioning commands.

7.3 "Synchronization axis" technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

UpdateFlag / UpdateCounter variable

Please take note: Updating technology DBs (Page 791)

ErrorID variable

Shows the ErrorID of the last error or warning detected at the synchronization axis.

An error entry can be acknowledged by calling the technology function "MC_Reset" (with *Axis =* number of the technology DB).

Please take note: Possible error messages and warnings (Page 804)

ErrorBuffer[0..2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

ErrorStatus.xxx variable

The bits in the *ErrorStatus.xxx* variable return information on axis errors.

Bit No.	Variable	Significance for the *TRUE* state
0	*ErrorStatus.SystemFault*	Internal system error
1	*ErrorStatus.ConfigFault*	Faulty technology object configuration
2	*ErrorStatus.UserFault*	The user program has caused an error by outputting an invalid command
3	*ErrorStatus.FaultDrive*	A drive or technology object reports an error
4	*ErrorStatus.Reserve4*	Not used
5	*ErrorStatus.FollowingWarning*	Dynamic following error monitoring function reports violation of warning limits
6	*ErrorStatus.FollowingError*	Dynamic following error monitoring function reports violation of window limits
7	*ErrorStatus.StandstillFault*	The axis has moved out of the standstill window or could not reach the standstill window within the time specified.
8	*ErrorStatus.PositioningError*	The axis failed to reach the positioning window within the specified time.
9	*ErrorStatus.SynchronOpError*	Synchronization tolerances exceeded
10	*ErrorStatus.DynamicError*	Dynamic response limits were exceeded
11	*ErrorStatus.ClampingError*	Clamping error The axis has exceeded the "Position tolerance after fixed stop detection" without receiving a new motion command. Position tolerance setting in S7T Config **Limits > "Fixed end stop" tab > "Position tolerance after fixed end stop detection" parameter**.
12	*ErrorStatus.SoftwareLimitPos*	Upper software limit switch reached or passed
13	*ErrorStatus.SoftwareLimitNeg*	Lower software limit switch reached or passed
14	*ErrorStatus.LimitSwitchActive*	Hardware limit switch is active
15	*ErrorStatus.SensorFreqViolation*	Encoder limit frequency exceeded.
16	*ErrorStatus.ReferenceNotFound*	Reference cam or zero pulse not found when homing
17	*ErrorStatus.ZeroMonitoring*	Zero pulse monitoring has detected an error (not a homing error)
18	*ErrorStatus.Overspeed*	Not used
19	*ErrorStatus.FollowObjectError*	Error at the synchronization object.
20	*ErrorStatus.SupImpFollowObjectError*	Error at the superimposing synchronization object.

Statusword.xxx variable

The variable *Statusword* indicates the current status of the synchronization axis.

Bit No.	Variable (system variable in S7T Config	Significance for the *TRUE* state
0	*Statusword.DriveEnabled* *(actormonitoring.power + actormonitoring.drivestate)*	The pulse enable and the drive enable are active. (Pulse enable and drive enable are not identical with enabling of the axes). This value is always *TRUE* at virtual axes. Pulse enable is not active in follow-up mode ("MC_Power" with *Mode = 3*).
1	*Statusword.HomingDone* *(positioningstate.homed)*	The axis was homed. The homing condition was satisfied. In active homing operations, the reference point coordinate and *Statusword.HomingDone* in the technology DB are set when the encoder zero mark is detected. However, the axis is still decelerating, or moving by the value of the reference point offset. However, the axis is already homed. Technology function "MC_Home" only reports *Done=TRUE* after the axis has reached a standstill.
2	*Statusword.Done*	No motion command is being executed ("MC_Power" does not represent a motion command in this case).
3	*Statusword.SuperImposedCommand*	A superimposed motion is active (for example "MC_MoveSuperImposed")
4	*Statusword.Error*	At least one error has occurred (variable *ErrorID = 8xxx*). In the event of warnings, the *Statusword.Error = FALSE* status is retained.
5	*Statusword.Errorstop* *(errorreaction <> NONE)*	The axis is/was stopped due to an error event; the technology object may be disabled. Eliminate the cause and acknowledge the error.
6	*Statusword.Stopping*	Active MC_Stop command at the axis. New motion commands are rejected. The status of the motion command is shown in the variables *Statusword.Standstill, Statusword.ConstantVelocity, Statusword.Accelerating* or *Statusword.Decelerating.* The response may occur with delay.
7	*Statusword.Standstill* *(motionstatedata.stillstandvelocity)*	The axis velocity (absolute value) is lower than the configured standstill limit.
8	*Statusword.PositioningCommand* *(poscommand.state)*	A positioning command is active at the axis (may also be superimposed.)
9	*Statusword.SpeedCommand* *(movecommand.state)*	Active speed command at the axis. Examples: • "MC_MoveVelocity" • "MC_MoveToEndPos" (before the fixed stop is detected)
10	*Statusword.SynchrCommand*	Axis in synchronous operation (active "MC_GearIn" or "MC_CamIn")
11	*Statusword.Homing* *(homimgcommand.state)*	The bit is set at the start, and reset at the end, of "MC_Home."
12	*Statusword.FollowUpControl* *(control (inverted))*	Set as long as follow-up mode is active. This is the case if the axis is disabled with "MC_Power" or enabled with *Mode = 3*.

Bit No.	Variable (system variable in S7T Config	Significance for the *TRUE* state
13	*Statusword.ConstantVelocity* *(motionstatedata.motionstate)*	Set as long as the speed setpoint remains constant.
14	*Statusword.Accelerating* *(motionstatedata.motionstate)*	Axis is accelerating (increasing drive power).
15	*Statusword.Decelerating* *(motionstatedata.motionstate)*	Axis is decelerating (declining drive power).
16	*Statusword.RequestRestart*	Axis parameters have been changed and will not be activated until the next restart is completed ("MC_Reset" with Restart = TRUE).
17	*Statusword.Simulation* *(simulation)*	Axis in simulation mode. You achieve this with "MC_Power" by setting *Mode = 4*.
18	*Statusword.CyclicInterface* *(actormonitoring.cyclicinterface)*	At virtual axes: The value of this variable is always *TRUE*.. At electrical axes: Cyclic communication between the controller and the drive is active. No active drive errors. At hydraulic axes: The I/O for control signal output and encoder signals are detected on DP(DRIVE) and can be used by the axis technology object. If several axis technology objects are configured at the same I/O address, *Statusword.CyclicInterface* returns *TRUE* at all relevant technology DBs. The status indicated is independent of any technology object access to the I/O.
19	*Statusword.EncoderValid* *(sensordata.sensordata[current encoder].state = VALID)*	The actual position value of the encoder that is relevant to the axis is valid (of particular importance in the startup phase of absolute encoders).
20	*Statusword.SpeedMode* *(speedmode)*	*TRUE* = speed-controlled mode *FALSE* = position-controlled mode Speed-controlled mode is activated at the technology function "MC_MoveVelocity" by setting input parameter *PositionControl = FALSE.*
21	*Statusword.TorqueLimiting* *(torquelimitingcommand. torquelimitingstate)*	The torque of the axis was limited by an "MC_SetTorqueLimit" or "MC_MoveToEndPos" command, and the drive is operating at this limit. The drive must support torque limiting (message frame 10x.)
22	*Statusword.SupImpSynchrCommand*	Axis in superimposed synchronous operation (active "MC_GearInSuperImposed" or "MC_CamInSuperImposed").
23	*Statusword.TorqueLimitingCommand* *(torquelimitingcommand.state)*	A torque-reducing command ("MC_SetTorqueLimit" or "MC_MoveToEndPos") is active at the axis.
24	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
25	*Statusword.TOdeactivated*	Technology object was deactivated
26	*Statusword.HWLimitSwitchMinus*	The HW limit switch has approached in the negative direction of motion
27	*Statusword.HWLimitSwitchPlus*	The HW limit switch has approached in the positive direction of motion

CommandVelocity variable

(S7T Config system variable *motionstatedata.commandvelocity*)

Shows the current setpoint value specification for the axis velocity (allowance is made for the current *SpeedOverride* value).

CommandAcceleration variable

(S7T Config system variable *motionstatedata.commandacceleration*)

Shows the current setpoint value specification for axis acceleration (allowance is made for the current *AccelerationOverride* value).

ActualVelocity variable

(S7T Config system variable *motionstatedata.actualvelocity*)

The variable returns the current velocity of the axis.

ActualAcceleration variable

(S7T Config system variable *motionstatedata.actualacceleration*)

Returns the current acceleration of the axis.

SpeedOverride variable

(S7T Config system variable *override.velocity*)

Returns the percentage of the currently effective SpeedOverride. The percentage is reciprocal to the velocity setpoint of the axis. The default is 100%.

The *SpeedOverride* value can be set within the range from 0% to 200% using the MC_WriteParameter technology function.

AccelerationOverride variable

(S7T Config system variable *override.acceleration*)

This variable returns the percentage of a currently effective *AccelerationOverride* value. The percentage is reciprocal to the acceleration setpoint of the axis. Default value = 100 %.

The *AccelerationOverride* value can be set within the range from 1 % *to* 1000 % using the *MC_WriteParameter* technology function.

ActualPosition variable

(S7T Config system variable *positioningstate.actualposition*)

Shows the current axis position.

CommandPosition variable

(S7T Config system variable *positioningstate.commandposition*)

Indicates the current position setpoint for positioning operations.

TargetPosition variable

Indicates the target position for the current command. This indicated value only applies to active positioning commands.

Distance variable

(S7T Config system variable *poscommand.distancetogo*)

Shows the current distance for relative/absolute positioning. This indicated value only applies to active positioning commands.

DecelerationDistance variable

(S7T Config system variable *poscommand.decelerationdistance*)

The variable returns the current stop distance of the axis. This indicated value only applies to active positioning commands.

FollowingError variable

(S7T Config system variable *servodata.followingerror*)

Indicates the current following error value of the axis. This value is only valid during the execution of a positioning command or in synchronous operation.

SyncStatus variable

(S7T Config system variable *<synchronization object>.state*)

Value	Significance	Comment
0	No synchronous operation	Active synchronous mode is "No synchronous operation."
1	Geared synchronous motion	Active synchronous mode is "Gearing."
2	Curve synchronization	Active synchronous mode is "Camming."

Direction variable

(S7T Config system variable *<synchronization object>.effectivedata.gearingsettings.direction*)

The technology CPU uses the *RatioNumerator* and *RatioDenominator* input parameters to calculate the gear ratio and transfer direction for gearing mode.

The transfer direction of the gear is defined by the *Direction* variable.

Value	Significance	Comment
5	positive	The leading and following axis move in the same direction.
4	negative	The leading and following axis move in opposite direction.

Camming rules out a contradictory sense of direction, that is, the value *5* will be set in camming mode.

NumGear / DenomGear variable

(S7T Config system variable *<synchronization object>.effectivedata.gearingsettings.numerator / <synchronization object>.effectivedata.gearingsettings.denominator*)

The variables return the gear ratio of basic synchronism. The *NumGear* variable returns the leading axis coefficient, and the *DenomGear* variable returns the following axis coefficient (with reference to basic synchronous operation). The indicated values apply only to basic synchronism.

Example:

At a gear ratio of 17:25 between the leading and following axes, the *NumGear* variable shows the value *17*, and the *DenomGear* variable shows the value *25*.

PhaseShift variable

Shows the phase offset between the position setpoints of the leading axis and following axis operating in basic synchronism. The value indicated is only valid during basic synchronism.

SupImpSyncStatus variable

(S7T Config system variable *<superimposed synchronization object>.state*)

Value	Significance	Comment
0	No synchronous operation	Active mode of the superimposing synchronization object is "No synchronism"
1	Geared synchronous motion	Active mode of the superimposing synchronization object is "Gearing"
2	Curve synchronization	Active mode of the superimposing synchronization object is "Camming"

SupImpDirection variable

(S7T Config system variable *<superimposed synchronization object>.effectivedata.gearingsettings.direction*)

The Technology CPU uses input parameters *RatioNumerator* and *RatioDenominator* to calculate the gear ratio and transfer direction for superimposed gearing.

The transfer direction of the gear is defined by the *Direction* variable.

Value	Significance	Comment
5	positive	The leading and following axis move in the same direction.
4	negative	The leading and following axis move in opposite direction.

Superimposed camming rules out a contradictory sense of direction, that is, the value *5* will be set in superimposed camming mode.

SupImpNumGear / SupImpDenomGear variable

(S7T Config system variable *<superimposed synchronization object>.effectivedata.gearingsettings.numerator / <superimposed synchronization object>.effectivedata.gearingsettings.denominator*)

These variables return the gear ratio for superimposed synchronism. The *SupImpNumGear* variable shows the coefficient of the leading axis, and the *SupImpDenomGear* shows the coefficient of the following axis (with reference to the superimposed coordinate system of the following axis). The indicated values apply only to active superimposed synchronism.

SupImpPhaseShift variable

The variable indicates the offset between the position setpoints of the leading axis and the superimposing coordinate system of the following axis. This value is only applies to active superimposed synchronism.

7.4 "External encoder" technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

UpdateFlag/UpdateCounter variable

Please take note: Updating technology DBs (Page 791)

ErrorID variable

Shows the ErrorID of the last detected error or warning of the external encoder.

An error entry can be acknowledged by calling the technology function "MC_Reset" (with *Axis* = number of the technology DB).

Please take note: Possible error messages and warnings (Page 815)

ErrorBuffer[0..2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

ErrorStatus.xxx variable

The bits in variable *ErrorStatus.xxx* return information on errors at the external encoder.

Bit No.	Variable	Significance for the *TRUE* state
0	*ErrorStatus.SystemFault*	Internal system error
1	*ErrorStatus.ConfigFault*	Faulty technology object configuration
2	*ErrorStatus.UserFault*	The user program has caused an error by outputting an invalid command
3	*ErrorStatus.FaultDrive*	The external encoder or technology object reports an error
4	*ErrorStatus.Reserve4*	Not used
5	*ErrorStatus.FollowingWarning*	Dynamic following error monitoring function reports violation of warning limits
6	*ErrorStatus.FollowingError*	Dynamic following error monitoring function reports violation of window limits
7	*ErrorStatus.StandstillFault*	The external encoder has moved out the standstill window, or could not reach the standstill window in the specified time.
8	*ErrorStatus.PositioningError*	The axis failed to reach the positioning window within the specified time.
9	*ErrorStatus.SynchronOpError*	Not used
10	*ErrorStatus.DynamicError*	Dynamic response limits were exceeded
11	*ErrorStatus.ClampingError*	Clamping error The axis has exceeded the "Position tolerance after fixed stop detection" without receiving a new motion command. Position tolerance setting in S7T Config **Limits > "Fixed end stop" tab > "Position tolerance after fixed end stop detection" parameter**.
12	*ErrorStatus.SoftwareLimitPos*	Not used
13	*ErrorStatus.SoftwareLimitNeg*	Not used
14	*ErrorStatus.LimitSwitchActive*	Not used
15	*ErrorStatus.SensorFreqViolation*	Encoder limit frequency exceeded.
16	*ErrorStatus.ReferenceNotFound*	Reference cam or zero pulse not found when homing
17	*ErrorStatus.ZeroMonitoring*	Zero pulse monitoring has detected an error (not a homing error)
18	*ErrorStatus.Overspeed*	Not used
19	*ErrorStatus.FollowObjectError*	Not used
20	*ErrorStatus.SupImpFollowObjectError*	Not used

Statusword.xxx variable

The *Statusword* variable indicates the current status of the external encoder.

These values are no longer updated if the external encoder is disabled. The values shown correspond with the last status registered at the time the enable signal was reset.

Bit No.	Variable (System variable in S7T Config)	Significance for the *TRUE* state
0	*Statusword.DriveEnabled* *(control)*	The external encoder is enabled (*Mode = 1* at the input parameter of the "MC_ExternalEncoder")
1	*Statusword.HomingDone* *(syncstate)*	The axis is homed, that is, "MC_ExternalEncoder" was successfully completed in *Mode= 2* to *6*
2	*Statusword.Done*	No job being executed.
3	*Statusword.SuperImposedCommand*	Irrelevant to the external encoder.
4	*Statusword.Error*	At least one error has occurred (variable *ErrorID* = *8xxx*). In case of warnings the *Statusword.Error = FALSE* status is retained.
5	*Statusword.Errorstop* *(errorreaction <> NONE)*	The external encoder will be / was locked due to an error. Eliminate the cause and acknowledge the error.
6	*Statusword.Stopping*	Irrelevant to the external encoder.
7	*Statusword.Standstill* *(motionstate.stillstandvelocity)*	The velocity at the external encoder (absolute value) is lower than the set standstill limit.
8	*Statusword.PositioningCommand*	Irrelevant to the external encoder.
9	*Statusword.SpeedCommand*	Irrelevant to the external encoder.
10	*Statusword.SynchrCommand*	Irrelevant to the external encoder.
11	*Statusword.Homing* *(synccommand.state)*	The bit is set at the start and reset at the end of homing ("MC_ExternalEncoder" *Mode = 2* to *6*)
12	*Statusword.FollowUpControl*	Irrelevant to the external encoder.
13	*Statusword.ConstantVelocity*	Irrelevant to the external encoder.
14	*Statusword.Accelerating*	Irrelevant to the external encoder.
15	*Statusword.Decelerating*	Irrelevant to the external encoder.
16	*Statusword.RequestRestart*	External encoder parameters have been changed and will not be activated until the next restart is completed ("MC_Reset" with *Restart* = *TRUE*).
17	*Statusword.Simulation*	Irrelevant to the external encoder.
18	*Statusword.CyclicInterface* *(sensormonitoring.cyclicInterface)*	Cyclic communication between the controller and the encoder is active.
19	*Statusword.EncoderValid* *(ExternalEncoder.sensordata.state = VALID)*	The actual position value of the external encoder is valid (of particular importance in the startup phase of absolute encoders.)
20	*Statusword.SpeedMode*	Always *FALSE* for the external encoder.

Bit No.	Variable (System variable in S7T Config)	Significance for the *TRUE* state
21	*Statusword.TorqueLimiting*	Irrelevant to the external encoder.
22	*Statusword.SupImpSynchrCommand*	Irrelevant to the external encoder.
23	*Statusword.TorqueLimitingCommand*	A torque-reducing command ("MC_SetTorqueLimit" or "MC_MoveToEndPos") is active at the axis.
24	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
25	*Statusword.ToDeactivated*	Technology object was deactivated

ActualPosition variable

(S7T Config system variable *motionstate.position*)

Shows the current position of the external encoder.

The position is no longer updated if the external encoder is disabled. The indicated position corresponds with the last position before the enable signal was reset.

ActualVelocity variable

(S7T Config system variable *motionstate.velocity*)

Shows the current velocity of the external encoder.

The velocity value is no longer updated when the external encoder is disabled. The indicated velocity corresponds with the last velocity before the enable signal was reset.

7.5 "Cam disk" technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

ErrorID variable

Shows the ErrorID of the last error or warning detected at the cam.

An error entry can be acknowledged by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

Please take note: Possible error messages and warnings (Page 820)

ErrorBuffer[0..2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

Status variable

Value	Significance	Comment
0	Blank	The cam does not contain any interpolation points or segments.
1	Interpolation points/segments inserted	The cam contains interpolation points/segments.
2	Interpolated	The cam is interpolated, meaning that it is ready for synchronous operation.
3	Error	There is an error.

UserCount variable

This variable returns the number of technology objects that are currently using the cam.

Statusword.xxx variable

The variable *Statusword* indicates the current status of the cam disk.

Bit No.	Variable	Significance for the *TRUE* state
0	*Statusword.Error*	At least one error has occurred (variable *ErrorID = 8xxx*). In the event of warnings, the *Statusword.Error = FALSE* status is retained.
1	*Statusword.RequestRestart*	Cam parameters have been changed and will not be activated until the next restart is completed (technology function "MC_Reset" with *Restart = TRUE*).
2	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
3	*Statusword.ToDeactivated*	Technology object was deactivated

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

7.6 "Measuring input" technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

The "Measuring input" technology DB contains information relating to the "Measuring input" technology object and to command execution. The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

UpdateFlag / UpdateCounter variable

Please take note: Updating technology DBs (Page 791)

ErrorID variable

The variable returns the ErrorID of the last error or warning detected at the measuring input.

An error entry can be acknowledged by calling the technology function "MC_Reset" (with *Axis =* number of the technology DB).

Please take note: Possible error messages and warnings (Page 431)

ErrorBuffer[0..2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

Status variable

Indicates the operating mode of the measuring input. The following operating modes can be displayed:

Value	Significance	Comment
0	Init	Measuring input in idle state
1	WaitForTrigger	Waiting for edges at the measuring input (measured values not recorded yet
2	DataValid	Measurement completed, values logged and valid
3	Error	Error detected; values invalid

MeasureValue1 variable

(S7T Config system variable *measuredvalue1*)

Returns the "Measured value 1" of the measuring input. This value is valid if status = *2* (operating state.)

MeasureValue2 variable

(S7T Config system variable *measuredvalue2*)

Returns the "Measured value 2" of the measuring input. This value is valid if status = *2* (operating state.)

The "Measured value 2" is only logged in modes (*Mode 3* and *4)* of technology function "MC_MeasuringInput."

MeasureStart / MeasureEnd variable

(S7T Config system variable *effectivedata.measuringrangestartposition / effectivedata.measuringrangeendposition*)

The *MeasureStart* variable returns the low limit, and the *MeasureEnd* variable measures the high limit of the measuring area. Only the measured values within these limits are logged. When the *MeasureStart* value equals the *MeasureEnd* value, the measuring area limits are ignored.

The *MeasureStart / MeasureEnd* variables shown at the DB correspond with the values at the input parameters of the relevant technology function "MC_MeasuringInput."

Statusword.xxx variable

The variable *Statusword* indicates the current status of the measuring input.

Bit No.	Variable	Significance for the *TRUE* state
0	*Statusword.Error*	At least one error has occurred (variable *ErrorID = 8xxx*). In the event of warnings, the *Statusword.Error = FALSE* status is retained.
1	*Statusword.RequestRestart*	Measuring input parameters have been changed and will not be activated until the next restart is completed (technology function "MC_Reset" with *Restart = TRUE*).
2	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
3	*Statusword.ToDeactivated*	Technology object was deactivated

Reservexx variable

The variables *Reserve02* to *Reserve10* are reserved for internal functions. They do not contain any user-relevant information.

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

7.7 "Output cam" technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

ErrorID variable

The variable returns the ErrorID of the last error or warning detected at the output cam.

An error entry can be acknowledged by calling the technology function "MC_Reset" (with *Axis* = number of the technology DB).

Please take note: Possible error messages and warnings (Page 827)

ErrorBuffer[0 to 2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis* = number of the technology DB).

State variable

(S7T Config system variable *state*)

Value	Significance	Comment
0	OFF	The current activation state of the output cam is "OFF".
1	ON	The current activation state of the cam is "ON."

CamType variable

Value	Significance	Comment
0	Position-based cam	The output cam is operated as position-based cam.
1	Time-based cam	The output cam is operated as time-based cam.
2	Uni-directional output cam	The output cam is operated as a uni-directional output cam.

Mode variable

Value	Significance	Comment
1	disabled	The current output cam operating state is "Cam disabled"
2	normal	The current operating state output cam is "Cam enabled (output not inverted)"
3	Inverse	The current operating state of the output cam is "Cam enabled (output inverted)"
4	Always on	The current operating state of the output cam is "Cam always on."

Direction variable

(S7T Config system variable *effectivedata.forcedirection*)

The variable returns the effective output cam direction of the most recent MC_CamSwitch or MC_CamSwitchTimes. This variable is not changed in the following "MC_CamSwitch" and "MC_CamSwitchTime" modes:

- *Mode = 1* (output cam disabled)
- *Mode = 4* (output cam always on)

Value	Significance
1	Positive effective direction
2	Positive and negative effective direction , meaning that the effective direction is irrelevant
3	Negative effective direction
4	The current rotational direction of the axis is the effective direction

OnPosition variable

(S7T Config system variable *effectivedata.switchonposition*)

This variable returns the current starting position.

OffPosition variable

(S7T Config system variable *effectivedata.switchoffposition*)

The variable returns the current end position.

Duration variable

(S7T Config system variable *effectivedata.onduration*)

The variable returns the pulse duration of a time-based cam. The physical unit corresponds with the physical unit of the "time" at the corresponding axis.

Delay variable

(S7T Config system variable *effectivedata.activationtime / effectivedata.deactivationtime*)

Indicates the current offset (derivative action time) of the cam activation/deactivation times.

Value < *0*: advanced activation

Value > *0*: retarded activation

Hysteresis variable

(S7T Config system variable *effectivedata.noswitchingrange*)

The variable returns the actual value of the hysteresis.

Statusword.xxx variable

The variable *Statusword* indicates the current status of the output cam.

Bit No.	Variable	Significance for the *TRUE* state
0	*Statusword.Error*	At least one error has occurred (variable *ErrorID = 8xxx*). In the event of warnings, the Statusword.Error = FALSE status is retained.
1	*Statusword.RequestRestart*	Output cam parameters have been changed and will not be activated until the next restart is completed (technology function "MC_Reset" with *Restart = TRUE*).
2	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
3	*Statusword.ToDeactivated*	Technology object was deactivated

Reservexx variable

The variables *Reserve02* to *Reserve10* are reserved for internal functions. They do not contain any user-relevant information.

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

7.8 "Cam track" technology DB

Valid for Integrated Technology with firmware version V4.1.x

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is non-retentive, irrespective of data retention settings in the CPU.

UpdateFlag / UpdateCounter variable

Please take note: Updating technology DBs (Page 791)

ErrorID variable

Returns the ErrorID of the error or warning most recently detected at the cam track.

An error entry can be acknowledged by calling the technology function "MC_Reset" (with *Axis =* number of the technology DB).

Please take note: Possible error messages and warnings (Page 831)

ErrorBuffer[0 to 2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

Statusword.xxx variable

The variable *Statusword* indicates the current status of the cam track.

Bit No.	Variable	Significance for the *TRUE* state
0	*Statusword.Error*	At least one error has occurred (variable *ErrorID = 8xxx*). In the event of warnings, the *Statusword.Error = FALSE* status is retained.
1	*Statusword.RequestRestart*	Cam track parameters have been changed and will not be activated until the next restart is completed (technology function "MC_Reset" with *Restart = TRUE*).
2	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.
3	*Statusword.TOdeactivated*	Technology object was deactivated

State variable

(S7T Config system variable *effectivedata.state*)

The variable indicates the current switching status of the output cam output.

Value	Significance	Comment
0	OFF	The current activation state of the output cam output is "OFF".
1	ON	The current activation state of the output cam output is "ON."

CamTrackType variable

(S7T Config configuration data *OctType.Camtracktype*)

The variable indicates the cam type of the cam track.

Value	Significance	Comment
0	Position-based cam	The individual output cams of the cam track operate as position-based cams.
1	Time-based cam	The individual output cams of the cam track operate as time-based cams.
2	Time-based output cam with maximum ON time	The individual output cams of the cam track operate as time-based cams. If the set maximum ON length is exceeded, the individual output cams are aborted prematurely.

Control variable

(S7T Config system variable *control*)

The variable indicates the internal status of the cam track.

Value	Significance	Comment
1	ACTIVE	The cam track is active
2	INACTIVE	The cam track is inactive
3	ACTIVE_AND_WAITING_ FOR_CAM_TRACK_ OUTPUT_INACTIVE	Applies to the *CommandMode = 3* input parameter at the "MC_CamTrack" technology function. • Cam track is enabled: A cam track command is still active. A new cam track command becomes active when the output cam output or all the individual output cams are inactive. • The cam track is disabled: The cam track command remains active until the output cam output or all the individual output cams are inactive.
4	ACTIVE_AND_WAITING_ FOR_NEXT_CYCLE	Applies to the *CommandMode = 2* input parameter at the "MC_CamTrack" technology function. Cam track is deactivated: The cam track is deactivated during entry of the cam track cycle and its repetitions.
5	INACTIVE_AND_ WAITING_FOR_NEXT_ CYCLE	Applies to the *CommandMode = 2* input parameter at the "MC_CamTrack" technology function. Cam track is enabled: The cam track is activated during entry of the cam track cycle and its repetitions.

CyclicMode variable

The variable indicates whether the cam track is in cyclic mode.

Value	Significance	Comment
1	Cyclic	The cam track is used cyclically.
2	Non-cyclic	The cam track is not used cyclically.

InvertOutput variable

(S7T Config configuration data *OctTechnologicalCfg.invertOutput*)

The variable indicates whether the cam track is in inverted mode.

Value	Significance	Comment
1	Not inverted	Output cam output and individual output cam are not inverted
2	inverted	Output cam output and individual output cam are inverted

RefrencePosition variable

(S7T Config system variable *effectivedata.axisreferenceposition*)

The variable shows the reference position of the cam track position *0* at the axis.

CamTrackLength variable

(S7T Config system variable *effectivedata.camtracklength*)

The variable returns the active cam track length.

ActivationTime variable

(S7T Config system variable *effectivedata.activationtime*)

The variable shows the effective delay or actuation time during activation.

DeactivationTime variable

(S7T Config system variable *effectivedata.deactivationtime*)

The variable shows the effective delay or actuation time during deactivation.

Hysteresis variable

(S7T Config system variable *effectivedata.hysteresisrange*)

The variable returns the actual value of the hysteresis.

CamTrackPosition variable

(S7T Config system variable *effectiveData.camtrackposition*)

The variable indicates the relative axis position of the cam track.

SingleCamState[0 ... 31] variable

(S7T Config system variable *singlecamstate*)

The variable indicates the activation status of the individual output cams 0 to 31.

Value	Significance	Comment
FALSE	Individual output cam not active	The individual output cam is not active. Individual output cams that are switched to invalid also show *FALSE*.
TRUE	Individual output cam active	The individual output cam is active.

7.9 "Trace" technology DB

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

You can use the TraceTool in S7T Config to record variables of the integrated technology in graphic real-time format. The "Trace" DB forms the interface between TraceTool and the user program.

From the application program, 2 values of the data type DINT, 2 values of the data type DWORD, and 4 values of the data type REAL can be written to the respective variables of the trace DB for tracing. The *ErrorID* variable may not be overwritten by the user program.

The integrated technology reads the DB data within the technology DB update cycle. The technology DB update cycle can be set in S7T Config using the **Target system > Set system clocks** command.

The DB is **non-retentive**, irrespective of data retention settings in the CPU.

ErrorID variable

Returns the ErrorID of the most recently detected error.

The cause of error must be eliminated. The error can not be acknowledged.

Please take note: Possible error messages and warnings (Page 834)

S7_TraceDINT[0..1] variable

The *S7_TraceDINT* variable contains an ARRAY with two elements of the data type DINT.

These values can be selected for tracing when you select the signal in *Technology* > *userdata1* > *user1* and *user2* in the TraceTool of S7T Config.

S7_TraceDWORD[0..1] variable

The *S7_TraceDWORD* variable contains an ARRAY with two elements of the data type DWORD. Use the variables if you want to trace bit information (such as status word or error word) in the S7T trace tool. You can select the bits in the usual SIMATIC numbering in the trace tool.

These values can be selected for tracing when you select the signal in *Technology* > *userdata1* > *user3* and *user4* in the TraceTool of S7T Config.

S7_TraceREAL[0..3] variable

The *S7_TraceREAL* variable contains an ARRAY with four elements of the data type REAL.

These values can be selected for tracing when you select the signal in *Technology* > *userdata1* > *user5, user6, user7* and *user8* in the TraceTool of S7T Config.

7.10 "MCDevice" technology DB

Valid for Integrated Technology with firmware V3.2.x or higher

This section describes applications with firmware V3.0.x and V3.1.x.

The "MCDevice" technology DB contains general information of the integrated technology. Information on the various technology objects is found in the corresponding DBs.

The integrated technology writes the data to the DB. The DB is read-only in the STEP 7 user program.

The DB is **non-retentive**, irrespective of data retention settings in the CPU.

UpdateFlag / UpdateCounter variable

Please take note: Updating technology DBs

ErrorID variable

Returns the ErrorID of the most recently detected error. The cause of error must be eliminated. The error can not be acknowledged.

Please take note: Possible error messages and warnings (Page 834)

ErrorBuffer[0..2] variable

Memory for the first three errors and warnings received. The first error is written to ARRAY element *0*, the second to ARRAY element *1*, and so forth.

The content of this memory can be cleared by calling technology function "MC_Reset" (with *Axis =* number of the technology DB).

Reserve34 variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

MaxLoopDuration variable

Returns the maximum command execution time of integrated technology.

The entry will be deleted during startup of the Technology CPU, or at the call of FB "MC_Reset" (with *Axis = MCDevice-DB; Restart = TRUE* or *FALSE*).

CmdLoopDuration variable

The variable returns the mean value by seconds of the command execution time of integrated technology.

The entry will be deleted during startup of the Technology CPU, or at the call of FB "MC_Reset" (with *Axis = MCDevice-DB; Restart = TRUE* or *FALSE*).

StationLifeList[0..127] variable

The ARRAY of this variable lists all partner stations available on PROFIBUS DP(DRIVE). The number of the ARRAY element corresponds to the PROFIBUS address. The value *1* in an element indicates that the partner station is available, and the value *0* indicates that it is not available.

DoneFlag[1..32] variable

The ARRAY elements 1 to 32 can be assigned to the *DoneFlag* input parameter of the function blocks.

Value of the *DoneFlag* at the FB	ARRAY element used	
0		A *DoneFlag* is not generated
1	1	A *DoneFlag[1]* is generated
…	…	...
32	32	A *DoneFlag[32]* is generated

Each time a command of the relevant technology function is completed, the value in *DoneFlag* is inverted (*FALSE* becomes *TRUE* and *TRUE* becomes *FALSE*) . The *DoneFlag* is only set as long as the initiated command can be monitored. The command can no longer be monitored if a subsequent command uses the same instance.

DIStatus[0..3] variable

The ARRAY elements *0* to *3* return the status of the integrated digital inputs of CPU 31xT.

DOStatus[0..7] variable

The ARRAY elements *0* to *7* return the status of the integrated digital outputs of the technology CPU. The outputs of the integrated technology can be assigned to individual output cams in S7T Config, for example.

Statusword.xxx variable

The variable *Statusword* indicates the current status of the synchronization axis.

Bit No.	Variable	Significance for the *TRUE* state
0	*Reserve01*	The variable is reserved for internal functions. It does not contain any user-relevant information.
1	*Reserve02*	The variable is reserved for internal functions. It does not contain any user-relevant information.
2	*Statusword.RequestStartUp*	The "Reduction", "Offset" or "IPO synchronous" entries were changed in "Technology Objects Management". The changed values are not activated until the next startup of the controller.

TODBTaskOverflows variable

Shows the number of technology DB update overflows. The maximum value returned is *FFFFFFFF*.

The entry will be deleted during startup of the Technology CPU, or at the call of technology function "MC_Reset" (with *Axis = MCDevice-DB; Restart = TRUE* or *FALSE*).

Note

You also set the basic clock for the technology DB updates in S7T Config by selecting the **Target system > Set system clocks** command.

A time setting which is too short can lead to an overflow during the technology DB update. When operating with integrated technology with firmware V3.2x or higher, the CPU will no longer go into STOP.

An "overflow" occurs when the active technology DB update is not yet completed and thus prevents the start of a new update cycle. The new update command will be discarded.

The number of overflows, and therefore the number of failed technology DB updates, can be read from the *TODBTaskOverflows* variable.

Reservexx variable

The variables *Reserve02* to *Reserve10* are reserved for internal functions. They do not contain any user-relevant information.

TO_Adaption variable

The variable is reserved for internal functions. It does not contain any user-relevant information.

7.11 Updating technology DBs

Valid for Integrated Technology with firmware V3.2.x or higher

This section describes applications with firmware V3.0.x and V3.1.x.

The integrated technology updates the technology DBs asynchronous to the cyclic user program. The technology DB update cycle can be set in S7T Config by selecting the **Target system > Set system clocks** command. The update cycle can be set to a multiple of the interpolator cycle.

Integrate technology with firmware V3.2.x supports distribution of the technology DB update cycle to different system clocks (exception: technology DB "Trace"). The technology DB update cycle represents the basic update cycle.

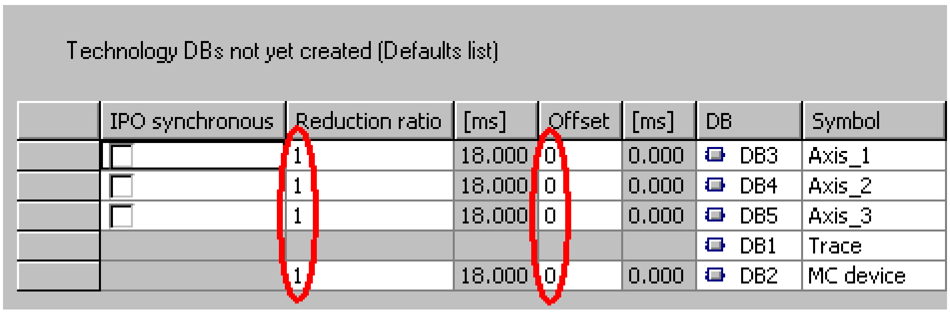
Note

You define the basic clock of the update in S7T Config by selecting the **Target system > Set system clocks** command.

An "overflow" occurs when a new Technology DB update is initiated while a previous update cycle is not yet completed. When operating with integrated technology firmware V3.1.x or earlier, the Technology CPU will go into STOP when it detects a technology DB update overflow. Integrated technology with firmware V3.2.x or higher tolerates this overflow, meaning that the non-executable update is discarded, and technology synchronization interrupt OB 65 will not be called.

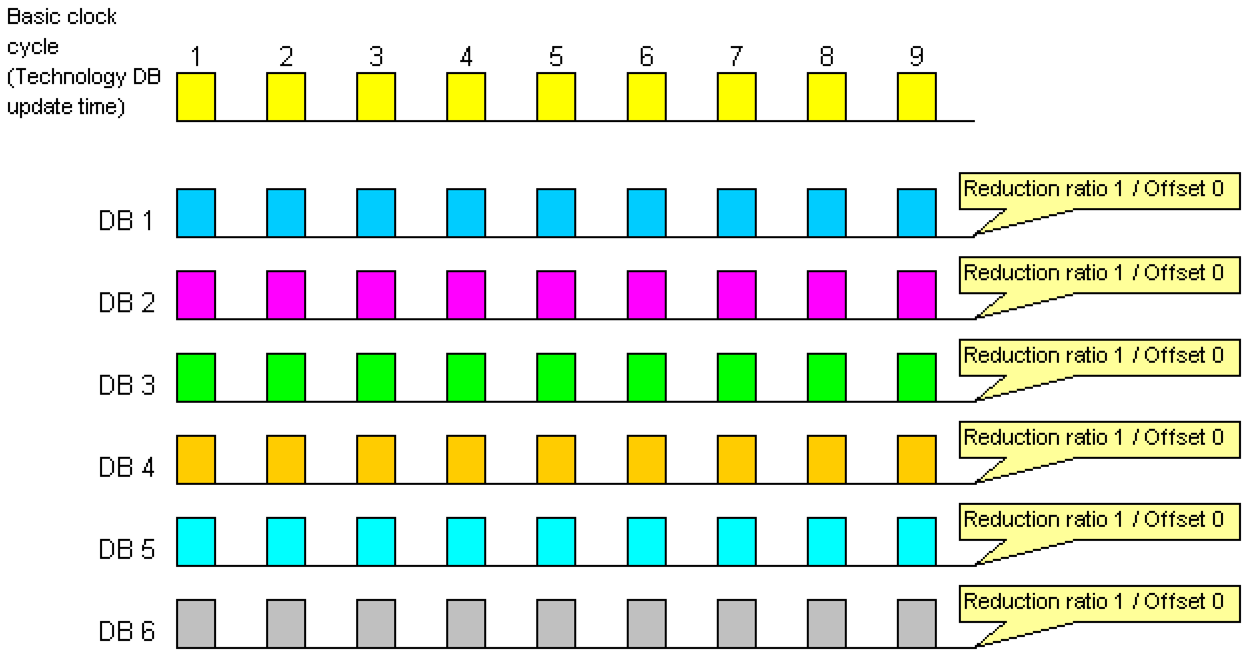
The number of overflows, and therefore the failure rate of Technology DB updates, is shown in the *TODBTaskOverflows* variable at the "MCDevice" technology DB.

With reference to this basic clock, you can distribute the update of the various technology DBs to several basic clocks by programming a step-down ratio and an offset in "Technology Objects Management". Activate the expert mode by selecting the **View > Expert mode** command.


Technology DBs not yet created (Defaults list)

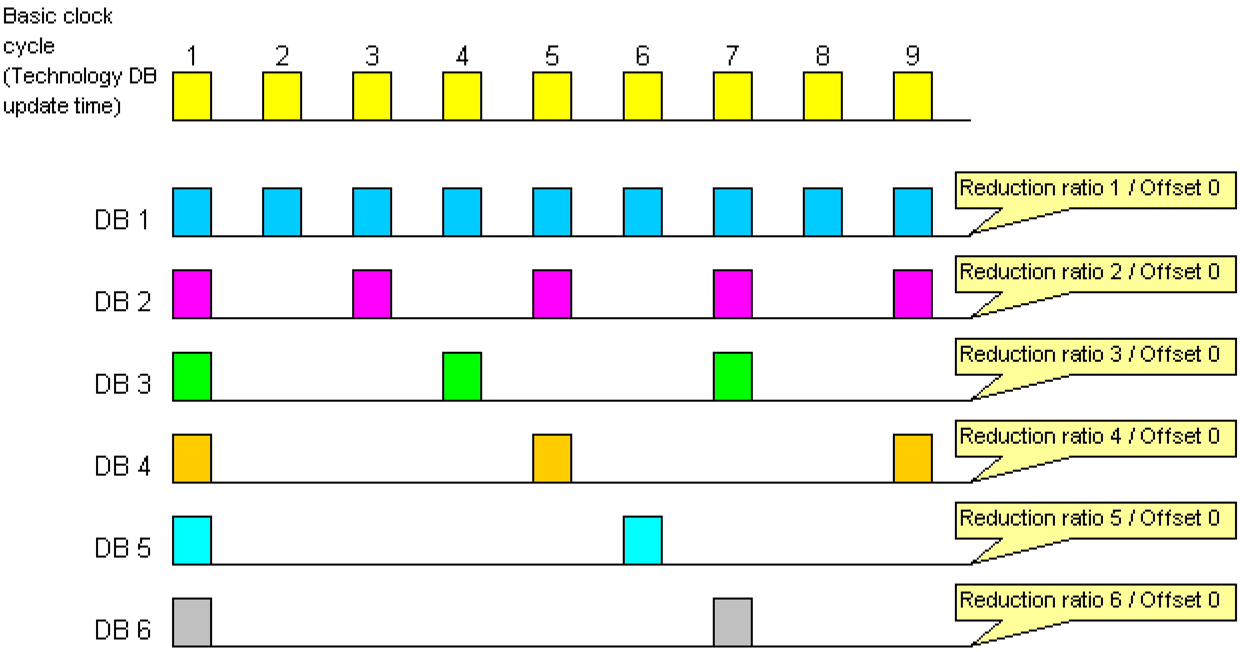
	IPO synchronous	Reduction ratio	[ms]	Offset	[ms]	DB	Symbol
	☐	1	18.000	0	0.000	DB3	Axis_1
	☐	1	18.000	0	0.000	DB4	Axis_2
	☐	1	18.000	0	0.000	DB5	Axis_3
						DB1	Trace
		1	18.000	0	0.000	DB2	MC device

Update of technology DBs with default settings



A reduction ratio of *1* and offset of *0* are set by default for the technology DBs. All technology DBs are updated within the basic clock cycle (technology DB update cycle). The selected basic clock cycle of the technology DB update must be of sufficient length to allow the update of all technology DBs.

Update of technology DBs with graded step-down ratio and uniform offset

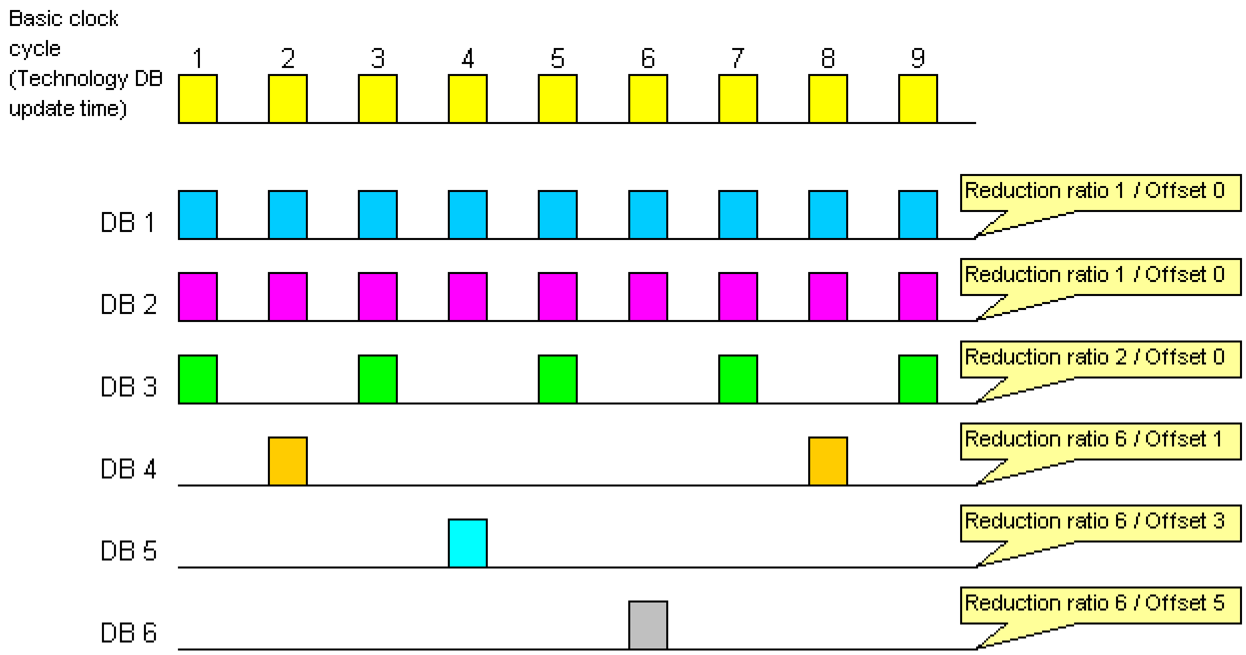


The step-down ratio determines in which basic clock the technology DB is to be updated.

- A reduction ratio of *1* means that the technology DB is updated within each basic cycle clock.
- A reduction ratio of *2* means that the technology DB is updated at every second basic cycle clock, and so forth.

By comparison to the default setting this improves distribution of technology DB updates. However, the basic cycle clocks *1, 5, 7,* and *9* will still generate high system load.

Example of an update with graded step-down ratio and offset



The example shows that a graded step-down ratio and offset can be used to limit update load to three technology DBs per basic clock. Compared to the example shown earlier this allows doubling of the basic clock. The update time of DB1 and DB2 is halved. The update time of DB3 remains the same. The update time of DB3 to DB6 is tripled.

Note

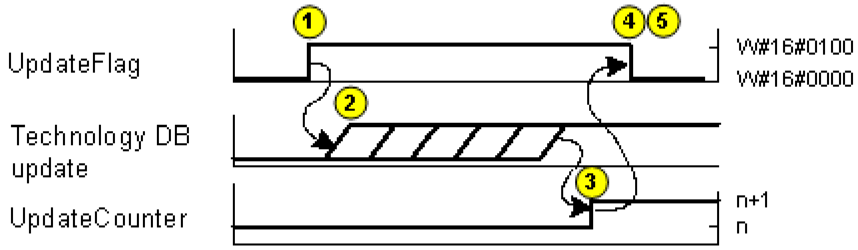
Use the information below in order to optimize the step-down ratio and offset:

- Set the duration of the cycles to at least such a high value that they can be evaluated in the user program.
- Select the same step-down ratio and offset for technology DBs which you want to evaluate consistently to each other
- Create a diagram of your update schedule which shows you the cyclic distribution of updates
- The time required for the update of a technology DB depends on the DB length Cyclic distribution of technology DBs of greater length (for example, axis technology DBs) will lead to a more even distribution of load compared to the distribution of short technology DBs

---

"UpdateFlag", "UpdateCounter" and technology synchronization interrupt OB (OB65)

The diagram below shows the sequence of "UpdateFlag", "UpdateCounter" of a technology DB.



1	The technology DB update is initiated by the transition of the "UpdateFlag" from *W#16#0000* to *W#16#0100*..
2	All technology DBs affected in this update cycle are updated. The integrated technology always writes the entire technology DB.
3	The "UpdateCounter" increments its value by the count of 1. All "UpdateCounter" values relevant to this update cycle are incremented.
4	The "UpdateFlag" changes from *W#16#0100* to *W#16#0000* for all technology DBs concerned.
5	The technology synchronization interrupt OB65 is called. For details on local data (start info in OB65), refer to the STEP 7 Online Help. OB 65 can be copied from the "Organization Blocks" folder of the "Standard Library" to the project.

Note

Note the following:

- The technology synchronization interrupt is called in each basic clock, regardless whether or not a technology DB is updated within this basic clock
- The technology DBs update and the generation of the feedback signals of the technology functions are handled asynchronously The data are therefore inconsistent

  After the call of the *Done* output parameter, wait until the "UpdateCounter" is incremented before you analyze the technology data block (to determine the position, for example).

Consistent evaluation of the technology DBs using the technology synchronization interrupt OB 65

As the technology synchronization interrupt OB is called after each update of the technology DBs, the relevant actual values will be available at the time this OB is called. This function allows you to evaluate technology DBs synchronously to the integrated technology. It is therefore advisable to use the technology synchronization interrupt OB in order to ensure consistent evaluation of technology DB data.

If the user program contains a long technology synchronization interrupt OB, and the technology DB update cycle is short, a new technology DB update cycle may be started before the technology synchronization interrupt OB is terminated. The data just read will be overwritten again straight-away.

Use the "UpdateCounter" and the "UpdateFlag" of the technology DBs to verify consistent evaluation of the technology DBs. The response of the "UpdateFlag" and of the "UpdateCounter" is shown in the diagram above:

Checking the consistent evaluation of the technology DB

Based on the procedure described below, check whether a new technology DB update cycle was started while the operation which is relevant to consistency is active at OB 65. The technology DB update starts at the "MCDevice" DB. However, in order to ensure consistency, the "UpdateCounter" and the "UpdateFlag" of each technology DB to be evaluated have to be verified. Procedure:

1. At the start of OB 65, save each value of the "UpdateCounter" functions of the technology DBs to be evaluated.
2. Evaluate the consistent data, or transfer the technology DBs to be evaluated to another working area
3. Check the "UpdateFlag". If the "UpdateFlag" has a value of *W#16#0100,* , consistent evaluation of this technology data block is no longer guaranteed.
4. Compare the "UpdateCounter" with the previously saved value. If the "UpdateCounter" was incremented by the count of *1*, consistent evaluation of the technology DB is no longer guaranteed
5. Repeat steps 3 and 4 for all technology DBs to be evaluated

   If the technology DBs have identical "UpdateCounter" value, the values in the technology DBs originate from the same update cycle.

Counter any consistency issues by taking the following measures:

- Extend the technology DB update time.
- Evaluate the technology DBs at the start of OB 65.
- Reduce the number of instructions to be evaluated.

Note

The *ErrorID*, *ErrorStatus*, and *ErrorBuffer* entries are updated asynchronously, that is, independently of the status of the "UpdateFlag".

7.12 ErrorID - Technology DBs

7.12.1 ErrorIDs – Axis technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

Warnings

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config *
0010	Failure of a non position-controlled encoder	Failure of an encoder of a non-active data set	The failed encoder can be identified in S7T from the alarm number 20015.	20015
0020	Restart to activate the parameter changes	A parameter was changed and is not applied immediately	Activate the parameter changes by disabling and then initializing the technology object ("MC_Reset" with *Restart* = *TRUE*)	20010
0021	Dynamic response values are limited	The dynamic response values of the command (velocity, acceleration, deceleration, or jerk) are being limited because they exceed the set limits. Any superimposing commands which are active at the technology object may also limit dynamic values. The warning can only be acknowledged if only one motion command remains active.	Check the cause of the excess setpoint values: configured maximum values; speed setpoint interface; encoder connection; mechanical configuration.	40002, 40003, 40004, 50003, 50005
0022	Internal warning	Temporary internal problem	For detailed information, refer to the alarm view of S7T Config	20009, 40109, 50010
0023	Maximum acceleration limits the holding time	The defined holding time cannot be maintained. The holding time is prolonged because it cannot be achieved due to the effective maximum acceleration.	Increase the holding time or the maximum acceleration setting in your configuration.	40013
0026	Position limited to software limit switch	The target position of a positioning command was outside the range of the software limit switch.	Check the calculated target position.	40105
* Interrupts in S7T Config which may be associated with the warning indicated.				

Errors

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8001	Fatal internal error	Project/software faulty/inconsistent; cannot be rectified by the user.	Check consistency of the project. Recompile all data and download these to the module. Read the error code from the alarm view in S7T Config and then inform the Siemens Hotline.	20001, 50015
8002	Internal error	• Version conflict • Firmware error • Inconsistent software	Check consistency of the project. Recompile all data and download these to the module. Evaluate the alarm view of S7T Config and inform the Siemens Hotline.	30004, 30005, 30010, 30011, 30015
8003	Fatal configuration error	Faulty axis configuration. A hardware driver, or the device it addresses, may not be available	• Check the plausibility of changed parameters. Examples: Leadscrew pitch is *0*; homing beyond encoder zero mark is not possible with absolute encoders.) • Check the connected devices and their configuration in HW Config. • Verify that the message frame type used is compatible with the technology object. • Download the technology once again to the target system.	20004, 20006
8004	Configuration error	Incomplete or faulty configuration of a technology object.	Check the plausibility of changed parameters. Evaluate the detailed information in the alarm view of S7T Config.	20003, 20011
8005	Command memory is full	The command can not be executed due to insufficient command capacity - Cause: • The technology object can not save additional commands	In S7T Config, increase the "maximum number of active command IDs." Set the value in the Expert List with the *numberOfMaxBuffered CommandId* confiugration data element. This setting also increases the command handling capacity of the technology object.	30003
8006	Insufficient memory	Integrated Technology is out of memory.	Reduce the number of technology objects, or the number of interpolation points or the polynomials in the cam disks.	20002

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8008	Invalid technology DB	The Technology rejects the DB because this does not have a valid internal ID. Perhaps the DB was generated in a copy operation.	Recreate the DB in the "Technology Objects Management" dialog box and download the configuration data again (establish online / offline consistency).	-
8009	The technology DB is disabled because it was replaced with a newer instance	The DB is no longer assigned to the technology object, because a newer instance with a different DB number has been downloaded.	Delete the DB	-
800A	HW driver not available	The driver of a physical device is unavailable or in use by another technology object.	Check whether the logical device was available at the time the request was output, or whether it was in use by another technology object (TO). Connect a suitable device. Check the device with regard to its function, connections and wiring. Check the topology in HW Config. Compare the configuration in HW Config with that of the technology object.	20014
8010	Drive failed	Failure of the drive, or of an external device, or communication is disrupted, as a result of faulty configuration or parameter data.	Check the device functions, connections and wiring. Check the configuration in HW Config, based on current message frame settings.	20005, 50001
8012	Homing error	The requested homing mode is not supported. Examples: Homing with absolute encoder; homing velocity = *0*.	Check the settings for axis homing in terms of inconsistency and invalid values.	40103
8013	Hardware limit switch is active	• The hardware limit switch has responded. • The polarity of the switch was reversed. • Wrong direction of retraction	Return the drive to its operating range. Check the limit switch connections.	50007
8014	Software limit switch actuated	A motion was stopped when the software limit switch was reached	Check the motion program.	40106, 40111
8015	Software limit switch overshot	The software limit switch was overtraveled during motion execution.	Check the motion program.	40107, 40112
8016	Following error out of limits	The offset between the position setpoint and the actual position of the axis has exceeded configured limits.	Check the transfer direction of the encoder, the settings of the position control loop, and the following error monitoring settings.	50102

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8017	Following error has exceeded the warning limit	The offset between the position setpoint and the actual position of the axis has exceeded configured warning limits.	Check the transfer direction of the encoder, the settings of the position control loop, and the following error monitoring settings.	50103
8018	Standstill monitoring error	The axis has moved out of the standstill window or could not reach the standstill window within the time specified.	Check the functions of the speed and position control loops, and the settings for standstill monitoring.	50107, 50008
8019	Positioning monitoring error	The axis failed to reach the positioning window within the specified time.	Check the functions of the speed and position control loops, and the settings for positioning monitoring.	50106
801A	Synchronous operation monitoring error	The following axis of a synchronous grouping cannot operate within configured tolerance limits, meaning that the offset between the position setpoint and actual value is too high.	Check the synchronization and desynchronization parameters, the gear ratio, and the configured units. Increase the set tolerance, if appropriate.	40110, 40201
801B	Clamping monitoring error	Clamping error. The axis has exceeded the "Position tolerance after fixed stop detection" without receiving a new motion command. (Position tolerance setting in S7T Config **Limits > "Fixed end stop" tab > "Position tolerance after fixed end stop detection" parameter**.)	Check: • The mechanical end stop • The function of the position control loop • The clamping monitoring settings in S7T Config	50108
801C	Dynamic limits are exceeded	The internal monitoring system has intervened in the dynamic response process, because user input would violate limit values or lead to an impermissible reversal or overshoot of the axis.	Particularly when setting the values of overriding motions, make sure you do not set these too low in terms of acceleration, deceleration, and jerk. With superimposed motions, always make allowances for superimposed values at the dynamic response parameters.	40012, 40202
801D	Reference cam or encoder zero mark not found	• Reference cam or encoder zero mark outside the permitted range. • A limit position monitoring function has responded. • The speed was too high.	Check the valid ranges (max. distance between reference cam and encoder zero mark, for example) and the limit position monitoring system, and reduce the shutdown speed.	40101, 40102

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
801F	Dynamic control loop out of range	• The closed-loop position or speed control system was not optimized. • The transfer direction of the encoder is incorrect. • Assigned parameters for following error monitoring are too sensitive • The discrepancy between the speed setpoint and actual speed value is too large.	• Check the transfer direction of the encoder. • Check the control loop parameters. • Modify the parameters of following error monitoring.	50101
8020	Setpoint limiting active	The speed setpoint is limited by the manipulated variable limiting.	• Check the maximum values of manipulated variable limiting in the "Manipulated variable limiting" field in the "Static controller data" tab of the **Axis > Control** dialog box. • Check the configuration of the speed setpoint interface • Check the mechanical configuration. • Check the encoder connection	50016
8030	Limit frequency of the measuring system exceeded	The pulse rate of the encoder signals is out of the permitted range.	Check the electrical connection of the encoder. Check whether the configured maximum speed is too low.	50002
8031	Actual value monitoring error	• Incremental encoder overrange • The movement per pulse of the position controller exceeds the modulo length of the axis	Check and adjust the configuration data for the adaptation of the actual value to mechanical requirements.	50001, 50013
8032	Cannot shift the coordinate system	The programmed coordinate system shift (set or correct the actual position value) leads to a violation of the internal traversing limit. The coordinate system is not shifted.	Check the programmed shift of the coordinate system.	30014
8033	Encoder zero mark monitoring triggered	The number of increments between encoder zero marks is incorrect.	Check the electrical connection and the set encoder resolution.	50006
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing.	Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).	40005
8041	Axis not homed	The active motion command requires a homed axis.	Home the axis ("MC_Home") If this command is to be executed at an axis which is not homed, then set axis homing to "Homing required = no"	40108

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8042	Invalid change of the axis status	The command for changing the axis status was rejected, because:	Carry out the following measures, depending on the cause:	40001
		• The current operating phase is not completed yet.	• Repeat the command	
		• A mode transition is not possible.	• First carry out a rest.	
8043	Invalid command parameter value or default value	One or several invalid values at the input parameters in the technology function of the command, or invalid default parameters.	Check the command and adjust invalid parameters. You should also observe the currently used default values.	30001
8045	Command not allowed in current state	• A command with higher priority is active (for example "MC_Stop") • Technology object disabled or not ready. • Received measurement command when homing	Acknowledge all queued errors of the technology object. Wait for initialization of the technology object (after restart). Enable the technology object. "Deactivate "MC_Stop." Do not start any measuring commands during homing.	30006, 30009
8046	Command not supported by the drive	• The connected drive does not support this type of command, for example, a specific homing method. • A measuring-input command is active at the axis.	Check and adjust the drive settings and configuration data.	50012
8047	Cannot acknowledge this error	It is not possible to acknowledge the error with "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in a restart phase.	Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.	30007
8048	Unable to restart	The technology object is active or a queued error prevents restart.	Eliminate and acknowledge all queued errors and disable the technology object.	20012

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8049	Illegal object combination	Faulty combination of technology objects. Examples:	Carry out the following measures, depending on the cause:	20007
		• Axis_2 was set as the leading axis in synchronous operation for Axis_1; at the same time, Axis_1 was set as the leading axis for Axis_2.	• Set a defined leading/following axis combination.	
		• The axes involved in synchronous operation are processed at different run levels.	• Assign the leading and following axis to the same run level (usually the interpolator cycle clock).	
804A	Required object connection is missing	• No allowances have been made in the configuration for the required combination of leading and following axes and cam disks. • The configured cam disk was not assigned to the axis as valve profile in S7T Config.	Select the required leading axes, cam disks, or valve profiles from the "Configuration" section of the synchronization object of the following axis.	30008
804B	Limit switch active; invalid command	A hardware limit switch is active. Only motion commands in direction of the operating range are allowed.	Check the mechanical configuration and the electrical connections. Correct any program errors or use the software limit switches.	50009
8082	Invalid *Zero* value in parameter	The command requested contains a parameter with *Zero* value. This value, however, must always be unequal *Zero*.	Check the parameters and default values. Observe in particular the limits and dynamic values (for example, max. speed or deceleration of a positioning operation.)	40006, 40007, 40008, 40009, 40010, 40011
8085	Impermissible end limit configuration	Implausible position values set at the SW limit switch. One of the values violates the permissible range or the start value is greater than the end value. The start value can be larger than the end value if you do not follow the correct order when reassigning parameters.	Adjust the setting of the software limit switch. After the first change of the position, the position value of the positive switch must be higher than that of the negative switch.	40104
* Interrupts in S7T Config that may be associated with the indicated error				

7.12.2 ErrorIDs – Synchronization technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

Warnings

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
0010	Failure of a non position-controlled encoder	Failure of an encoder of a non-active data set	The failed encoder can be identified in S7T from the alarm number 20015.	-	20015
0020	Restart to activate the parameter changes	A parameter was changed and is not applied immediately	Apply the parameter changes by disabling and then initializing the technology object ("MC_Reset" with *Restart* = *TRUE*)	20010	20010
0021	Dynamic response values are limited	• The dynamic response values of the command (velocity, acceleration, deceleration, or jerk) are being limited because they exceed the set limits.	• Check the cause of the excess setpoint values: configured maximum values; speed setpoint interface; encoder connection; mechanical configuration.	40002, 40003, 40004	40002, 40003, 40004, 50003, 50005
		• Any superimposing commands which are active at the technology object may also limit dynamic values. The warning cannot be acknowledged until the resulting dynamic response of the superimposed commands is no longer limited.	• Acknowledge the warning when the superimposing motion has ended.		
0022	Internal warning	Temporary internal problem	For detailed information, refer to the alarm view of S7T Config	50103, 50104, 50105, 50106, 50108, 50109	20009, 40109, 50010

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
0023	Maximum acceleration limits the holding time	The defined holding time cannot be maintained. The holding time is prolonged because it cannot be achieved due to the effective maximum acceleration.	Increase the holding time or the maximum acceleration setting in your configuration.	40013	0023
0024	No long-term stability of gearing	The numerator and denominator values of the gear ratio are too high. This may result in mathematical rounding errors in long-term operation.	Modify the gear ratio, or correct any inaccuracies by running a new synchronization.	50008	-
0026	Position limited to software limit switch	The target position of a positioning command was outside the range of the software limit switch.	Check the calculated target position.	-	40105
* Interrupts in S7T Config which may be associated with the warning indicated.					

Errors

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8001	Fatal internal error	Project/software faulty/inconsistent; cannot be rectified by the user.	Check consistency of the project. Recompile all data and download these to the module. Read the error code from the alarm view in S7T Config and then inform the Siemens Hotline.	20001	20001, 50015
8002	Internal error	• Version conflict • Firmware error • Inconsistent software	Check consistency of the project. Recompile all data and download these to the module. Evaluate the alarm view of S7T Config and inform the Siemens Hotline.	30004, 30005, 30010, 30011, 30015	30004, 30005, 30010, 30011, 30015

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8003	Fatal configuration error	Faulty axis configuration. A hardware driver, or the device it addresses, may not be available	• Check the plausibility of changed parameters. Examples: Leadscrew pitch is *0*; homing beyond encoder zero mark is not possible with absolute encoders.) • Check the connected devices and their configuration in HW Config. • Verify that the message frame type used is compatible with the technology object. • Download the technology once again to the target system.	20004, 20006	20004, 20006
8004	Configuration error	Incomplete or faulty configuration of a technology object.	Check the plausibility of changed parameters. Evaluate the detailed information in the alarm view of S7T Config.	20003, 20011	20003, 20011
8005	Command memory is full	The command can not be executed due to insufficient command capacity - Cause: • The technology object can not save additional commands	In S7T Config, increase the "maximum number of active command IDs." Set the value in the Expert List with the *numberOfMaxBuffered CommandId* configuration data element. This setting also increases the command handling capacity of the technology object.	30003	30003
8006	Insufficient memory	Integrated Technology is out of memory.	Reduce the number of technology objects, or the number of interpolation points or the polynomials in the cam disks.	20002	20002
8008	Invalid technology DB	The Technology rejects the DB because this does not have a valid internal ID. Perhaps the DB was generated in a copy operation.	Recreate the DB in the "Technology Objects Management" dialog box and download the configuration data again (establish online / offline consistency).	-	-

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8009	The technology DB is disabled because it was replaced with a newer instance	The DB is no longer assigned to the technology object, because a newer instance with a different DB number has been downloaded.	Delete the DB	-	-
800A	HW driver not available	The driver of a physical device is unavailable or in use by another technology object.	• Check whether the logical device was available at the time the request was output, or whether it was in use by another technology object (TO). • Connect a suitable device. Check the device with regard to its function, connections and wiring. • Check the topology in HW Config. • Compare the configuration in HW Config with that of the technology object.	20014	20014
8010	Drive failed	Failure of the drive, or of an external device, or communication is disrupted, as a result of faulty configuration or parameter data.	Check the device functions, connections and wiring. Check the configuration in HW Config, based on current message frame settings.	20005	20005, 50001
8012	Homing error	The requested homing mode is not supported. Examples: Homing with absolute encoder; homing velocity = *0*.	Check the settings for axis homing in terms of inconsistency and invalid values.	-	40103
8013	Hardware limit switch is active	• The hardware limit switch has responded. • The polarity of the switch was reversed. • Wrong direction of retraction	Return the drive to its operating range. Check the limit switch connections.	-	50007
8014	Software limit switch actuated	A motion was stopped when the software limit switch was reached.	Check the motion program.	-	40106, 40111
8015	Software limit switch overshot	The software limit switch was overtraveled during motion execution.	Check the motion program.	-	40107, 40112

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8016	Following error out of limits	The offset between the position setpoint and the actual position of the axis has exceeded configured limits.	Check the transfer direction of the encoder, the settings of the position control loop, and the following error monitoring settings.	-	50102
8017	Following error has exceeded the warning limit	The offset between the position setpoint and the actual position of the axis has exceeded configured warning limits.	Check the transfer direction of the encoder, the settings of the position control loop, and the following error monitoring settings.	-	50103
8018	Standstill monitoring error	The axis has moved out of the standstill window or could not reach the standstill window within the time specified.	Check the functions of the speed and position control loops, and the settings for standstill monitoring.		50107, 50008
8019	Positioning monitoring error	The axis failed to reach the positioning window within the specified time.	Check the functions of the speed and position control loops, and the settings for positioning monitoring.	-	50106
801A	Synchronous operation monitoring error	The following axis of a synchronous grouping cannot operate within configured tolerance limits, meaning that the offset between the position setpoint and actual value is too high.	Check the synchronization and desynchronization parameters, the gear ratio, and the configured units. Increase the set tolerance, if appropriate.	-	40110, 40201
801B	Clamping monitoring error	Clamping error. The axis has exceeded the "Position tolerance after fixed end stop detection" without having received a new motion command. (Position tolerance setting in S7T Config **Limits > "Fixed end stop" tab > "Position tolerance after fixed end stop detection" parameter**.)	Check: • The mechanical end stop • The function of the position control loop • The clamping monitoring settings in S7T Config	-	50108

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
801C	Dynamic limits are exceeded	• The internal monitoring system has intervened in the dynamic response process, because user input would violate limit values or lead to an impermissible reversal or overshoot of the axis.	• Particularly when setting the values of release motions, make sure you do not set these too low in terms of acceleration, deceleration and jerk. With superimposed motions, always make allowances for superimposed values at the dynamic response parameters.	-	40012
		• The dynamic values returned by the leading axis during synchronization are higher than the values defined at the following axis. The dynamic values set at the following axis for synchronization will be ignored, and the following axis is synchronized with the leading axis accordingly.	• Change the value at configuration parameter *SyncingMotion. synchronizing Adaption* of the synchronization object to *NO* if the following axis is to be synchronized based on its set dynamic values. You can also change the configuration data element using parameter number *4337* at the "MC_WriteParameter" technology function.		
801D	Reference cam or encoder zero mark not found	• Reference cam or encoder zero mark outside the permitted range. • A limit position monitoring function has responded. • The speed was too high.	Check the valid ranges (max. distance between reference cam and encoder zero mark, for example) and the limit position monitoring system, and reduce the shutdown speed.	-	40101, 40102
801F	Dynamic control loop out of range	• The closed-loop position or speed control system was not optimized. • The transfer direction of the encoder is incorrect. • Assigned parameters for following error monitoring are too sensitive • The discrepancy between the speed setpoint and actual speed value is too large.	• Check the transfer direction of the encoder. • Check the control loop parameters. • Modify the parameters of following error monitoring.	-	50101

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8030	Limit frequency of the measuring system exceeded	The pulse rate of the encoder signals is out of the permitted range.	Check the electrical connection of the encoder. Check whether the configured maximum speed is too low.	-	50002
8031	Actual value monitoring error	• Incremental encoder overrange • The movement per pulse of the position controller exceeds the modulo length of the axis	Check and adjust the configuration data for the adaptation of the actual value to mechanical requirements.	-	50001, 50013
8032	Cannot shift the coordinate system	The programmed coordinate system shift (set or correct the actual position value) leads to a violation of the internal traversing limit. The coordinate system is not shifted.	Check the programmed shift of the coordinate system.	-	30014
8033	Encoder zero mark monitoring triggered	The number of increments between encoder zero marks is incorrect.	Check the electrical connection and the set encoder resolution.	-	50006
8040	The axis / external encoder are disabled, or the wrong mode is set	The enable signal required for a motion command is missing.	Eliminate and acknowledge all queued errors and then enable the relevant axis mode (for example position-controlled).	-	40005
8041	Axis not homed	The active motion command requires a homed axis.	Home the axis ("MC_Home") If this command is to be executed at an axis which is not homed, then set axis homing to "Homing required = no"	-	40108
8042	Invalid change of the axis status	The command for changing the axis status was rejected, because:	Carry out the following measures, depending on the cause:	-	40001
		• The current operating phase is not completed yet.	• Repeat the command		
		• A mode transition is not possible.	• First carry out a reset.		
8043	Invalid command parameter value or default value	One or several invalid values at the input parameters in the technology function of the command, or invalid default parameters.	Check the command and adjust invalid parameters. You should also observe the currently used default values.	30001, 50107	30001

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8045	Command not allowed in current state	• A command with higher priority is active (for example "MC_Stop"). • Technology object disabled or not ready. • Received measurement command when homing	Acknowledge all queued errors of the technology object. Wait for initialization of the technology object (after restart). Enable the technology object. "Deactivate "MC_Stop." Do not start any measuring commands during homing.	30006	30006, 30009
8046	Command not supported by the drive	• The connected drive does not support this type of command, for example, a specific homing method. • A measuring-input command is active at the axis.	Check and adjust the drive settings and configuration data.	-	50012
8047	Cannot acknowledge this error	It is not possible to acknowledge the error by calling technology function "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in a restart phase.	Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.	30007	30007
8048	Unable to restart	The technology object is active or a queued error prevents restart.	Eliminate and acknowledge all queued errors and disable the technology object.	20012	20012

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8049	Illegal object combination	Faulty combination of technology objects. Examples:	Carry out the following measures, depending on the cause:	20007	20007
		• Axis_2 was set as the leading axis in synchronous operation for Axis_1; at the same time, Axis_1 was set as the leading axis for Axis_2.	• Set a defined leading/following axis combination.		
		• The axes involved in synchronous operation are processed at different run levels.	• Assign the leading and following axis to the same run level (usually the interpolator cycle clock).		
804A	Required object connection is missing	• No allowances have been made in the configuration for the required combination of leading and following axes and cam disks. • The configured cam disk was not assigned to the axis as valve profile in S7T Config.	Select the required leading axes, cam disks, or valve profiles from the "Configuration" section of the synchronization object of the following axis.	30008	30008
804B	Limit switch active; invalid command	A hardware limit switch is active. Only motion commands in direction of the operating range are allowed.	Check the mechanical configuration and the electrical connections. Correct any program errors or use the software limit switches.	-	50009
8063	Cam is not interpolated	At the start of camming, it was detected that the cam was not interpolated yet.	Check the current cam status at the technology DB. Add the missing points or segments, and then interpolate the cam.	50002	-

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronize d axis
8064	Failed to access the cam	• The cam disk specified does not exist.	• Check the cam disk and all possible interconnections with the axes (configuration of synchronism).	50001, 50004	-
		• The cam disk is not interconnected with synchronous operation.	• In order to be able to edit the cam disk, the cam disk may neither be active in cam synchronization, nor be used as valve profile of a hydraulic axis.		
		• The cam disk is already active.	• If camming is started relative to the leading axis the value of system variable *userdefault. cammingsettings. camstartposition master* at the synchronization object must lie within the definition range of the cam disk.		
		• The position of the leading axis is outside the cam definition range	• In order to execute non-cyclic cam synchronization with absolute reference to the leading axis, the current position of the master must be within the definition range of the cam. Make allowances for the cam shift caused by input parameter *MasterOffset*.		
8065	Range violation for cam	A specified cam interpolation point violates the defined cam range.	Check the start and end positions of cam synchronization and desynchronization.	50003	-
8072	Error occurred while activating/deactivating the synchronous operation	This error is caused by an invalid status of the leading axis during synchronization and desynchronization. Example: The leading axis reverses during position-based synchronization.	Check the motion of the leading axis, or select a different synchronization strategy.	50007	-

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config * Following object	Interrupts in S7T Config * Synchronized axis
8073	Violation of leading axis dynamics in synchronous operation	The dynamic values at the leading axis are too high during synchronization and desynchronization (for example acceleration values too high)	Check the motion of the leading axis, or select a different synchronization strategy.	50009	-
8074	The leading axis is not configured for synchronous operation	• The axis connected to the "Master" terminal was not defined as leading axis in the following axis configuration. • The leading axis is not enabled (applies only for the external encoder). • The leading axis is operated in speed-controlled mode.	Adjust the configuration of the following axis or use the configured leading axes. Operate the leading axis in position-controlled mode or enable it.	50101, 50102	-
8077	Deactivation of synchronous operation aborted	Deactivation of synchronism has been rejected because the current synchronization mode (gear / cam gear) does not agree with the deactivation type.	Deactivate camming with MC_CamOut and gearing with MC_GearOut", or use global instructions ("MC_Halt", "MC_Stop", etc.)	50005	-
8078	Superimposed synchronous operation is not possible	Superimposed synchronous operation was configured for the axis in the expert list.	Correct the settings in the expert list.	-	-
8082	Invalid *Zero* value in parameter	The command requested contains a parameter with *Zero* value. This value, however, must always be unequal *Zero*.	Check the parameters and default values. Observe in particular the limits and dynamic values (for example, max. speed or deceleration of a positioning operation.)	40006, 40007, 40008, 40009, 40010, 40011	40006, 40007, 40008, 40009, 40010, 40011
8085	Impermissible end limit configuration	The position values set at the SW limit switch are not plausible. One of the values violates the permissible range or the start value is greater than the end value. The latter may occur if you neglect the order by which parameters are changed.	Adjust the setting of the software limit switch. After the first change of position, the position value of the positive switch must be higher than that of the negative switch.	-	40104

* Interrupts in S7T Config that may be associated with the indicated error

7.12.3 ErrorIDs – External encoder technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

Warnings

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config *
0020	Restart to activate the parameter changes	A parameter was changed and is not applied immediately	Apply the parameter changes by disabling and then initializing the technology object ("MC_Reset" with *Restart = TRUE*)	20010
0021	Dynamic response values are limited	The dynamic response values of the command (velocity, acceleration, deceleration, or jerk) are being limited because they exceed the set limits.	Check the cause of the excess setpoint values: configured maximum values; speed setpoint interface; encoder connection; mechanical configuration.	50003, 50005
* Interrupts in S7T Config which may be associated with the warning indicated.				

Errors

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8001	Fatal internal error	Project/software faulty/inconsistent; cannot be rectified by the user.	Check consistency of the project. Recompile all data and download these to the module. Read the error code from the alarm view in S7T Config and then inform the Siemens Hotline.	20001, 50015
8002	Internal error	• Version conflict • Firmware error • Inconsistent software	Check consistency of the project. Recompile all data and download these to the module. Evaluate the alarm view of S7T Config and inform the Siemens Hotline.	30004, 30005, 30010, 30011, 30015

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8003	Fatal configuration error	Faulty axis configuration. A hardware driver, or the device it addresses, may not be available	• Check the plausibility of changed parameters. Examples: Leadscrew pitch is *0*; homing beyond encoder zero mark is not possible with absolute encoders.) • Check the connected devices and their configuration in HW Config. • Verify that the message frame type used is compatible with the technology object. • Download the technology once again to the target system.	20004, 20006
8004	Configuration error	Incomplete or faulty configuration of a technology object.	Check the plausibility of changed parameters. Evaluate the detailed information in the alarm view of S7T Config.	20003, 20011
8005	Command memory is full	The command can not be executed due to insufficient command capacity - Cause: • The technology object can not save additional commands	In S7T Config, increase the "maximum number of active command IDs." Set the value in the Expert List with the *numberOfMaxBuffered CommandId* configuration data element. This setting also increases the command handling capacity of the technology object.	30003
8006	Insufficient memory	Integrated Technology is out of memory.	Reduce the number of technology objects, or the number of interpolation points or the polynomials in the cam disks.	20002
8008	Invalid technology DB	The Technology rejects the DB because this does not have a valid internal ID. Perhaps the DB was generated in a copy operation.	Recreate the DB in the "Technology Objects Management" dialog box and download the configuration data again (establish online / offline consistency).	-
8009	The technology DB is disabled because it was replaced with a newer instance	The DB is no longer assigned to the technology object, because a newer instance with a different DB number has been downloaded.	Delete the DB	-

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
800A	HW driver not available	The driver of a physical device is unavailable or in use by another technology object.	• Check whether the logical device was available at the time the request was output, or whether it was in use by another technology object (TO). • Connect a suitable device. Check the device with regard to its function, connections and wiring. • Check the topology in HW Config. • Compare the configuration in HW Config with that of the technology object.	20014
8010	Drive failed	Failure of the drive, or of an external device, or communication is disrupted, as a result of faulty configuration or parameter data.	Check the device functions, connections and wiring. Check the configuration in HW Config, based on current message frame settings.	20005, 50001
8012	Homing error	The requested homing mode is not supported. Examples: Homing with absolute encoder; homing velocity = *0*	Check the settings for axis homing in terms of inconsistency and invalid values.	40103
8013	Hardware limit switch is active	• The hardware limit switch has responded. • The polarity of the switch was reversed. • Wrong direction of retraction	Return the drive to its operating range. Check the limit switch connections.	50007
8016	Following error out of limits	The offset between the position setpoint and the actual position of the axis has exceeded configured limits.	Check the transfer direction of the encoder, the settings of the position control loop, and the following error monitoring settings.	50102
8017	Following error has exceeded the warning limit	The offset between the position setpoint and the actual position of the axis has exceeded configured warning limits.	Check the transfer direction of the encoder, the settings of the position control loop, and the following error monitoring settings.	50103
8018	Standstill monitoring error	The axis has moved out of the standstill window or could not reach the standstill window within the time specified.	Check the functions of the speed and position control loops, and the settings for standstill monitoring.	50107, 50008
8019	Positioning monitoring error	The axis failed to reach the positioning window within the specified time.	Check the functions of the speed and position control loops, and the settings for positioning monitoring.	50106

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
801B	Clamping monitoring error	Clamping error. The axis has exceeded the "Position tolerance after fixed stop detection" without receiving a new motion command. (Position tolerance setting in S7T Config **Limits > "Fixed end stop" tab > "Position tolerance after fixed end stop detection" parameter**.)	Check: • The mechanical end stop • The function of the position control loop • The clamping monitoring settings in S7T Config	50108
801D	Reference cam or encoder zero mark not found	• Reference cam or encoder zero mark outside the permitted range. • A limit position monitoring function has responded. • The speed was too high.	Check the valid ranges (max. distance between reference cam and encoder zero mark, for example) and the limit position monitoring system, and reduce the shutdown speed.	40102
801F	Dynamic control loop out of range	• The closed-loop position or speed control system was not optimized. • The transfer direction of the encoder is incorrect. • Assigned parameters for following error monitoring are too sensitive • The discrepancy between the speed setpoint and actual speed value is too large.	• Check the transfer direction of the encoder. • Check the control loop parameters. • Modify the parameters of following error monitoring.	50101
8030	Limit frequency of the measuring system exceeded	The pulse rate of the encoder signals is out of the permitted range.	Check the electrical connection of the encoder. Check whether the configured maximum speed is too low.	50002
8031	Actual value monitoring error	• Incremental encoder overrange • The movement per pulse of the position controller exceeds the modulo length of the axis.	Check and adjust the configuration data for the adaptation of the actual value to mechanical requirements	50001, 50013
8032	Cannot shift the coordinate system	The programmed coordinate system shift (set or correct the actual position value) leads to a violation of the internal traversing limit. The coordinate system is not shifted.	Check the programmed shift of the coordinate system.	30014
8033	Encoder zero mark monitoring triggered	The number of increments between encoder zero marks is incorrect.	Check the electrical connection and the set encoder resolution.	50006
8040	The axis / external encoder are disabled, or the wrong mode is set	The required command enable signal is missing.	Eliminate and acknowledge all queued errors and then enable the axis for the relevant mode.	40005

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8043	Invalid command parameter value or default value	One or several invalid values at the input parameters in the technology function of the command, or invalid default parameters.	Check the command and adjust invalid parameters. You should also observe the currently used default values.	30001
8045	Command not allowed in current state	• A command with higher priority is active (for example "MC_Stop"). • Technology object disabled or not ready. • Received measurement command when homing	Acknowledge all queued errors of the technology object. Wait for initialization of the technology object (after restart). Enable the technology object. "Deactivate "MC_Stop." Do not start any measuring commands in the homing phase.	30006, 30009
8046	Command not supported by the drive	• The connected drive does not support this type of command, for example, a specific homing method. • A measuring-input command is active at the external encoder.	Check and adjust the drive settings and configuration data.	50012
8047	Cannot acknowledge this error	It is not possible to acknowledge the error with "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in a restart phase.	Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.	30007
8048	Unable to restart	The technology object is active or a queued error prevents restart.	Eliminate and acknowledge all queued errors and disable the technology object.	20012
8049	Illegal object combination	Faulty combination of technology objects. Examples:	Carry out the following measures, depending on the cause:	20007
		• Axis_2 was set as the leading axis in synchronous operation for Axis_1; at the same time, Axis_1 was set as the leading axis for Axis_2.	• Set a defined leading/following axis combination.	
		• The axes involved in synchronous operation are processed at different run levels.	• Assign the leading and following axis to the same run level (usually the interpolator cycle clock).	

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
804A	Required object connection is missing	No allowances have been made in the configuration for the required combination of leading and following axes and cam disks.	Set the required leading axes and cams in the "Configuration" section of the synchronous object of the following axis.	30008
804B	Limit switch active; invalid command	A hardware limit switch is active. Only motion commands in direction of the operating range are allowed.	Check the mechanical configuration and the electrical connections. Correct any program errors or use the software limit switches.	50009
* Interrupts in S7T Config that may be associated with the indicated error				

7.12.4 ErrorIDs – Cam technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

Warnings

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config *
0020	Restart to activate the parameter changes	A parameter was changed and is not applied immediately	Apply the parameter changes by disabling and then initializing the technology object ("MC_Reset" with *Restart* = *TRUE*)	20010
0022	Internal warning	Temporary internal problem	For detailed information, refer to the alarm view of S7T Config	40003
0027	Interpolation properties cannot be maintained	Discontinuity of the position or gradient was detected at the transition between cam segments	Move the interpolation limits. Check the cam disk profile. Change the criteria for the interpolation of segments.	40008
* Interrupts in S7T Config which may be associated with the warning indicated.				

Errors

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8001	Fatal internal error	Project/software faulty/inconsistent; cannot be rectified by the user.	Check consistency of the project. Recompile all data and download these to the module. Read the error code from the alarm view in S7T Config and then inform the Siemens Hotline.	20001
8002	Internal error	• Version conflict • Firmware error • Inconsistent software	Check consistency of the project. Recompile all data and download these to the module. Evaluate the alarm view of S7T Config and inform the Siemens Hotline.	30004, 30005, 30010, 30011, 30015
8003	Fatal configuration error	Faulty axis configuration. A hardware driver, or the device it addresses, may not be available	• Check the plausibility of changed parameters. Examples: Leadscrew pitch is *0*; homing beyond encoder zero mark is not possible with absolute encoders.) • Check the connected devices and their configuration in HW Config. • Verify that the message frame type used is compatible with the technology object. • Download the technology once again to the target system.	20004, 20006
8004	Parameter assignment error	Incomplete or faulty configuration of a technology object.	Check the plausibility of changed parameters. Evaluate the detailed information in the alarm view of S7T Config.	20003, 20011
8005	Command memory is full	The command can not be executed due to insufficient command capacity - Cause: • The technology object can not save additional commands	In S7T Config, increase the "maximum number of active command IDs." Set the value in the Expert List at configuration parameter *numberOfMaxBuffered CommandId*. This setting also increases the command handling capacity of the technology object.	30003
8006	Insufficient memory	Integrated Technology is out of memory.	Reduce the number of technology objects, or the number of interpolation points or the polynomials in the cam disks.	20002
8008	Invalid technology DB	The Technology rejects the DB because this does not have a valid internal ID. Perhaps the DB was generated in a copy operation.	Recreate the DB in the "Technology Objects Management" dialog box and download the configuration data again (establish online / offline consistency).	-

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8009	The technology DB is disabled because it was replaced with a newer instance	The DB is no longer assigned to the technology object, because a newer instance with a different DB number has been downloaded.	Delete the DB	-
800A	HW driver not available	The driver of a physical device is unavailable or in use by another technology object.	• Check whether the logical device was available at the time the request was output, or whether it was in use by another technology object (TO). • Connect a suitable device. Check the device with regard to its function, connections and wiring. • Check the topology in HW Config. • Compare the configuration in HW Config with that of the technology object.	20014
8010	Drive failed	Failure of the drive, or of an external device, or communication is disrupted, as a result of faulty configuration or parameter data.	Check the device functions, connections and wiring. Check the configuration in HW Config, based on current message frame settings.	20005
8043	Invalid command parameter value or default value	One or several invalid values at the input parameters in the technology function of the command, or invalid default parameters.	Check the command and adjust invalid parameters. You should also observe the currently used default values.	30001
8045	Command not allowed in current state	• A command with higher priority is active (for example "MC_Stop"). • Technology object disabled or not ready. • Received measurement command when homing.	• Acknowledge all queued errors of the technology object. • Wait for initialization of the technology object (after restart). • Enable the technology object. • "Deactivate "MC_Stop." • Do not start any measuring commands during homing.	30006, 30009
8047	Cannot acknowledge this error	It is not possible to acknowledge the error by calling technology function "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in the restart phase.	Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.	30007

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8048	Unable to restart	The technology object is active or a queued error prevents restart.	Eliminate and acknowledge all queued errors and disable the technology object.	20012
8049	Illegal object combination	Faulty combination of technology objects. Examples:	Carry out the following measures, depending on the cause:	20007
		• Axis_2 was set as the leading axis in synchronous operation for Axis_1; at the same time, Axis_1 was set as the leading axis for Axis_2.	• Set a defined leading/following axis combination.	
		• The axes involved in synchronous operation are processed in different priority classes.	• Assign the leading and following axis to the same priority class (usually the interpolator cycle clock).	
804A	Required object connection is missing	No allowances have been made in the configuration for the required combination of leading and following axes and cam disks.	Set the required leading axes and cams in the "Configuration" section of the synchronous object of the following axis.	30008
8051	Read / write access denied.	An attempt was made to delete a cam, or to adjust it while it was in use.	Necessary procedure: • Curve synchronization Terminate synchronous operation of the cam, and then edit it. • Valve profile Assign a different cam to the hydraulic axis with "MC_SetCharacteristic."	40001, 40002
8060	Error when adding a cam segment	The new cam segment is faulty, for example, its start and end points are identical.	Check the values defining the new cam segment.	40004
8061	Cam point out of the domain/range	The leading/following axis position defined at "MC_GetCamPoint" does not exist in the domain/range of the cam.	Set a valid position at "MC_GetCamPoint".	40005, 40006
8062	Access to an interpolated cam not allowed	An invalid command was output to an already interpolated cam. Example: "MC_CamSectorAdd"	You can not adjust a cam while it is in use. Before you do so, delete it with "MC_CamClear."	40007
* Interrupts in S7T Config that may be associated with the indicated error				

7.12.5 ErrorIDs – Measuring input technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

Warnings

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config *
0020	Restart to activate the parameter changes	Changes to a parameter are not applied directly	Apply the parameter changes by disabling and then initializing the technology object ("MC_Reset" with *Restart* = *TRUE*)	20010
* Interrupts in S7T Config which may be associated with the warning indicated.				

Errors

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8001	Fatal internal error	Project/software faulty/inconsistent; cannot be rectified by the user.	Check consistency of the project. Recompile all data and download these to the module. Read the error code from the alarm view in S7T Config and then inform the Siemens Hotline.	20001
8002	Internal error	• Version conflict • Firmware error • Inconsistent software	Check consistency of the project. Recompile all data and download these to the module. Evaluate the alarm view of S7T Config and inform the Siemens Hotline.	30004, 30005, 30010, 30011, 30015
8003	Fatal configuration error	Faulty axis configuration. A hardware driver, or the device it addresses, may not be available	• Check the plausibility of changed parameters. Examples: Leadscrew pitch is *0*; homing beyond encoder zero mark is not possible with absolute encoders.) • Check the connected devices and their configuration in HW Config. • Verify that the message frame type used is compatible with the technology object. • Download the technology once again to the target system.	20004, 20006, 40006
8004	Parameter assignment error	Incomplete or faulty configuration of a technology object.	Check the plausibility of changed parameters. Evaluate the detailed information in the alarm view of S7T Config.	20003, 20011

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8005	Command memory is full	The command can not be executed due to insufficient command capacity - Cause: • The technology object can not save additional commands	In S7T Config, increase the "maximum number of active command IDs." Set the value in the Expert List at configuration parameter *numberOfMaxBuffered CommandId*. This setting also increases the command handling capacity of the technology object.	30003
8006	Insufficient memory	Integrated Technology is out of memory.	Reduce the number of technology objects, or the number of interpolation points or the polynomials in the cam disks.	20002
8008	Invalid technology DB	The Technology rejects the DB because this does not have a valid internal ID. Perhaps the DB was generated in a copy operation.	Recreate the DB in the "Technology Objects Management" dialog box and download the configuration data again (establish online / offline consistency).	-
8009	The technology DB is disabled because it was replaced with a newer instance	The DB is no longer assigned to the technology object, because a newer instance with a different DB number has been downloaded.	Delete the DB	-
800A	HW driver not available	The driver of a physical device is unavailable or in use by another technology object.	• Check whether the logical device was available at the time the request was output, or whether it was in use by another technology object (TO). • Connect a suitable device. Check the device with regard to its function, connections and wiring. • Check the topology in HW Config. • Compare the configuration in HW Config with that of the technology object.	20014
8010	Drive failed	Failure of the drive, or of an external device, or communication is disrupted, as a result of faulty configuration or parameter data.	Check the device functions, connections and wiring. Check the configuration in HW Config, based on current message frame settings.	20005
8034	Could not record a measured value in this range	The expected edge of the sensor signal was not detected within the specified measuring range.	Check the measuring range and the electrical function of the measuring input.	40003, 40004

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8035	Measuring command canceled	Causes: • Drive or encoder error • Operation aborted by the recording device • Multiple measuring inputs access the same encoder	Eliminate the encoder / drive error. Do not access the same encoder in parallel with several measuring inputs.	40005, 40007, 40008
8043	Invalid command parameter value or default value	One or several invalid values at the input parameters in the technology function of the command, or invalid default parameters.	Check the command and adjust invalid parameters. You should also observe the currently used default values.	30001
8044	Command not supported by the technology object	The technology DB parameterized at the technology function belongs to a technology object which does not support this type of command. Example: Measurements at a virtual axis are not possible.	Use the correct object type. Adjust the axis configuration.	40001
8045	Command not allowed in current state	• A command with higher priority is active (for example "MC_Stop"). • Technology object disabled or not ready. • Received measurement command when homing.	• Acknowledge all queued errors of the technology object. • Wait for initialization of the technology object (after restart). • Enable the technology object. • "Deactivate "MC_Stop." • Do not start any measuring commands in the homing phase.	30006, 30009, 40002
8047	Cannot acknowledge this error	It is not possible to acknowledge the error by calling technology function "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in a restart phase.	Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.	30007
8048	Unable to restart	The technology object is active or a queued error prevents restart.	Eliminate and acknowledge all queued errors and disable the technology object.	20012

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8049	Illegal object combination	Faulty combination of technology objects. Examples:	Carry out the following measures, depending on the cause:	20007
		• Axis_2 was set as the leading axis in synchronous operation for Axis_1; at the same time, Axis_1 was set as the leading axis for Axis_2.	• Set a defined leading/following axis combination.	
		• The axes involved in synchronous operation are processed in different priority classes.	• Assign the leading and following axis to the same priority class (usually the interpolator cycle clock).	
804A	Required object connection is missing	No allowances have been made in the configuration for the required combination of leading and following axes and cam disks.	Set the required leading axes and cams in the "Configuration" section of the synchronous object of the following axis.	30008
* Interrupts in S7T Config that may be associated with the indicated error				

7.12.6 ErrorIDs – Output cam technology DB

Valid for Integrated Technology with firmware version V4.1.x

This section describes applications with firmware versions V3.0.x , V3.1.x, V3.2.x.

Warnings

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config *
0020	Restart to activate the parameter changes	A parameter was changed and is not applied immediately	Apply the parameter changes by disabling and then initializing the technology object ("MC_Reset" technology function with *Restart = TRUE*)	20010
* Interrupts in S7T Config which may be associated with the warning indicated.				

Errors

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8001	Fatal internal error	Project/software faulty/inconsistent; cannot be rectified by the user.	Check consistency of the project. Recompile all data and download these to the module. Read the error code from the alarm view in S7T Config and then inform the Siemens Hotline.	20001
8002	Internal error	• Version conflict • Firmware error • Inconsistent software	Check consistency of the project. Recompile all data and download these to the module. Evaluate the alarm view of S7T Config and inform the Siemens Hotline.	30004, 30005, 30010, 30011, 30015
8003	Fatal configuration error	Faulty axis configuration. A hardware driver, or the device it addresses, may not be available	• Check the plausibility of changed parameters. Examples: Leadscrew pitch is *0*; homing beyond encoder zero mark is not possible with absolute encoders.) • Check the connected devices and their configuration in HW Config. • Verify that the message frame type used is compatible with the technology object. • Download the technology once again to the target system.	20004, 20006
8004	Parameter assignment error	Incomplete or faulty configuration of a technology object.	Check the plausibility of changed parameters. Evaluate the detailed information in the alarm view of S7T Config.	20003, 20011
8005	Command memory is full	The command can not be executed due to insufficient command capacity - Cause: • The technology object can not save additional commands	In S7T Config, increase the "maximum number of active command IDs." Set the value in the Expert List at configuration parameter *numberOfMaxBuffered CommandId*. This setting also increases the command handling capacity of the technology object.	30003
8006	Insufficient memory	Integrated Technology is out of memory.	Reduce the number of technology objects, or the number of interpolation points or the polynomials in the cam disks.	20002

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8008	Invalid technology DB	The Technology rejects the DB because this does not have a valid internal ID. Perhaps the DB was generated in a copy operation.	Recreate the DB in the "Technology Objects Management" dialog box and download the configuration data again (establish online / offline consistency).	-
8009	The technology DB is disabled because it was replaced with a newer instance	The DB is no longer assigned to the technology object, because a newer instance with a different DB number has been downloaded.	Delete the DB	-
800A	HW driver not available	The driver of a physical device is not available or is being used by another technology object.	• Check whether the logical device was available at the time the request was output, or whether it was in use by another technology object (TO). • Connect a suitable device. Check the device with regard to its function, connections and wiring. • Check the topology in HW Config. • Compare the configuration in HW Config with that of the technology object.	20014
8010	Drive failed	Failure of the drive, or of an external device, or communication is disrupted, as a result of faulty configuration or parameter data.	Check the device functions, connections and wiring. Check the configuration in HW Config, based on current message frame settings.	20005
8021	Invalid cam position	Invalid actual value at the cam.	• Restart of the corresponding axis or external encoder. • Failure of the output of actual values at the encoder system (due to drive failure, for example.)	40005
8043	Invalid command parameter value or default value	One or several invalid values at the input parameters in the technology function of the command, or invalid default parameters. Example: • The switching position of an output cam is outside the operating range of the axis.	Check the command and adjust invalid parameters. You should also observe the currently used default values.	30001, 40003

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8045	Command not allowed in current state	• A command with higher priority is active (for example "MC_Stop"). • Technology object disabled or not ready. • Received measurement command when homing.	• Acknowledge all queued errors of the technology object. • Wait for initialization of the technology object (after Restart). • Enable the technology object. • "Deactivate "MC_Stop." • Do not start any measuring commands in the homing phase.	30006, 30009
8047	Cannot acknowledge this error	It is not possible to acknowledge the error by calling technology function "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in a restart phase.	Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.	30007
8048	Unable to restart	The technology object is active or a queued error prevents restart.	Eliminate and acknowledge all queued errors and disable the technology object.	20012
8049	Illegal object combination	Faulty combination of technology objects. Examples:	Carry out the following measures, depending on the cause:	20007
		• Axis_2 was set as the leading axis in synchronous operation for Axis_1; at the same time, Axis_1 was set as the leading axis for Axis_2.	• Set a defined leading/following axis combination.	
		• The axes involved in synchronous operation are processed at different run levels.	• Assign the leading and following axis to the same run level (usually the interpolator cycle clock).	
804A	Required object connection is missing	No allowances have been made in the configuration for the required combination of leading and following axes and cam disks.	Set the required leading axes and cams in the "Configuration" section of the synchronous object of the following axis.	30008
8087	Illegal output cam position	At least one output cam position is illegal. The output cam was moved beyond the limits of the operating range.	Check the output cam positions. Reduce the current delay/rate time.	40001
* Interrupts in S7T Config that may be associated with the indicated error				

7.12.7 ErrorIDs - Cam track technology DB

Valid for Integrated Technology with firmware version V4.1.x

Warnings

ErrorID	Warning message	Possible causes	Remedy	Interrupts in S7T Config *
0020	Restart to activate the parameter changes	A parameter was changed and is not applied immediately	Apply the parameter changes by disabling and then initializing the technology object ("MC_Reset" technology function with *Restart* = *TRUE*)	20010
* Interrupts in S7T Config which may be associated with the warning indicated.				

Errors

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8001	Fatal internal error	Project/software faulty/inconsistent; cannot be rectified by the user.	Check consistency of the project. Recompile all data and download these to the module. Read the error code from the alarm view in S7T Config and then inform the Siemens Hotline.	20001
8002	Internal error	• Version conflict • Firmware error • Inconsistent software	Check consistency of the project. Recompile all data and download these to the module. Evaluate the alarm view of S7T Config and inform the Siemens Hotline.	30004, 30005, 30010, 30011, 30015
8003	Fatal configuration error	Faulty axis configuration. A hardware driver, or the device it addresses, may not be available	• Check the plausibility of changed parameters. Examples: Leadscrew pitch is *0*; homing beyond encoder zero mark is not possible with absolute encoders.) • Check the connected devices and their configuration in HW Config. • Verify that the message frame type used is compatible with the technology object. • Download the technology once again to the target system.	20004, 20006

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8004	Parameter assignment error	Incomplete or faulty configuration of a technology object.	Check the plausibility of changed parameters. Evaluate the detailed information in the alarm view of S7T Config.	20003, 20011
8005	Command memory is full	The command can not be executed due to insufficient command capacity - Cause: • The technology object can not save additional commands	In S7T Config, increase the "maximum number of active command IDs." Set the value in the Expert List at configuration parameter *numberOfMaxBuffered CommandId*. This setting also increases the command handling capacity of the technology object.	30003
8006	Insufficient memory	Integrated Technology is out of memory.	Reduce the number of technology objects, or the number of interpolation points or the polynomials in the cam disks.	20002
800A	HW driver not available	The driver of a physical device is not available or is being used by another technology object.	• Check whether the logical device was available at the time the request was output, or whether it was in use by another technology object (TO). • Connect a suitable device. Check the device with regard to its function, connections and wiring. • Check the topology in HW Config. • Compare the configuration in HW Config with that of the technology object.	20014
8010	Drive failed	Failure of the drive, or of an external device, or communication is disrupted, as a result of faulty configuration or parameter data.	Check the device functions, connections and wiring. Check the configuration in HW Config, based on current message frame settings.	20005
8021	Invalid cam position	Invalid actual value at the cam.	• Restart of the corresponding axis or external encoder. • Failure of the output of actual values at the encoder system (due to drive failure, for example.)	40005

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
8043	Invalid command parameter value or default value	One or several invalid values at the input parameters in the technology function of the command, or invalid default parameters. Example: • The switching position of an individual output cam is outside the operating range of the axis. • The cam track length *CamTrackLength* has the value *0* or *<0.*	Check the command and adjust invalid parameters. You should also observe the currently used default values.	30001, 40003
8045	Command not allowed in current state	• A command with higher priority is active (for example "MC_Stop"). • Technology object disabled or not ready. • Received measurement command when homing.	• Acknowledge all queued errors of the technology object. • Wait for initialization of the technology object (after Restart). • Enable the technology object. • "Deactivate "MC_Stop." • Do not start any measuring commands in the homing phase.	30006, 30009
8047	Cannot acknowledge this error	It is not possible to acknowledge the error by calling technology function "MC_Reset". Examples: • An axis has been brought to standstill, and the error causing this can not be acknowledged. • Further errors occur during acknowledgment or in a restart phase.	Eliminate all causes of error and retry to acknowledge. Please note that you must acknowledge particular errors by cycling power OFF and ON.	30007
8048	Unable to restart	The technology object is active or a queued error prevents restart.	Eliminate and acknowledge all queued errors and disable the technology object.	20012
8049	Illegal object combination	Faulty combination of technology objects. Examples:	Carry out the following measures, depending on the cause:	20007
		• Axis_2 was set as the leading axis in synchronous operation for Axis_1; at the same time, Axis_1 was set as the leading axis for Axis_2.	• Set a defined leading/following axis combination.	
		• The axes involved in synchronous operation are processed in different priority classes.	• Assign the leading and following axis to the same run level (usually the interpolator cycle clock).	

ErrorID	error message	Possible causes	Remedy	Interrupts in S7T Config *
804A	Required object connection is missing	No allowances have been made in the configuration for the required combination of leading and following axes and cam disks.	Set the required leading axes and cams in the "Configuration" section of the synchronous object of the following axis.	30008
8087	Illegal output cam position	At least one individual output cam position is illegal.	Check the individual output cam positions. Reduce the current delay/rate time.	40001
* Interrupts in S7T Config that may be associated with the indicated error				

7.12.8 ErrorIDs – MCDevice/Trace technology DB

Valid for Integrated Technology with firmware V3.1.x or higher

This section describes applications with firmware V3.0.x

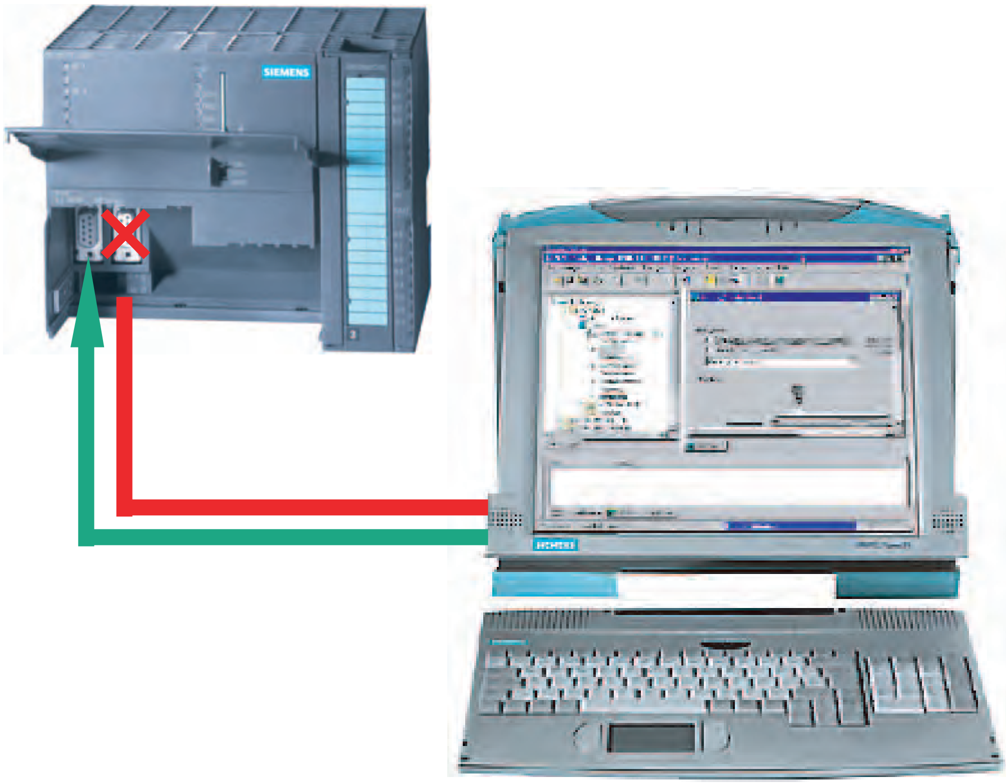
Errors

ErrorID	Error message	Possible causes	Remedy	Interrupts in S7T Config *
8008	Invalid technology DB	The Technology rejects the DB because this does not have a valid internal ID. Perhaps the DB was generated in a copy operation.	Recreate the DB in the "Technology Objects Management" dialog box and download the configuration data again (establish online / offline consistency).	
8009	The technology DB is disabled because it was replaced with a newer instance	The DB is no longer assigned to the technology object, because a newer instance with a different DB number has been downloaded.	Delete the DB	
* Interrupts in S7T Config that may be associated with the indicated error				

Loading, testing and diagnostics 8

8.1 Connecting the programming device

All configuration data and programs are created on the PG or PC.



Always use the online cable to interconnect the programming device with the X1 interface of the technology CPU.

Programming devices such as text-based displays and OPs may not be connected to the X3 interface (DP(DRIVE)). Any operation of these devices will jeopardize isochronous operation on DP(DRIVE).

As an option, you may connect the programming device to a CP in your system. Make allowances for prolonged load times when connecting to a CP, or when setting a constant bus cycle at the X1 interface.

Note

For information on routing to drives, refer to the corresponding device manual.

For additional information on routing, refer to the entry ID in the FAQ on drive technology: 26710833. You can retrieve the entry on the Internet pages of Siemens AG, Automation and Drives, Service & Support.

8.2 Download commands

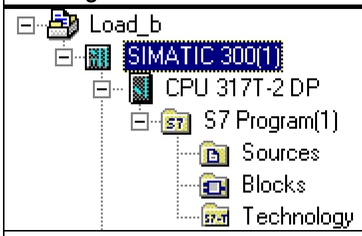
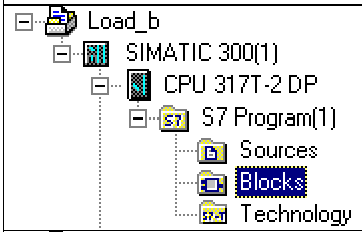
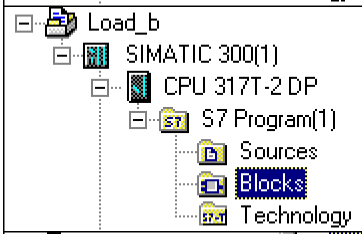
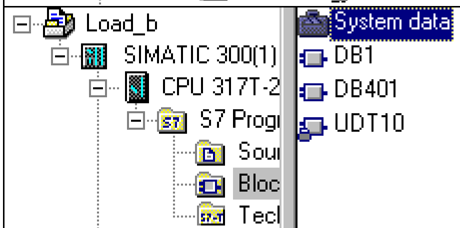
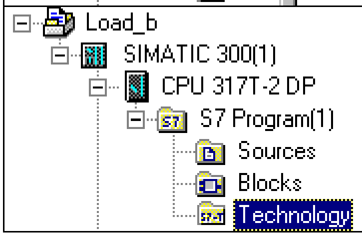
8.2.1 Load commands in STEP 7

You can always download your configuration data to the target system when working in SIMATIC Manager, HW Config or in S7T Config, that is, to your Technology CPU. Objects you can download:

User program (OB, FC, FB, DB)	Program blocks and DBs from the blocks folder of your project.
System data	System data contain the permanent and programmed parameters of the CPU, that is, the HW configuration and all configuration data of integrated technology.
Hardware configuration	The hardware configuration data are created in HW Config and reflect the HW configuration of the station.
Technology	These data comprise the configuration data of integrated technology and of the technology objects.
Firmware of integrated technology	The current firmware of integrated technology is packed to SDBs and saved to load memory of your Technology CPU. The firmware download is based on the conditions listed below: • The "Generate technology data" check box was set at the time "Save and compile" was executed. This check box is available in HW Config, properties dialog box, **Technology > "Technology system data" tab**. All firmware data are deleted from load memory if the technology system data were not generated in the "Save and compile" operation (check box deactivated.) • The current firmware of integrated technology is not yet saved to load memory. If it already exists in load memory, the firmware will not be downloaded again.
Drive configuration	In S7T Config, you can configure the MICROMASTER and SINAMICS drives with integrated STARTER.

Below you will find a listing of objects you can download with the various tools:

Download in "SIMATIC Manager"

Marking	Menu command	What is downloaded
	Target system > Download	• User Program • Hardware configuration • Technology • Firmware of integrated technology [1)]
Selection has no effect	PLC > Upload Station to PG	• User Program • Hardware configuration • Technology (backup copy only; can not be edited in S7T Config) • Firmware of integrated technology
	Target system > Download	• User Program • Hardware configuration • Technology • Firmware of integrated technology [1)]
	PLC > Download user program to memory card	• User Program • Hardware configuration • Technology • Firmware of integrated technology [1)] The contents of load memory will be deleted before the download starts.
	Target system > Download	• Hardware configuration • Technology • Firmware of integrated technology [1)]
	No download	The block folder and the "technology objects" icon are only representative objects used for configuring the technology.

Load commands of the "HW Config" tool

Marking	Menu command	What is downloaded
No selection required / possible	Target system -> Download to Module	• Hardware configuration • Technology • Firmware of integrated technology [1]

Load commands in the "S7T Config" tool

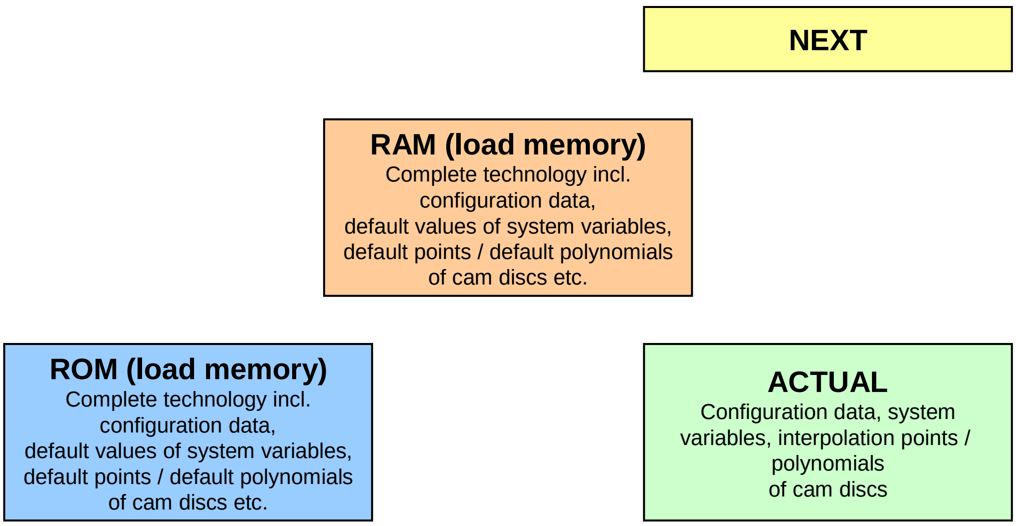
[1] If the integrated technology firmware version is identical in load memory of the technology CPU, the technology CPU does not automatically perform a memory reset after the download.

All download operations described can also be directed to the MMC in the MMC interface of the programming device / PC. The card can be used to reduce download times. The same applies to the Compact Flash Card of the Technology CPU MICROBOX T.

8.2.2 Load commands in S7T Config - Memory organization

8.2.2.1 Memory organization of integrated technology

The diagram below shows the memory areas associated with integrated technology:



ROM (load memory)

"ROM" memory forms part of load memory in the Technology CPU. For CPU 31xT, this memory is located on the Micro Memory Card (MMC), and for the Microbox T, it is located on the Compact Flash Card. The entire "Technology" is stored in this memory area. This includes the following data:

- Configuration data (defined in S7T Config)
- Default values of system variables
- Default interpolation points/polynomials (configured in S7T Config) of the cam disks.

RAM (load memory)

The "RAM" memory area is also located in load memory of the Technology CPU. The entire "Technology" is loaded to this memory area. This includes the following data:

- Configuration data (defined in S7T Config)
- Default values of system variables
- Default interpolation points/polynomials (configured in S7T Config) of the cam disks.

ACTUAL

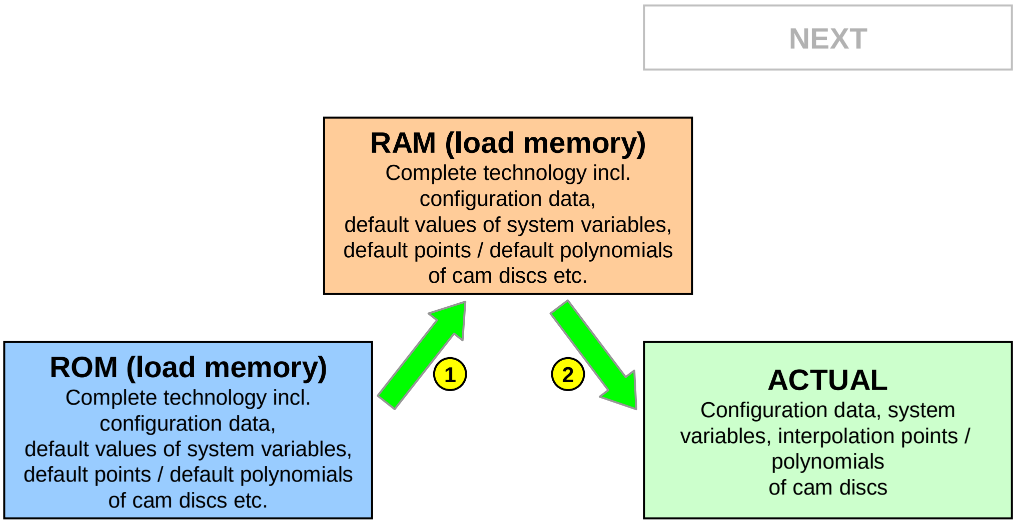
The "ACTUAL" memory area is a component of integrated technology. The "ACTUAL" memory area is used to save and process actual runtime configuration data, system variables, and the interpolation points and polynomials of the cam disks.

NEXT

The "NEXT" memory area is also a component of integrated technology. This memory area is used for the interim storage of changes which are not immediately activated.

8.2.2.2 POWER ON and CPU memory reset

The representation below shows which memory areas will be loaded after POWER ON and after a memory reset of the Technology CPU.

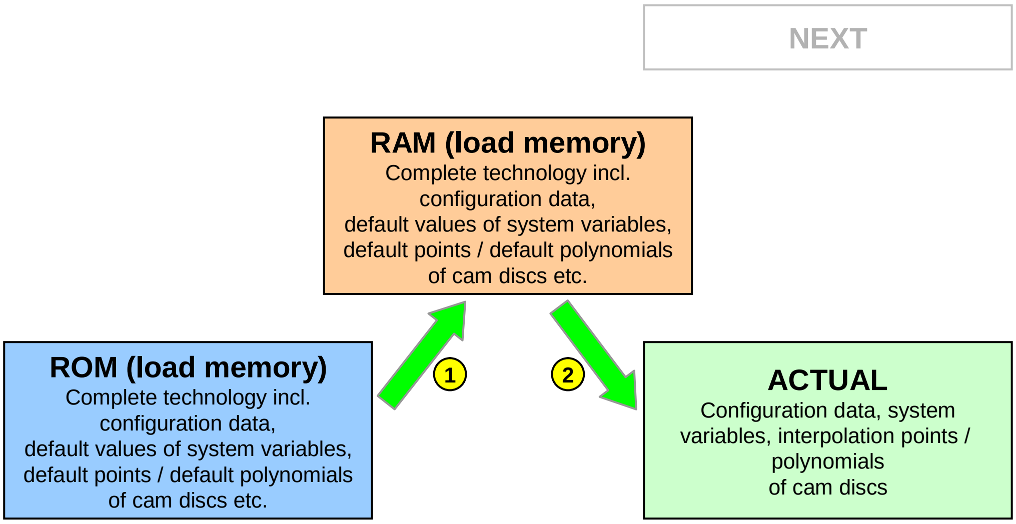


① The complete technology will be transferred from ROM to RAM.

② The configuration data, the default values of system variables, including the default interpolation points/polynomials of the cam disks are transferred from "RAM" to "ACTUAL."

8.2.2.3 Restarting a technology object

The illustration below shows which memory areas are copied to a technology object after a restart (for example, by calling technology function "MC_Reset", input parameter *Restart = TRUE*).



① The configuration data and the default values of system variables for the relevant technology object are transferred from "ROM" to" RAM."

If the technology object is a cam disk, the default interpolation points/polynomials will be included.

② The configuration data, the default values of system variables, including the default interpolation points/polynomials of the cam disks are transferred from "RAM" to "ACTUAL."

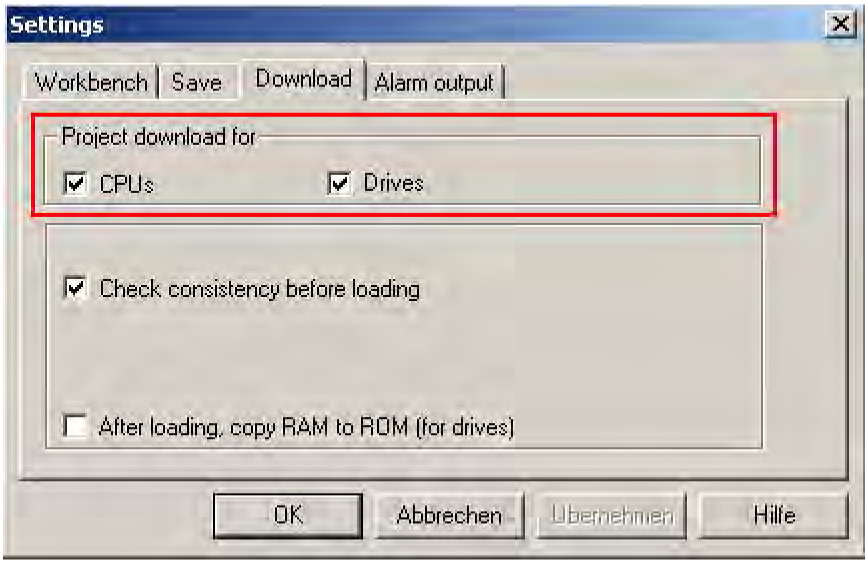
8.2.2.4 Load commands in S7T Config

Target system / Target device

S7T Config distinguishes between the "Target system" and "Target device" objects when handling download commands.

"Target device" is always the object selected in the Navigator. If the destination device is a drive, all nested drive objects will be included.

Define the "Target system" by selecting the **Options > Customize** command.



Define the target devices of the "Target system" in the "Download" tab.

- set the "CPUs" check box if the Technology CPU should belong to the "Target system"
- activate the "Drives" check box if all drives listed in the Navigator should belong to the "Target system"

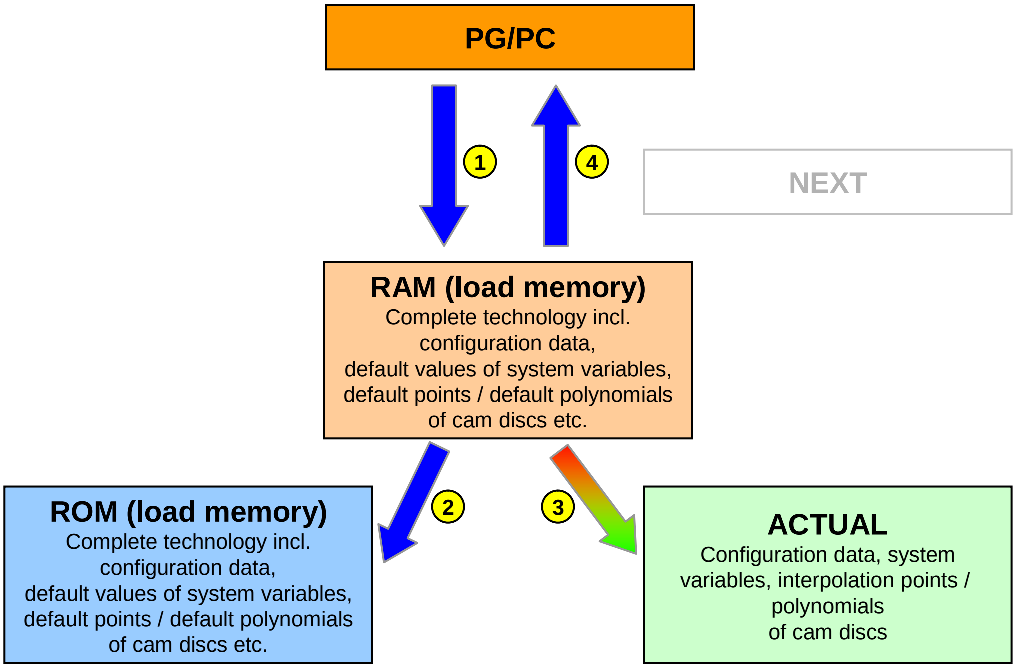
Note

The target devices of the "Target system" must be online.

Define the target devices for the online connection by selecting the **Target system > Select target device** command.

Download to the Technology CPU

The diagram below shows the reaction in memory to commands of S7T Config to download data to the Technology CPU.



Menu commands or shortcut menu commands:

- **Project > Download to target system**
- **Target system > Download > Project to target system**
- **Target system > Download > To target device** (if the technology is selected in the Navigator)

Menu command **Target system > Upload > Configuration data to PG**

① The delta data of the project will be downloaded from the programming device / PC to "RAM."

② The delta data of the project are downloaded from "RAM" to "ROM."

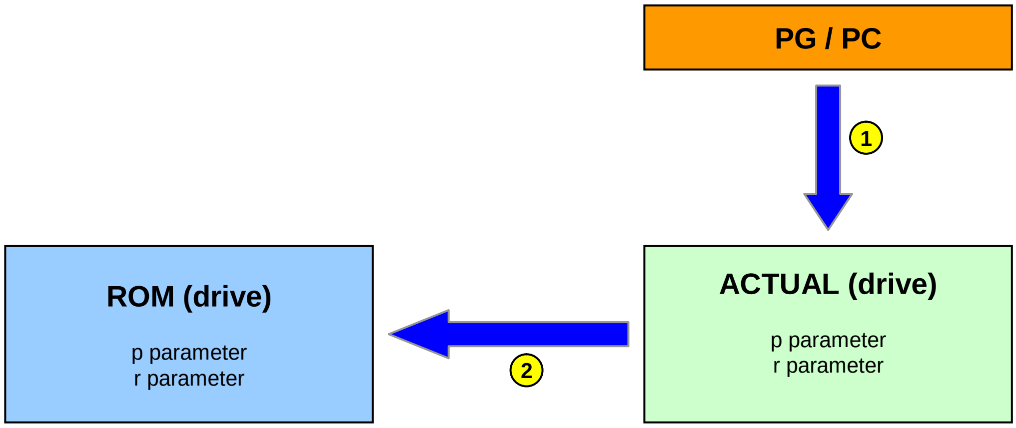
③ At the transition of the Technology CPU from STOP to RUN, the configuration data, the default values of the system variables, and the default interpolation points/polynomials are transferred from "RAM" to "ACTUAL."

Menu command **Target system > Upload > Configuration data to PG**

④ This command uploads the configuration data from "RAM" to the project data on the programming device / PC.

Download to drive

The diagram below shows the reaction in memory to commands of S7T Config to download data to the drive.



Menu commands or shortcut menu commands:

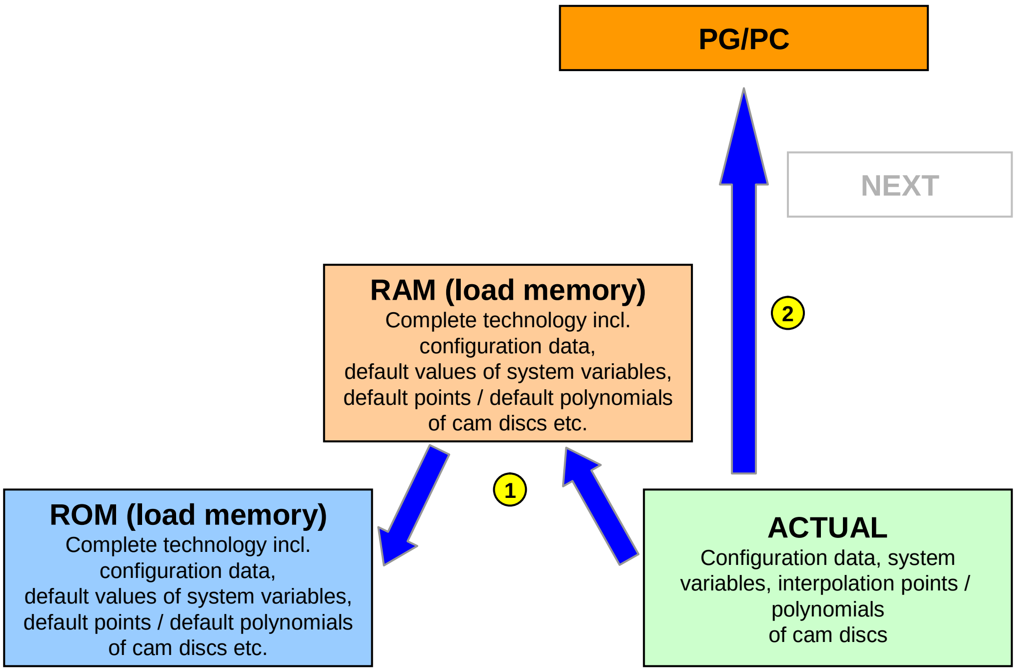
- **Target device > Download to target device**
- **Target system > Download > To target device**

① Downloads drive-specific project data from the PG/PC to "ACTUAL" of the drive.

② Transfers drive-specific project data from "ACTUAL" to "ROM" of the drive.

8.2.2.5 Copy Actual to ROM

The diagram below shows which memory areas are copied by selecting the **Target system > Copy Actual to ROM,** command.



① The configuration data are transferred from "Actual " to "RAM," and then from "RAM" to "ROM."

② The configuration data defined in item 1 will also be uploaded to the programming device / PC when you set the "Upload configuration data to PG" in the dialog box of the menu.

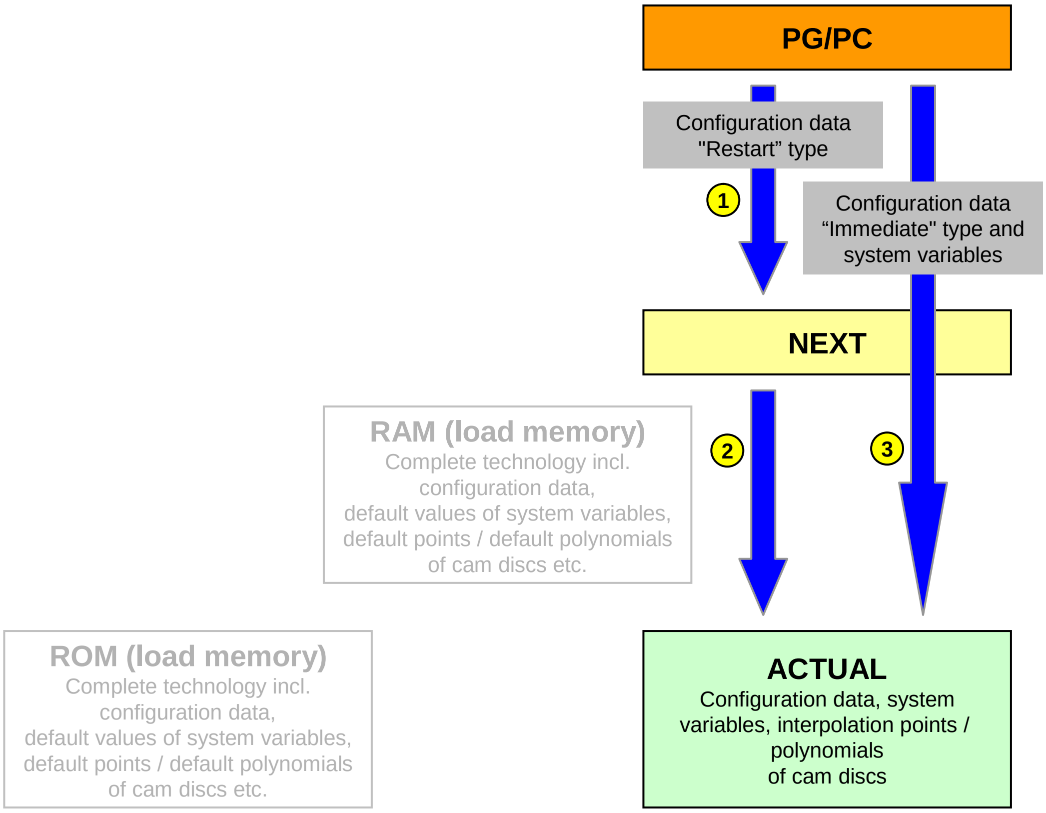
8.2.2.6 Online changes in S7T Config

The diagrams below demonstrate the effects of online changes on the memory areas of integrated technology.

Online changes in dialog boxes, in the expert list of the technology objects, and by calling the technology function "MC_WriteParameter"

In online mode of S7T Config, the modified value will be accepted and transferred to the memory areas shown below when you close a field in the dialog boxes or in the expert list. This also applies to write operations of the technology function "MC_WriteParameter".

You must distinguish between changed configuration data of the type "Restart" and "Immediate." You can identify the type in online mode by the entries in the "Effects" column of the expert list, or by the entries in the parameters list.



① Configuration data of the type "Restart" which were modified in online mode are initially downloaded to the "NEXT" memory area.

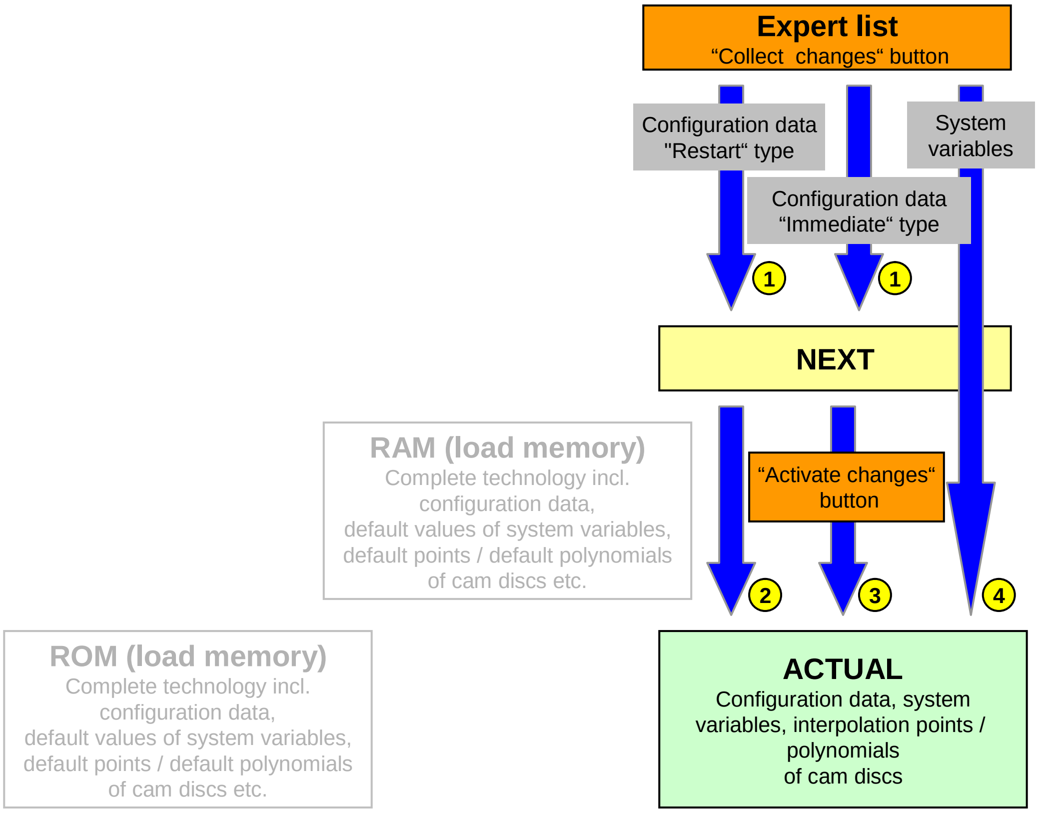
② When a technology object "Restart" is executed (example: by setting input parameter *Restart = TRUE* using technology function "MC_Reset", the changed value is transferred from memory area "NEXT" to "ACTUAL."

③ Changed values of configuration data of the "Immediate" type and of system variables will be downloaded directly to "ACTUAL." Memory area "NEXT" will be by-passed.

Online changes using the "Collect changes" button function

In online mode, the expert list of the technology objects features the "Collect changes" and "Activate changes" buttons. Use the "Collect changes" button to maintain consistency when editing selected configuration data.

By clicking "Collect changes" you open the "Next value" column in the expert list of configuration data and gray out the "Actual value" column.



① Any values changed in the "Next value" column will be transferred immediately to memory area "NEXT." This applies to configuration data of the type "Restart" and "Immediate."

② If a "Restart" is applied to the technology object (for example by setting the input parameter *Restart = TRUE* using the "MC_Reset" technology function), you transfer the changed value from memory area "NEXT" to "ACTUAL" (applies only to configuration data of the "Restart" type.)

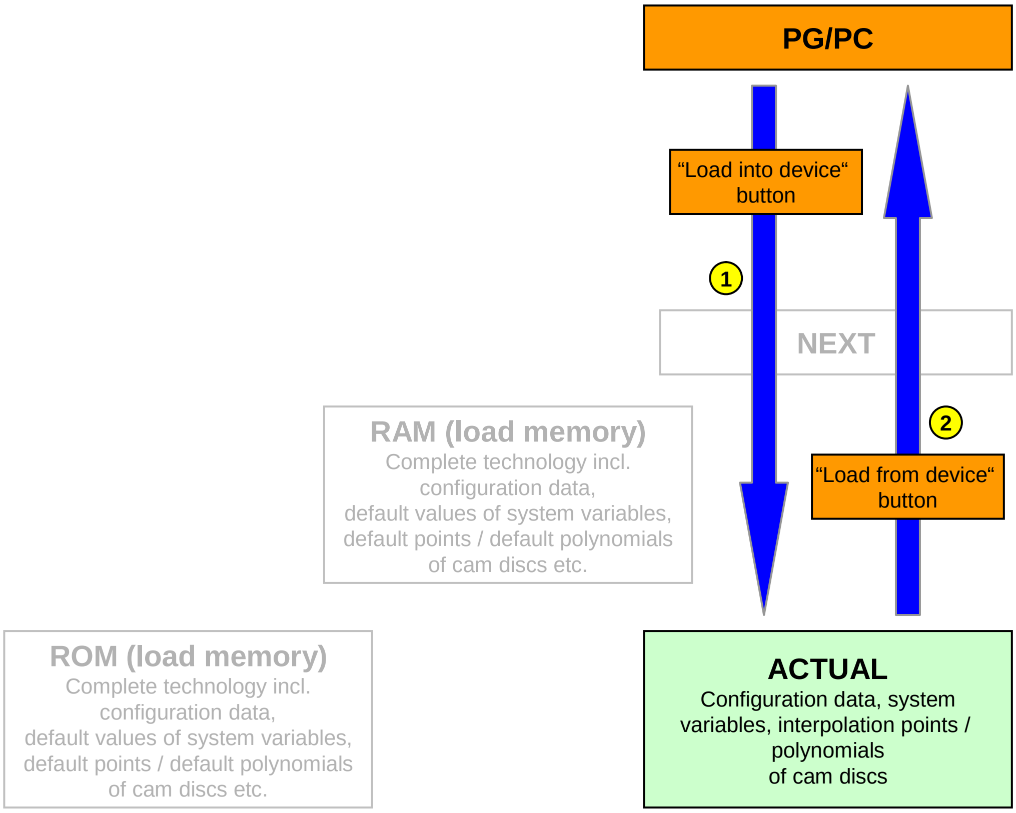
③ When "Collect changes" is active, all configuration data of the "Immediate" type will also be transferred to memory area "NEXT."

Click "Activate changes" to apply all collected changes to the integrated technology.

④ Changed values of system variables are downloaded directly to "ACTUAL." Memory area "NEXT" will be by-passed.

Online changes of a cam disk

In online mode of S7T Config, you can transfer cam disks from the cam disk creation tool "CamEdit" to "ACTUAL" and vice versa. You can only transfer cam disks to "ACTUAL" which are currently not in use by the Technology CPU.



① Click "Download to device" to interpolate the cam disk represented in CamEdit, and to download it to memory area "ACTUAL" of the integrated technology.

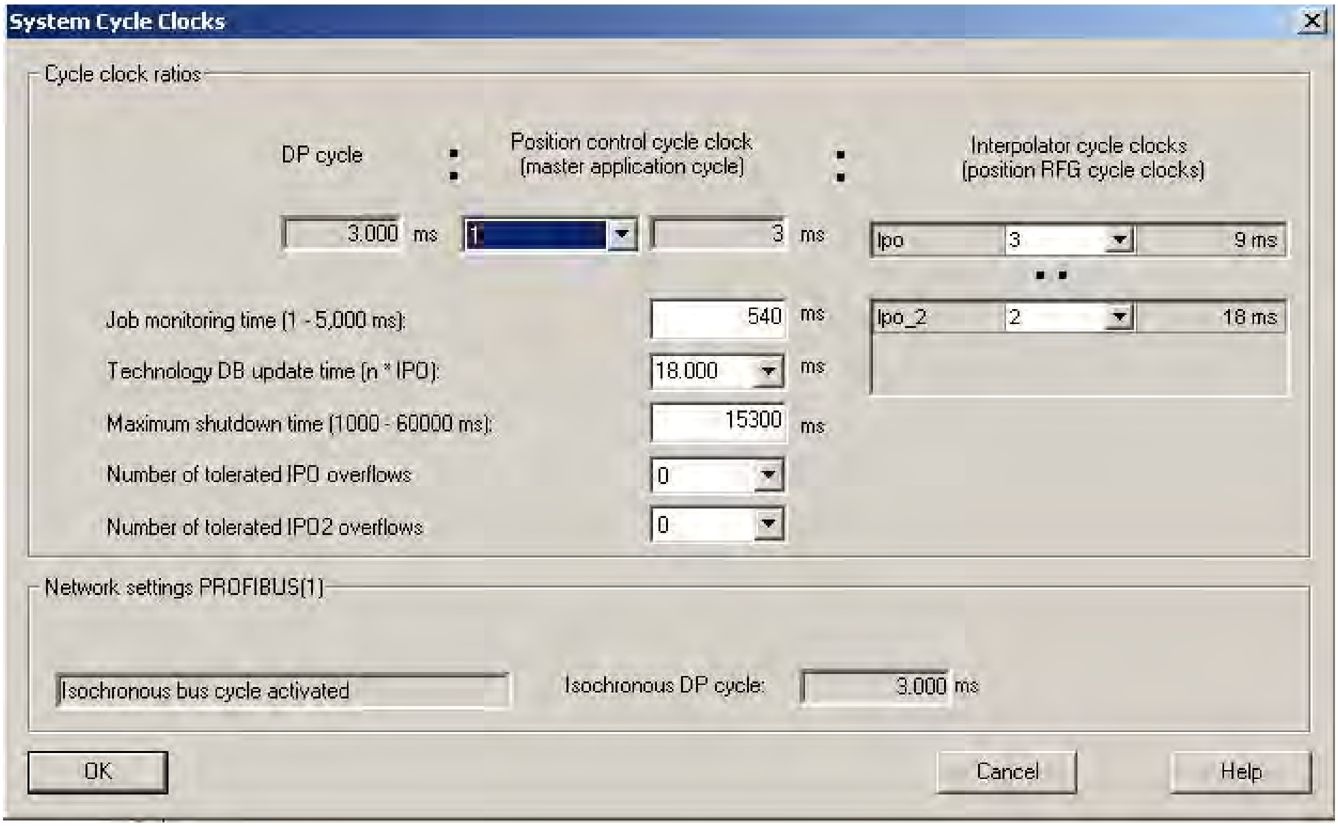
② Click "Upload from device" to upload the interpolated cam disk from memory area "ACTUAL" to CamEdit.

8.3 Technology system clocks

8.3.1 Setting the technology system cycle clocks

The integrated technology processes your tasks in single processing cycles. Based on the task, you can adjust the technology system clocks to suit your requirements and to enhance performance of the Motion Control application.

Select the technology in S7T Config, and then select **Target system > Set system clocks.**



Cycle clock ratios

Set the clock of integrated technology in the "Cycle clock ratio". Select a suitable ratio from the corresponding drop-down list. The absolute cycle time is displayed in the column on the right side. Cycle times defined in this dialog box:

- Bus clock
  The bus clock cannot be set in this dialog box. The "bus clock" is set in HW Config as "equidistant DP cycle" in the DP(DRIVE) settings. The equidistant DP cycle of the DP(DRIVE) interface forms the basic clock of the technology system clocks.
  The DP cycle clock must be must be an integral multiple of 0.250 ms. Change the value in HW Config if this is not the case.
- Servo (position control cycle)
  The system also calculates the axis position control within this cycle. The position control cycle can be set to an integral multiple of the DP cycle. The technology CPU supports the factors *1* and *2*. Select a multiplication factor from the drop-down list box in order to set the required position control cycle.
  Usually, the factor *1* should be entered. Dynamic performance of the control system will deteriorate if you set the factor *2*. However, you will release computing time for processing other tasks. The maximum and average command execution times can be read from the *MaxLoopDuration* and *CmdLoopDuration* variables of the "MCDevice" technology data block.

  The reduction ratio between the DP cycle and the position control cycle must also be set at the drive as "master application cycle." This setting is required for the mutual sign-of-life monitoring. For further information, refer to the drive documentation.
- Ipo (interpolator cycle)
  Axis, external encoders, output cams and measuring inputs are computed by default within the "Interpolator cycle". Select a multiplication factor from the drop-down list box in order to set the required interpolator cycle
- Ipo_2 (Interpolator cycle 2)"Interpolator cycle 2" is used to control motions of axes with lower priority. This value determines the time pattern for the calculation of axes, external encoders, output cams and measuring inputs of a lower priority class. Select a factor from the drop-down list box in order to set the required interpolator cycle 2.

Technology parameterization

Click the selection button of the drop-down box on the left edge of the dialog box to view the contents of this area. Input boxes in this area:

- Command monitoring time (Job_Processing_Task)
  Set this time to determine the maximum execution time of all concurrent Motion Control commands.
  If the command monitoring time is exceeded, the Technology CPU changes to the safe operation mode STOP. An error message is entered in the diagnostic buffer.
- Technology DB update time
  Select an entry from the drop-down list to determine the basic cycle used by integrated technology to update the technology DBs (also see "Updating technology DBs" (Page 791)). Frequent update operations will prolong the execution of Motion Control commands (see Command monitoring time).
  We speak of an "overflow" situation when a new Technology DB update is initiated while a previous update cycle is not yet completed. When operating with integrated technology firmware V3.1.x or earlier, the Technology CPU will go into STOP when it detects a technology DB update overflow. Integrated technology with firmware V3.2.x or higher tolerates this overflow, meaning that the non-executable update is discarded, and technology synchronization interrupt OB 65 will not be called. The number of overflows, and therefore the failure rate of Technology DB updates, can be read from the *TODBTaskOverflows* variable at Technology DB "MCDevice".
- Maximum shutdown time
  Enter the shutdown monitoring time for integrated technology. If the shutdown of integrated technology is not completed within this time the technology CPU automatically goes into STOP.
- Number of tolerated IPO overflows / IPO2 overflows
  An "IPO overflow" occurs when the interpolator execution time exceeds the configured interpolator cycle. In this case the following interpolator cannot be started in the selected cycle and will be omitted. If the processing of the interpolator is completed at the next cycle, it will be started as usual.
  Each omitted interpolator will be added to the number of "IPO overflows." Select the number of successive "IPO overflows" to be tolerated before the Technology CPU goes into STOP operation mode from the the drop-down list box. The number of overflow can be set separately for IPO and IPO2.

Network settings

Check your DP(DRIVE) settings in HW Config in this area.

8.3.2 Checking the load on integrated technology

The typical use of memory by technology objects and cam disk IPOs is described in the "CPU Data" documentation of your Technology CPU. Determine the approximate percentile value of memory allocation based on the values described in this documentation. This approximate value should normally not exceed 80 %.

You can check current memory load in S7T Config.

Checking the load on integrated technology

How to check load on integrated technology.

1. Download all technology objects required to the Technology CPU.

   Verify that the cam disks contain all interpolation points required. When changing cams dynamically in runtime using the "MC_CamClear", "MC_CamSectorAdd," and "MC_CamInterpolate" technology functions, use these functions to create the interpolation points.
2. Change to "Online" mode in S7T Config.
3. Select the **Target system > Device diagnostics** command in S7T Config.

   The "Device diagnostics" dialog box opens.
4. Select the "System load" tab.

This tab shows current memory load and "CPU load due to system tasks" of integrated technology. The indicated memory allocation value should normally not exceed 80 %.

"CPU load due to system tasks" indicates the load on integrated technology caused by the position control and interpolator cycles (ipo and ipo2) by percentage (for example, 40%.) The remaining time slice (60%, for example) is available for Technology DB updates and command processing.

You can modify CPU load by adjusting the system clocks:

- Shorter system clocks increase control performance, but reduce the time slice for command processing. Technology DB updates may require a longer clock.
- Longer system clocks, however, increase computing time for command processing and Technology DB updates. This reduces control performance.

The value of "CPU load due to system tasks" should normally lie between 40% and 60%.

8.4 Optimizing the position controller

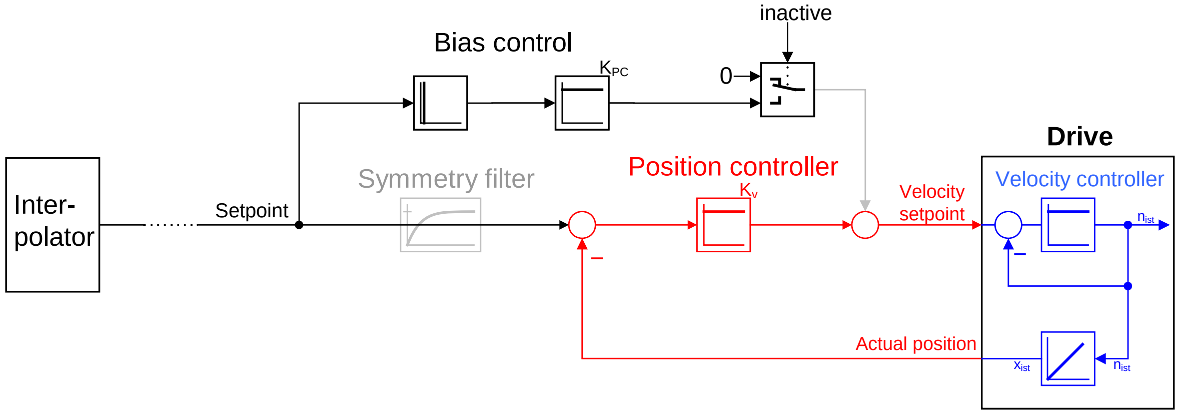
8.4.1 Optimizing the position controller - Overview

Tune the position controller of your axes if you set the focus on precise drive positioning without oscillation. The description below is a practical example which can be varied as required.

8.4.2 Preparing for tuning

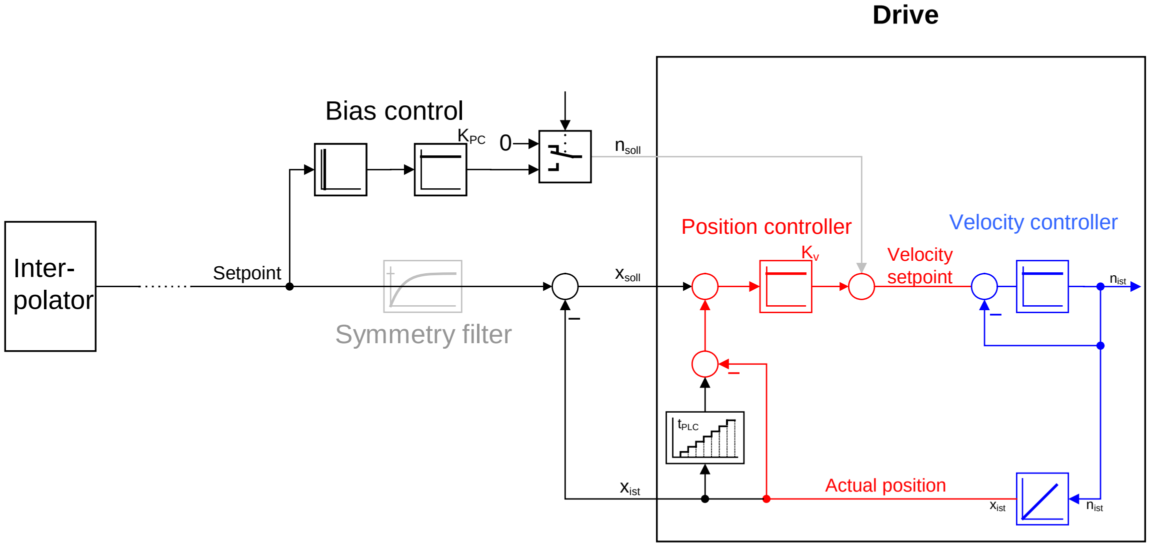
The position controller must be reduced to its basic functionality for tuning, that is, the "pre-control" and "balancing filter" functions must be disabled.

Block diagram of a position control loop without DSC



In a system operating a position control loop without DSC (Dynamic Servo Control), the system calculates the position controller in the Technology CPU, and the speed controller of the drive. The parameters are calculated in different cycles.

Block diagram of a position control loop with DSC



For position control with DSC (Dynamic Servo Control), the system calculates both the position controller and the speed controller in the drive. This allows you to set significantly higher gain factors (Kv) for the position controller. The position and speed control parameters are calculated within the same cycle. Hence, the position control and its subordinate control loop, meaning the speed controller, are calculated at precisely the same speed.

Do not disable Dynamic Servo Control (DSC) when tuning if you want to operate the drive in tuned state with DSC.

Prerequisites

- You configured the axis, with the exception of acceleration limits and input of the controller settings.
- You commissioned the lower-level control loops, and set a speed control loop that is free of overshoot when applying pre-control settings.
- The drive, encoder and technology object scaling match.

Tune the position controller by making the settings outlined in the next section. Make a note of all originally set values before you change these in order to be able to undo these settings after tuning.

- Disable speed pre-control
  Deactivate the "Pre-control" check box on the "Static controller data" of the **Axis > Control** dialog box.
- Fine interpolator "constant acceleration interpolation"
  elect "constant acceleration interpolation" from the "Fine interpolator" drop-down list box. This drop-down list box is available in the **"Static controller data" tab of the Axis > Control** dialog box.
- Disable drift compensation
  Deactivate the "Drift compensation" check box in the "Static controller data" tab of the **Axis > Control** dialog box.

- **Disable friction compensation**
  Deactivate the "Enable friction compensation" check box in the "Friction compensation" tab of the **Axis > Control** dialog box. This setting is only available in expert mode.
- **Deactivate balancing filter**
  Change the value of configuration data element *TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct.PV_Controller.balanceFilterMode* to *Off*.
  Alternatively, the balancing filter can be deactivated with technology function "MC_WriteParameter" by means of parameter number *2035* with the value *89*.
- **Disable following error monitoring**
  Deactivate the "dynamic following error monitoring active" check box in the "Following error monitoring" tab of the **Axis > Monitoring** dialog box.
- **Positioning window / Standstill window**
  Enter a high value in the "Positioning window" and "Standstill window" input box of the **Axis > Monitoring** dialog box. A high value prevents any reaction of these monitoring functions during tuning.

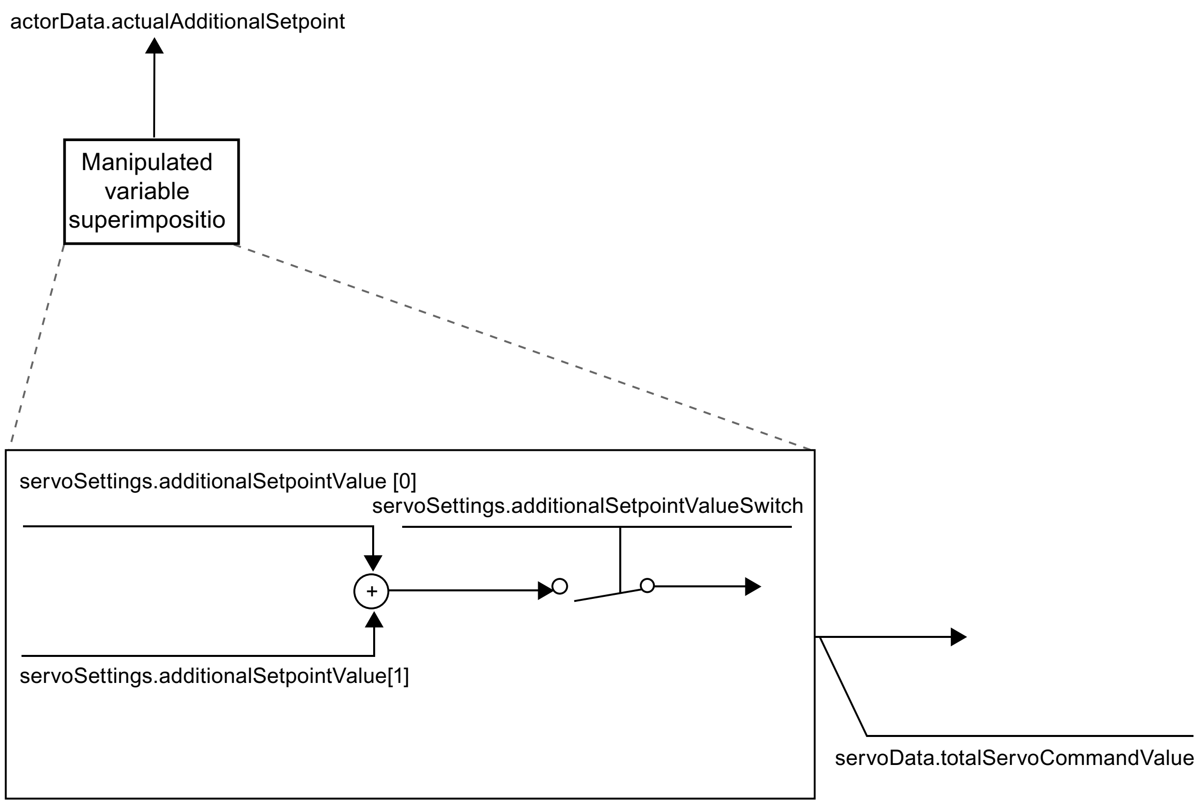
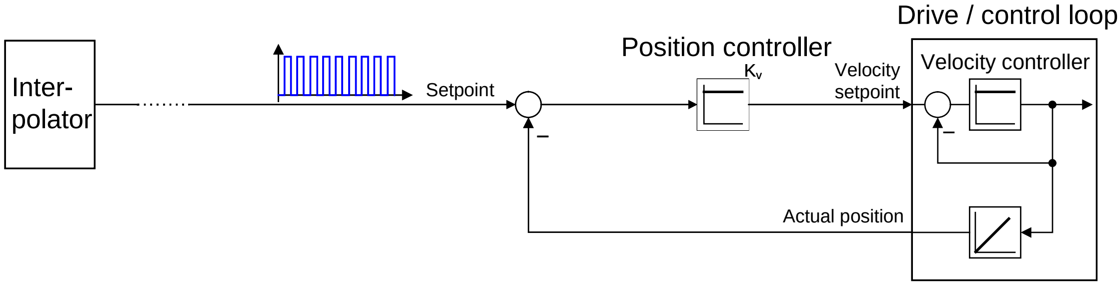


Figure 8-1 test

8.4.3 User program for controller tuning

Implement a test routine in the user program which sets step response values at the position controller.



The example below shows a practical function block. You can copy its contents from the Online Help to STEP 7 as STL source file.

The function block generates a step function. Using technology function "MC_MoveRelative," the axis position is increased by 100 mm, and is then returned to its initial position.

Input parameters of FB 130 PositionControl

Parameters	Data type	Description
Axis_No	INT	Enter the number of your axis Technology DB
Enable_Axis	BOOL	You can enable and disable the axis by setting this parameter *TRUE* = Enable axis *FALSE* = Disable axis
Start_Function	BOOL	*TRUE* = Start function *FALSE* = Stop function
Reset_Axis	BOOL	Use this parameter to "Reset" the axis
Restart_Axis	BOOL	Use this parameter to "Restart" the axis

Prerequisites:

Your project must contain the following technology functions. Copy the technology function from the block library of S7-Technology to your block folder:

- FB 401 MC_Power
- FB 402MC_Reset
- FB 411MC_MoveRelative

To generate the function block:

Copy the block code shown below to an STL source file. Change the block number of FB 130 if an FB 130 already exists in your project. Compile the block from the STL source.

STL source code

```
FUNCTION_BLOCK FB 130
TITLE = PositionControl
VERSION : 1.0
VAR_INPUT
   Axis_No : INT ;
   Enable_Axis : BOOL ;
   Start_Function : BOOL ;
   Reset_Axis : BOOL ;
   Restart_Axis : BOOL ;
END_VAR
VAR
   MC_Power_Axis : FB 401; // MC_Power
   MC_Reset_Axis : FB 402; // MC_Reset
   MC_Move_Axis : FB 411; // MC_MoveRelative
   Start_Optimation : BOOL ;
   Forward : BOOL ;
   Backward : BOOL ;
   Forward_Store : BOOL ;
END_VAR
BEGIN
NETWORK
TITLE =Enable Axis
      U       #Enable_Axis;
      =       #MC_Power_Axis.Enable;
      L       #Axis_No;
      T       #MC_Power_Axis.Axis;
NETWORK
TITLE =Reset / Restart Axis
      L       #Axis_No;
      T       #MC_Reset_Axis.Axis;
      U       #Reset_Axis;
      O       #Restart_Axis;
      =       #MC_Reset_Axis.Execute;
      U       #Restart_Axis;
      =       #MC_Reset_Axis.Restart;
NETWORK
TITLE =Generation of Test function for Optimization of Axis
      L       #Axis_No;
      T       #MC_Move_Axis.Axis;
      U       #Start_Function;
      FP      #Start_Optimization;
      O(      ;
      U       #Start_Function;
```

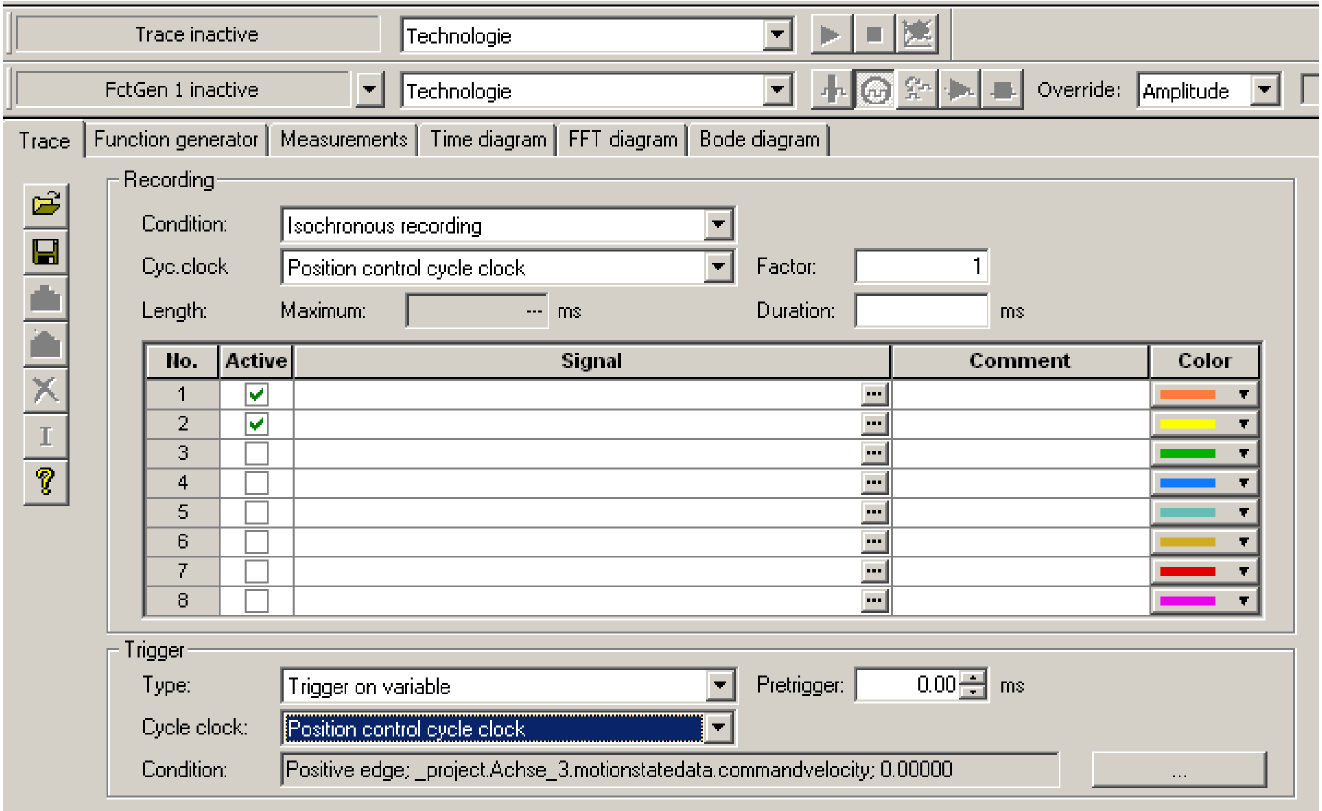
FUNCTION_BLOCK FB 130

```
      U       #MC_Move_Axis.Done;
      UN      #Forward_Store;
      )       ;
      =       #Forward;
      U       #Forward;
      S       #Forward_Store;
      U       #Forward;
      SPB     FWD:
      U       #Start_Function;
      U       #Forward_Store;
      U       #MC_Move_Axis.Done;
      =       #Backward;
      U       #Backward;
      R       #Forward_Store;
      U       #Backward;
      SPB     BWD;
      SPA     GO:
FWD: L        1.000000e+002;
      T       #MC_Move_Axis.Distance;
      L       1.000000e+002;
      T       #MC_Move_Axis.Velocity;
      L       1.000000e+004;
      T       #MC_Move_Axis.Acceleration;
      T       #MC_Move_Axis.Deceleration;
      SPA     GO:
BWD: L        -1.000000e+002;
      T       #MC_Move_Axis.Distance;
      L       1.000000e+002;
      T       #MC_Move_Axis.Velocity;
      L       1.000000e+004;
      T       #MC_Move_Axis.Acceleration;
      T       #MC_Move_Axis.Deceleration;
GO: U         #Forward;
      O       #Backward;
      =       #MC_Move_Axis.Execute;
NETWORK
TITLE =Call Motion Commands
      CALL #MC_Power_Axis ;
      CALL #MC_Reset_Axis ;
      CALL #MC_Move_Axis ;
END_FUNCTION_BLOCK
```

8.4.4 Optimizing the position controller

In this procedure you adapt the gain factor Kv of the position controller. Record and analyze the speed profile of the axis using the TraceTool of S7T Config during positioning. The user program described earlier controls the axis motion between two positions.

Settings in TraceTool



Set the following values in TraceTool of S7T Config:

"Signals" drop-down list

Signals	Value
Signal 1	*_to."AxisName".motionstatedata. commandvelocity*
Signal 2	*_to."AxisName".motionstatedata. actualvelocity*

The recording signals are available in the "Project name" > Technology name > TO > "Axis name" path of the corresponding selection window.

"Recording" drop-down list

Recording	Value
Condition	*Isochronous recording*
Clock	*Position control cycle*
Factor	*1*
Duration	Approx. 5000 ms or longer

"Trigger" drop-down list

Trigger	Value
Type	*Trigger on variable - positive edge*
Variable	*_to*."AxisName".*motionstatedata.commandvelocity*
Clock	*Position control cycle*
Pretrigger	Approx. 6 ms or longer
threshold value	Approx. 0.01000 (this value is used as trigger threshold)

The variables are located in the "Project name" > Technology name > TO > "Axis name" path of the corresponding selection window.

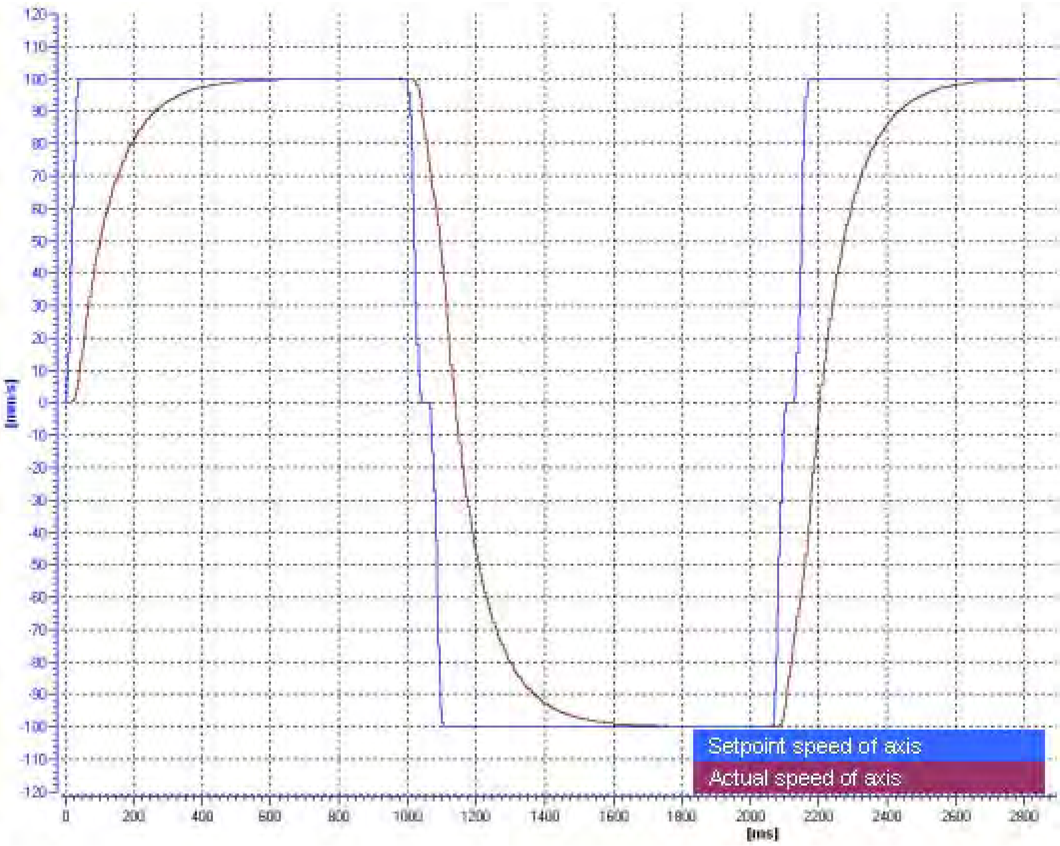
Analysis of the recorded Trace trends and adaptation of the Kv factor

In the trends shown below, we assume a low gain factor Kv with iterative adaptation for position controller tuning.

Set the Kv factor in the user program via parameter *2033* of technology function "MC_WriteParameter" or in configuration data element *TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct.PV_Controller.kv* .

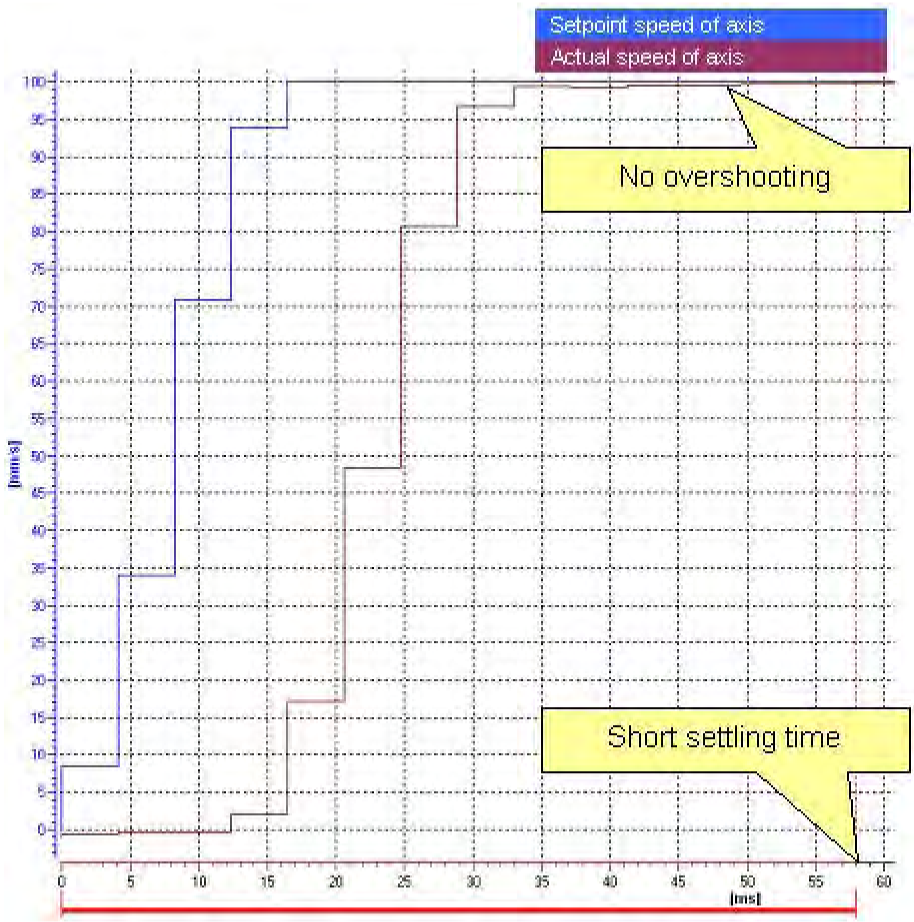
The Kv factor can also be set in the "Static controller data" tab of the **Axis > Control** dialog box.

Trend profile - gain factor Kv too low



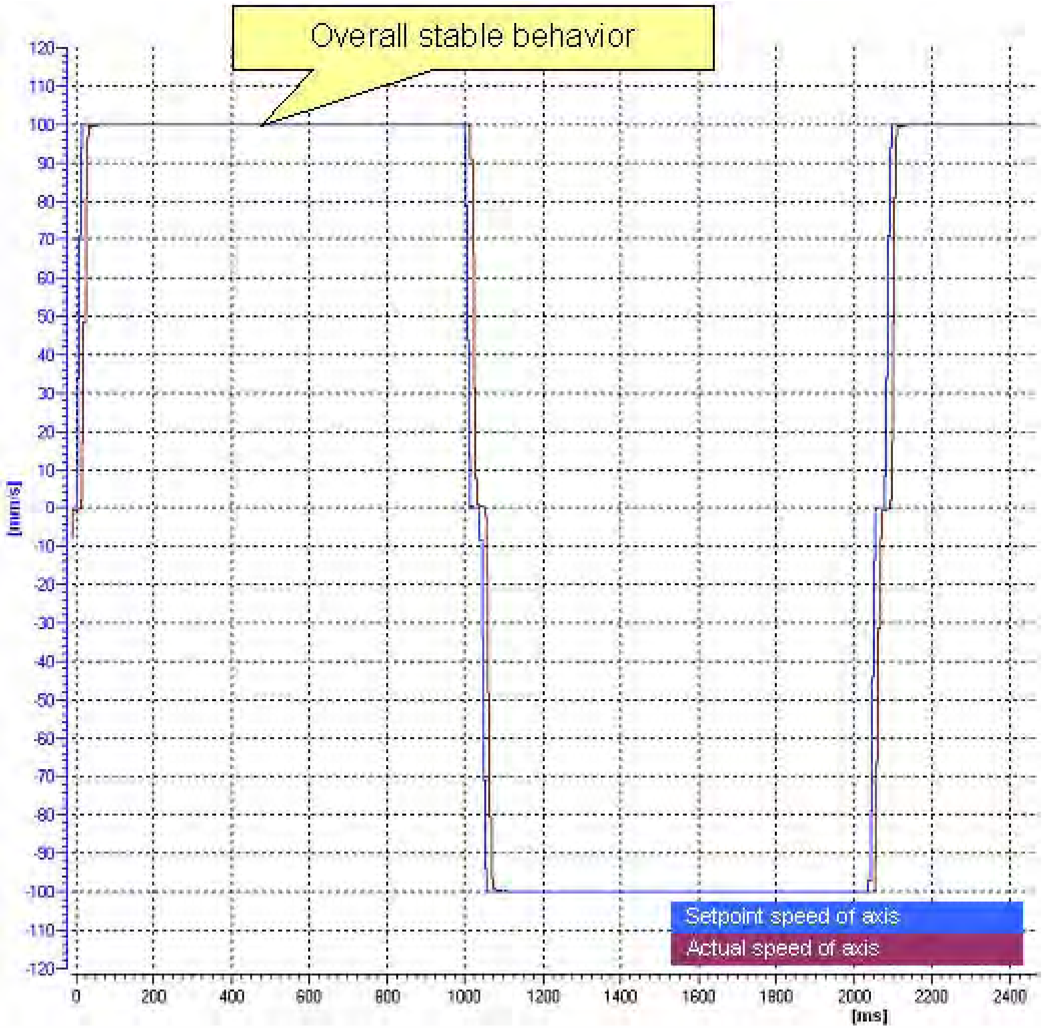
Although this trend profile does not show any overshoot, it discloses significant settling times. The Kv factor must be increased to tune the position controller.

Trend profile - optimized gain factor Kv (detail view)



When increasing the Kv factor, keep an eye on the following properties of the trend profile:

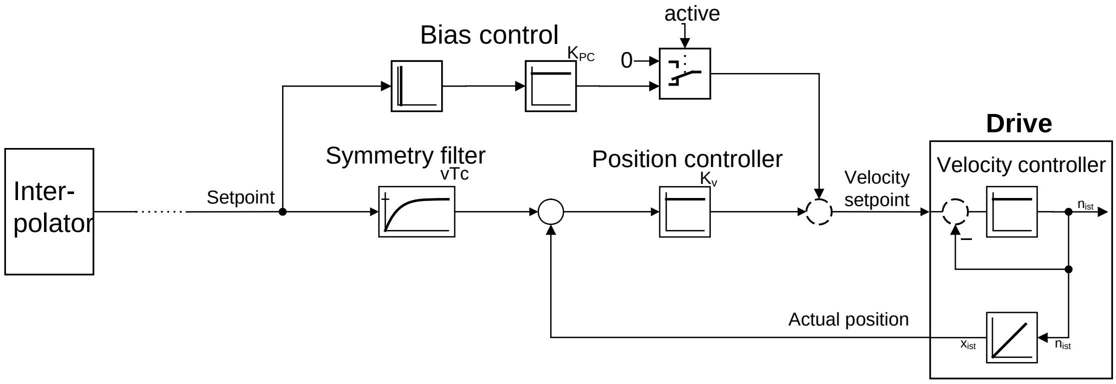
- The curve shows a short settling time.
- The trend does not disclose any reversal of the actual value.
- There is no overshoot when the setpoint is activated.
- The trend profile shows an overall stable response (no oscillation in trend profile.)



8.4.5 Tuning the balancing filter

After having tuned the gain factor Kv, you continue by tuning the time constant vTc (velocity Time constant) of the balancing filter.

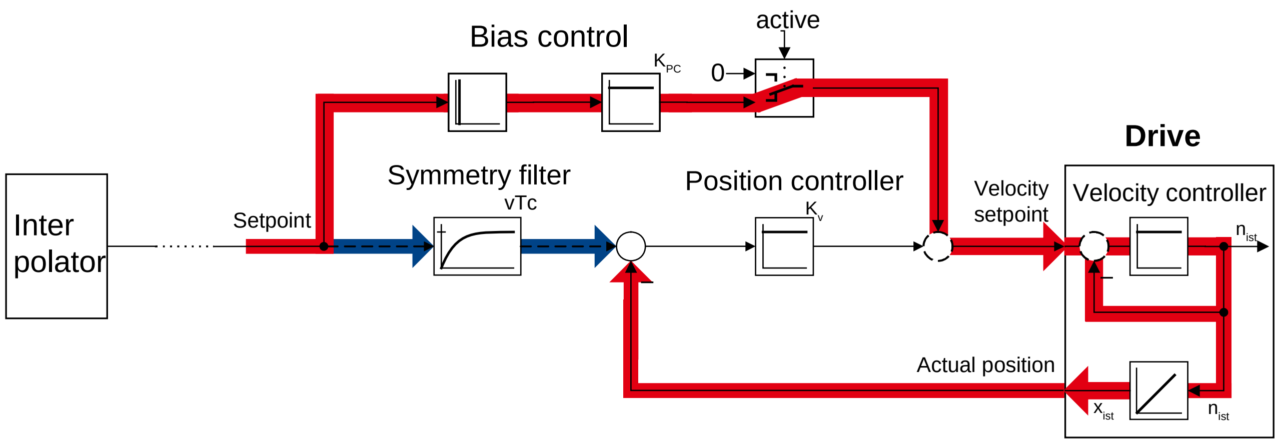
Block diagram of the position control



Setpoint changes require fast and precise settling of the controller. The Technology CPU thus controls setpoint deviations using pre-control and the position controller. The balancing filter compensates for the runtime delay of the pre-control function.

- **Pre-control**
  Differential function for high-speed reaction to setpoint changes. The pre-control function immediately settles the greater part of a setpoint change.
- **Position controller**
  The operation of a position controller is slower due to its p-action, however, with significantly higher precision. Its task is to fine adjust the residual control error the pre-control function has not yet settled. The error in a well-tuned control loop is small compared to the setpoint change.
- **Balancing filter**
  The balancing filter compensates for runtime delay caused by the pre-control circuit.

Function of the balancing filter



The pre-control function has a differential effect on changing setpoints, meaning that the resultant value is transferred to the drive at considerable speed. The speed controller in the drive corrects the portion of the setpoint change output by the pre-control; the drive moves. The actual value change is fed back to the position controller.

The position controller now has to correct the residual error between the defined position setpoint and the difference of the actual value generated by the pre-control. The position controller thus requires the setpoint at the correct time.

The balancing filter has to compensate for the runtime delay caused by the pre-control circuit. This requires a delayed transfer of the setpoint to the position controller.

Tuning the balancing filter

Use the user program described above and the TraceTool to tune the balancing filter. By contrast to previous changes made, change the settings for the signals to be logged:

Recording	Value
Signal 1	*_project."axis name".servodata. controllerdifference*
Signal 2	*_project."axis name".servodata. followingerror*

The balancing filter and pre-control must be activated before you can tune the balancing filter.

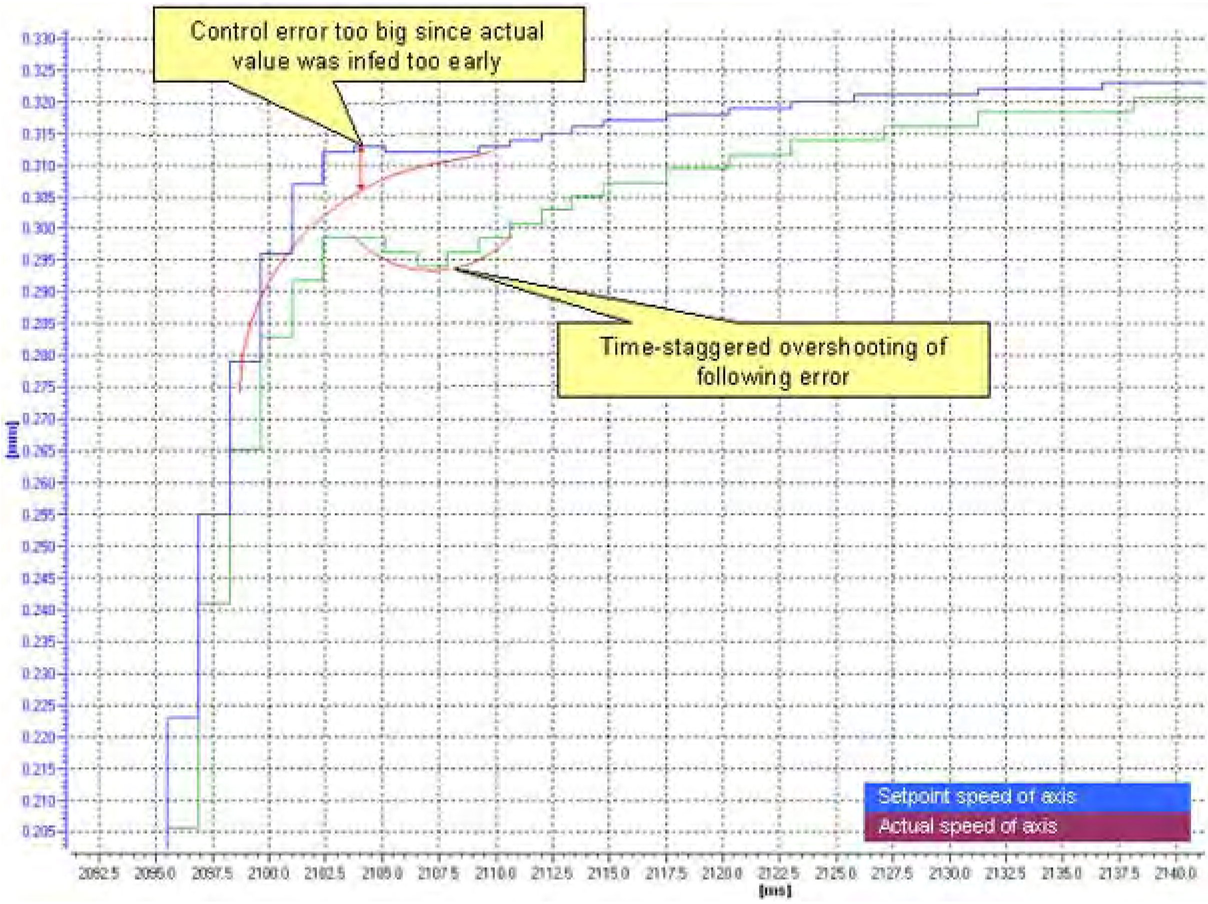
- **Activate balancing filter**
  Change the value of configuration data element *TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct.PV_Controller.balanceFilterMode* to the original value *Mode_1* or *Mode_2*.
  Alternatively, the balancing filter can be reset by using the parameter number *2035* of technology function "MC_WriteParameter." Select the value *90* for *Mode_1*, or the value *91* for *Mode_2*.
- **Enable speed pre-control**
  Activate the "Pre-control" check box on the "Static controller data" of the **Axis > Control** dialog box.

The objective of balancing filter tuning is to correct the time slice required by the pre-control.

- **Set time constant vTc**
  Set time constant vTc (velocity Time constant) in the user program via parameter number *2063* of technology function "MC_WriteParameter" or in configuration data element *TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicData.velocityTimeConstant* .

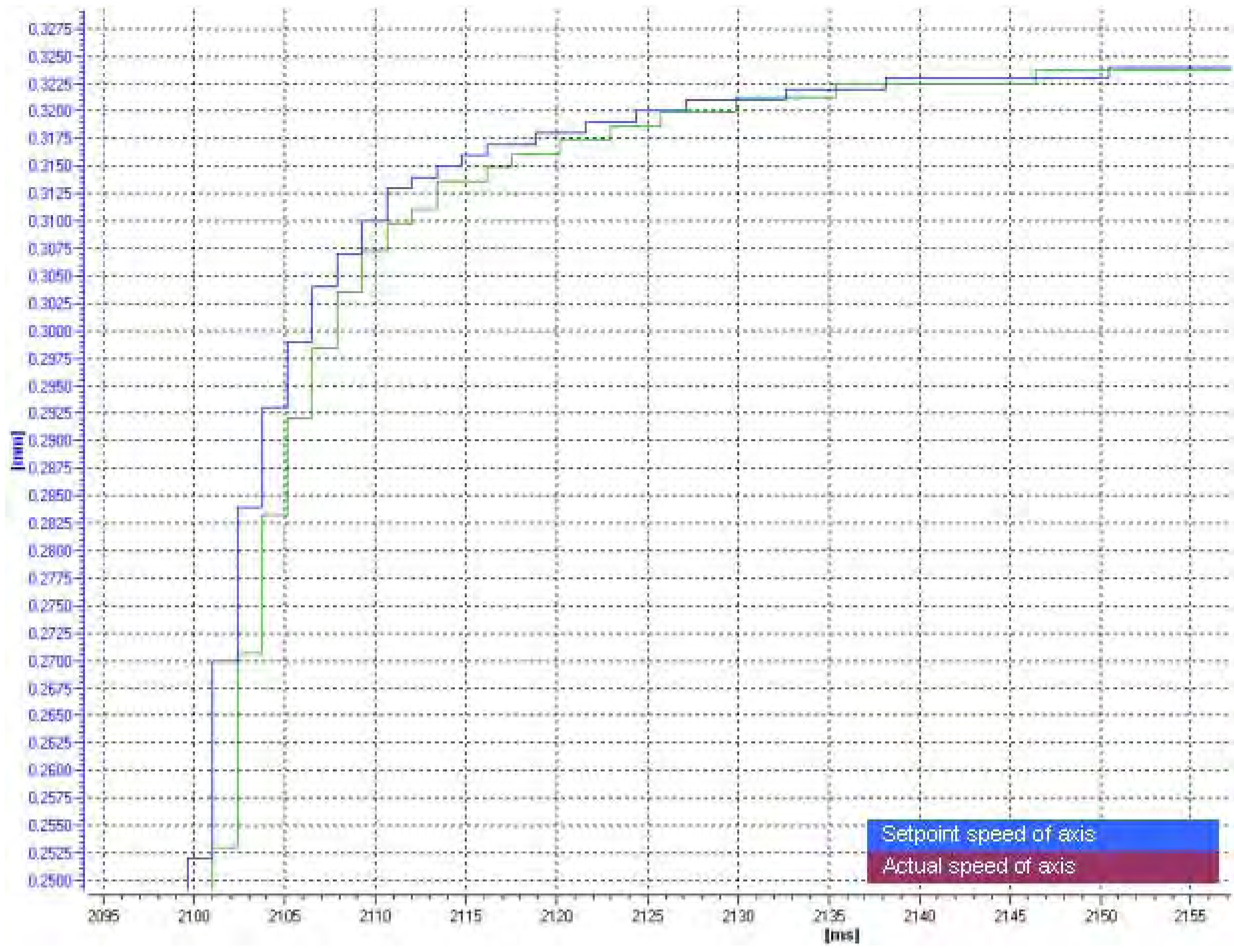
  You can also set the time constant vTc (velocity Time constant) on the "Dynamic controller data tab of the **Axis > Control** dialog box. The tab is visible when the "Expert" check box is set in the dialog box. Edit the value at parameter "Substitution time speed control loop" in this tab.

Example of a time constant vTc setting of insufficient length



The setpoint is output too early to the position control. The difference between the setpoint and actual value is too high, because the setpoint leads the returned actual value. The value output by the position controller in the control variable is therefore too high; the speed controller overreacts and the following error will deteriorate.

Example of an ideal setting of time constant vTc



The balancing filter is tuned to optimum performance. The position controller receives the setpoint in time with the actual value. The position controller only has to correct the slight deviations the pre-control has not yet corrected.

Tuning of the position controller is successfully completed when you have tuned the balancing filter.

Note

Note the following:

- The following error cannot assume a zero value if the constant velocity phase is configured without overshoot!
- As the position control cycle is included in the calculation of the equivalent time constant of the speed control loop (vTc), the maximum value of this time constant amounts to *16* position controller clocks. The dead time element has a maximum buffer length *16*.
- Do not forget to restore the original values after having completed the changes described in the chapter "Preparing for tuning".
- Make sure that you also change the tuned values of the Kv factor and of time constant vTc in the offline project

---

8.4.6 Calculating the equivalent time constant of the position control loop

The equivalent time constant of the position control loop comes into effect in the following situations:

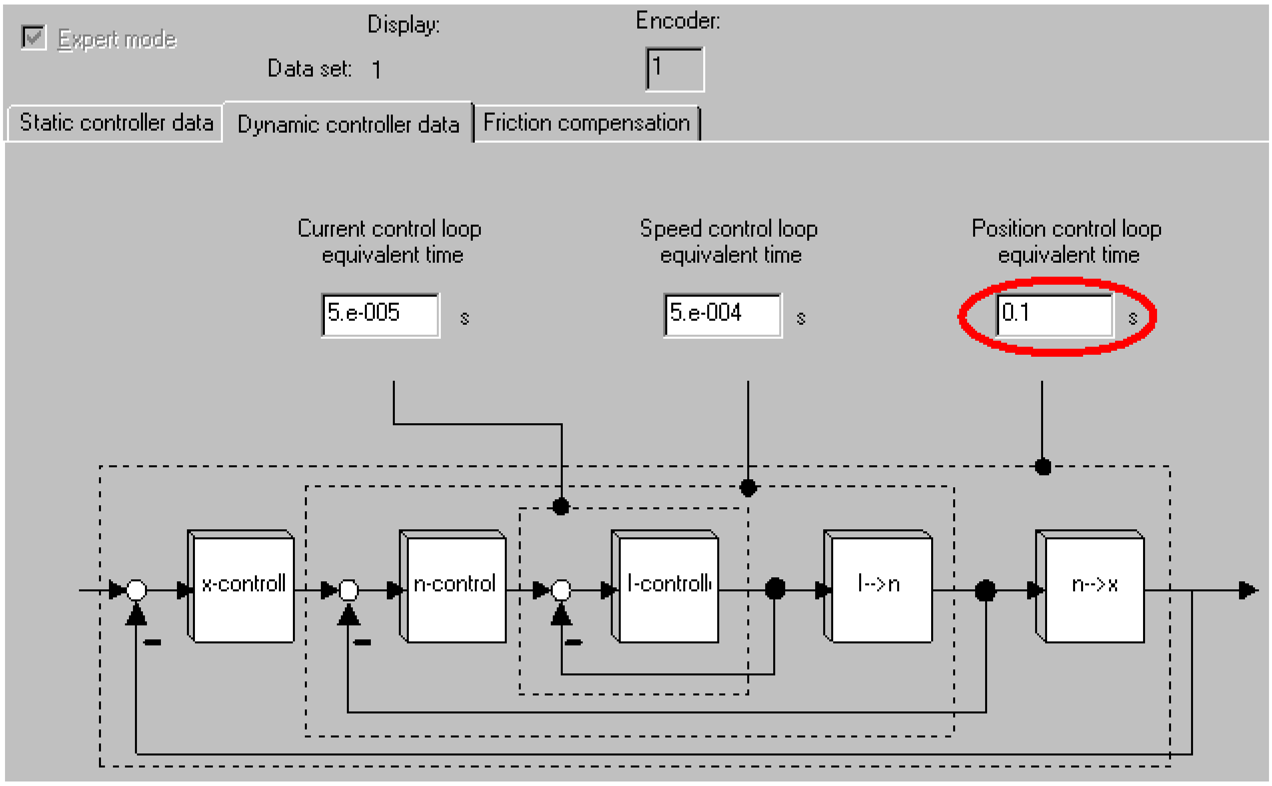
- Emergency Off by setting *StopMode = 0* at the "MC_Power" technology function
- Changeover from position-controlled to speed-controlled mode by setting *PositionControl = FALSE* at the ""MC_MoveVelocity" technology function.

NOTICE
If the equivalent time constant of the position control loop has not been tuned adequately, the axis may perform unwanted compensation motions in the situations described earlier.

To determine the equivalent time constant of the position control circuit pTc:

Proportional-action controller with precontrol	vTc <> *0*	pTc = vTc + Tservo/*2*
	vTc = *0*	pTc = *0*
Proportional-action controller without precontrol		pTc = 1/Kv

The equivalent time constant used in this case for the speed control circuit (vTc) can be determined as described in the "Tuning balancing filters" section.



Enter the calculated equivalent time constant for the position control loop thus determined in the "Dynamic controller data" tab of the **Axis > Control** dialog box. This tab can only be selected if expert mode is set.

The equivalent time of the position control loop can also be set in the S7T Config expert list in configuration data element *TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicData.velocityTimeConstant*.

Note

Use an emergency off command to verify the calculated value for the equivalent time constant of the position control loop. To execute the emergency off command:

1. "MC_Power" *StopMode = 0*
2. Falling edge at *Enable*of "MC_Power"

Increase the value of the equivalent time constant for the position control loop if the axis shows a tendency towards overshooting.

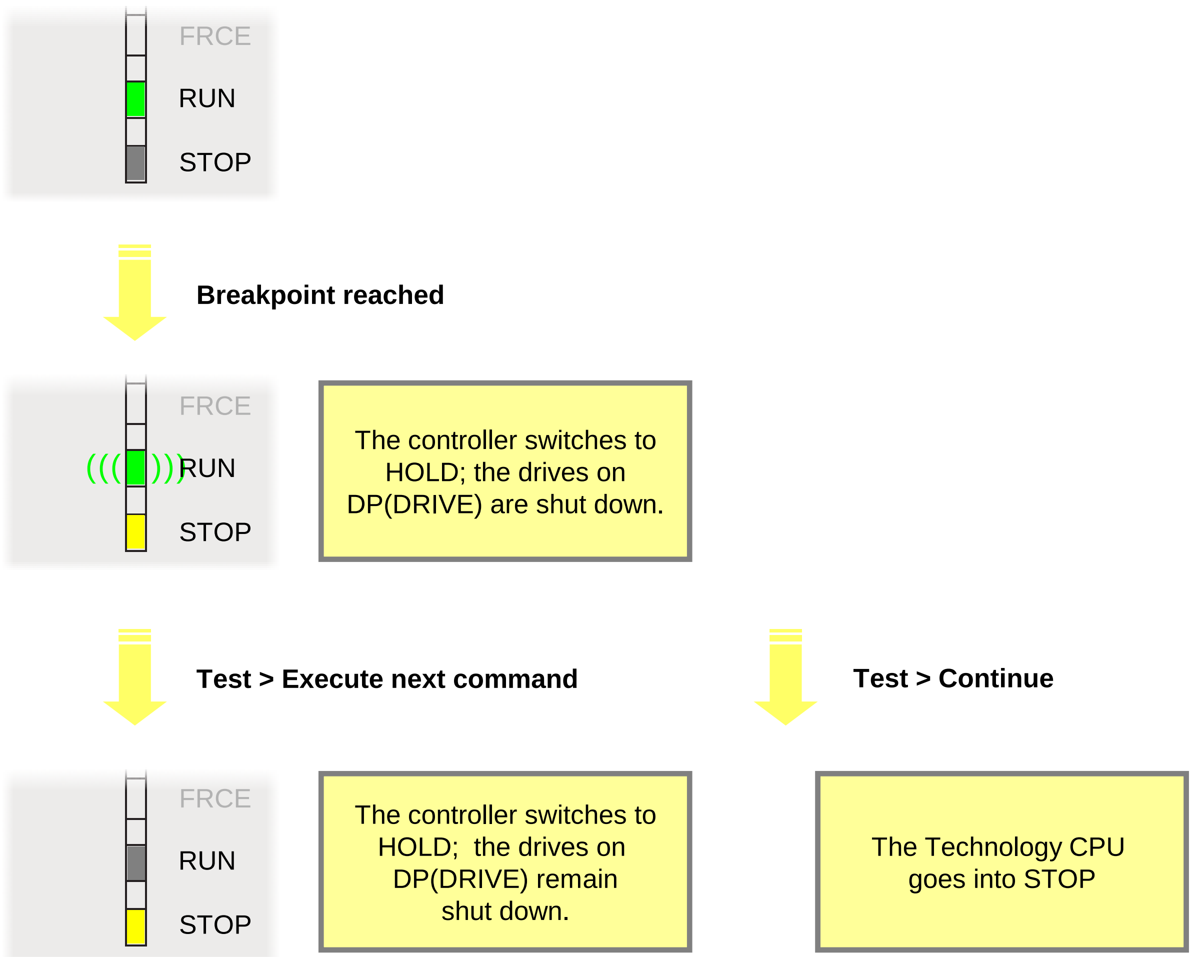
8.5 Testing with breakpoints

8.5.1 Testing with breakpoints

You can test your user program in the program editor of STEP 7 with breakpoints. Refer to your STEP 7 documentation for additional information.

Safety considerations result in various special features at the Technology CPU shown in the diagram below:

RUN state



Breakpoint reached

After the breakpoint is reached, the PLC changes as usual to "HOLD"; the "RUN" LED flashes at 0.5 Hz, the "STOP" LED is lit permanently. In "HOLD" mode, the outputs of the PLC and of its integrated technology are shut down.

Menu command **Test > Execute next statement**

After having reached the breakpoint, you can continue the test of your user program by selecting the **Test > Execute next statement** command.

Test > Resume command

The Technology CPU changes to "STOP" when you execute the **Test > Resume** command. "Restart" the Technology CPU in order to resume "RUN" mode.

8.6 Testing using the Watch Table

8.6.1 Using the Watch Table to monitor system variables

Use the Watch Table in S7T Config to monitor or control system variables of the technology objects. The Watch Table in S7T Config can be compared with the variable table of STEP 7.

You can group different variables (system variables of several technology objects, user variables of several programs) in a Watch Table and monitor this group.

Note

Monitoring of variables in the Watch Table requires an online connection.

How to create a Watch Table:

1. Open the Navigator and select the "MONITOR" element.
2. Select the **Insert > Watch table** command.
3. Enter the name of the Watch Table.
4. Confirm with "OK".

To add a variable to the Watch Table:

1. In the Navigator, select the element containing the variable you want to add to the Watch Table (the variable table is shown in the symbol browser).
2. In the symbol browser, select the rows containing the variables to be added to the Watch Table.
3. Select **Move to watch table** from the context menu.
4. Repeat steps 1 to 3 as often as necessary.

To monitor variables in the watch table:

1. Establish an online connection to the technology CPU.
2. Open the Watch Table.
3. Monitor the current value of the variable in the "Status value" column.

To control variables in the Watch Table:

1. Establish an online connection to the technology CPU.
2. Enter the value to be controlled in the "Control value" column. The check box in the cell will be set.
3. Click "Control immediately" to transfer the values to be controlled to the Technology CPU. Reset the check box if you do not want to transfer the value to the Technology CPU.

8.7 Using the TraceTool for testing

8.7.1 Overview - TraceTool

You can use the TraceTool of S7T Config to trace and save signal profiles and the values of variables. The data recorded can be used to analyze motion sequences of the machine, and for troubleshooting in user programs, for example. In order to achieve better diagnostics results, you can assign defined setpoints to variables using the function generator, while simultaneously recording the values of other variables. Recorded measurements can be displayed and edited in a diagram.

For information about the user interface and operation of TraceTool, refer to the S7T Config Online Help.

8.7.2 Tracing values of the CPU user program

The technology data component "Trace" is the interface between the trace tool and the application program. From the application program, 2 values of the data type DINT, 2 values of the data type DWORD, and 4 values of the data type REAL can be written to the respective variables of the trace DB for tracing. The *ErrorID* variable may not be overwritten by the user program.

The integrated technology reads the DB data within the technology DB update cycle. The technology DB update cycle can be set in S7T Config using the **Target system > Set system clocks** command.

The data block is non-retentive, irrespective of the retention settings in the CPU.

S7_TraceDINT[0..1] variable

The *S7_TraceDINT* variable contains an ARRAY with two elements of the data type DINT.

These values can be selected for tracing when you select the signal in *Technology > userdata1 > user1* and *user2* in the TraceTool of S7T Config.

S7_TraceDWORD[0..1] variable

The *S7_TraceDWORD* variable contains an ARRAY with two elements of the data type DWORD. Use the variables if you want to trace bit information (such as status word or error word) in the S7T trace tool. You can select the bits in the usual SIMATIC numbering in the trace tool.

These values can be selected for tracing when you select the signal in *Technology > userdata1 > user3* and *user4* in the TraceTool of S7T Config.

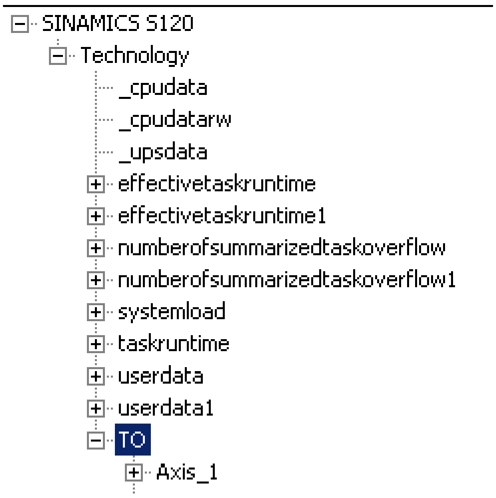
S7_TraceREAL[0..3] variable

The *S7_TraceREAL* variable contains an ARRAY with four elements of the data type REAL.

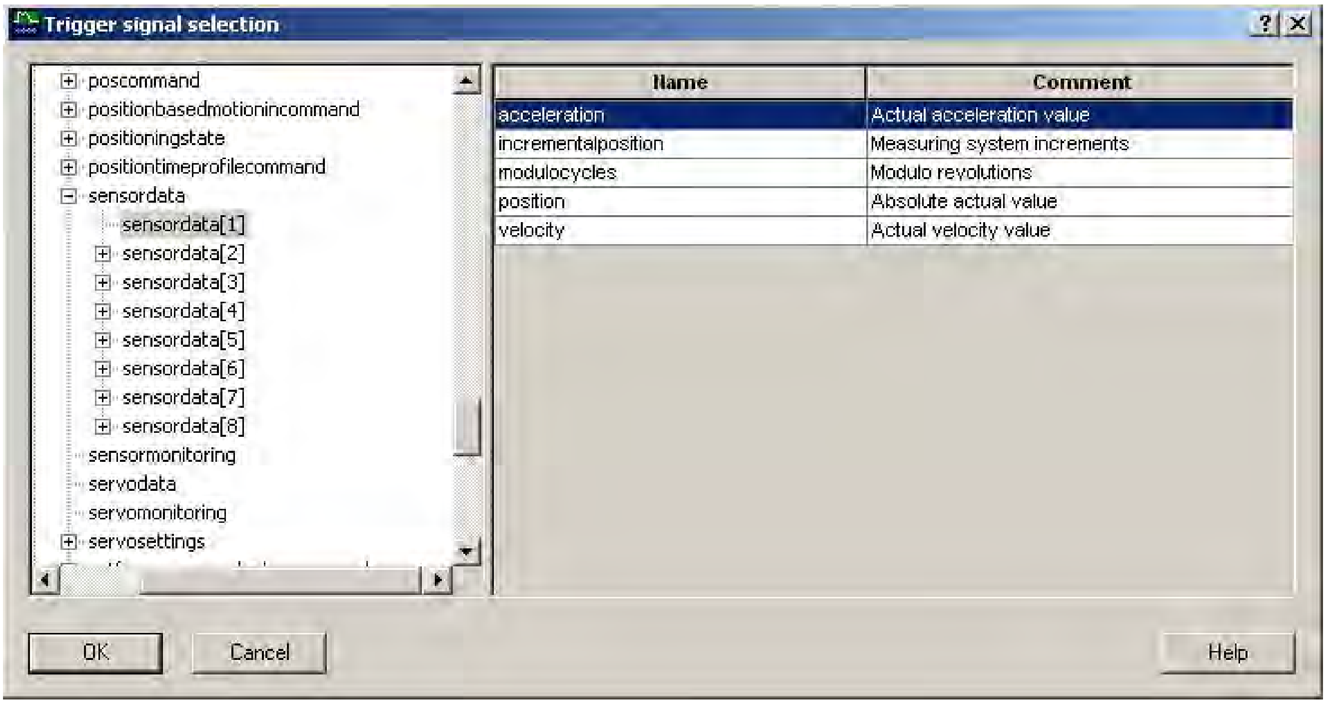
These values can be selected for tracing when you select the signal in *Technology > userdata1 > user5, user6, user7* and *user8* in the TraceTool of S7T Config.

8.7.3 Trace of Controller Data

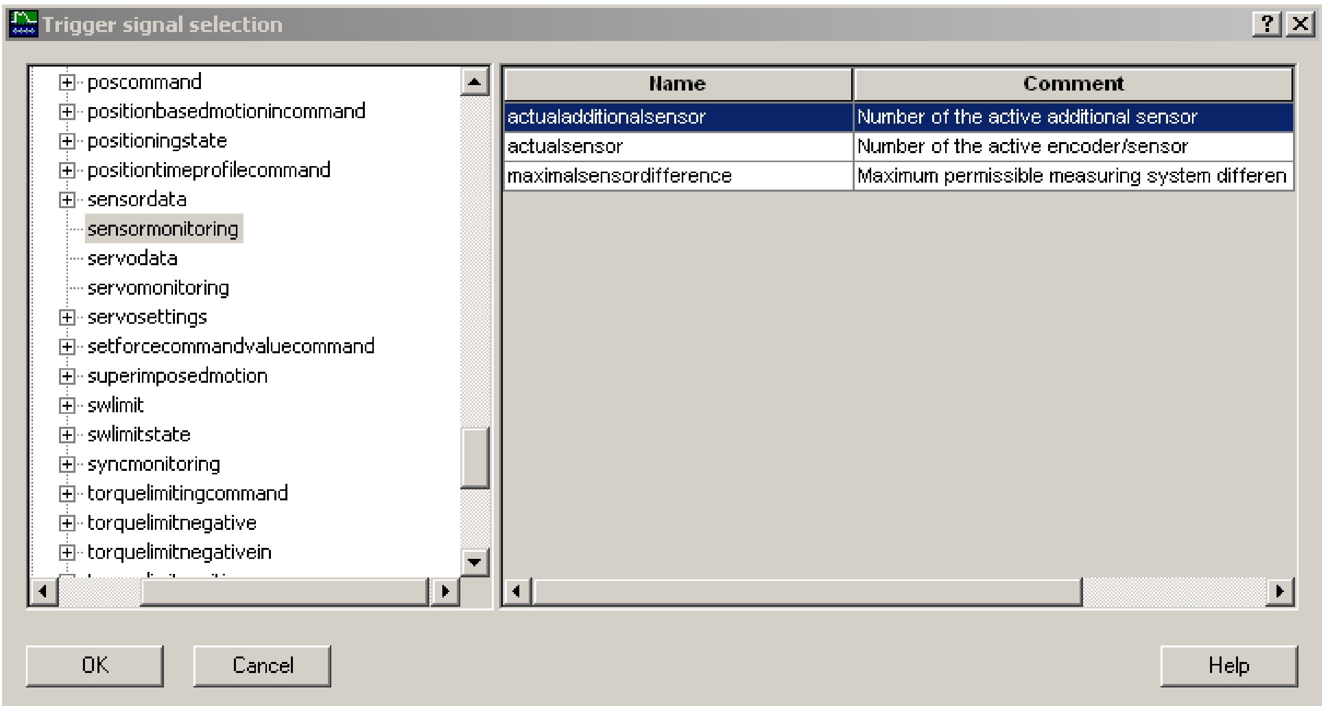
The trace tool is especially valuable for monitoring the dynamic controller data. In the following, some interesting trace values are listed, which will help you optimize controllers. The trace values can be viewed in the signal selection of Trace by selecting **Project name > Technology > TO > Axis name**:



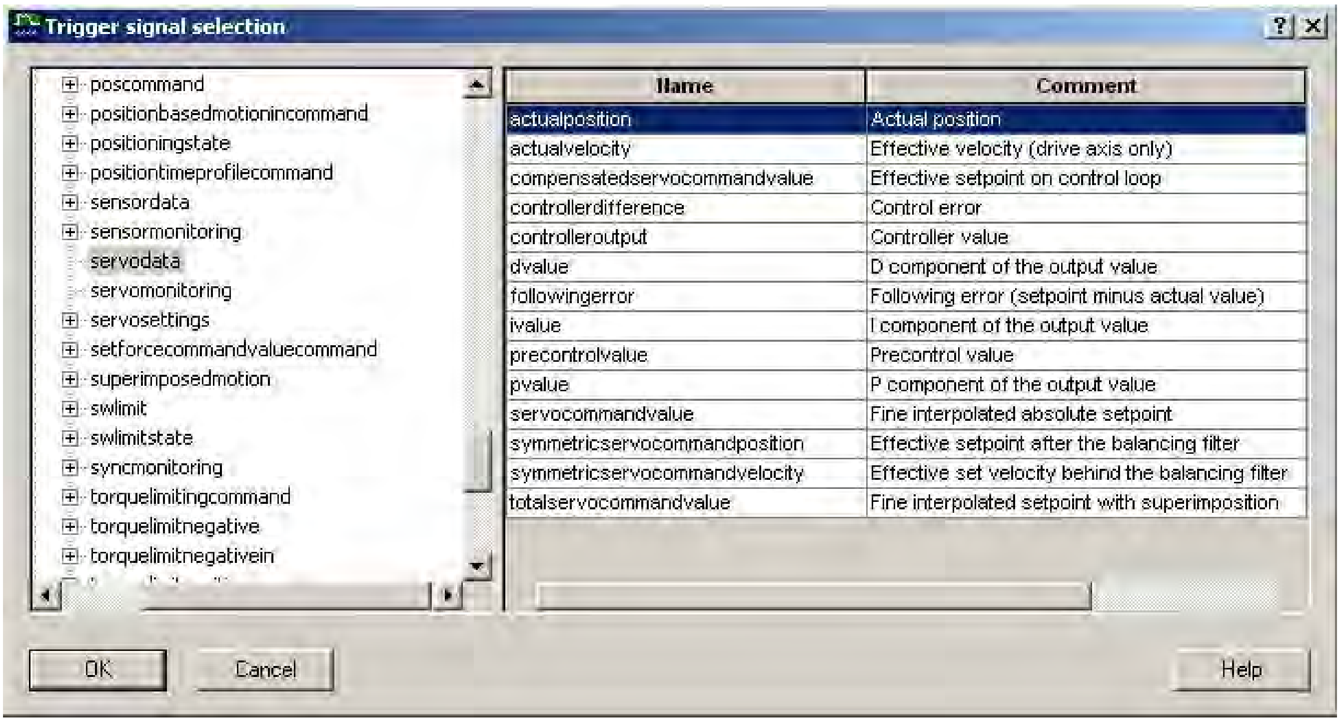
sensordata



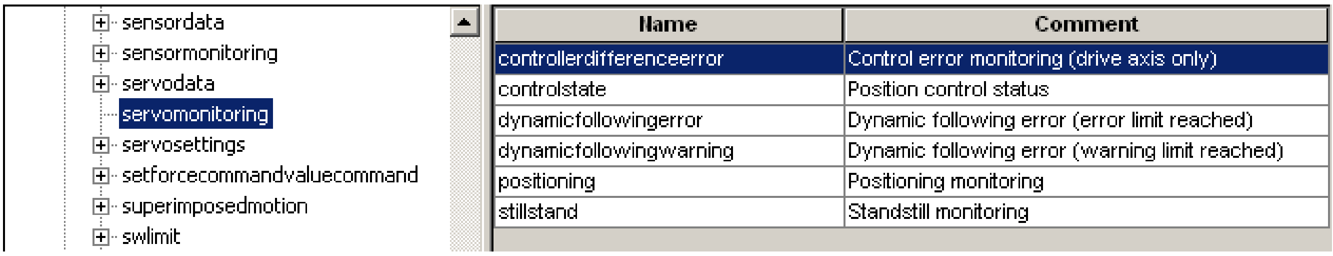
sensormonitoring



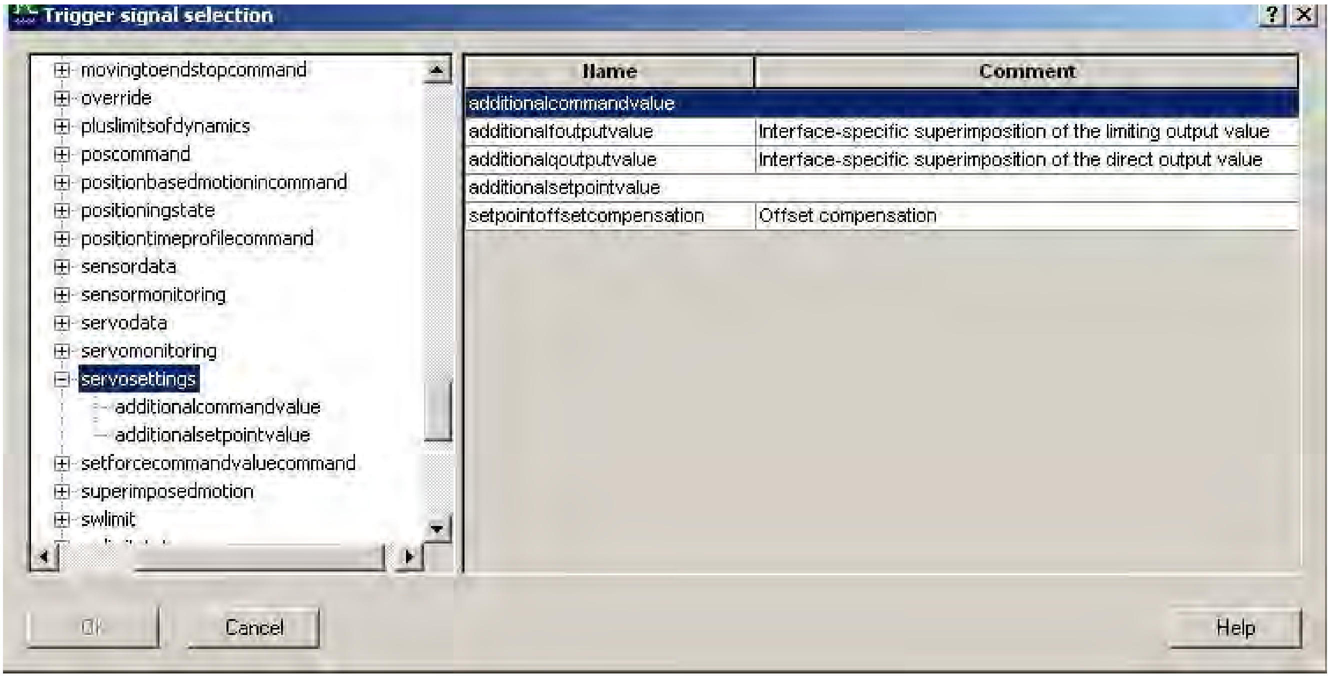
servodata



servomonitoring



servosettings



8.8 Testing using the control panel

8.8.1 Introduction - Control panel

The control panel is used to control and monitor individual axes and drives. You can use it to move drives or axes. This allows the commissioning engineer:

- To test each plant component individually before he initiates program driven, coordinated axis motions
- To simulate error conditions in order to test whether the various axes can be operated at all from the control panel, or whether this section is already posing problems
- Moving the axis quickly to another position, independent on the program: "Retract the axis"
- To traverse the axes for optimization purposes (controller optimization)
- perform active homing
- to position in relative / absolute mode
- To set and reset the axis enable signal
- To trigger an axis reset
- To adjust the absolute value encoder by adding the absolute value encoder offset ("Homing" dialog box)

Note

While the control panel is active and S7T Config has control priority, you can only edit configuration data that become immediately active.

Control priority must be returned before you edit any configuration data which require a RESTART. Control priority must be requested again after RESTART

⚠ WARNING
Use the axis control panel only in the commissioning phase and for test purposes.

8.8.2 Starting the Axis Control Panel

Verify that the following conditions are met:

- The axis is configured.
- S7T Config is in online mode.
- The technology was downloaded to the target system.

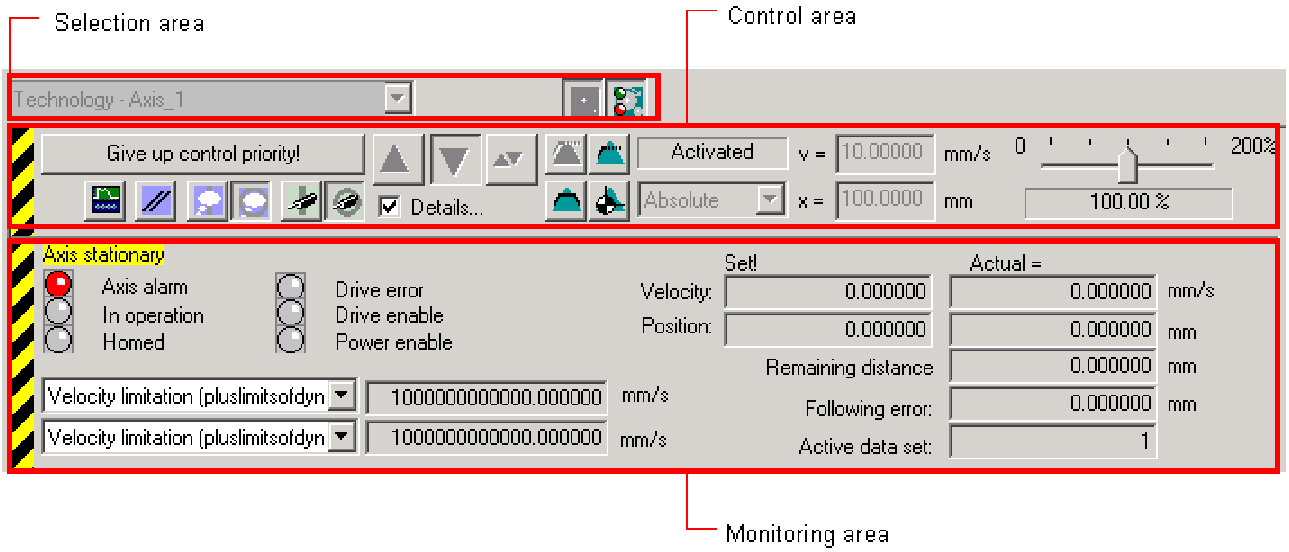
To start the axis control panel:

1. In Navigator, open the "AXES" folder.
2. Select the required axis.
3. Select **Edit > Open Object**.
4. Select the **Axis > Control Panel** command.

The "Control panel" tab opens in the detail view. The control panel is now in monitoring mode.

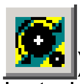
8.8.3 Layout of the Axis Control Panel

The axis control panel is organized by several areas:



Axis control panel (the CP is in control mode)

- **Selection area**
  You can hide or show the other areas by pressing the corresponding buttons.
  The axis can be selected in monitoring mode. All configured axes of the technology can be selected.

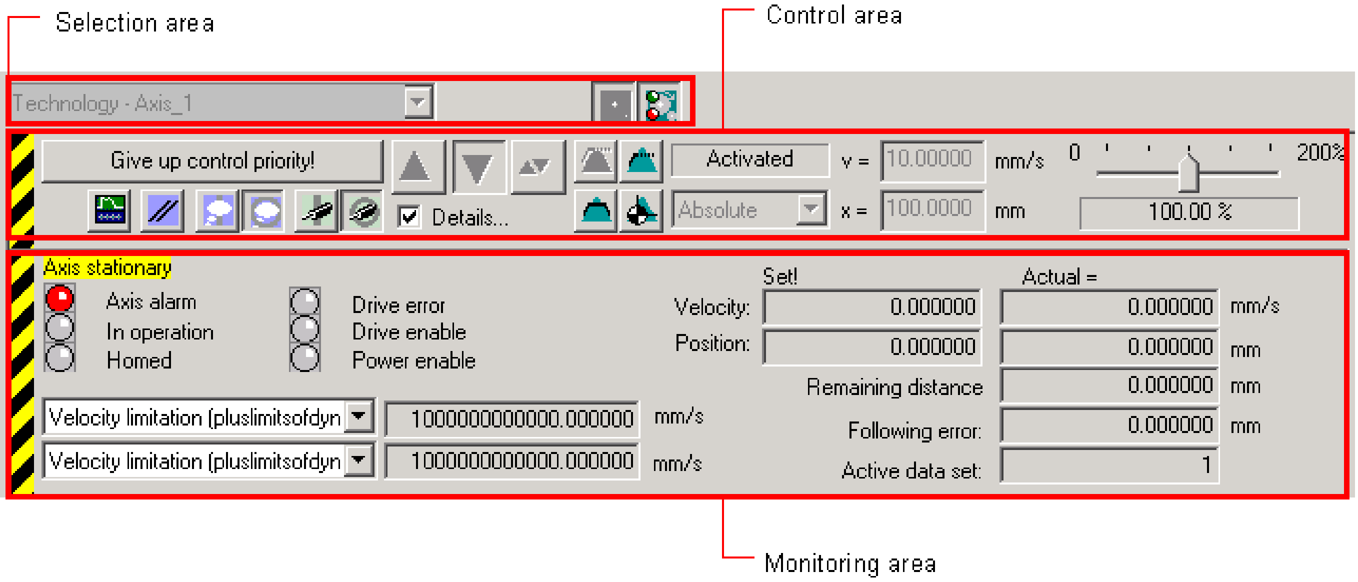
- **Control area** (Show and hide by pressing the button ).
  In this area you can assume control priority at the selected axis, or transfer motion commands to the axis.
  Only the "Trace Editor" button is active in monitoring mode.

- **Monitoring area** (show and hide using the button )
  The values of selected axes can be monitored in this area:
  - in monitoring mode:
    You can test axis motions which are initiated in the user program, for example.
  - In control mode:
    You can test motions which are initiated by command output to the axis.

8.8.4 Monitoring the axis values using the control panel

You monitor values of the selected axis on this area in monitoring mode and control mode:

- in monitoring mode (directly after having started the control panel, for example):
  You can test motions which are initiated by a program, for example.
- in control mode (after having assumed control priority):
  You can test motions which are initiated by command output to the axis.



Monitoring area of the control panel

The view is determined by the technology of the selected axis (speed-controlled, positioning or synchronization axis):

- Speed setpoint and actual value
- Position setpoint and actual value
- Distance-to-go
- Following error
- Two further system variables which you can select from the drop-down list box.

The motion state of the axis, for example, "Axis in stop" or "Axis accelerating", is visualized in a text box with a colored background.

The colored “LEDs" keep you informed of:

- Alarms and enable states of the axis:
  - Active technology alarm (red)
    Select the "Alarms" tab on the detail view for further information.
  - Active (green)
    The "LED" indicates whether following mode is cancelled. Axis commands can only be executed in this state.
  - Axis is homed (green).
- Monitoring functions and enable signals of the drive:
  - Active drive error (red)
  - Drive enable (green)
  - Power unit enable (pulse enable - green)

8.8.5 Assuming control priority

Verify that the following conditions are satisfied:

- The Technology CPU is in STOP mode.
- Select a STEP 7 variable table and activate the **Variable > Enable Peripheral Outputs** command

⚠ WARNING
You must strictly observe pertinent safety regulations. Use the axis control panel in control mode only if programming device / CPU lifesign monitoring is enabled and a suitably short monitoring time is set! The axis may otherwise perform uncontrollable motions if problems occur in the communication between the programming device and the Technology CPU.

Follow the steps outlined below:

1. Click "Assume control priority !" in the control area.
   A warning message box is output.
2. Read the warning message in S7T Config!
3. Do not change life-sign monitoring defaults.

   Changes should only be made in exceptional circumstances and in observance of all danger warnings.
4. Check the emergency stop setting. Change it if necessary.
   – Click "Extend>>".
   – For reasons of safety, you should set the "Emergency-Stop when changing to another Windows application" check box. Monitoring is not relevant to the function.
   – Set the key which triggers the emergency stop.

     <Spacebar>

     <Esc>

The CP is in control mode after it has assumed control priority. The previously grayed out buttons and fields on the control area are now active.

Note

When the control panel is in control mode, a space bar signal will be interpreted as a control signal to stop the axis. This also affects other applications running on the programming device (word processors, for example). If you assign the space bar to the emergency-stop function, its function is no longer available for entering space characters in a word processor.

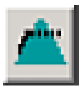
8.8.6 Controlling an axis

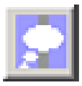
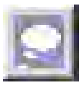
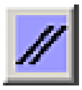
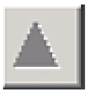
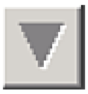
Verify that the following conditions are satisfied:

- The axis CP is in control mode.
- The CP has assumed control priority.

The buttons on the control area are active.

Control area of the axis CP
Control elements in the control area of the axis CP:

Command	Control element	Description
Set / reset enable signals		This command toggles the enable signals at the axis. States • Drive enabled • Pulse enable (power enable) issued Are indicated at the corresponding "LEDs" on the monitoring area. To permit the motion commands to be processed on the axis, all enables must be issued and the follow-up operation cancelled. For speed-controlled axes, the position control enable is ignored.
Details	Check-box	When this check box is set, certain commands will open a separate parameter input box. If the check box is deactivated you can enter selected parameters in the input fields next to it.
Speed-controlled axis motion		This command specifies that the axis is started speed-controlled. A speed setpoint is preset to which the axis accelerates along a configured ramp. The command is available to all axis types. Positioning and synchronization axes are speed-controlled. Start the motion by clicking "Start" or "Jog".
Position-controlled axis motion		This command specifies that the axis is started position-controlled. A speed setpoint is preset to which the axis accelerates along a configured ramp. The command is possible for position and following axes. Start the motion by clicking "Start" or "Jog".
Relative / absolute positioning		The command positions the axis (position or following axis). The position specification can be absolute or relative. Modulo axes can also be positioned along the "shortest path". You start the motion by clicking "Start".

Command	Control element	Description
Axis homing		With an absolute measuring system, homing is only required once during commissioning. When this is completed, the position value is known when the machine is switched on. In the case of an incremental measuring system, the machine must be homed every time it is switched on. You start the motion by clicking "Start".
-	Input boxes	Input boxes used to enter the position and speed setpoints when the "Details" check box is reset
Velocity override	Slider control	The preset speed is multiplied by the set override factor (0% to 200%). For safety reasons, the value is set to *100* when the control panel is started. This value affects all subsequent commands.
Simulation ON		This command sets axis simulation mode. Setpoint output will be suppressed.
Simulation OFF		This command returns the axis from simulation mode to normal operation. The setpoint last valid in the simulation will be activated for output. **Caution:** As the drive is disabled in simulation mode, it may perform an adjusting movement when you exit simulation mode. You can avoid these adjusting motions by moving the axis to its initial position while you are still in simulation mode.
Reset axis		The axis is reset to the initial state. The command should be used when, because of incorrect parameterization, the axis is in an inexactly defined state.
Start motion		The last motion command configured (speed preset, start axis with position control, position axis or home axis) will be started.
Stop motion		The motion started last is stopped.
Jogging		"Speed preset" and "Position-controlled start of the axis motion" commands can also be executed in jog mode. The motion continues as long as you keep the right mouse button pressed on the button.
Execute trace recording		Starts the TraceTool.

NOTICE
Take all safety measures before you put the axis into operation.

To move an axis using the axis CP:

1. Click "Details".
2. Click "Enable axis".

   On the window shown, check that all enable signals are set, and that following mode is disabled. Confirm with "OK".
3. Click a motion command button ("Speed preset", "Start axis positioncontrolled", "Position axis" or "Home axis").

   This opens the next dialog box. There, enter your parameters. Confirm with "OK".
4. Set an appropriate override value at the "Speed Override" shift register.
5. Click "Start".

   This starts the motion.

   You can modify the speed override factor while the motion is active.
6. You can stop the motion by clicking "Stop," or by pressing the space bar.

   You can resume the motion by clicking "Start".
7. Repeat steps 3 to 6 as often as required.

   Note

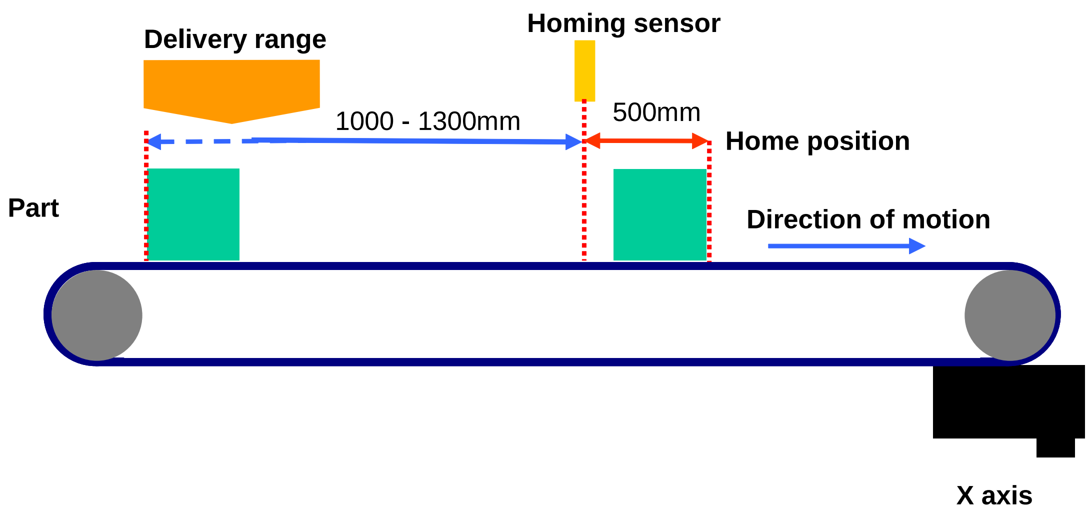
   When using the "Speed preset" and "Start axis position controlled" commands, you can also start the motion by clicking "Jog". Keep the left mouse button pressed to run the motion as long as necessary. The motion stops when you release the mouse button.

   You can also stop the axis by pressing the space bar.

Appendix 9

9.1 Application examples

9.1.1 Application example "Positioning with target sensor"



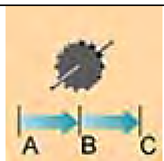
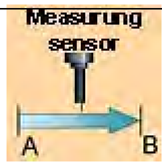
The example shows a typical application of a positioning task on a conveyor.

A workpiece is to be moved to a defined end position. The discharge area is variable, so that the current position of the part is not known at the outset of the motion.

A target sensor attached at a distance of 500 mm downstream of the end position enables precise positioning of the part. The target sensor is connected to a digital input of the drive. The actual position value of 500 mm will be set at the conveyor when the sensor is triggered. The part position and measuring input data are saved without noticeable delay thanks to the integrated technology of the Technology CPU. The actual position value is known and positioning can be completed at high accuracy.

Technology Objects

In S7T Config, you configure the technology objects which you can use to control the conveyor drive and the target sensor. Technology objects used in this example:

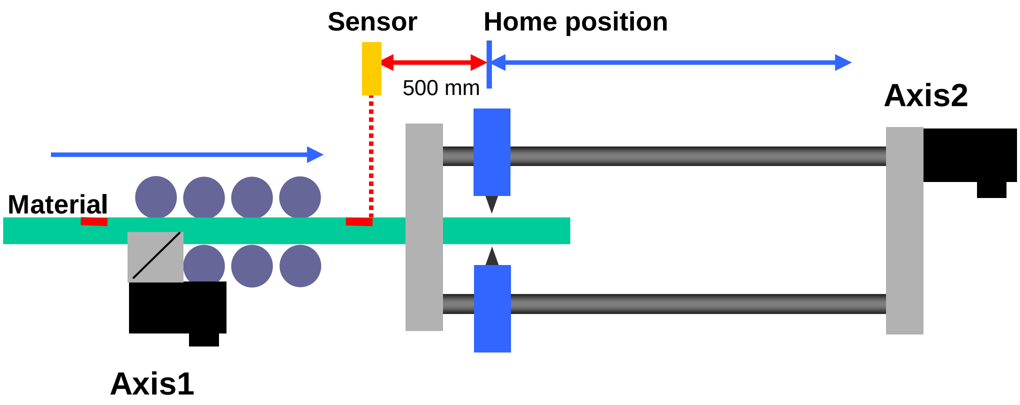
Technology objects: • Positioning axis (conveyor) • Measuring input (target sensor)		

Technology functions

The technology functions required to solve this task are executed by calling the function blocks listed below in the user program:

	Technology function	Task/description
1.	MC_Power	Enable axis
2.	MC_MeasuringInput	Set the parameters and enable the measuring input (target sensor)
3.	MC_MoveVelocity	Move the axis with speed preset
4.	MC_Home	The position is corrected after the target sensor has been detected (actual position value – target sensor position). The position value will therefore be reset to zero when the target sensor is detected.
5.	MC_MoveAbsolute	Absolute positioning of the axis to 500 mm.

9.1.2 Application example "Flying shears"



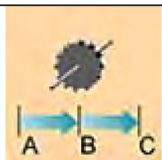
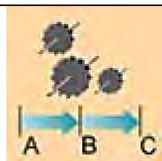
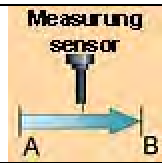
This example shows a typical application for cutting continuously produced material to length.

The incoming material is fed continuously by the feed axis (Axis1) and cut to length "on-the-fly" by the shearing axis (Axis2).

Pressure marks applied to the material indicate the cutting position. A sensor registers the position of the pressure mark and sets the actual value of the feed axis (Axis1) to -500 mm. The system starts synchronous operation of the shearing axis when it detects the sensor. The shear blades approach the pressure mark while the system performs this synchronization. The material is cut immediately after the shearing axis has reached synchronism. The shearing axis returns to its home position after having completed the cut and waits for the next pressure mark.

Technology Objects

You configure the Technology objects in S7T Config. Technology objects used in this example:

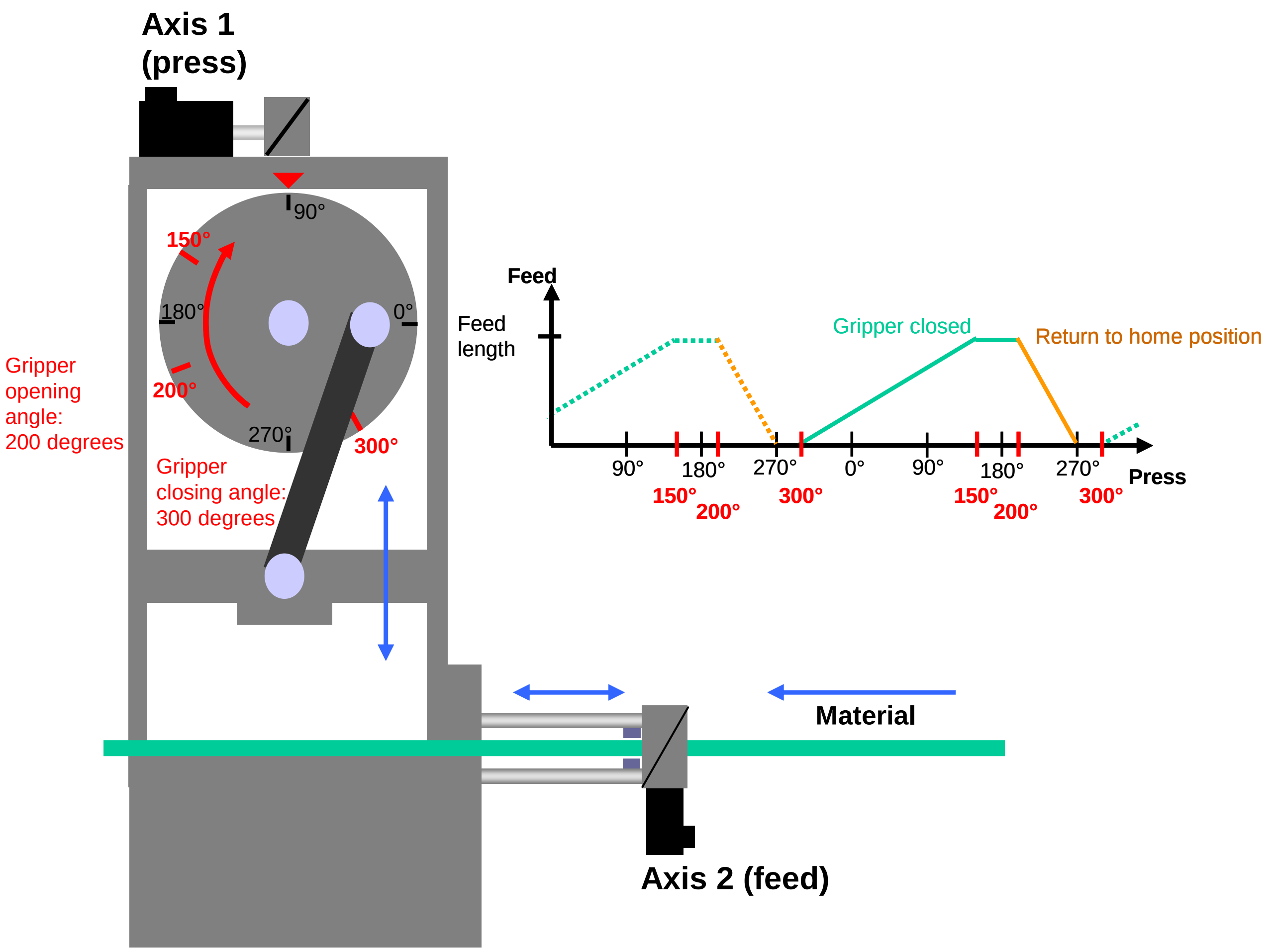
Technology objects: • Positioning axis (feed axis) • Synchronization axis (shearing axis) • Measuring input (sensor)			

Technology functions

The technology functions required to solve this task are executed by calling the function blocks listed below in the user program:

	Technology function	Task/description
1.	MC_Power	Enable feed axis
2.	MC_Power	Enable shearing axis
3.	MC_Home	Home the shearing axis in basic position to position value 0 mm
4.	MC_MeasuringInput	Set the parameters and enable the sensor
5.	MC_MoveVelocity	Move the feed axis at preset speed
6.	MC_Home	A position value of -500 mm will be set at the feed axis when the sensor detects the pressure mark. "MC_Home" in Mode 4 (position override): Xact = Xact - "sensor position" - 500 mm
7.	MC_GearIn	Synchronous operation of the shearing axis starts when the sensor detects the pressure mark
8.	MC_GearIn	The material is cut when the "InGear" value is returned (gear synchronism reached)
9.	MC_MoveAbsolute	The shearing axis returns to its home position
		Further cyclic processing at steps 5 to 10

9.1.3 Application example "Gripper feed"



The example shows a press to which the material is fed by means of a gripper feed mechanism.

The gripper closes at an angular position of 300 ° and then pushes the material into the press position at angular position 150 °.

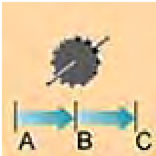
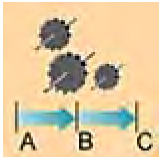
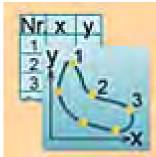
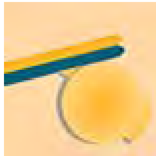
The gripper remains closed during the press operation and opens again at 200 °.

The gripper feed returns from angular position 200° to its home position at 270 °.

The motion of the feed axis is directly proportional to the rotary movement of the axis. The feed axis is therefore coupled to the press axis by means of a cam disk technology. The gripper release and closing motion is controlled by means of a cam switch function (close at 300 °, release at 200 °).

Technology Objects

You configure the Technology objects in S7T Config. Technology objects used in this example:

Technology Objects				
• Positioning axis (press axis) • Synchronization axis (feed axis) • Cam disk (to couple the feed and press axes) • Output cam (gripper control)				

Technology functions

The technology functions required to solve this task are executed by calling the function blocks listed below in the user program:

	Technology function	Task/description
1.	MC_Power	Enable feed axis
2.	MC_Power	Enable press axis
3.	MC_Home	Homing of the press axis at angular position 0 °
4.	MC_Home	Homing of the feed axis in home position
5.	MC_CamIn	The feed and pressure axes are coupled by means of a cam disk
6.	MC_CamSwitch	Setup of a position-based cam which controls gripper clamping
7.	MC_MoveVelocity	Move the press axis at preset speed
		Active production process
		Press shut-down:
a.	MC_MoveAbsolute	Move press axis to angular position 0 °
b.	MC_Halt	"Normal stop" feed axis
c.	MC_Halt	"Normal stop" press axis

9.2 Questions, tips and tricks

9.2.1 List - Questions, Tips and Tricks

9.2.2 How can I utilize the new performance features?

Question:

How can I utilize the performance features of S7-Technology described in "What's new in S7-Technology"?

Answer:

In order to utilize the new performance features of S7-Technology you must install the firmware versions listed in the diagram. Start by checking the firmware versions of integrated technology and of the CPU.

	S7-Technology V1.0	S7-Technology V2.0	S7-Technology V2.0 + SP1	S7-Technology V3.0	S7-Technology V3.0 + SP1	S7-Technology V3.0 + SP2
CPU 31xT-2 DP						
FW-CPU	V2.1	V2.3	V2.3	V2.4	V2.4	V2.4
FW-Technology	V3.0	V3.1	V3.1	V3.2	V3.2	V3.2
S7T Config	V3.0	V3.1	V3.2	V3.2	V3.2	V4.0
Microbox T						
WinLC T					V4.2	V4.2
FW-Technology					V3.2	V3.2
S7T Config					V3.2	V4.0

Perform the upgrade steps as described below.

Upgrade to a new version of S7-Technology

The upgrade described below applies to all upgrades of S7-Technology V3.x, V4.x, etc.

1. Upgrading CPU firmware (Page 896)
2. Upgrading technology firmware (Page 897)
3. Converting the technology and upgrading the technology packages (Page 901)
4. Recreating technology DBs (Page 905)
5. Updating technology functions (Page 907)
6. Download to the PLC (Page 908)

Note

The memory requirements for the project data may increase due to the upgrade. Keep a larger Micro Memory Card (MMC) at hand if only small memory reserves are available on your current MMC.

Upgrade to a Service Pack of S7-Technology

The description below covers the upgrade from a version of S7-Technology to the corresponding Service Pack (for example, upgrade from V3.0 to V3.0 + SP2).

1. Converting the technology (Page 899)
2. Updating technology functions (Page 907)
3. Download to the PLC (Page 908)

Note

Create a backup copy of your project data before you run the upgrade.

Do not open any additional dialog windows such as the expert list when you perform the upgrades.

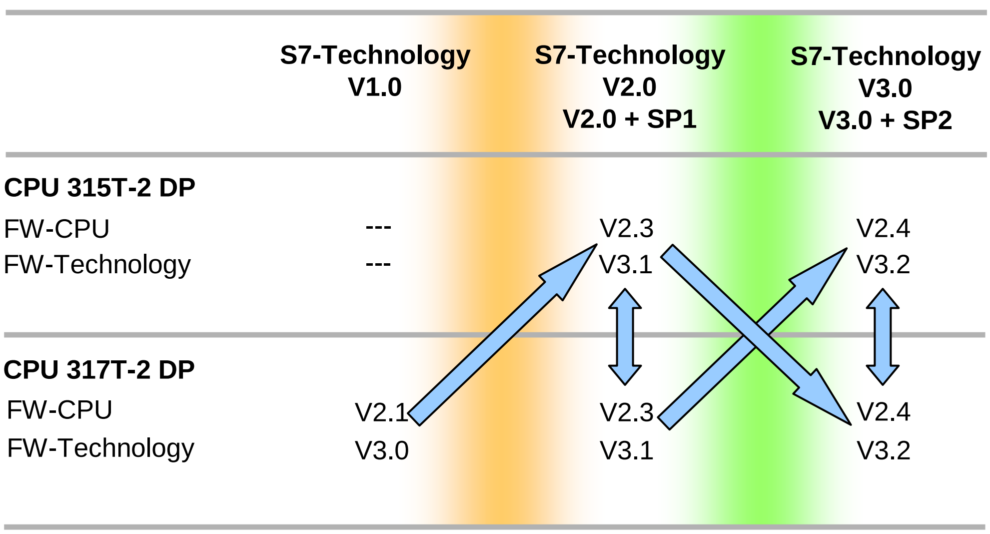
9.2.3 How do I replace the Technology CPU with a different type?

Question:

Can I replace the Technology CPU with a different type and reuse its configuration data?

Answer:

The Technology CPUs listed below support interchanging and reuse of their configuration data:



Certain versions may be omitted if you replace the Technology CPU.

Note

Perform the following steps before you replace the Technology CPU if you saved the current project in SIMATIC Manager using the **File > Save as** command and set the "With reorganization (slow)" check box.

1. Start S7T Config.
2. Follow any information and instructions shown. Close S7T Config afterwards.

Replacing the Technology CPU with a CPU and technology firmware of the same version

1. Replace the technology CPU in HW Config (as described in the chapter "How to upgrade the firmware of the integrated technology")
2. Select **Station > Save**
3. Close HW Config.
4. In SIMATIC manager, double-click "Technological objects" Open S7T Config by selecting the **Options > Configure technology** command.
5. Select the **Project > Save and compile all** menu command.
6. Close S7T Config.
7. Download the project data to the PLC (as described in the chapter "What to download to the PLC after the upgrade").

Replacing the Technology CPU with a CPU and technology with different firmware version

To replace the Technology CPU, proceed as with the upgrade. For detailed information, refer to the chapter "How to utilize the new performance features".

Note

Make allowances for any different data volumes when you replace the Technology CPU.

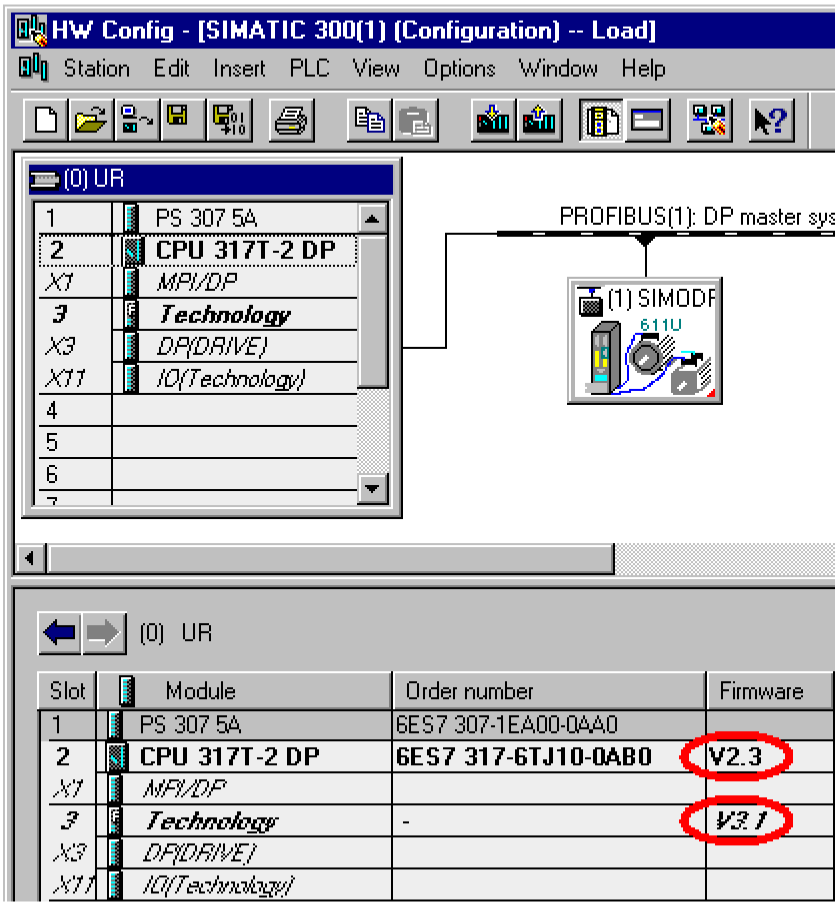
9.2.4 How can I identify the firmware versions?

Question:

The documentation frequently refers to different firmware versions. How do I identify the firmware version of the Technology CPU or Technology used in my system?

Answer:

Open HW Config, and then mark the top station window (represented below by an example of a 317T-2 DP CPU). The station window at the bottom shows you the firmware version of the CPU and of the Technology.



Note

Open HW Config in Online mode if you are uncertain of consistency between offline and online data.

9.2.5 How do I upgrade the firmware of CPU 31xT-2 DP?

Question:

The new performance features offered in S7-Technology may require an upgrade of the firmware of CPU 31xT-2 DP. How can I upgrade the firmware of CPU 31xT-2 DP?

Answer:

Check the relevant firmware versions before you upgrade the CPU firmware:

- You identify the firmware version of CPU 31xT-2 DP as described in the chapter "Identifying firmware versions".
- The firmware version required for the upgrade is specified in the chapter "How can I utilize the new performance features?".

The firmware upgrade is carried out from Micro Memory Card. Components required:

- A programming device / PC with STEP 7 Basis and Micro Memory Card adapter.
- A Micro Memory Card with a minimum capacity of 8 MB.

You can download the current firmware update from our Internet Service & Support pages:

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

Select the **PLC > Update Operating System** command in SIMATIC Manager. For details on procedures, refer to the STEP 7 Online Help.

As of Bootblock version A030705 of CPU 31xT, the firmware update is also possible without Micro Memory Card online (via networks). You recognize the Bootblock version in the diagnostics buffer of the CPU 31xT.

Note

Note the differences in the hardware versions of CPU 31xT-2 DP.

The hardware versions differ, for example, in terms of memory load of the integrated technology. For further information, refer to relevant the "CPU Data" manual, Chapter 6.6 "Memory in the integrated technology of the CPU".

9.2.6 How do I upgrade the firmware of integrated technology?

Question:

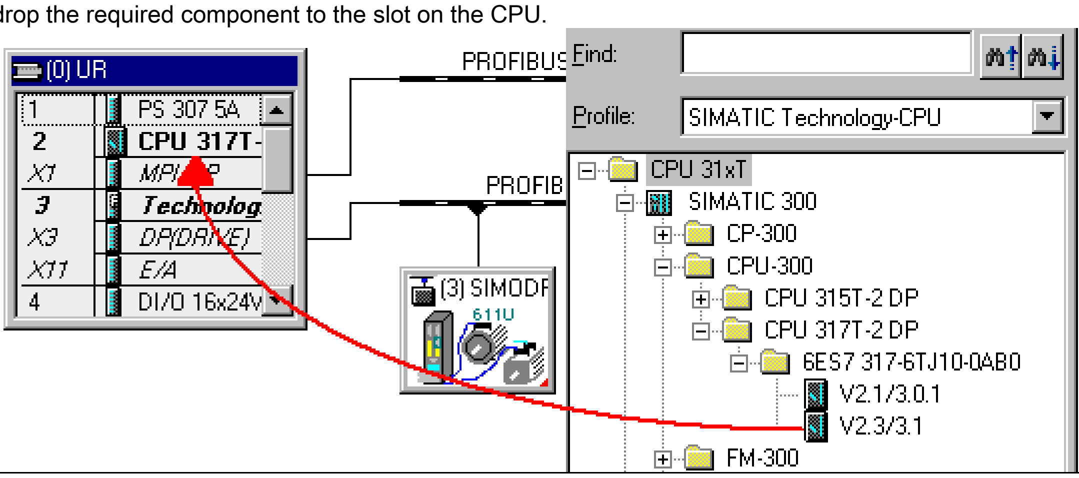
You may need to upgrade the firmware of integrated technology in order to be able to use the new performance features of S7-Technology. How can I upgrade the firmware of integrated technology?

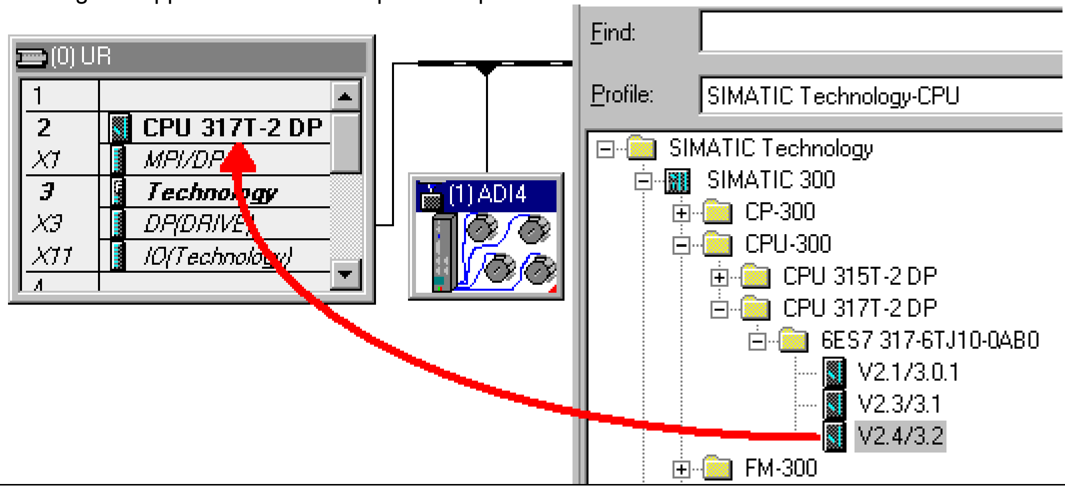
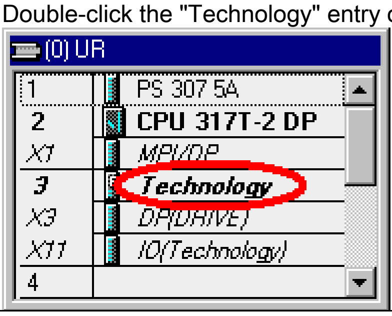
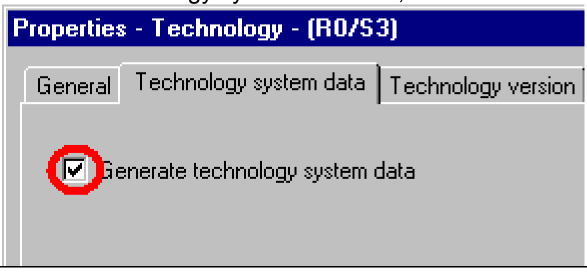
Answer:

Check the relevant firmware versions before you upgrade the firmware of integrated technology:

- You can identify the firmware version of the integrated technology as described in the chapter "How can I identify the firmware versions?".
- The firmware version required for the upgrade is specified in the chapter "How can I utilize the new performance features?".

Proceed as follows to upgrade the firmware: The diagrams below show the upgrade of a CPU 317T-2 DP. The representation in HW Config may differ slightly according to the Technology CPU used:

1.	Open HW Config.
2.	From the hardware catalog, select the "SIMATIC Technology CPU" profile.
3.	Open the folder which contains the Technology CPU or firmware version as shown in the diagram below. Drag-and-drop the required component to the slot on the CPU. 

4.	A dialog box appears. Confirm component replacement with "Yes". 
5.	Double-click the "Technology" entry of the technology CPU. 
6.	The "Properties - Technology" dialog box opens. On the "Technology system data" tab, set the "Generate technology system data" check box. 
7.	Click "OK" to close the "Properties - Technology" dialog box.
8.	Select **Station > Save**
9.	Close HW Config.

9.2.7 How do I convert the technology?

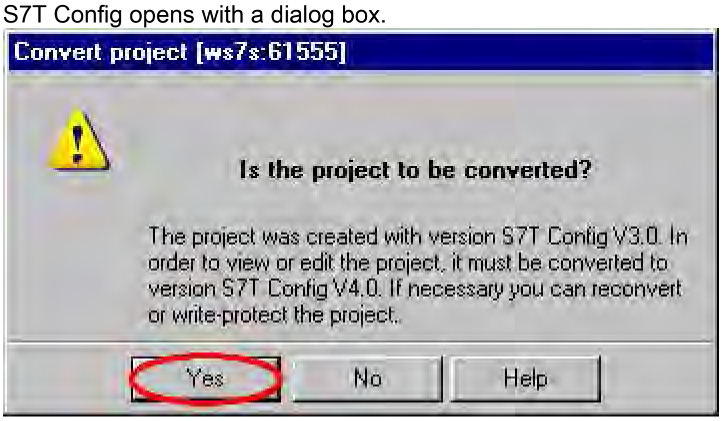
Question:

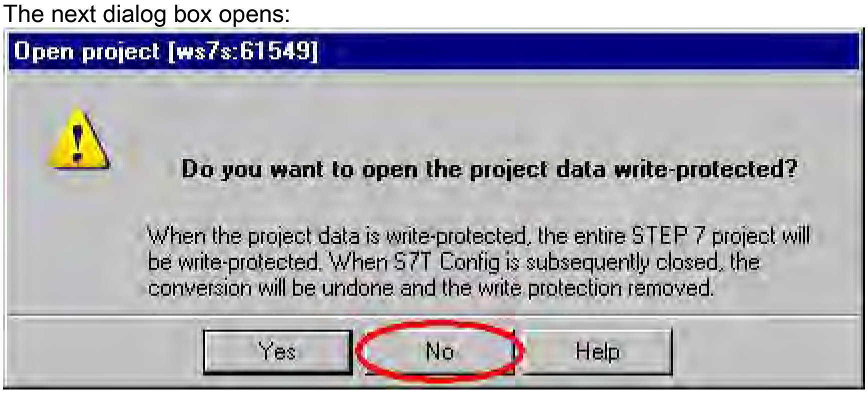
A new version of S7-Technology may require conversion of the technology. How can I convert the technology?

Answer:

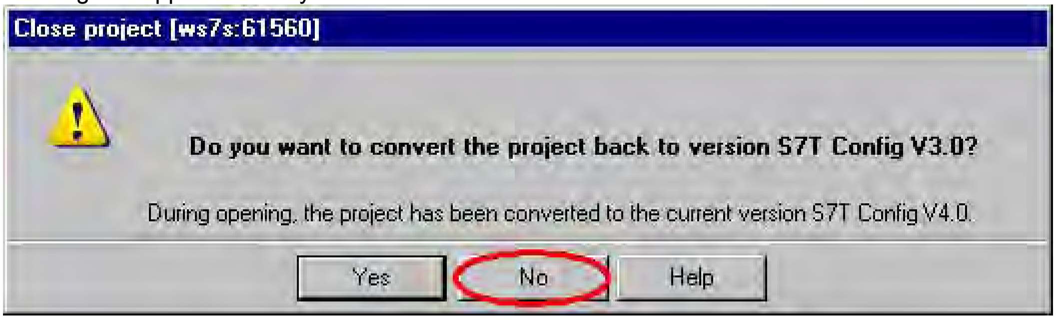
Configurations which may require a conversion of the technology are specified in the chapter "How can I utilize the new performance features".

Convert the technology as described below:
(Depending on the conversion, the text content and the version designations can vary in the illustrations)

<table>
<tr><td>1.</td><td>Select "Technological objects" in SIMATIC Manager and then select Options > Configure technology</td></tr>
<tr><td>2.</td><td>S7T Config opens with a dialog box.


Confirm this dialog box with "Yes". The project is now converted.</td></tr>
<tr><td>3.</td><td>The next dialog box opens:


Confirm the question with "No".</td></tr>
<tr><td>4.</td><td>Select Project > Save and compile all</td></tr>
<tr><td>5.</td><td>Close S7T Config.</td></tr>
</table>

6.	A dialog box appears when you close it.  Confirm it with "No".

9.2.8 How do I convert the technology and upgrade the technology packages?

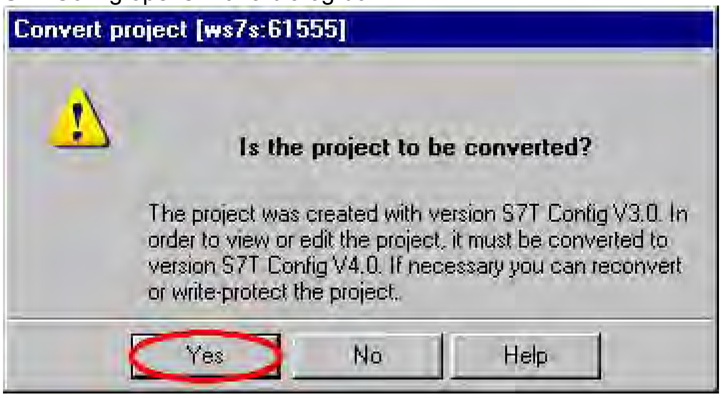
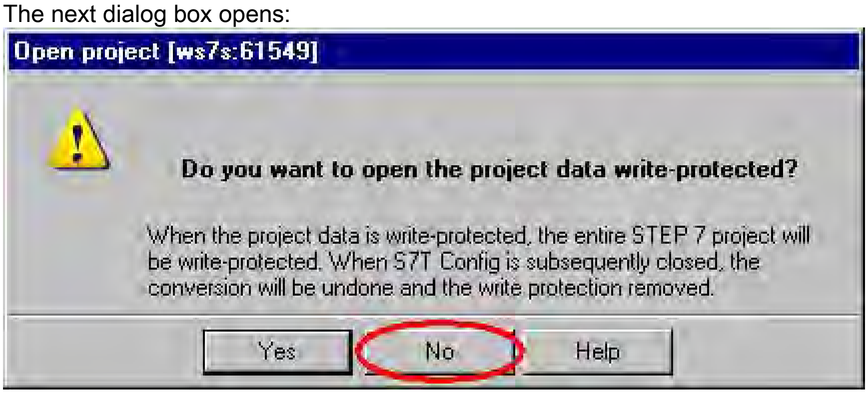
Question:

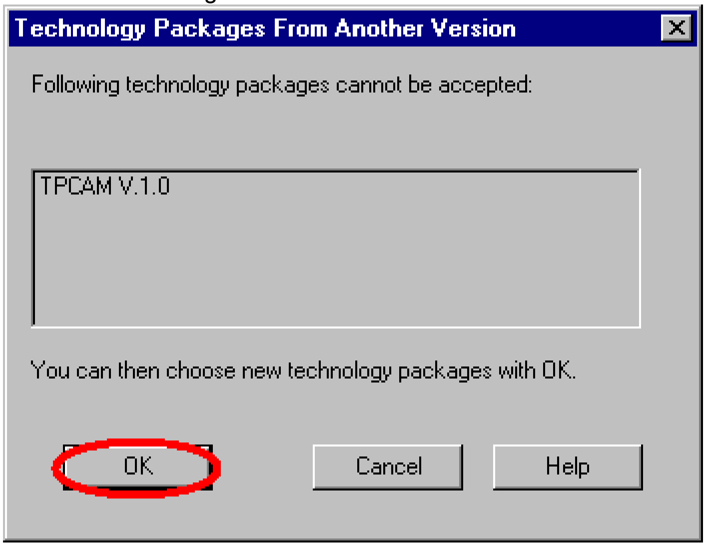
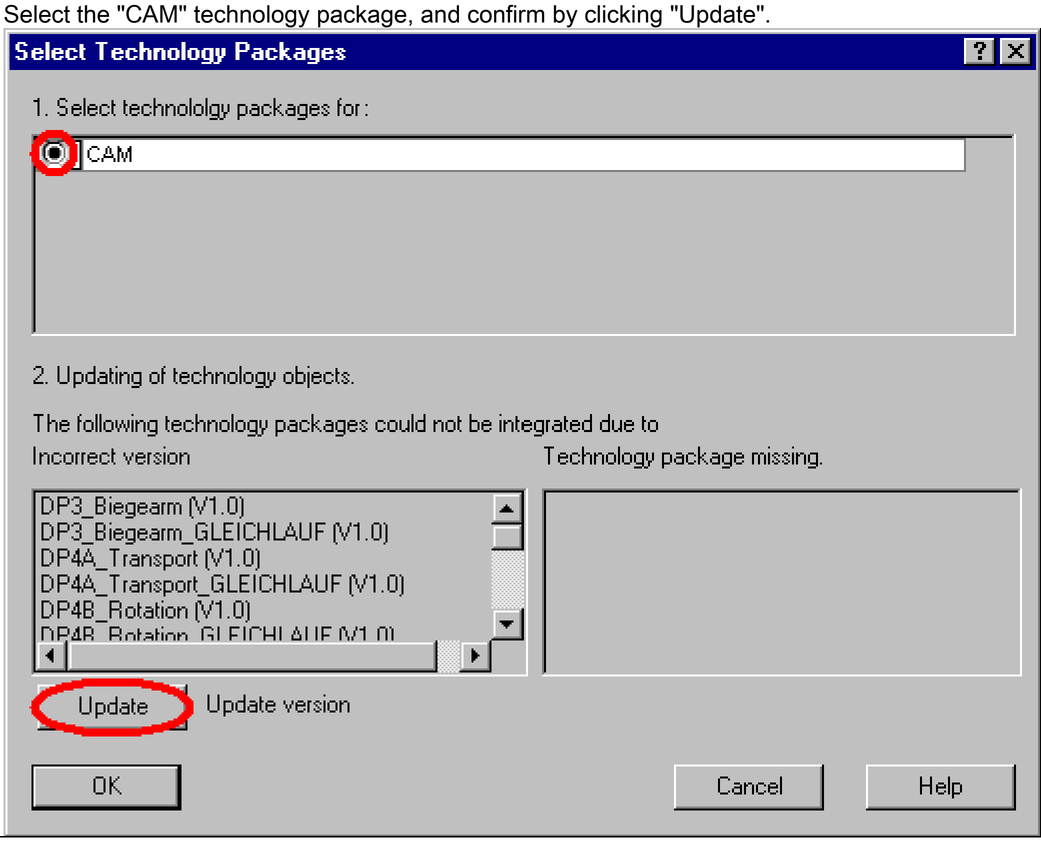
You may need to convert the technology and upgrade the technology packages when you implement a new version of S7-Technology. How can I do this?

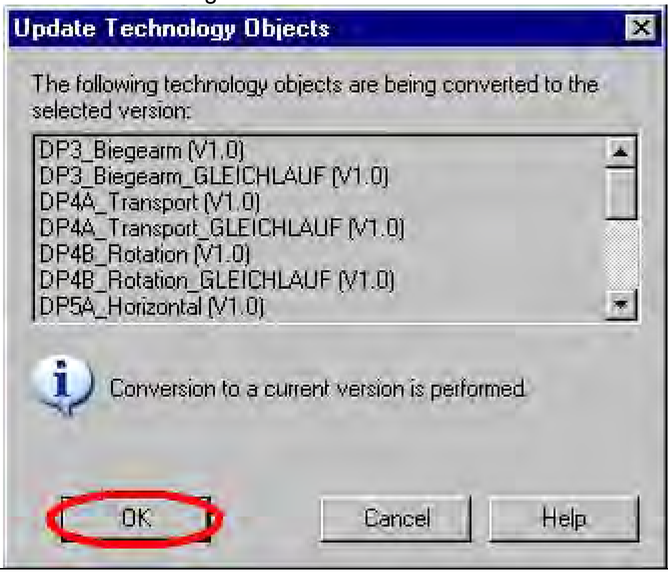
Answer:

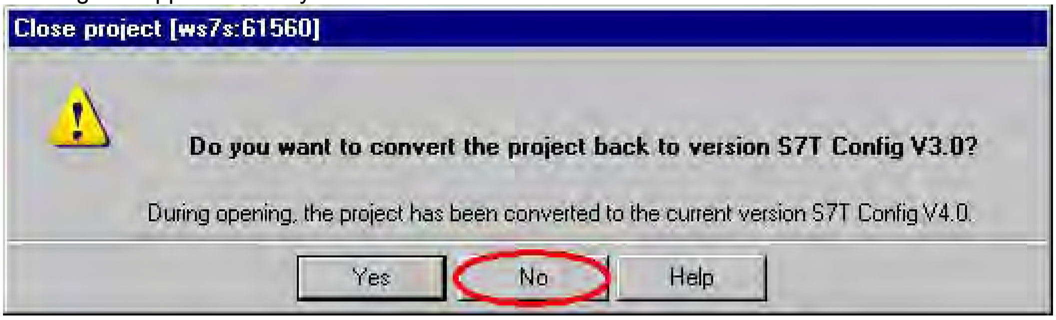
The chapter "How to utilize the new performance features" specifies configurations which may require these tasks.

Procedure:
(Depending on the conversion, the text content and the version designations can vary in the illustrations)

1.	Select "Technological objects" in SIMATIC Manager and then select **Options > Configure technology**.
2.	S7T Config opens with a dialog box.  Confirm this dialog box with "Yes". The project is now converted.
3.	The next dialog box opens:  Confirm the question with "No". The project is automatically saved and compiled in the next step. Certain warning and error messages may be generated because of the incomplete upgrade operation. Ignore such warnings and errors and confirm the message box with "OK".

4.	Select the technology from the Navigator and then select the **Edit > Select technology packages** command.
5.	Confirm the message box with "OK".  The "Select technology packages" dialog box opens.
6.	Select the "CAM" technology package, and confirm by clicking "Update". 

7.	Confirm the message box with "OK". 
8.	Confirm the next message with "Yes". The system updates the technology packages.
	Notes for converting older projects The conversion changes the units at some of the system variables. For this reason, the detail view of S7T Config outputs warnings during conversion, indicating that certain system variables (*userdefaultqfaxis.maxderivative.foutput*, for example) could not be set. You can ignore the warnings since these system variables are not used in the technology CPU.
9.	Confirm the "Update of technology objects completed" message with "OK".
10.	Close the "Select technology packages" dialog box with "OK".
11.	Select **Project > Save and compile all**
11a.	Continue with step 12 if no error messages appear. Continue with step 11b if error messages appear.
11b.	Select the **Project > Check consistency** command to locate the errors.
	Notes for converting older projects Possible causes of the error messages: • At the axes and external encoders, configuration data element *TypeOfAxis.NumberOfEncoders.Encoder1.DriverInfo. EncoderNumberOnDevice* does not have a value of *0*. Exception: An external encoder returns an additional actual value for the axis. A prerequisite is the use of standard message frames *4* or *6*, or of message frames *103, 104*, or *106*. Remedy: Set the configuration data element to *0*. • At the position-controlled axes without encoders, configuration data element *TypeOfAxis.NumberOfDataSets.DataSet_1.EncoderNumber.EncoderNumber* does not have a value of *0*. Remedy: Set the configuration data element to *0*.
11c.	Eliminate the causes of error, and continue at step 11. You can also save the project by selecting the **Project > Save** command and continue at step 12 if you want to eliminate the causes of error at a later time.

12.	Close S7T Config.
13.	A dialog box appears when you close it.  Confirm it with "No".

9.2.9 How do I recreate the technology DBs?

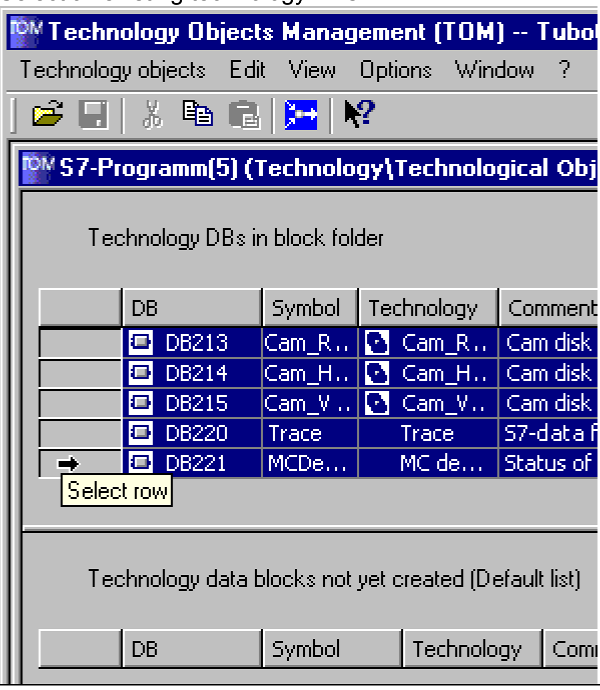
Question:

It may be necessary to recreate the technology DBs after you implemented a new version of S7-Technology. How can I recreate the technology DBs?

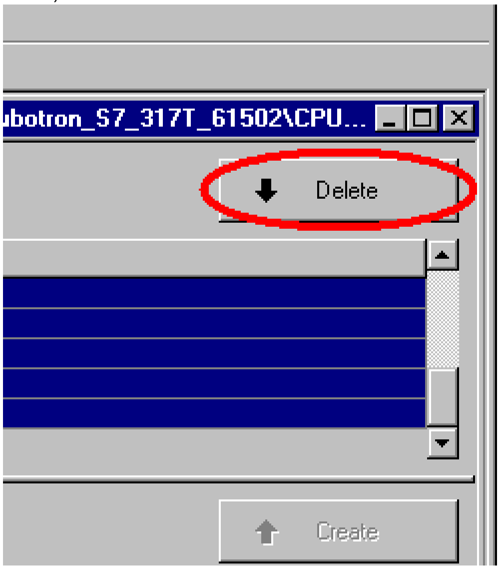
Answer:

The configurations which may require the recreation of technology DBs are defined in the chapter "How to utilize the new performance features".

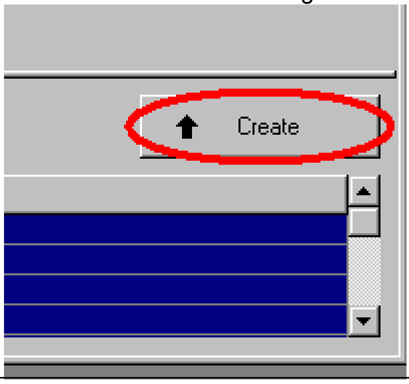
Proceed as follows to recreate the technology DBs:

1.	Run "Technology Objects Management"
2.	Select all existing technology DBs 

3. Next, click "Delete"



After this deletion, all technology DBs are marked in the area of the technology DBs which are not generated yet.

4.	Click "Generate" in order to generate the technology DBs with the new structure. 
5.	Close "Technology Objects Management"

9.2.10 How do I update the technology functions after an upgrade?

Question:

How do I update the technology functions after an upgrade?

Answer:

To use the features of the upgraded version, the technology functions have to be updated in the project's block folder. Procedure:

1.	Select the "Blocks" object in the block folder of the SIMATIC Manager and select the shortcut menu command **Check block consistency**.
2.	Clean up any displayed errors.
3.	Select the menu command **File > Open** in the SIMATIC Manager and select the "Libraries" tab. Open the "S7-Tech" library and the directory of the corresponding firmware version.
4.	Update all technology functions of your block folder with the corresponding technology functions of the library folder.
5.	Select the "Blocks" object in the block folder of the SIMATIC Manager and again select the shortcut menu command **Check block consistency**.
6.	Clean up any displayed errors.

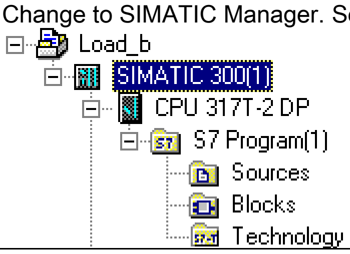
9.2.11 What do I have to download to the PLC after an upgrade?

Question:

Which data do I have to download to the PLC after an upgrade?

Answer:

Download the user program, the HW configuration, the technology and the firmware of integrated technology to the Technology CPU after the upgrade.
Procedure:

1.	Change to SIMATIC Manager. Select the upgraded SIMATIC station in your project. 
2.	Select **PLC > Download** The technology CPU automatically performs a memory reset after the download [1]. The new performance features are available after you completed the download.

[1] If the integrated technology firmware version is identical in load memory of the technology CPU, the technology CPU does not automatically perform a memory reset after the download.

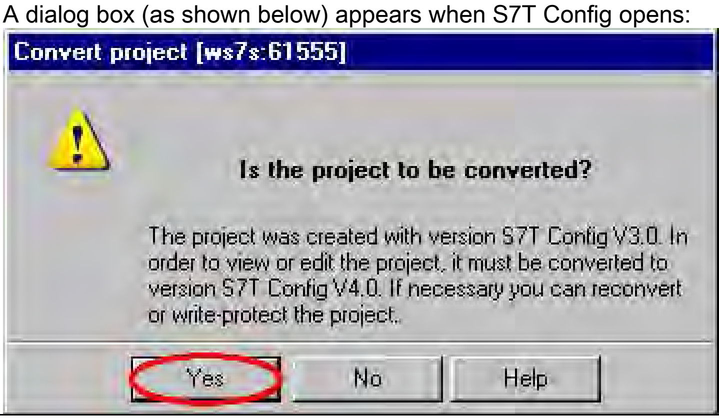
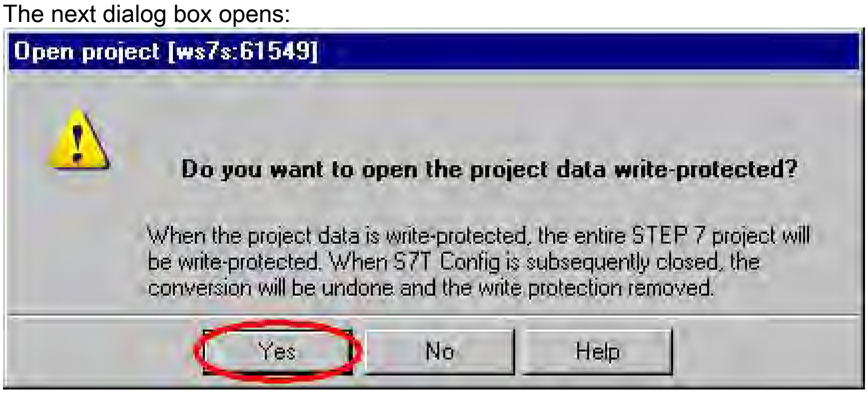
9.2.12 How do I analyze a project from a previous version using S7-Technology?

Question:

The current project was created in a previous version of S7-Technology. Can I analyze a project from a previous version using the current S7-Technology?

Answer:

Yes. You can analyze a project from a previous version using your current S7-Technology. Write protect your project data. Procedure:

<table>
<tr><td>1.</td><td>Select "Technological objects" and the Options > Configure technology command in SIMATIC Manager.</td></tr>
<tr><td>2.</td><td>A dialog box (as shown below) appears when S7T Config opens:

</td></tr>
<tr><td>3.</td><td>Convert the project in order to be able to view it in S7T Config. Confirm the question with "Yes".</td></tr>
<tr><td>4.</td><td>The next dialog box opens:


Confirm the question with "Yes".
The entire STEP 7 project will inherit the write protection setting. The conversion is undone and the write protection is cleared if you close S7T Config at a later time.</td></tr>
<tr><td>5.</td><td>Carry out the required analysis.</td></tr>
<tr><td>6.</td><td>Close S7T Config.</td></tr>
</table>

Note

Meaning of the Navigator icons during online analysis:

- Objects with green icons are consistent
- Objects with red/green icons have an indefinite state, that is, their "consistent" or "inconsistent" state can not be determined.

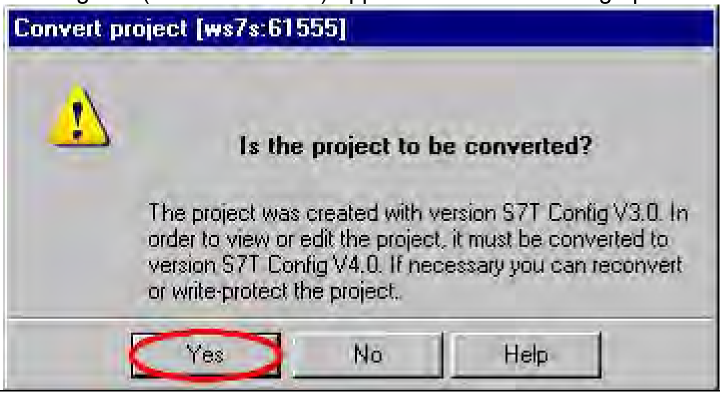
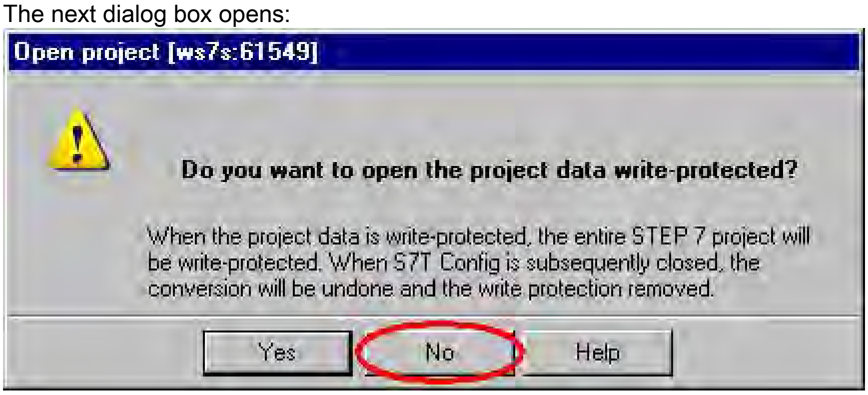
9.2.13 How do I change a project from a previous version using S7-Technology?

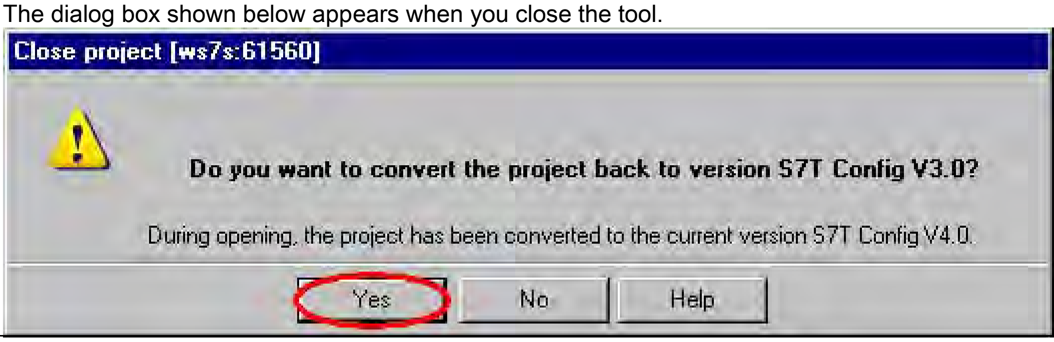
Question:

The current project was created in a previous version of S7-Technology. Can I change the contents of the project using the current S7-Technology and maintain down compatibility with the original version?

Answer:

Yes, you can change the project so that it retains its compatibility with the original version. Procedure:

1.	Select "Technological objects" and the **Options > Configure technology** command in SIMATIC Manager.
2.	A dialog box (as shown below) appears when S7T Config opens: 
3.	You need to convert the project to edit it in S7T Config. Confirm the question with "Yes".
4.	The next dialog box opens:  Confirm the question with "No".
5.	Make your relevant changes.
6.	Select **Project > Save and compile all**
6a.	If you have not recreated or deleted any technology objects, continue at step 7. If you have re-inserted or deleted technology objects, open "Technology Objects Management" (by double-clicking "Technological objects" in SIMATIC Manager) .
6b.	Create or delete the corresponding technology DBs as described in "Creating and managing technology DBs" .
6c.	Close "Technology Objects Management".
7.	Close S7T Config.

8.	The dialog box shown below appears when you close the tool. 
9.	Confirm the question with "Yes" in order to reconvert the project to its original version.

Note

The offline/online comparison of converted or reconverted projects may return inconsistency of identical same project data.

Reason: Internal data structure adjustments in later versions.

Remedy: Carry out the following steps to correct the displayed inconsistencies:

1. "Save and compile all" in S7T Config
2. Download the project data to the Technology CPU

Carry out these steps in the relevant version of S7-Technology.

If you forgot to reconvert the project before you closed S7T Config you can catch up with requirements by calling the **Project > Save in old project format** command in S7T Config.

9.2.14 What do I have to observe when using "Save as" with reorganization?

Open S7T Config if the following conditions are given:

- After module replacement
- After a station was copied
- After having executed the **File > Save as** command with the "With reorganization (slow)" check box checked

Follow the instructions on your screen and then close S7T Config.

If you created, saved and reorganized a project created in a previous version of S7-Technology, the system may output a message reporting that certain technology objects could not be copied.

You can ignore this message, because the technology objects were in actual fact copied.

9.2.15 Why do certain system DBs have a different online / offline time stamp?

Question:

Certain SDBs in the project have a different time stamp compared to the SDBs in the technology CPU.

Does this time stamp difference pose a problem?

Answer:

System data blocks type *3311* are assigned the time stamp of the integrated real-time clock of the CPU if you download the technology to the Technology CPU using the **Target system> Download > Technology to target system** command in S7T Config. A different time stamp as such does not pose a problem at the moment.

System data blocks type *3311* of the Technology CPU are assigned the time stamp of the integrated real-time clock of the CPU at the time of their download. Online / offline SDBs with identical contents may be assigned different time stamps. If all drives are marked in green color in the online view of S7T Config you can presume that your technology configuration is consistent.

Note

If "Automatic summertime/wintertime" is set in Windows the time stamp of the system data displays an additional hour during summertime.

9.2.16 STEP 7 reports "Insufficient memory space"

Question:

During the download of blocks to the Technology CPU, STEP 7 returns the message "Insufficient memory space on the module. Do you want to compress load memory?" The message persists after you compressed the data and sufficient work memory is available.

Answer:

The cause of this message may be:

- The Technology CPU contains "orphan" technology DBs which are not listed in your project. You can identify technology DBs based on the creation language "TechObj".

Check the configuration by selecting the **View > Details** command in in SIMATIC Manager. Delete all unused technology DBs from the Technology CPU.

9.2.17 The CPU goes into STOP sporadically as a result of timeout

Question:

Although you have tuned the system clocks, the CPU sporadically goes into STOP when operating in continuous mode due to timeout of the integrated technology.

Answer:

System load at the integrated technology fluctuates based on the command rate and possibly occurring errors. If the selected system clocks are too short a timeout may be triggered at the integrated technology which causes a CPU STOP.

If this happens, extend the system clocks so that the system no longer reacts with timeout errors over a longer period of time.

9.2.18 Errors occur when the "Save and compile all" function is executed for the technology data

Question:

Errors occur when the "Save and compile all" function is executed for the technology data in S7T Config However, I have only activated the default parameter values?

Answer:

For reasons of safety, all safety- and function-relevant parameters or addresses are assigned default values in S7T Config in order to prevent execution of the "Save and compile all" function.

This is why you should always exercise due care when setting safety- and function-relevant parameters, and verify their values and logical context.

9.3 Expert list in S7T Config

9.3.1 Using the Expert List

Experience in using the Expert List in S7T Config is prerequisite for handling the settings described in the chapters below. The expert list in S7T Config provides read / write access to all configuration data and system variables of a technology object. The list includes data which cannot be set in the wizards or in the programming windows.

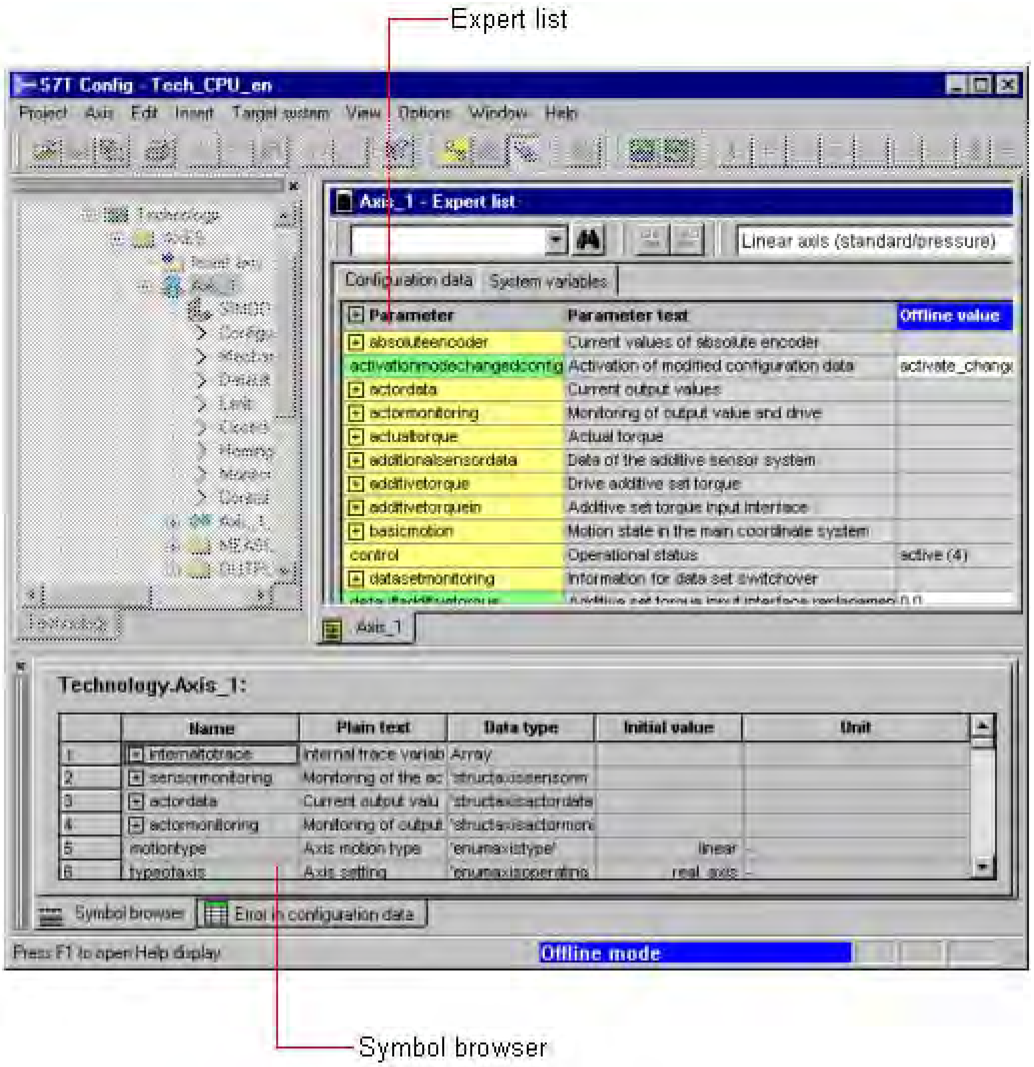
Separate expert lists are available for each configured technology object.

⚠ CAUTION
The expert mode is intended for experienced users. Incorrect input may result in uncontrolled response of the drive! The system does not check the plausibility of your entries! Incorrect changes may lead to inconsistency of the technology object.

Calling the Expert List:

1. Select a technology object (axis, for example) from the Navigator to view or edit its configuration data or system variables.
2. Select the **Expert > Expert list** command from the shortcut menu.

The Expert List opens in the working area.



The left pane of the Expert List shows the tree structure of the system variables and of configuration data. The right pane outputs a tabular view of the system variables and configuration data.

Select an element from the tree structure to view the corresponding parameter values in this table.

Values which require a restart, and all subsequently changed configuration data, are represented in brown color until you have performed the restart.

Click the value you want to edit. Type in a new value or select a symbolic name (grayed out values can not be edited.) Save your changes with ENTER. Set the "Collect changes" check box to change a group of values. Make your changes, and then confirm your entries by clicking "Activate." All your changes will be saved and accepted.

Effectiveness of changes in Online mode

- Modified system variables are applied immediately.
- Changes in configuration data are applied according to the setting at the "Effectiveness" column.
  - "Immediately"

    Values are activated when you press ENTER or click "Activate".
  - "restart"

    Values are only activated after you restart the technology object by calling the "MC_Reset" technology function.
  - "download"

    Values cannot be edited in online mode. Change to Offline mode, edit the technology, and then download your changes to the target system.
- Changes to system variables and configuration data are retained when the Technology CPU changes from STOP to RUN.
- Changes to system variables and configuration data are non-retentive and are lost when power is cycled OFF and ON.

Select the technology from the Navigator, and then select the **Target device > Copy current data to ROM** in order to save the changes of configuration data to non-volatile memory.

In order to save the values permanently to your project, make your changes in Offline mode, save and compile the technology data, and then download these to the target device. You can also upload the changes to configuration data to the project data of the technology by executing the **Target system > Upload > Configuration data to PG** command. Next, save and compile the technology data by calling the menu command **Project > Save and Compile All**

Effectiveness of changes in Offline mode

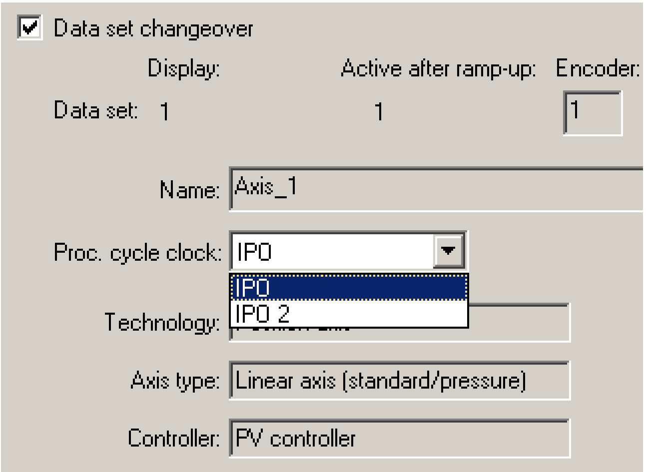
- Changes in Offline mode changes only affect the technology project data. In order to activate the changes in the Technology CPU, save and compile the technology data by calling the **Project > Save and Compile All** command and then download these data to the target system (menu command **Target system > Download > Technology Data to Target System**).

9.3.2 Axis settings

9.3.2.1 Assigning interpolator cycle 2

Axes, external encoders, output cams and measuring inputs are processed by default within the interpolator cycle. Technology objects of a lower priority class can also be assigned interpolator cycle 2.

Select the the required interpolator cycle from the "Execution cycle" drop-down list box in the **Axis > Configuration** dialog box.



The synchronization axis and the corresponding synchronization object must be operated within the same processing cycle. Any change of the processing cycle of a synchronization axis also has to be applied to the synchronization object:

1. In Navigator of S7T Config, select the synchronization object of the corresponding synchronization axis.
2. Select the **Expert > Expert list** command from the shortcut menu.
3. You can change the processing cycle clock in the *Execution.executionlevel* configuration data element.

Note

The interpolator cycle may not be assigned online!

"Save and compile all" technology data, and then download system data to the technology CPU in SIMATIC Manager.

9.3.2.2 Torque settings

You can reduce torque of the drive motor by setting the "MC_SetTorqueLimit" and "MC_MoveToEndPos" technology functions (not for following error detection). The torque is defined in N/m. To do so, you must set the reference torque for the motor in configuration data element "*TypeOfAxis.SetPointDriverInfo.DriveData.maxTorque*".

Percentile reduction

In order to override the absolute value with a reduction percentage, set a reference torque of 100.0 N/m:

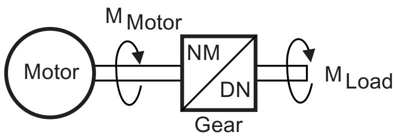
Configuration data element: *TypeOfAxis.SetPointDriverInfo.DriveData.maxTorque = 100.0* Nm

You reduce the maximum torque of the motor to 25% by setting parameter *MaxTorque = 25.0* at the call of "MC_SetTorqueLimit."

Calculating torque on the load side

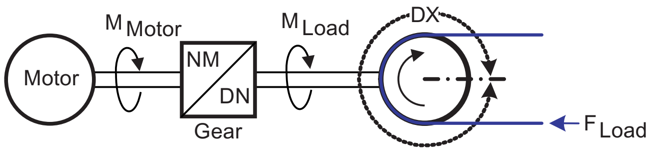
You can calculate the motor torque on load side, making allowances for gear load and leadscrew pitches. Frictional losses will be ignored in the following section.

Rotary axes:



$$M_{Load} = M_{Motor} \cdot \frac{NM}{DN}$$

Linear axes (here, the force is calculated):



$$F_{Load} = M_{Motor} \cdot \frac{NM}{DN} \cdot \frac{2\pi}{DX}$$

M_{Motor}	Drive motor torque in [N/m]
M_{Load}	Torque at the load side of the gear in [N/m]
F_{Load}	Load force in [N]
NM	Number of motor revolutions
DN	Number of load revolutions
DX	Load feed per revolution (leadscrew pitch, for example) in [m]

9.3.3 Monitoring functions

9.3.3.1 Encoder monitoring functions

- Monitoring the zero mark of incremental encoders

  You can activate the function for monitoring the number of increments between two encoder zero marks. An alarm is triggered if the encoder zero mark is not reached within the defined distance.

  In homing mode with encoder zero mark the distance is monitored after the axis has passed the reference cam.

- Valid changes to the actual value of an absolute encoder

  The user can activate the function for monitoring valid changes to the actual value of an absolute encoder.

- Actual velocity

  The permissible maximum value of the actual velocity can be monitored. If the maximum value is exceeded, system variable *sensordata.sensormonitoring.velocity* is output as *limitexceeded*. The velocity is not limited to this value.

9.3.3.2 Encoder limit frequency monitoring

The system monitors the limit frequency of the encoders. It also monitors the actual speed /actual acceleration in order to detect errors in the control loop of the drives. An error is reported at the technology DB if the actual value overshoots the encoder limit frequency. The error is indicated at ErrorStatus "SensorFreqViolation" of the Technology DB.

9.4 Technology parameters

9.4.1 Technology parameters 0001 to 0999 - Axes

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter *ParameterNumber* of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Description	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum" data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Supported for	In the case of axis parameters specifies the type of axis for which the parameter applies.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
1	-	Name in DB-Param ("Normal" display mode)	Setpoints.Position.Target position
		Name in DB-Param ("Expert" display mode)	*positioningstate.commandposition*
		Description	Axis position setpoint
		Parameter type	system variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity.Software limit switch.Negative
		Name in DB-Param ("Expert" display mode)	*swlimit.minusposition*
		Description	Negative software limit switch
		Parameter type	system variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity.Software limit switch.Positive
		Name in DB-Param ("Expert" display mode)	*swlimit.plusposition*
		Description	Positive software limit switch
		Parameter type	system variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity.Software limit switch.Activation
		Name in DB-Param ("Expert" display mode)	*swlimit.state*
		Description	Activation of software limit switch
		Parameter type	system variables
		Data type	DINT EnumActiveInactive (Page 1040)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
6	1-16	Name in DB-Param ("Normal" display mode)	Monitoring.Following error monitoring.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicFollowing.enable*
		Description	Activation of dynamic following error monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
7	1-16	Name in DB-Param ("Normal" display mode)	Monitoring.Following error monitoring.Maximum
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicFollowing.maxPositionTolerance*
		Description	Upper threshold of the following error characteristic
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
8	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity.Velocity.Maximum
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.MaxVelocity.maximum*
		Description	Maximum permissible speed
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
10	-	Name in DB-Param ("Normal" display mode)	Actual values.Motion.Velocity
		Name in DB-Param ("Expert" display mode)	*motionstatedata.actualvelocity*
		Description	Actual axis velocity
		Parameter type	system variables
		Data type	REAL
		Min.	-1E12
		Max.	+1E12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
11	-	Name in DB-Param ("Normal" display mode)	Setpoints.Motion.Velocity
		Name in DB-Param ("Expert" display mode)	*motionstatedata.commandvelocity*
		Description	Axis velocity setpoint
		Parameter type	system variables
		Data type	REAL
		Min.	-1E12
		Max.	+1E12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
12	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Acceleration.Maximum
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.MaxAcceleration.maximum*
		Description	Maximum value of the permissible acceleration
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E12
		Max.	+1E12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
16	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Jerk.Maximum
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.MaxJerk.maximum*
		Description	Maximum value of the permissible jerk
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E12
		Max.	+1E12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

9.4.2 Technology parameters 1000 to 1999 - Axes

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter *ParameterNumber* of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Description	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum..." data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.

Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Supported for	In the case of axis parameters specifies the type of axis for which the parameter applies.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
1100	-	Name in DB-Param ("Normal" display mode)	Mechanics.Spindle settings.Spindle pitch for each revolution of the axis
		Name in DB-Param ("Expert" display mode)	*LeadScrew.pitchVal*
		Description	Leadscrew pitch for each revolution of the axis
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	+1E12
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1110	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Extrapolation time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationTime*
		Description	Extrapolation time of the master axis
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	+1E12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1111	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Filter.Mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.Filter.Mode*
		Description	Selection criterion for actual value smoothing
		Parameter type	Configuration data element
		Data type	DINT EnumAxisFilterMode (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1112	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Filter.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.Filter.Enable*
		Description	Activation status
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1113	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Filter.Time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.Filter.TimeConstant*
		Description	Time constant for PT1 smoothing and averaging
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1114	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Tolerance range.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ToleranceRange.Enable*
		Description	Activation of the tolerance range
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1115	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Tolerance range.Tolerance window
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ToleranceRange.Value*
		Description	Tolerance window
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1116	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Velocity signal
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis Extrapolation.extrapolatedVelocitySwitch*
		Description	Selection of the velocity master value
		Parameter type	Configuration data element
		Data type	DINT EnumAxisExtrapolatedVelocitySwitch (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1120	-	Name in DB-Param ("Normal" display mode)	Homing.Distance to the bero
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Homing.beroDistance*
		Description	Maximum distance to the homing output cam / external zero mark
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1121	-	Name in DB-Param ("Normal" display mode)	Homing.Approach direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Homing.direction*
		Description	Homing direction
		Parameter type	Configuration data element
		Data type	DINT EnumDirectionType (Page 1048)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1122	-	Name in DB-Param ("Normal" display mode)	Homing.Bero monitoring activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Homing.enableBeroDistance*
		Description	Status of monitoring mode activation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1123	-	Name in DB-Param ("Normal" display mode)	Homing.Homing required
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Homing.referencingNecessary*
		Description	Homing required
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1130	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Position filter.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.enable*
		Description	Activation of filter
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Synchronized axis
		Firmware version	
1131	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Position filter.T1
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T1*
		Description	First time constant
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1132	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Position filter.T2
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T2*
		Description	Second time constant
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1201	-	Name in DB-Param ("Normal" display mode)	Mechanics.Modulo.Activation
		Name in DB-Param ("Expert" display mode)	*Modulo.state*
		Description	Activation of the modulo settings
		Parameter type	Configuration data element
		Data type	DINT EnumActiveInactive (Page 1040)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1202	-	Name in DB-Param ("Normal" display mode)	Mechanics.Modulo.Length
		Name in DB-Param ("Expert" display mode)	*Modulo.length*
		Description	Modulo length
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1203	-	Name in DB-Param ("Normal" display mode)	Mechanics.Modulo.Starting value
		Name in DB-Param ("Expert" display mode)	*Modulo.startValue*
		Description	Modulo starting value
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1211	-	Name in DB-Param ("Normal" display mode)	Monitoring.Acceleration.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.ActualAccelerationMonitoring.enable*
		Description	Activation of the actual acceleration monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1212	-	Name in DB-Param ("Normal" display mode)	Monitoring.Acceleration.Maximum value
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.ActualAccelerationMonitoring.maximum*
		Description	Maximal permissible actual value of acceleration
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1221	-	Name in DB-Param ("Normal" display mode)	Monitoring.Velocity.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.ActualVelocityMonitoring.enable*
		Description	Activation of the actual velocity monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1222	-	Name in DB-Param ("Normal" display mode)	Monitoring.Velocity.Maximum value
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.ActualVelocityMonitoring.maximum*
		Description	Maximum permissible actual velocity
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1231	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Drift compensation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Drift.enable*
		Description	Activation of drift compensation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1241	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity.Emergency stop delay
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.EmergencyRampGenerator.maxDeceleration*
		Description	Delay of the deceleration ramp
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1251	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Fine interpolator type
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.FineInterpolator._type*
		Description	Fine interpolator type
		Parameter type	Configuration data element
		Data type	DINT EnumAxisFineInterpolatorMode (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1261	-	Name in DB-Param ("Normal" display mode)	Position control.Friction compensation.Amplitude
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Friction.amplitude*
		Description	Amplitude of the friction compensation characteristic
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1262	-	Name in DB-Param ("Normal" display mode)	Position control.Friction compensation.Decay time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Friction.decayTime*
		Description	Decay time constant
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1263	-	Name in DB-Param ("Normal" display mode)	Position control.Friction compensation.Standstill decay time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Friction.delayTimeStandStill*
		Description	Delay time for standstill signal
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	2^31 * Servotime * 9
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1264	-	Name in DB-Param ("Normal" display mode)	Position control.Friction compensation.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Friction.enable*
		Description	Activation of friction compensation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1265	-	Name in DB-Param ("Normal" display mode)	Position control.Friction compensation.Standstill maximum velocity
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Friction.maxVeloStandStill*
		Description	Maximum value of standstill signal
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1271	-	Name in DB-Param ("Normal" display mode)	Monitoring.Synchronous operation monitoring.Actual value tolerance
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.GearingPosTolerance.actualValueTolerance*
		Description	Permissible actual value tolerance
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
1272	-	Name in DB-Param ("Normal" display mode)	Monitoring.Synchronous operation monitoring.Setpoint tolerance
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.GearingPosTolerance.commandValueTolerance*
		Description	Permissible setpoint tolerance
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
1273	-	Name in DB-Param ("Normal" display mode)	Monitoring.Synchronous operation monitoring.Activation of actual value monitoring
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.GearingPosTolerance.enableActualValue*
		Description	Monitoring actual value tolerance
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
1274	-	Name in DB-Param ("Normal" display mode)	Monitoring.Synchronous operation monitoring.Activation of setpoint monitoring
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.GearingPosTolerance.enableCommandValue*
		Description	Monitoring setpoint tolerance
		Parameter type	Configuration data element
		Data type	DINT EnumGearingPosToleranceCommandValue (Page 1050)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
1275	-	Name in DB-Param ("Normal" display mode)	Monitoring.Synchronous operation monitoring.Message to master
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.GearingPosTolerance.enableErrorReporting*
		Description	Error report of tolerance monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumErrorReporting (Page 1048)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
1281	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity. Hardware limit switch.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.HWEndPos.enable*
		Description	Activation of the hardware limit switch
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1284	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity. Hardware limit switch.Mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.HWEndPos.mode*
		Description	Position of the hardware limit switch relative to the axis traversing range.
		Parameter type	Configuration data element
		Data type	DINT EnumMountSwitch (Page 1053)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1291	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Acceleration.Activation of the monitoring
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.MaxAcceleration.enableSetPointMonitoring*
		Description	Activation of manipulated variable monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1301	-	Name in DB-Param ("Normal" display mode)	Configuration.Dead zone compensation.Center point
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NeutralBand.centreValue*
		Description	Center point of the dead band range
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1302	-	Name in DB-Param ("Normal" display mode)	Configuration.Dead zone compensation.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NeutralBand.enable*
		Description	Activation of dead band compensation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1303	-	Name in DB-Param ("Normal" display mode)	Configuration.Dead zone compensation.Expansion
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NeutralBand.range*
		Description	Expansion of the dead band range
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1311	-	Name in DB-Param ("Normal" display mode)	Monitoring.Positioning and standstill monitoring.Message delay time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.PositionMonitoring.posWinTolDelayTime*
		Description	Message activation delay
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1312	-	Name in DB-Param ("Normal" display mode)	Monitoring.Positioning and standstill monitoring.Tolerance window delay time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.PositionMonitoring.posWinTolTime*
		Description	Delay time to go until the tolerance window is reached
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1313	-	Name in DB-Param ("Normal" display mode)	Monitoring.Positioning and standstill monitoring.Tolerance
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.PositionMonitoring.tolerance*
		Description	Width of the positioning window
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1321	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Reversal of direction of rotation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SetPointDriverInfo.InvertSetPoint.invSetPoint*
		Description	Activation of the reversal of direction of rotation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1331	-	Name in DB-Param ("Normal" display mode)	Configuration.Actual value smoothing.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SmoothingFilter.enable*
		Description	Activation of the filter functionality
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1332	-	Name in DB-Param ("Normal" display mode)	Configuration.Actual value smoothing.Mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SmoothingFilter.mode*
		Description	Calculation method for actual value smoothing
		Parameter type	Configuration data element
		Data type	DINT EnumAxisFilterMode (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1333	-	Name in DB-Param ("Normal" display mode)	Configuration.Actual value smoothing.Time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SmoothingFilter.timeConstant*
		Description	Time constant for PT1 smoothing
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1341	-	Name in DB-Param ("Normal" display mode)	Monitoring.Positioning and standstill monitoring.Standstill message delay time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillMonitoring.delayTimeToActivate*
		Description	Message activation delay
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
1342	-	Name in DB-Param ("Normal" display mode)	Monitoring.Positioning and standstill monitoring.Position tolerance in standstill
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillMonitoring.stillStandTolerance*
		Description	Permissible tolerance of standstill position
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1351	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Message delay time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.delayTimeToActivate*
		Description	Delay time for the triggering of the standstill signal
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	214748
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1352	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Degree of filtering
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.filterDegree*
		Description	Degree of filtering
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	15
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1353	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Filter frequency
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.filterFrequency*
		Description	Filter frequency
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1354	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Velocity limit
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.maxVeloStandStill*
		Description	Velocity limit for standstill signal
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1400	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Manipulated variable limit.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SpeedLimitation.Enable*
		Description	Activation status of the speed range limit
		Parameter type	Configuration data element
		Data type	DINT EnumAxisFilterMode (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
1401	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Manipulated variable limit.Lower limit
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SpeedLimitation.MinSpeed*
		Description	Lower limit for the speed range
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
1402	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Manipulated variable limit.Upper limit
		Name in DB-Param ("Expert" display mode)	TypeOfAxis.SpeedLimitation.MaxSpeed
		Description	Upper limit for the speed range
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

9.4.3 Technology parameters 2000 to 2999 - Axes

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter *ParameterNumbe* of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Significance	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum" data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Supported for	In the case of axis parameters specifies the type of axis for which the parameter applies.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
2000	-	Name in DB-Param ("Normal" display mode)	Configuration.Data block switching.Data blocks count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfData Sets.numberOfDataSets*
		Significance	Number of data records
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	16
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
2001	-	Name in DB-Param ("Normal" display mode)	Configuration.Data block switching.Data block initialization
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.initDataSet*
		Significance	Initialization data record
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	16
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
2002	-	Name in DB-Param ("Normal" display mode)	Configuration.Data block switching.Data block switching mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfData Sets.changeMode*
		Significance	Mode of data record change
		Parameter type	Configuration data element
		Data type	DINT EnumChangeMode (Page 1047)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2003	-	Name in DB-Param ("Normal" display mode)	Configuration.Data block switching.Rise limitation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.smoothingTimeByChangeDifference*
		Significance	Time constant for smoothing manipulated variable jumps
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
2010	1-16	Name in DB-Param ("Normal" display mode)	Limits.Fixed stop.Following error
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ClampingMonitoring.followingErrorDeviation*
		Significance	Declaration of the following error required to detect the fixed end stop
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2011	1-16	Name in DB-Param ("Normal" display mode)	Limits.Fixed stop.Position tolerance
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ClampingMonitoring.positionTolerance*
		Significance	Permissible offset between the actual value to the setpoint in clamped state
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2012	1-16	Name in DB-Param ("Normal" display mode)	Limits.Fixed stop.Fixed stop detection mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ClampingMonitoring. recognitionMode*
		Significance	Torque monitoring activation mode
		Parameter type	Configuration data element
		Data type	DINT EnumRecognitionMode (Page 1054)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2020	1-16	Name in DB-Param ("Normal" display mode)	Mechanics.Load gearbox.Load revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfData Sets.DataSet_x.Gear denFactor*
		Significance	Gearbox factor denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
2021	1-16	Name in DB-Param ("Normal" display mode)	Mechanics.Load gearbox.Motor revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfData Sets.DataSet_x.Gear.numFactor*
		Significance	Gearbox factor numerator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2031	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PV controller.DSC activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PV_Controller.enableDSC*
		Significance	Activation of DSC
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2032	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PV controller.Weighting factor
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PV_Controller.kpc*
		Significance	Pre-control weighting
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	150.0
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2033	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PV controller.kp
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PV_Controller.kv*
		Significance	P-controller gain
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	< < 1E+7
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2034	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PV controller.Activation of the precontrol
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PV_Controller.preCon*
		Significance	Activation of pre-control
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2035	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PV controller.Activation/mode balancing filter
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PV_Controller.balanceFilterMode*
		Significance	Balancing filter mode
		Parameter type	Configuration data element
		Data type	DINT EnumBalanceFilterMode (Page 1045)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2041	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PD controller.kp
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PD_Controller.kp*
		Significance	P-controller gain
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	100
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2042	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PD controller.Delay time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct.PD_Controller.decayTime*
		Significance	Time constant of the DT1 element
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2043	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PD controller.kd
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct.PD_Controller.kd*
		Significance	Gain of the DT1 element
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2051	1-16	Name in DB-Param ("Normal" display mode)	Position control.Dynamic compensation.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicComp.enable*
		Significance	Activation of dynamic compensation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2052	1-16	Name in DB-Param ("Normal" display mode)	Position control.Dynamic compensation.First time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicComp.T1*
		Significance	First time constant
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2053	1-16	Name in DB-Param ("Normal" display mode)	Position control.Dynamic compensation.Second time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicComp.T2*
		Significance	Second time constant
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2061	1-16	Name in DB-Param ("Normal" display mode)	Position control.Dynamic controller data.Position control loop time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicData. positionTimeConstant*
		Significance	Equivalent time constant of the position control loop
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2062	1-16	Name in DB-Param ("Normal" display mode)	Position control.Dynamic controller data. Torque control loop time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicData. torqueTimeConstant*
		Significance	Equivalent time constant of the torque control loop
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
2063	1-16	Name in DB-Param ("Normal" display mode)	Position control.Dynamic controller data. Velocity control loop time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicData. velocityTimeConstant*
		Significance	Equivalent time constant of the speed control loop
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
2071	1-16	Name in DB-Param ("Normal" display mode)	Monitoring.Following error monitoring.Minimum
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicFollowing. minPositionTolerance*
		Significance	Low threshold of the following error characteristic
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2072	1-16	Name in DB-Param ("Normal" display mode)	Monitoring.Following error monitoring.Minimum velocity
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicFollowing. minVelocity*
		Significance	Lower velocity threshold of the following error characteristic
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2073	1-16	Name in DB-Param ("Normal" display mode)	Monitoring.Following error monitoring.Warning limit
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.DynamicFollowing. warningLimit*
		Significance	Warning limits of following error monitoring
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	100.0
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2081	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.kp
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.kp*
		Significance	Proportional gain
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2082	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.Weighting factor
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.kpc*
		Significance	Pre-control weighting
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	150
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2083	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.ki
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.ki*
		Significance	Gain of the integrator
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2084	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.Activation of the precontrol
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.preCon*
		Significance	Activation of pre-control
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2085	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.kd
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.kd*
		Significance	Differential gain
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2086	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.Delay time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.DecayTime*
		Significance	Delay time of the DT1 element
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
2087	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.Integrator limitation activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.EnableAntiWindUp*
		Significance	Activation of the integrator limitation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
2088	1-16	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.PID controller.Activation/mode balancing filter
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DataSet_x.ControllerStruct. PID_Controller.BalanceFilterMode*
		Significance	Balancing filter setting
		Parameter type	Configuration data element
		Data type	DINT EnumBalanceFilterMode (Page 1045)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

9.4.4 Technology parameters 3000 to 3999 - Axes

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter *ParameterNumber* of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Description	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum" data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Supported for	In the case of axis parameters specifies the type of axis for which the parameter applies.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
3000	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Encoder count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.numberOfEncoders*
		Description	Number of measuring systems
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	8
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3010	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Encoder attachment type
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AssemblyBase. assemblyBase*
		Description	Attachment type of the encoder
		Parameter type	Configuration data element
		Data type	DINT EnumAxisEncoderAssemblyType (Page 1041)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3011	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Motor side measuring gearbox.Encoder revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AdaptDrive.denFactor*
		Description	Denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3012	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Motor side measuring gearbox.Motor revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AdaptDrive.numFactor*
		Description	Counters
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3013	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Load side measuring gearbox.Encoder revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AdaptExtern.denFactor*
		Description	Denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3014	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Load side measuring gearbox.Load revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AdaptExtern.numFactor*
		Description	Counters
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3015	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.External measuring gearbox.Path per measuring wheel rotation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.PathPerRevolution. length*
		Description	Path for each measuring wheel rotation
		Parameter type	Configuration data element
		Data type	REAL
		Min.	> 0.0
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3016	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.External measuring gearbox.Encoder revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AdaptLoad.denFactor*
		Description	Denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3017	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.External measuring gearbox.Measuring wheel revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AdaptLoad.numFactor*
		Description	Counters
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3021	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Absolute encoder.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AbsBackLash.enable*
		Description	Status of the activation of backlash compensation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3022	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Absolute encoder.Direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AbsBackLash._type*
		Description	Direction of backlash compensation
		Parameter type	Configuration data element
		Data type	DINT EnumBackLashType (Page 1045)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3023	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Absolute encoder.Preferred position
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AbsBackLash. startUpDifference*
		Description	Preferred position
		Parameter type	Configuration data element
		Data type	DINT EnumBackLashDiff (Page 1044)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3024	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Absolute encoder.Value
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AbsBackLash.length*
		Description	Value
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3025	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Absolute encoder.Velocity
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AbsBackLash.velocity*
		Description	Backlash compensation velocity
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3031	1-8	Name in DB-Param ("Normal" display mode)	Homing.Offset
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.AbsHomingEncoder.absShift*
		Description	Offset of the absolute encoder
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3041	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Filter.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.Filter.enable*
		Description	Activation status of the filter functionality
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3042	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Filter.Time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.Filter.timeConstant*
		Description	Time constant for PT1 smoothing (actual value smoothing)
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3043	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Encoder count direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_X.InversCountDirection. encoderFeedbackPolarity*
		Description	Activation of the count direction inversion
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3051	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Incremental encoder.Direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncBackLash._type*
		Description	Effective direction of backlash compensation
		Parameter type	Configuration data element
		Data type	DINT EnumBackLashType (Page 1045)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3052	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Incremental encoder.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncBackLash.enable*
		Description	Status of the activation of backlash compensation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3053	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Incremental encoder.Value
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncBackLash.length*
		Description	Backlash value
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3054	1-8	Name in DB-Param ("Normal" display mode)	Mechanics.Backlash on reversal compensation.Incremental encoder.Velocity
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncBackLash.velocity*
		Description	Velocity of traversing at backlash compensation
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
3061	1-8	Name in DB-Param ("Normal" display mode)	Homing.Active homing.Approach direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncHomingEncoder.approachDirection*
		Description	Reference approach direction
		Parameter type	Configuration data element
		Data type	DINT EnumAxisApproachDirection (Page 1041)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3063	1-8	Name in DB-Param ("Normal" display mode)	Homing.Zero mark activation monitoring
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncHomingEncoder.enableZeroMarkDistance*
		Description	Activation of the encoder zero mark monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3064	1-8	Name in DB-Param ("Normal" display mode)	Homing.Active homing.Homing mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncHomingEncoder.homingMode*
		Description	Homing mode - active homing
		Parameter type	Configuration data element
		Data type	DINT EnumAxisHomingMode (Page 1043)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3066	1-8	Name in DB-Param ("Normal" display mode)	Homing.Passive homing.Approach direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncHomingEncoder.passiveApproachDirection*
		Description	Expected approach direction
		Parameter type	Configuration data element
		Data type	DINT EnumAxisPassiveApproachDirection (Page 1043)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3068	1-8	Name in DB-Param ("Normal" display mode)	Homing.Passive homing.Homing mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOf Encoders.Encoder_x.IncHomingEncoder.passiveHomingMode*
		Description	Homing mode - passive homing
		Parameter type	Configuration data element
		Data type	DINT EnumAxisPassiveHomingMode (Page 1044)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3070	1-8	Name in DB-Param ("Normal" display mode)	Homing.Active homing.Reference point offset
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncHomingEncoder.proceedShiftPos*
		Description	Reference point offset at incremental encoder
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3071	1-8	Name in DB-Param ("Normal" display mode)	Homing.Distance to zero mark
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_x.IncHomingEncoder.zeroMarkDistance*
		Description	Maximum distance between homing output cam and encoder zero mark
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3072	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Homing.Actual value change
		Name in DB-Param ("Expert" display mode)	*homingcommand.positiondifference*
		Description	Actual value change during last homing
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3073	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Activate position filter
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_X.AnalogSensor. PositionFilter.enable*
		Description	Activation of filter
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3074	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Time constant position filter
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_X.AnalogSensor. PositionFilter.timeConstant*
		Description	Time constant for PT1 smoothing
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
3075	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Weighting factor
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_X.AnalogSensor. ConversionData.factor*
		Description	Weighting factor of the analog measured value
		Parameter type	Configuration data element
		Data type	REAL
		Min.	> 0.0
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
3076	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Offset
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_X.AnalogSensor. ConversionData.offset*
		Description	Offset
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

9.4.5 Technology parameters 4000 to 4999 - Axes

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter "ParameterNumber" of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.

Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Significance	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum" data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Supported for	In the case of axis parameters specifies the type of axis for which the parameter applies.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
4001	-	Name in DB-Param ("Normal" display mode)	Configuration.Accept changes
		Name in DB-Param ("Expert" display mode)	*activationmodechanged configdata*
		Significance	Activation of modified configuration data
		Parameter type	System variables
		Data type	DINT EnumToActivationModeSetConfigData (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4010	1-8	Name in DB-Param ("Normal" display mode)	Homing.Absolute value encoder adjustment.Total offset
		Name in DB-Param ("Expert" display mode)	*absoluteencoder_x.totaloffsetvalue*
		Significance	Measuring system state of the absolute value encoder, calculated total offset
		Parameter type	System variables
		Data type	2x DWORD
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4011	1-8	Name in DB-Param ("Normal" display mode)	Homing.Absolute value encoder adjustment.Measuring system status
		Name in DB-Param ("Expert" display mode)	*absoluteencoder_x.activationstate*
		Significance	Measuring system state of the absolute value encoder, including calculated total offset
		Parameter type	System variables
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4020	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Data block switching.Active data block
		Name in DB-Param ("Expert" display mode)	*datasetmonitoring.actualdataset*
		Significance	Currently active data record number
		Parameter type	System variables
		Data type	DINT
		Min.	0
		Max.	16
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4030	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Motion.Overlaid movement.Position
		Name in DB-Param ("Expert" display mode)	*superimposedmotion.position*
		Significance	Position in the superimposing coordinate system
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4031	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Motion.Overlaid movement.Velocity
		Name in DB-Param ("Expert" display mode)	*superimposedmotion.velocity*
		Significance	Velocity in the superimposing coordinate system
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4040	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Motion.Base motion.Position
		Name in DB-Param ("Expert" display mode)	*basicmotion.position*
		Significance	Position in the base coordinate system
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4041	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Motion.Base motion.Velocity
		Name in DB-Param ("Expert" display mode)	*basicmotion.velocity*
		Significance	Velocity in the base coordinate system
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4050	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Sensor.Status
		Name in DB-Param ("Expert" display mode)	*sensordata_x.state*
		Significance	Measuring system status
		Parameter type	System variables
		Data type	DINT EnumSensorState (Page 1054)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4121	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Negative direction of travel.Negative acceleration
		Name in DB-Param ("Expert" display mode)	*minuslimitsofdynamics.negativeaccel*
		Significance	Limits of negative acceleration/deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4122	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Negative direction of travel.Negative jerk
		Name in DB-Param ("Expert" display mode)	*minuslimitsofdynamics.negativeacceljerk*
		Significance	Jerk limit at the end of acceleration and at the start of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4123	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Negative direction of travel.Positive acceleration
		Name in DB-Param ("Expert" display mode)	minuslimitsofdynamics.positiveaccel
		Significance	Limits of positive acceleration/deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4124	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Negative direction of travel.Positive jerk
		Name in DB-Param ("Expert" display mode)	minuslimitsofdynamics.positiveacceljerk
		Significance	Jerk limits at the start of acceleration and at the end of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4125	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity.Negative direction of travel.Velocity
		Name in DB-Param ("Expert" display mode)	*minuslimitsofdynamics.velocity*
		Significance	Velocity limiting
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4141	-	Name in DB-Param ("Normal" display mode)	Setpoints.Override.Acceleration
		Name in DB-Param ("Expert" display mode)	*override.acceleration*
		Significance	Acceleration override as a percentage
		Parameter type	System variables
		Data type	REAL
		Min.	1.0
		Max.	1000.0
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4142	-	Name in DB-Param ("Normal" display mode)	Setpoints.Override.Velocity
		Name in DB-Param ("Expert" display mode)	*override.velocity*
		Significance	Velocity override as a percentage
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	200.0
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4151	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Positive direction of travel.Negative acceleration
		Name in DB-Param ("Expert" display mode)	*pluslimitsofdynamics.negativeaccel*
		Significance	Limits of negative acceleration/deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4152	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Negative direction of travel.Negative jerk
		Name in DB-Param ("Expert" display mode)	*pluslimitsofdynamics.negativeacceljerk*
		Significance	Jerk limit at the end of acceleration and at the start of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4153	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Negative direction of travel.Positive acceleration
		Name in DB-Param ("Expert" display mode)	*pluslimitsofdynamics.positiveaccel*
		Significance	Limits of positive acceleration/deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4154	-	Name in DB-Param ("Normal" display mode)	Limits.Dynamic response.Negative direction of travel.Positive jerk
		Name in DB-Param ("Expert" display mode)	*pluslimitsofdynamics.positiveacceljerk*
		Significance	Jerk limits at the start of acceleration and at the end of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4155	-	Name in DB-Param ("Normal" display mode)	Limits.Position and velocity.Positive direction of travel.Velocity
		Name in DB-Param ("Expert" display mode)	*pluslimitsofdynamics.velocity*
		Significance	Velocity limiting
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4162	-	Name in DB-Param ("Normal" display mode)	Actual values.Commands.Positioning.Distance to go
		Name in DB-Param ("Expert" display mode)	*poscommand.distancetogo*
		Significance	Current remaining distance to target
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4173	-	Name in DB-Param ("Normal" display mode)	Actual values.Homing.Reference position coordinates
		Name in DB-Param ("Expert" display mode)	*positioningstate.homeposition*
		Significance	Reference point coordinate
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4174	-	Name in DB-Param ("Normal" display mode)	Actual values.Homing.Bero status
		Name in DB-Param ("Expert" display mode)	*homingcommand.berostate*
		Significance	Reference output cam active (homing with encoder zero mark and reference output cam)
		Parameter type	System variables
		Data type	DINT EnumActiveInactive (Page 1040)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4180	-	Name in DB-Param ("Normal" display mode)	Homing.Reference position coordinates
		Name in DB-Param ("Expert" display mode)	*userdefaulthoming.homeposition*
		Significance	Reference point coordinate
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4181	-	Name in DB-Param ("Normal" display mode)	Homing.Approach speed
		Name in DB-Param ("Expert" display mode)	*userdefaulthoming.homingapproachvelocity*
		Significance	Reference point approach velocity
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	10000.0
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4182	-	Name in DB-Param ("Normal" display mode)	Homing.Creep speed
		Name in DB-Param ("Expert" display mode)	*userdefaulthoming.homingentryvelocity*
		Significance	Reference point entry velocity
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	2000.0
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4183	-	Name in DB-Param ("Normal" display mode)	Homing.Shutdown speed
		Name in DB-Param ("Expert" display mode)	*userdefaulthoming.homingreducedvelocity*
		Significance	Reference point shutdown speed
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1000.0
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4190	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Offset compensation
		Name in DB-Param ("Expert" display mode)	servosettings.setpointoffsetcompensation
		Significance	Offset compensation of the analog interface
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4231	-	Name in DB-Param ("Normal" display mode)	Limits.Fixed stop.Clamping value
		Name in DB-Param ("Expert" display mode)	*userdefaultclamping.clampingvalue*
		Significance	Clamping value (for example traversing to fixed end stop)
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4241	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Direction
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.direction*
		Significance	Selection of direction
		Parameter type	System variables
		Data type	DINT EnumDirection (Page 1048)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4242	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Negative acceleration
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.negativeaccel*
		Significance	Negative acceleration/deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4243	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Jerk at end of deceleration
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.negativeaccelendjerk*
		Significance	Jerk limit at the end of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4244	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Jerk at start of deceleration
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.negativeaccelstartjerk*
		Significance	Jerk limit at the start of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4245	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Positive acceleration
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.positiveaccel*
		Significance	Positive acceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4246	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Jerk at end of acceleration
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.positiveaccelendjerk*
		Significance	Jerk limit at the end of acceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4247	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Jerk at start of acceleration
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.positiveaccelstartjerk*
		Significance	Jerk limit at the start of acceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4248	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Velocity profile
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.profile*
		Significance	Speed profile type
		Parameter type	System variables
		Data type	DINT EnumProfile (Page 1053)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4249	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Stopping time
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.stoptime*
		Significance	Time specified for emergency stop
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4250	-	Name in DB-Param ("Normal" display mode)	Default.Default value.Dynamics.Velocity
		Name in DB-Param ("Expert" display mode)	*userdefaultdynamics.velocity*
		Significance	Velocity specification
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4262	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Cam synchronization.Master offset
		Name in DB-Param ("Expert" display mode)	*userdefault.cammingsettings.camstartpositionmaster*
		Significance	Offset for the cam start position with "relative" master
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4266	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Cam synchronization.Synchronization mode
		Name in DB-Param ("Expert" display mode)	*userdefault.cammingsettings.synchronizingmode*
		Significance	Synchronization criterion for cam disk
		Parameter type	System variables
		Data type	DINT EnumSyncModeCamming (Page 1055)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4267	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Cam synchronization.Desynchronization mode
		Name in DB-Param ("Expert" display mode)	*userdefault.cammingsettings.syncoffmode*
		Significance	Cam disk desynchronization criterion
		Parameter type	System variables
		Data type	DINT EnumSyncOffModeCamming (Page 1056)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4268	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Cam synchronization.Synchronization master position
		Name in DB-Param ("Expert" display mode)	*userdefault.cammingsettings.syncpositionmaster*
		Significance	Master position for synchronization
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4269	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Cam synchronization.Synchronization slave position
		Name in DB-Param ("Expert" display mode)	*userdefault.cammingsettings.syncpositionslave*
		Significance	Slave position for synchronization
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4270	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Cam synchronization.Slave synchronization direction
		Name in DB-Param ("Expert" display mode)	*userdefault.cammingsettings.synchronizingDirection*
		Significance	Direction of slave synchronization
		Parameter type	System variables
		Data type	DINT EnumFollowingObjectSynchronizingDirection (Page 1049)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4286	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Gear synchronization.Synchronization criteria
		Name in DB-Param ("Expert" display mode)	*userdefault.gearingsettings.synchronizingmode*
		Significance	Gearing synchronization criterion
		Parameter type	System variables
		Data type	DINT EnumSyncModeGearing (Page 1055)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4287	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Gear synchronization.Desynchronization criteria
		Name in DB-Param ("Expert" display mode)	*userdefault.gearingsettings.syncoffmode*
		Significance	Gear desynchronization criterion
		Parameter type	System variables
		Data type	DINT EnumSyncOffModeGearing (Page 1056)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4288	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Gear synchronization.Master position
		Name in DB-Param ("Expert" display mode)	*userdefault.gearingsettings.syncpositionmaster*
		Significance	Synchronization position of the master
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4289	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Gear synchronization.Slave position
		Name in DB-Param ("Expert" display mode)	*userdefault.gearingsettings.syncpositionslave*
		Significance	Synchronization position of the slave
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4291	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Gear synchronization.Slave synchronization direction
		Name in DB-Param ("Expert" display mode)	*userdefault.gearingsettings.syncronizingDirection*
		Significance	Direction of slave synchronization
		Parameter type	System variables
		Data type	DINT EnumFollowingObjectSynchronizingDirection (Page 1049)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronization axis
		Firmware version	
4292	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Gear synchronization.Look ahead
		Name in DB-Param ("Expert" display mode)	*userdefault.gearingsettings.synchronizewithlookahead*
		Description	Associated Master Look ahead
		Parameter type	System variable
		Data type	DINT EnumFollowingObjectSynchronizeWithLookAhead (Page 1049)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4301	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Negative acceleration
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.negativeaccel*
		Significance	Negative acceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4302	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Jerk at end of deceleration
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.negativeaccelendjerk*
		Significance	Jerk limit at the end of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4303	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Jerk at start of deceleration
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.negativeaccelstartjerk*
		Significance	Jerk limit at the start of deceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4304	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Positive acceleration
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.positiveaccel*
		Significance	Positive acceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4305	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Jerk at end of acceleration
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.positiveaccelendjerk*
		Significance	Jerk limit at the end of acceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4306	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Jerk at start of acceleration
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.positiveaccelstartjerk*
		Significance	Jerk limit at the start of acceleration
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4307	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Velocity
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.velocity*
		Significance	Velocity
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4308	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Dynamics.Velocity profile
		Name in DB-Param ("Expert" display mode)	*userdefault.syncdynamics.velocityprofile*
		Significance	Type of velocity profile
		Parameter type	System variables
		Data type	DINT EnumProfile (Page 1053)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4321	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Desynchronization master position
		Name in DB-Param ("Expert" display mode)	*userdefault.syncoffpositions.master*
		Significance	Desynchronization position of the master
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4322	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Desynchronization slave position
		Name in DB-Param ("Expert" display mode)	*userdefault.syncoffpositions.slave*
		Significance	Desynchronization position of the slave
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4331	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Synchronization path
		Name in DB-Param ("Expert" display mode)	*userdefault.syncprofile.synclength*
		Significance	Synchronization length for the specific synchronization profile of a leading axis
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4332	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Desynchronization path
		Name in DB-Param ("Expert" display mode)	*userdefault.syncprofile.syncofflength*
		Significance	Desynchronization length for the specific synchronization profile of a leading axis
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4333	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Desynchronization reference
		Name in DB-Param ("Expert" display mode)	*userdefault.syncprofile.syncoffpositionreference*
		Significance	Position reference for desynchronization profile
		Parameter type	System variables
		Data type	DINT EnumSyncOffPositionReference (Page 1057)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4334	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Synchronization reference
		Name in DB-Param ("Expert" display mode)	*userdefault.syncprofile.syncpositionreference*
		Significance	Position reference for synchronization profile
		Parameter type	System variables
		Data type	DINT EnumSyncPositionReference (Page 1057)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4335	1-2	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Profile reference
		Name in DB-Param ("Expert" display mode)	userdefault.syncprofile.syncprofilereference
		Significance	Reference variable of synchronization profile
		Parameter type	System variables
		Data type	DINT EnumSyncProfileReference (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4336	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Absolut.Jerklimiting
		Name in DB-Param ("Expert" display mode)	*SyncinMotion.smoothAbsolute Synchronization*
		Significance	Jerk limits during synchronization in absolute synchronization mode
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4337	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Synchronization.Limiting
		Name in DB-Param ("Expert" display mode)	*SyncingMotion.synchronizingAdaption*
		Significance	Increase / limitation of default dynamic values
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Synchronized axis
		Firmware version	
4338	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Actual values.Master position
		Name in DB-Param ("Expert" display mode)	*currentsyncposition.master*
		Significance	Master value
		Parameter type	System variable
		Data type	REAL
		Min.	-3.402823E+38
		Max.	3.402823E+38
		Access	Read
		Active	-
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4339	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Actual values.Slave position
		Name in DB-Param ("Expert" display mode)	*currentsyncposition.slave*
		Significance	Slave value
		Parameter type	System variable
		Data type	REAL
		Min.	-3.402823E+38
		Max.	3.402823E+38
		Access	Read
		Active	-
		Supported for	Synchronized axis
		Firmware version	
4350	1-2	Name in DB-Param ("Normal" display mode)	Actual values.Synchronous operation.Cam synchronization.Master offset
		Name in DB-Param ("Expert" display mode)	*cammingadjustments.master.offset*
		Significance	Master offset
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Synchronized axis
		Firmware version	
4351	1-2	Name in DB-Param ("Normal" display mode)	Actual values.Synchronous operation.Cam synchronization.Master scale
		Name in DB-Param ("Expert" display mode)	*cammingadjustments.master.scale*
		Significance	Master scaling
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4360	1-2	Name in DB-Param ("Normal" display mode)	Actual values.Synchronous operation.Cam synchronization.Slave offset
		Name in DB-Param ("Expert" display mode)	*cammingadjustments.slave.offset*
		Significance	Slave offset
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Synchronized axis
		Firmware version	
4361	1-2	Name in DB-Param ("Normal" display mode)	Actual values.Synchronous operation.Cam synchronization.Slave scale
		Name in DB-Param ("Expert" display mode)	*cammingadjustments.slave.scale*
		Significance	Slave scaling
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Synchronized axis
		Firmware version	
4370	1-2	Name in DB-Param ("Normal" display mode)	Actual values.Synchronous operation.Gear synchronization.Master offset
		Name in DB-Param ("Expert" display mode)	*gearingadjustments.master.offset*
		Significance	Master offset
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Synchronized axis
		Firmware version	

No.	Index	Parameter description	
4400	-	Name in DB-Param ("Normal" display mode)	Actual values.Limits.Software limit switch.SW minus limit switch status
		Name in DB-Param ("Expert" display mode)	*swlimitstate.swlimitswitchminus*
		Significance	Software limit switch minus
		Parameter type	System variables
		Data type	DINT EnumLimitExceededOk (Page 1051)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4401	-	Name in DB-Param ("Normal" display mode)	Actual values.Limits.Software limit switch.SW plus limit switch status
		Name in DB-Param ("Expert" display mode)	*swlimitstate.swlimitswitchplus*
		Significance	Software limit switch plus
		Parameter type	System variables
		Data type	DINT EnumLimitExceededOk (Page 1051)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Positioning axis, synchronization axis
		Firmware version	
4410	-	Name in DB-Param ("Normal" display mode)	Actual values.Limits.Hardware limit switch.HW minus limit switch status
		Name in DB-Param ("Expert" display mode)	*sensormonitoring.hwlimitswitchminus*
		Significance	Hardware minus limit switch
		Parameter type	System variables
		Data type	DINT EnumLimitExceededOk (Page 1051)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4411	-	Name in DB-Param ("Normal" display mode)	Actual values.Limits.Hardware limit switch.HW plus limit switch status
		Name in DB-Param ("Expert" display mode)	*sensormonitoring.hwlimitswitchplus*
		Significance	Hardware plus limit switch
		Parameter type	System variables
		Data type	DINT EnumLimitExceededOk (Page 1051)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4412	-	Name in DB-Param ("Normal" display mode)	Actual values.Measuring system.Active encoder
		Name in DB-Param ("Expert" display mode)	*sensormonitoring.actualsensor*
		Significance	Number of the active encoder / sensor
		Parameter type	System variables
		Data type	DINT
		Min.	1
		Max.	8
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4500	-	Name in DB-Param ("Normal" display mode)	Setpoints.Drive.Control words
		Name in DB-Param ("Expert" display mode)	*drivedata.stw*
		Significance	Control words 1 and 2
		Parameter type	System variables
		Data type	DWORD
		Min.	-
		Max.	-
		Access	Read, Write
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
4501	-	Name in DB-Param ("Normal" display mode)	Actual values.Drive.Status words
		Name in DB-Param ("Expert" display mode)	*drivedata.zsw*
		Significance	Status words 1 and 2
		Parameter type	System variables
		Data type	DWORD
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
4502	-	Name in DB-Param ("Normal" display mode)	Configuration.Drive.Torque.Resolution
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SetPointDriverInfo.DriveData. torqueReductionGranularity*
		Significance	Fine resolution of the torque reduction
		Parameter type	Configuration data element
		Data type	DINT EnumAxisTorqueForceReductionGranularity (Page 1044)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

9.4.6 Technology parameters 5000 to 9999 - Axes

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter "ParameterNumber" of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Significance	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum" data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Supported for	Specifies the type of axis for which the parameter applies.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
5000	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Superposition.Setpoint
		Name in DB-Param ("Expert" display mode)	*servosettings.additionalqoutputvalue*
		Significance	Hydraulic axis: Manipulated variable superimposition
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5001	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Superposition.Activation
		Name in DB-Param ("Expert" display mode)	*servosettings.additionalqoutputvalueswitch*
		Significance	Hydraulic axis: Activate manipulated variable superimposition
		Parameter type	System variables
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5002	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Setpoints.Qoutputvalue
		Name in DB-Param ("Expert" display mode)	*actordata.qoutputvalue*
		Significance	Hydraulic axis: Manipulated variable relative to Q output as %
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
5003	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Sliding friction compensation.Dynamic.Setpoint
		Name in DB-Param ("Expert" display mode)	*actordata.slidingfriction compensationvalue*
		Significance	Hydraulic axis: Velocity-dependent sliding friction compensation value
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5004	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Sliding friction compensation.Dynamic.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SlidingFriction.enable*
		Significance	Hydraulic axis: Activation of the sliding-friction compensation
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5005	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Sliding friction compensation.Dynamic.Factor
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SlidingFriction.FactorMotionControl*
		Significance	Hydraulic axis: Factor of sliding friction compensation as a function of the velocity setpoint
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
5006	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Sliding friction compensation.Static.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.AdditionalOffset.enable*
		Significance	Hydraulic axis: Activation of the offset injection
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5007	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Sliding friction compensation.Static.Negative direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.AdditionalOffset.offsetMotionControlNegative*
		Significance	Hydraulic axis: Offset with negative direction of motion
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5008	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Sliding friction compensation.Static.Positive direction
		Name in DB-Param ("Expert" display mode)	TypeOfAxis.AdditionalOffset.offsetMotionControlPositive
		Significance	Hydraulic axis: Offset with positive direction of motion
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
5009	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Limits.Set value increasing
		Name in DB-Param ("Expert" display mode)	*userdefaultqfaxis.maxderivative.qoutput*
		Significance	Hydraulic axis: Limit of changes at the manipulated variable
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5010	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Sliding friction compensation.Static.Setpoint
		Name in DB-Param ("Expert" display mode)	*actordata.frictionadditionaloffsetvalue*
		Significance	Hydraulic axis: Direction-dependent sliding friction compensation value
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5011	-	Name in DB-Param ("Normal" display mode)	Position control.Static friction compensation.Activation
		Name in DB-Param ("Expert" display mode)	*actordata.frictioncompensation*
		Significance	Status of the activation of friction compensation
		Parameter type	System variables
		Data type	DINT EnumActiveInactive (Page 1040)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
5012	-	Name in DB-Param ("Normal" display mode)	Position control.Static friction compensation.Setpoint
		Name in DB-Param ("Expert" display mode)	*actordata.frictioncompensationvalue*
		Significance	Friction compensation value
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5013	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Limits.Set value limit exceeded
		Name in DB-Param ("Expert" display mode)	*actormonitoring.qoutputderivativelimitingstate*
		Significance	Hydraulic axis: Limit of the rise at the Q output
		Parameter type	System variables
		Data type	DINT EnumLimitExceededOk (Page 1051)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5014	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Valve active
		Name in DB-Param ("Expert" display mode)	*actormonitoring.qoutputstate*
		Significance	Hydraulic axis: Q output active
		Parameter type	System variables
		Data type	DINT EnumActiveInactive (Page 1040)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
5015	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Dynamic controller data.Hydraulic time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DynamicQFData. qOutputTimeConstant*
		Significance	Hydraulic axis: Equivalent time constant of the hydraulic control loop
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5016	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Dynamic controller data.Velocity control loop time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DynamicQFData. velocityTimeConstant*
		Significance	Hydraulic axis: Equivalent time constant of the speed control loop
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis
		Firmware version	
5017	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Dynamic controller data.Position control loop time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.DynamicQFData. positionTimeConstant*
		Significance	Hydraulic axis: Equivalent time constant of the position control loop
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
5018	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Invert setpoint.In front of characteristic.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.InvertQOutput.invSetPoint*
		Significance	Hydraulic axis: Activation of manipulated variable superimposition
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5019	-	Name in DB-Param ("Normal" display mode)	Hydraulic.Invert setpoint.After characteristic.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfDataSets.InvertSetPoint.Invert*
		Significance	Hydraulic axis: Activation of Q output inversion
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	
5020	1-8	Name in DB-Param ("Normal" display mode)	Actual values.Sensor.Velocity
		Name in DB-Param ("Expert" display mode)	*sensordata.sensordata_x.velocity*
		Significance	Actual velocity value, encoder value
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	-
		Supported for	Speed-controlled axis, positioning axis, synchronization axis
		Firmware version	

No.	Index	Parameter description	
5065	1-8	Name in DB-Param ("Normal" display mode)	Configuration.Releases
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.DriveControllConfig.ReleaseDisableMode*
		Significance	Bit to be reset ProfiDrive protocol
		Parameter type	Configuration data element
		Data type	WORD
		Min.	1
		Max.	127
		Access	Read, Write
		Active	Immediately
		Supported for	Positioning axis, synchronization axis
		Firmware version	

9.4.7 Technological parameters - External encoders

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter "ParameterNumber" of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Description	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum" data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
1100	-	Name in DB-Param ("Normal" display mode)	Mechanics.Spindle settings.Spindle pitch for each revolution of the axis
		Name in DB-Param ("Expert" display mode)	*LeadScrew.pitchVal*
		Description	Leadscrew pitch for each revolution of the axis
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	+1E12
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
1110	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Extrapolation time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationTime*
		Description	Extrapolation time of the master axis
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	+1E12
		Access	Read, Write
		Active	Immediately
		Firmware version	
1111	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Filter.Mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.Filter.Mode*
		Description	Selection criterion for actual value smoothing
		Parameter type	Configuration data element
		Data type	DINT EnumAxisFilterMode (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1112	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Filter.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.Filter.Enable*
		Description	Activation status
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
1113	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Filter.Time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.Filter.TimeConstant*
		Description	Time constant for PT1 smoothing
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
1114	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Tolerance range.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ToleranceRange.Enable*
		Description	Activation of the tolerance range
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
1115	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Tolerance range.Tolerance window
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ToleranceRange.Value*
		Description	Tolerance window
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
1116	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Velocity signal
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis Extrapolation.extrapolatedVelocitySwitch*
		Description	Selection of the velocity master value
		Parameter type	Configuration data element
		Data type	DINT EnumAxisExtrapolationVelocitySwitch (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
1130	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Position filter.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.enable*
		Description	Activation of filter
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1131	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Position filter.T1
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T1*
		Description	First time constant
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
1132	-	Name in DB-Param ("Normal" display mode)	Synchronous operation.Extrapolator.Position filter.T2
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Extrapolation.ExtrapolationPositionFilter.T2*
		Description	Second time constant
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
1201	-	Name in DB-Param ("Normal" display mode)	Mechanics.Modulo.Activation
		Name in DB-Param ("Expert" display mode)	*Modulo.state*
		Description	Activation of the modulo settings
		Parameter type	Configuration data element
		Data type	DINT EnumActiveInactive (Page 1040)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1202	-	Name in DB-Param ("Normal" display mode)	Mechanics.Modulo.Length
		Name in DB-Param ("Expert" display mode)	*Modulo.length*
		Description	Modulo length
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
1203	-	Name in DB-Param ("Normal" display mode)	Mechanics.Modulo.Starting value
		Name in DB-Param ("Expert" display mode)	*Modulo.startValue*
		Description	Modulo starting value
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Firmware version	
1221	-	Name in DB-Param ("Normal" display mode)	Monitoring.Velocity.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.ActualVelocityMonitoring.enable*
		Description	Activation of the actual velocity monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
1222	-	Name in DB-Param ("Normal" display mode)	Monitoring.Velocity.Maximum value
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.ActualVelocityMonitoring.maximum*
		Description	Maximum permissible actual velocity
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
1331	-	Name in DB-Param ("Normal" display mode)	Configuration.Actual value smoothing.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SmoothingFilter.enable*
		Description	Activation of the filter functionality
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
1332	-	Name in DB-Param ("Normal" display mode)	Configuration.Actual value smoothing.Mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SmoothingFilter.mode*
		Description	Calculation method for actual value smoothing
		Parameter type	Configuration data element
		Data type	DINT EnumAxisFilterMode (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1333	-	Name in DB-Param ("Normal" display mode)	Configuration.Actual value smoothing.Time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SmoothingFilter.timeConstant*
		Description	Time constant for PT1 smoothing
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
1351	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Message delay time
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.delayTimeToActivate*
		Description	Delay time for the triggering of the standstill signal
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
1352	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Degree of filtering
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.filterDegree*
		Description	Degree of filtering
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	15
		Access	Read, Write
		Active	Immediately
		Firmware version	
1353	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Filter frequency
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.filterFrequency*
		Description	Filter frequency
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
1354	-	Name in DB-Param ("Normal" display mode)	Monitoring.Standstill signal.Velocity limit
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.StandStillSignal.maxVeloStandStill*
		Description	Velocity limit for standstill signal
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
1400	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data. Manipulated variable limit.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SpeedLimitation.Enable*
		Description	Activation status of the speed range limit
		Parameter type	Configuration data element
		Data type	DINT EnumAxisFilterMode (Page 1042)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1401	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Manipulated variable limit.Lower limit
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SpeedLimitation.MinSpeed*
		Description	Lower limit for the speed range
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
1402	-	Name in DB-Param ("Normal" display mode)	Position control.Static controller data.Manipulated variable limit.Upper limit
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.SpeedLimitation.MaxSpeed*
		Description	Upper limit for the speed range
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
2020	-	Name in DB-Param ("Normal" display mode)	Mechanics.Load gearbox.Load revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Gear.denFactor*
		Description	Gearbox factor denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	
2021	-	Name in DB-Param ("Normal" display mode)	Mechanics.Load gearbox.Motor revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.Gear.numFactor*
		Description	Gearbox factor numerator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	0
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
3010	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Encoder attachment type
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AssemblyBase.assemblyBase*
		Description	Attachment type of the encoder
		Parameter type	Configuration data element
		Data type	DINT EnumAxisEncoderAssemblyType (Page 1041)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
3011	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Motor side measuring gearbox.Encoder revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AdaptDrive.denFactor*
		Description	Denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	
3012	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Motor side measuring gearbox.Motor revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AdaptDrive.numFactor*
		Description	Counters
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
3013	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Load side measuring gearbox.Encoder revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AdaptExtern.denFactor*
		Description	Denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	
3014	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.Load side measuring gearbox.Load revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AdaptExtern.numFactor*
		Description	Counters
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	
3015	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.External measuring gearbox.Path per measuring wheel rotation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.PathPerRevolution. ength*
		Description	Path for each measuring wheel rotation
		Parameter type	Configuration data element
		Data type	REAL
		Min.	> 0.0
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
3016	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.External measuring gearbox.Encoder revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AdaptLoad.denFactor*
		Description	Denominator
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	
3017	-	Name in DB-Param ("Normal" display mode)	Mechanics.Measuring system.External measuring gearbox.Measuring wheel revolutions count
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AdaptLoad.numFactor*
		Description	Counters
		Parameter type	Configuration data element
		Data type	DINT
		Min.	1
		Max.	2147483647
		Access	Read, Write
		Active	Restart
		Firmware version	
3031	-	Name in DB-Param ("Normal" display mode)	Homing.Offset
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AbsHomingEncoder.absShift*
		Description	Offset of the absolute encoder
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
3041	-	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Filter.Activation
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.Filter.enable*
		Description	Activation status of the filter functionality
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
3042	-	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Filter.Time constant
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.Filter.timeConstant*
		Description	Time constant for PT1 smoothing (actual value smoothing)
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
3043	-	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Encoder count direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_X.InversCountDirection. encoderFeedbackPolarity*
		Description	Activation of the count direction inversion
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
3063	-	Name in DB-Param ("Normal" display mode)	Homing.Zero mark activation monitoring
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.IncHomingEncoder.enableZeroMarkDistance*
		Description	Activation of the encoder zero mark monitoring
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
3066	-	Name in DB-Param ("Normal" display mode)	Homing.Passive homing.Approach direction
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.IncHomingEncoder.passiveApproachDirection*
		Description	Expected approach direction
		Parameter type	Configuration data element
		Data type	DINT EnumAxisPassiveApproachDirection (Page 1043)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
3068	-	Name in DB-Param ("Normal" display mode)	Homing.Passive homing.Homing mode
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOf Encoders.Encoder_1.IncHomingEncoder.passiveHomingMode*
		Description	Homing mode passive homing
		Parameter type	Configuration data element
		Data type	DINT EnumAxisPassiveHomingMode (Page 1044)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
3071	-	Name in DB-Param ("Normal" display mode)	Homing.Distance to zero mark
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.IncHomingEncoder.zeroMarkDistance*
		Description	Maximum distance between BERO and the zero mark
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
3073	-	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Activate position filter
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AnalogSensor.PositionFilter.enable*
		Description	Activation of filter
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
3074	-	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Time constant position filter
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AnalogSensor.PositionFilter.timeConstant*
		Description	Time constant for PT1 smoothing
		Parameter type	Configuration data element
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
3075	-	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Weighting factor
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AnalogSensor.ConversionData.factor*
		Description	Weighting factor of the analog measured value
		Parameter type	Configuration data element
		Data type	REAL
		Min.	> 0.0
		Max.	1E+12
		Access	Read, Write
		Active	Restart
		Firmware version	
3076	-	Name in DB-Param ("Normal" display mode)	Configuration.Measuring system.Analog encoder.Offset
		Name in DB-Param ("Expert" display mode)	*TypeOfAxis.NumberOfEncoders.Encoder_1.AnalogSensor.ConversionData.offset*
		Description	Offset
		Parameter type	Configuration data element
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
4001	-	Name in DB-Param ("Normal" display mode)	Configuration.Accept changes
		Name in DB-Param ("Expert" display mode)	*activationmodechanged.configdata*
		Description	Activation of modified configuration data
		Parameter type	System variables
		Data type	DINT EnumToActivationModeSetConfigData (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
4010	-	Name in DB-Param ("Normal" display mode)	Homing.Absolute value encoder adjustment.Total offset
		Name in DB-Param ("Expert" display mode)	*absoluteencoder.totaloffset value*
		Description	Measuring system state of the absolute value encoder, calculated total offset
		Parameter type	System variables
		Data type	2x DWORD
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
4011	-	Name in DB-Param ("Normal" display mode)	Homing.Absolute value encoder adjustment.Measuring system status
		Name in DB-Param ("Expert" display mode)	*absoluteencoder.activationstate*
		Description	Measuring system state of the absolute value encoder, including calculated total offset
		Parameter type	System variables
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Firmware version	
4341	-	Name in DB-Param ("Normal" display mode)	Default.Default values.Synchronization position
		Name in DB-Param ("Expert" display mode)	*userdefault.syncposition*
		Description	Synchronization position
		Parameter type	System variables
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
4410	-	Name in DB-Param ("Normal" display mode)	Actual values.Limits.Hardware limit switch.HW minus limit switch status
		Name in DB-Param ("Expert" display mode)	*sensormonitoring.hwlimitswitchminus*
		Description	Hardware minus limit switch
		Parameter type	System variables
		Data type	DINT EnumLimitExceededOk (Page 1051)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Firmware version	
4411	-	Name in DB-Param ("Normal" display mode)	Actual values.Limits.Hardware limit switch.HW plus limit switch status
		Name in DB-Param ("Expert" display mode)	*sensormonitoring.hwlimitswitchplus*
		Description	Hardware plus limit switch
		Parameter type	System variables
		Data type	DINT EnumLimitExceededOk (Page 1051)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Firmware version	

9.4.8 Technology parameters – Output cams

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter "ParameterNumber" of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Significance	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum..." data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
1361	-	Name in DB-Param ("Normal" display mode)	Configuration.Activate cam output
		Name in DB-Param ("Expert" display mode)	*LogAddress.enableOutput*
		Significance	Activation status of the output cam output. You cannot change the status at the hardware output of a high-speed output cam.
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1362	-	Name in DB-Param ("Normal" display mode)	Configuration.Logic operation
		Name in DB-Param ("Expert" display mode)	*LogAddress.logicOperation*
		Significance	Logic operation for output cam signals
		Parameter type	Configuration data element
		Data type	DINT EnumLogicOperation (Page 1051)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1371	-	Name in DB-Param ("Normal" display mode)	Configuration.Cam type
		Name in DB-Param ("Expert" display mode)	*OcaType._type*
		Description	Output cam type
		Parameter type	Configuration data element
		Data type	DINT EnumOutputCamType (Page 1053)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
4001	-	Name in DB-Param ("Normal" display mode)	Configuration.Accept changes
		Name in DB-Param ("Expert" display mode)	*activationmodechanged.configdata*
		Description	Activation of modified configuration data
		Parameter type	System variables
		Data type	DINT EnumToActivationModeSetConfigData (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	

9.4.9 Technology parameters - Cam track

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter "ParameterNumber" of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode

Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Description	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum..." data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.
Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
1361	-	Name in DB-Param ("Normal" display mode)	Configuration.Activate cam output
		Name in DB-Param ("Expert" display mode)	*LogAddress.enableOutput*
		Significance	Activation status of the output cam output. You cannot change the status at the hardware output of a high-speed output cam.
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
1371	-	Name in DB-Param ("Normal" display mode)	Configuration.Cam type
		Name in DB-Param ("Expert" display mode)	*OcaType._type*
		Description	Output cam type
		Parameter type	Configuration data element
		Data type	DINT EnumOutputCamType (Page 1053)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	

No.	Index	Parameter description	
1372	-	Name in DB-Param ("Normal" display mode)	Configuration.Cam type
		Name in DB-Param ("Expert" display mode)	OctType.camTrackType
		Description	Output cam type
		Parameter type	Configuration data element
		Data type	DINT EnumCamTrackType (Page 1047)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Restart
		Firmware version	
4001	-	Name in DB-Param ("Normal" display mode)	Configuration.Accept changes
		Name in DB-Param ("Expert" display mode)	*activationmodechanged.configdata*
		Description	Activation of modified configuration data
		Parameter type	System variable
		Data type	DINT EnumToActivationModeSetConfigData (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
5050	-	Name in DB-Param ("Normal" display mode)	Configuration.Inversion
		Name in DB-Param ("Expert" display mode)	*OctTechnologicalCfg.InvertOutput*
		Significance	Inversion of cam track output
		Parameter type	Configuration data element
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
5052	-	Name in DB-Param ("Normal" display mode)	Cam track.Output cam activation.Bit pattern
		Name in DB-Param ("Expert" display mode)	enablevalidcam
		Significance	Rapid suppression/enabling of individual output cams on an output cam track
		Parameter type	Configuration data element
		Data type	DWORD
		Min.	0
		Max.	FFFF FFFF
		Access	Read, Write
		Active	Immediately
		Firmware version	
5060	-	Name in DB-Param ("Normal" display mode)	Cam track.Individual output cam.Starting position
		Name in DB-Param ("Expert" display mode)	*Userdefault.singleCamsettings.cam.cam[1..32].startposition*
		Significance	Output cam start position
		Parameter type	System variable
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
5061	-	Name in DB-Param ("Normal" display mode)	Cam track.Individual output cam.End position
		Name in DB-Param ("Expert" display mode)	*Userdefault.singleCamsettings.cam.cam[1..32].Endposition*
		Significance	Output cam end position
		Parameter type	System variable
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	

No.	Index	Parameter description	
5062	-	Name in DB-Param ("Normal" display mode)	Cam track.Individual output cam.ON length
		Name in DB-Param ("Expert" display mode)	*Userdefault.singleCamsettings.cam.cam[1..32].maxLength*
		Significance	Time-based cam maximum activation length
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
5063	-	Name in DB-Param ("Normal" display mode)	Cam track.Individual output cam.ON duration
		Name in DB-Param ("Expert" display mode)	*Userdefault.singleCamsettings.cam.cam[1..32].Onduration*
		Significance	ON duration for time-based output cam
		Parameter type	System variables
		Data type	REAL
		Min.	0.0
		Max.	1E+12
		Access	Read, Write
		Active	Immediately
		Firmware version	
5064	-	Name in DB-Param ("Normal" display mode)	Cam track.Individual output cam.Validity
		Name in DB-Param ("Expert" display mode)	*Userdefault.singleCamsettings.cam.cam[1..32].Validity*
		Significance	Validity
		Parameter type	System variable
		Data type	DINT EnumYesNo (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	

9.4.10 Technology parameters - Cam disk

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter *ParameterNumber* of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Description	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum..." data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.

Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
4001	-	Name in DB-Param ("Normal" display mode)	Configuration.Accept changes
		Name in DB-Param ("Expert" display mode)	*activationmodechanged.configdata*
		Description	Activation of modified configuration data
		Parameter type	System variable
		Data type	DINT EnumToActivation ModeSet ConfigData (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	
4201	-	Name in DB-Param ("Normal" display mode)	Actual values.Cam type
		Name in DB-Param ("Expert" display mode)	*interpolation.camMode*
		Significance	Interpolation conditions
		Parameter type	System variables
		Data type	DINT EnumCamMode (Page 1046)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Firmware version	
4202	-	Name in DB-Param ("Normal" display mode)	Actual values.Interpolation type
		Name in DB-Param ("Expert" display mode)	*interpolation.interpolationmode*
		Significance	Interpolation type
		Parameter type	System variables
		Data type	DINT EnumCamInterpolationMode (Page 1045)
		Min.	-
		Max.	-
		Access	Read
		Active	-
		Firmware version	

No.	Index	Parameter description	
4203	-	Name in DB-Param ("Normal" display mode)	Actual values.Definition range.Starting point
		Name in DB-Param ("Expert" display mode)	*leadingrange.start*
		Significance	Start point of the range
		Parameter type	System variable
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	Immediately
		Firmware version	
4204	-	Name in DB-Param ("Normal" display mode)	Actual values.Definition range.End point
		Name in DB-Param ("Expert" display mode)	leadingrange.end
		Significance	End point of the range
		Parameter type	System variable
		Data type	REAL
		Min.	-1E+12
		Max.	1E+12
		Access	Read
		Active	Immediately
		Firmware version	
4211	-	Name in DB-Param ("Normal" display mode)	Default.Default values.Coordinate reference
		Name in DB-Param ("Expert" display mode)	*userdefault.campositionmode*
		Significance	Reference of coordinate settings
		Parameter type	System variables
		Data type	DINT EnumCamPositionMode (Page 1047)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	

9.4.11 Technology parameters – Measuring input

Use the information given in this chapter to read or to change parameters of the technology objects in the user program. The parameters of the technology objects can be read by using the "MC_ReadSysParameter" technology function and written by using the MC_WriteParameter" technology function.

Supplementary conditions for editing technological parameters:

- The Restart specified in the list is not required if you enter a new value that is equivalent to the old value at the parameter. Also, when changing parameters of virtual axes, you do not necessarily have to perform a Restart. Whether or not a Restart is required is indicated at the *Statusword.RequestRestart* variable of the corresponding Technology DB.
- The parameter defined by parameter number and index must be available and allow write access. The data record defined by the index must be enabled in the Technology CPU with the technology function "MC_ChangeDataset."

Key

No.	Number of the parameter for the input parameter *ParameterNumber* of the "MC_ReadSysParameter" and "MC_WriteParameter" technology functions
Index	The index can be used to access parameters existing several times. Multiple parameters exist in these areas: • Data sets (Dataset_1 to Dataset_16) • Encoders (Encoder_1 to Encoder_8) • Synchronous objects (1 for synchronous object, 2 for superimposed synchronous object) Indices can be used from firmware V3.1.x and higher of the integrated technology.
Name in DB-Param ("Normal" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Normal" display mode
Name in DB-Param ("Expert" display)	If the instance data block of the "MC_ReadSysParameter" or "MC_WriteParameter" technology function is opened in STEP 7, a configuration dialog box is displayed in DB-Param. Shows the parameter name in the "Expert" display mode This name corresponds to the name of the system variable or of the configuration data element in the Expert list of S7T Config.
Description	Description of the parameter
Parameter type	Reference to the parameter type "System variable" or "Configuration data" in the Expert list of S7T Config.
Data type	Specifies the data type of the parameter. The possible values for the "DINT Enum..." data type can be found in the "List of DINT values".
Min/Max	Specifies the upper and lower value of the parameter.
Access	Read: The parameter can be read. Write: The parameter can be changed.

Active	Immediately: The change in the parameter takes effect immediately. Restart: Parameter changes require a restart, meaning when the technology object is reinitialized. This is only possible when a technology object TO is disabled.
Firmware version	Firmware version of the integrated technology from which the parameter can be used.

No.	Index	Parameter description	
4001	-	Name in DB-Param ("Normal" display mode)	Configuration.Accept changes
		Name in DB-Param ("Expert" display mode)	*activationmodechanged.configdata*
		Description	Activation of modified configuration data
		Parameter type	System variable
		Data type	DINT EnumToActivationModeSetConfigData (Page 1058)
		Min.	-
		Max.	-
		Access	Read, Write
		Active	Immediately
		Firmware version	

9.4.12 DINT values

9.4.12.1 EnumActiveInactive

EnumActiveInactive

Values	Significance	ValueDint	S7T Config
INACTIVE	Inactive	0	61
ACTIVE	Active	1	4

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.2 EnumAxisApproachDirection

EnumAxisApproachDirection

Values	Significance	ValueDint	S7T Config
APPROACH_NEGATIVE	Approach to encoder zero mark in negative direction	15	0
APPROACH_POSITIVE	Approach to encoder zero mark in positive direction	16	1
EDGE_POS_SIDE_NEG	Positive edge at the negative side of the external zero mark	17	2
EDGE_POS_SIDE_POS	Positive edge at the positive side of the external zero mark	18	3
EDGE_NEG_SIDE_POS	Negative edge at the positive side of the external zero mark	19	4
EDGE_NEG_SIDE_NEG	Negative edge at the negative side of the external zero mark	20	5

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.3 EnumAxisEncoderAssemblyType

EnumAxisEncoderAssemblyType

Values	Significance	ValueDint	S7T Config
ASSEMBLY_BASE_DRIVE	Drive side	83	0
ASSEMBLY_BASE_LOAD	Load side	84	1
ASSEMBLY_BASE_EXTERN	External	85	2
ASSEMBLY_BASE_LINEAR	Linear	86	3

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.4 EnumAxisExtrapolatedVelocitySwitch

ExtrapolatedVelocitySwitch

Values	Description	ValueDint	S7T Config
DIFFERENTIATION	Differentiation of extrapolated master value	0	408
TRANSFER	Transfer of extrapolated velocity	1	409

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.5 EnumAxisFilterMode

EnumAxisFilterMode

Values	Significance	ValueDint	S7T Config
DEFAULT_MODE	Mean value as a function of the ratio between the IPO clock and servo clock	9	0
AVERAGING	Mean value	10	1
PT1	Smoothing with filter; adjustment using time constant	11	2

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.6 EnumAxisFineInterpolatorMode

EnumAxisFineInterpolatorMode

Values	Significance	ValueDint	S7T Config
DIRECT_MODE	No interpolation	12	0
LINEAR_MODE	Linear interpolation	13	1
CUBIC_MODE	Interpolation at constant velocity	14	2

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.7 EnumAxisHomingMode

EnumAxisHomingMode

Values	Significance	ValueDint	S7T Config
MODE_CAM_AND_ZM	Reference cam and encoder zero mark	22	1
MODE_ZM	Encoder zero mark only	23	2
MODE_CAM	External zero mark only	24	3
MODE_NO_REFERENCE	No mode	88	0

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.8 EnumAxisPassiveApproachDirection

EnumAxisPassiveApproachDirection

Values	Significance	ValueDint	S7T Config
APPROACH_NEGATIVE_ PASSIVE	Approach to encoder zero mark in negative direction	15	0
APPROACH_POSITIVE_ PASSIVE	Approach to encoder zero mark in positive direction	16	1
EDGE_POS_SIDE_ NEG_PASSIVE	Positive edge at the negative side of the external zero mark	17	2
EDGE_POS_SIDE_ POS_PASSIVE	Positive edge at the positive side of the external zero mark	18	3
EDGE_NEG_SIDE_ POS_PASSIVE	Negative edge at the positive side of the external zero mark	19	4
EDGE_NEG_SIDE_ NEG_PASSIVE	Negative edge at the negative side of the external zero mark	20	5
ACTUAL_DIRECTION_PASSIVE	Next signal edge	21	6

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.9 EnumAxisPassiveHomingMode

EnumAxisPassiveHomingMode

Values	Significance	ValueDint	S7T Config
CAM_AND_ZM_PASSIVE	Reference cam and encoder zero mark	22	1
ZM_PASSIVE	Encoder zero mark only	23	2
CAM_PASSIVE	External zero mark only	24	3
DEFAULT_PASSIVE	Default depends on encoder type.With encoder zero mark: ZM_PASSIVE without encoder zero mark: CAM_PASSIVE	25	4

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.10 EnumAxisTorqueForceReductionGranularity

EnumAxisTorqueForceReductionGranularity

Values	Description	ValueDint	S7T Config
BASIC	Resolution 1/100	1	16
STANDARD	Resolution 1/16384	0	0

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.11 EnumBackLashDiff

EnumBackLashDiff

Values	Significance	ValueDint	S7T Config
DIFF_POSITIVE	positive	26	0
DIFF_NEGATIVE	negative	27	1

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.12 EnumBackLashType

EnumBackLashType

Values	Significance	ValueDint	S7T Config
NEGATIVE	negative	4	1
POSITIVE	positive	5	0

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.13 EnumBalanceFilterMode

EnumBalanceFilterMode

Values	Significance	ValueDint	S7T Config
OFF	Balancing filter not active	89	0
MODE_1	Balancing filter active	90	1
MODE_2	Extended balancing filter active	91	2

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.14 EnumCamInterpolationMode

EnumCamInterpolationMode

Values	Significance	ValueDint	S7T Config
B_SPLINE	Approximation using Bezier splines	28	25
C_SPLINE	Interpolation using cubic splines	29	38
LINEAR	Linear interpolation	30	72

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.15 EnumCammingDirection

EnumCammingDirection

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last direction set in the program	3	45
NEGATIVE	inverse	4	85
POSITIVE	equidirectional	5	107

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.16 EnumCammingMode

EnumCammingMode

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
CYCLIC	Cyclic cam disk	31	35
NOCYCLIC	Non-cyclic cam disk	32	92

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.17 EnumCamMode

EnumCamMode

Values	Significance	ValueDint	S7T Config
CYCLIC_ABSOLUTE	Cyclic absolute cam disk	35	36
CYCLIC_RELATIVE	Cyclic relative cam disk	36	37
NO_CONSTRAINTS	Non-cyclic cam disk	37	96

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.18 EnumCamPositionMode

EnumCamPositionMode

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
ACTUAL	Notation with scaling and offset	33	7
BASIC	Notation without scaling and offset	34	16

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.19 EnumCamTrackType

EnumCamTrackType

Values	Description	ValueDint	S7T Config
TYPE_WAY	Position-based cam	60	0
TYPE_TIME	Time-based cam	61	1
TYPE_TIME_WITH_MAX_LENGTH	Time-based output cam with maximum ON length	64	4

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.20 EnumChangeMode

EnumChangeMode

Values	Significance	ValueDint	S7T Config
NEVER	No changeover	92	0
IN_POSITION	Changeover when axis in positioning window	93	1
IN_STANDSTILL	Changeover when axis below standstill velocity	94	3
IMMEDIATELY	Immediate changeover	95	7

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.21 EnumDirection

EnumDirection

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last direction set in the program	3	45
NEGATIVE	negative	4	85
POSITIVE	positive	5	107
BY_VALUE	derived from sign of velocity setpoint	38	24
SHORTEST_WAY	Shortest path	39	121

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.22 EnumDirectionType

EnumDirectionType

Values	Significance	ValueDint	S7T Config
NEGATIVE	Negative direction	4	1
POSITIVE	Positive direction	5	0

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.23 EnumErrorReporting

EnumErrorReporting

Values	Significance	ValueDint	S7T Config
NO_REPORTING	No	40	0
COMMAND_VALUE_ TOLERANCE	Setpoints	41	1
ACTUAL_VALUE_TOLERANCE	Actual values	42	2

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.24 EnumFollowingObjectSynchronizeWithLookAhead

EnumFollowingObjectSynchronizeWithLookAhead

Values	Description	ValueDint	S7T Config
STANDARD	Look ahead with s and v	112	358
EXTENDED_LOOK_AHEAD	Look ahead with s, v and a	113	359

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.25 EnumFollowingObjectSynchronizingDirection

EnumFollowingObjectSynchronizingDirection

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
SHORTEST_WAY	Shortest path without direction preset	96	121
SYSTEM_DEFINED	Compatibility mode	97	316
SAME_DIRECTION	Same direction as master	98	317
POSITIVE_DIRECTION	Positive synchronization direction	99	319
NEGATIVE_DIRECTION	Negative synchronization direction	100	320

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.26 EnumForceDirection

EnumForceDirection

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last effective direction set in the program	3	45
NEGATIVE	Negative effective direction	4	85
POSITIVE	Positive effective direction	5	107
BOTH	Positive and negative effective direction	43	20

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.27 EnumGearingDirection

EnumGearingDirection

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last direction set in the program	3	45
NEGATIVE	inverse	4	85
POSITIVE	equidirectional	5	107
BY_VALUE	Sign	38	24
CURRENT	Current slave direction	44	33
REVERSE	inverse to current slave direction	45	116

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.28 EnumGearingMode

EnumGearingMode

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed type	3	45
GEARING_WITH_FRACTION	Gear ratio as a function of the numerator to denominator ratio	46	55
GEARING_WITH_RATIO	Gear ratio as floating point number	47	56

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.29 EnumGearingPosToleranceCommandValue

EnumGearingPosToleranceCommandValue

Values	Significance	ValueDint	S7T Config
NO_ACTIVATE	No	48	0
WITHOUT_JERK	Without jerk	49	1
WITH_JERK	With jerk	50	2

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.30 EnumGearingType

EnumGearingType

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed type	3	45
ABSOLUTE	Absolute gearing	6	1
RELATIVE	Relative gearbox	7	115

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.31 EnumLogicOperation

EnumLogicOperation

Values	Significance	ValueDint	S7T Config
OPERATION_OR	Logical OR	51	0
OPERATION_AND	Logical AND	52	1

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.32 EnumLimitExceededOk

EnumLimitExceededOk

Values	Significance	ValueDint	S7T Config
LIMIT_EXCEEDED		101	71
O_K_	Within range	102	102

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.33 EnumMasterMode

EnumMasterMode

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed master mode	3	45
ABSOLUTE	Absolute reference to master	6	1
RELATIVE	Relative reference to master	7	115

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.34 EnumMeasuredEdge

EnumMeasuredEdge

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
FALLING_EDGE	Negative edge (high > low)	53	86
RISING_EDGE	Positive edge (low > high)	54	117
BOTH_EDGES	Measurement at both edge signals	55	224
BOTH_EDGES_FIRST_RISING	Measurement at both edges, starting with a positive edge (low to high)	56	225
BOTH_EDGES_FIRST_FALLING	Measurement at both edges, starting with a negative edge (high to low)	57	226

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.35 EnumMeasuringRangeMode

EnumMeasuringRangeMode

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
WITHOUT_SPECIFIC_AREA	Measurement without specified range	58	168
WITH_SPECIFIC_AREA	Measurement in specified area	59	170

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.36 EnumMountSwitch

EnumMountSwitch

Values	Significance	ValueDint	S7T Config
END_MOUNTED_SWITCH	Hardware limit switches outside the permissible traversing range are always active	103	0
FLEXIBLE_MOUNTED_SWITCH	Limit switch can be passed by the axis	104	1

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.37 EnumOutputCamType

EnumOutputCamType

Values	Significance	ValueDint	S7T Config
TYPE_REVERSE	Reversing output cam	63	3
TYPE_SWITCH	Uni-directional output cam	62	2
TYPE_TIME	Time-based cam	61	1
TYPE_WAY	Position-based cam	60	0

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.38 EnumProfile

EnumProfile

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed velocity profile	3	45
PARABOLIC	Parabolic velocity profile / not available	64	103
SINUSOIDAL	Sinusoidal velocity profile / not available	65	122
SMOOTH	Smooth acceleration curve	66	124
TRAPEZOIDAL	Trapezoidal velocity profile	67	146

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.39 EnumRecognitionMode

EnumRecognitionMode

Values	Significance	ValueDint	S7T Config
DO_NOT_CLAMP	Not detected	105	0
CLAMP_BY_FOLLOWING_ ERROR_DEVIATION	When following error is exceeded	106	1
CLAMP_WHEN_TORQUE_ LIMIT_REACHED	When torque is exceeded / not available (or to be verified)	107	2

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.40 EnumSensorState

EnumSensorState

Values	Significance	ValueDint	S7T Config
NOT_VALID	Invalid	204	109
WAIT_FOR_VALID	Wait for validation	205	110
VALID	Values are valid	206	111

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.41 EnumSlaveMode

EnumSlaveMode

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed slave mode	3	45
ABSOLUTE	Absolute reference to slave	6	1
RELATIVE	Relative reference to slave	7	115

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.42 EnumSyncModeCamming

EnumSyncModeCamming

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
IMMEDIATELY	Effective immediately	8	60
AT_THE_END_OF_CAM_CYCLE	Transition at the end of the active cam	68	14
NEXT_WITH_ REFERENCE	At the next leading axis position / not available	69	90
ON_MASTER_AND_SLAVE_ POSITION	Default synchronization position of the leading axis and following axis	70	99
ON_MASTER_POSITION	Default synchronization position of the leading axis	71	100
IMMEDIATELY_AND_SLAVE_ POSITION	Effective immediately and synchronous position of the slave axis	108	315

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.43 EnumSyncModeGearing

EnumSyncModeGearing

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
IMMEDIATELY	Effective immediately	8	60
NEXT_WITH_ REFERENCE	At the next leading axis position / not available	69	90
ON_MASTER_AND_SLAVE_ POSITION	Default synchronization position of the leading axis and following axis	70	99
ON_MASTER_POSITION	Default synchronization position of the leading axis	71	100
ON_SLAVE_POSITION	Specified by the synchronization position of the slave axis	72	101
IMMEDIATELY_AND_SLAVE_ POSITION	Effective immediately and synchronous position of the slave axis	108	315
IMMEDIATELY_AND_BE_ SYNCHRONOUS_AT_ MASTER_POSITION	Immediate absolute synchronization with time base/not available	74	370

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.44 EnumSyncOffModeCamming

EnumSyncOffModeCamming

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
IMMEDIATELY	Effective immediately	8	60
AT_THE_END_OF_CAM_CYCLE	End of cam cycle	68	14
ON_MASTER_POSITION	At position of the leading axis	71	100
ON_SLAVE_POSITION	At position of the following axis	72	101

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.45 EnumSyncOffModeGearing

EnumSyncOffModeGearing

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
IMMEDIATELY	Effective immediately	8	60
ON_MASTER_POSITION	Specified by the desynchronization position of the leading axis	71	100
ON_SLAVE_POSITION	Default desynchronization position of the following axis	72	101

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.46 EnumSyncOffPositionReference

EnumSyncOffPositionReference

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
AXIS_STOPPED_AT_ POSITION	Stop before desynchronization position	73	15
BEGIN_TO_STOP_WHEN_ POSITION_ REACHED	Stop after desynchronization position	74	18
STOP_SYMMETRIC_WITH_ POSITION	Stop symmetrically to desynchronization position	75	137

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.47 EnumSyncPositionReference

EnumSyncPositionReference

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
BE_SYNCHRONOUS_AT_ POSITION	Synchronize before synchronization position	76	19
SYNCHRONIZE_SYMMETRIC	Synchronize symmetrically to synchronization position	77	143
SYNCHRONIZE_WHEN_ POSITION _REACHED	Synchronize from synchronization position	78	144

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.48 EnumSyncProfileReference

EnumSyncProfileReference

Values	Significance	ValueDint	S7T Config
USER_DEFAULT	User default setting / standard	2	149
EFFECTIVE	Last programmed setting	3	45
RELATE_SYNC_PROFILE_ TO_ LEADING_VALUE	Synchronization profile specific to the leading axis	79	113
RELATE_SYNC_PROFILE_ TO_TIME	Time-related synchronization profile	80	114

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.49 EnumToActivationModeSetConfigData

EnumToActivationModeSetConfigData

Values	Significance	ValueDint	S7T Config
ACTIVATE_CHANGED_ CONFIG_DATA	Activate the configuration data immediately	81	291
COLLECT_CHANGED_ CONFIG_ DATA	Collect the configuration data, do not activate	82	292

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.4.12.50 EnumYesNo

EnumYesNo

Values	Significance	ValueDint	S7T Config
NO	Activation: No	0	91
YES	Activation: Yes	1	173

The "ValueDint" column shows the numerical value for the input and output parameters of the "MC_WriteParameter" and "MC_ReadSysParameter" technology functions. The "S7T Config" column contains the corresponding numerical value from the Expert list of S7T Config.

9.5 Additional information on the Internet

9.5.1 Additional information on the Internet

If your PG/PC provides Internet access functions you can download supplementary information from the Internet.

Additional information about the Technology CPU is also available at the Internet URL shown below, if your PG/PC is capable of connecting to the Internet:

Your Siemens contact partners

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

Service & Support

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

Technical Support for all A&D products

http://www.siemens.de/automation/support-request

SIMATIC documentation

http://www.ad.siemens.de/support

Training Center

http://www.sitrain.com

FAQs

http://support.automation.siemens.com

Index

A

B

C

D

E

F

G

H

N

O

P

Q

R

S

T

U

V

W

Z