SIEMENS

SIMATIC

STEP 7
S7-1200 Motion Control V6.0 in TIA Portal V15

Function Manual

Preface

Introduction 1

Basics for working with S7-1200 Motion Control 2

Guidelines on use of motion control 3

Using versions 4

Positioning axis technology object 5

Technology object command table 6

Download to CPU 7

Commissioning 8

Programming 9

Axis - Diagnostics 10

Instructions 11

Appendix 12

12/2017
A5E03790551-AF

Legal information

Warning notice system

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

⚠ DANGER

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

⚠ WARNING

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

⚠ CAUTION

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

NOTICE

indicates that property damage can result if proper precautions are not taken.

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

⚠ WARNING

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E03790551-AF
℗ 11/2017 Subject to change

Preface

Purpose of this manual

This document provides you with detailed information on S7-1200 Motion Control. The contents of this document correspond to the STEP 7 V15 online help with respect to contents and structure. Interaction with STEP 7 is required to understand many parts of this document.

This document is aimed at programmers of STEP 7 programs and at people who work in the areas of configuring, commissioning and servicing automation systems with motion control applications.

Required basic knowledge

General knowledge in the fields of automation engineering and motion control is required to understand this document.

It is also essential to be familiar with the use of computers or programming devices under the Windows operating system.

Because S7-1200 Motion Control is based on STEP 7, you need knowledge of working with STEP 7.

Scope of this manual

This manual is valid for STEP 7 V15.

Conventions

This documentation contains pictures of the devices described. The pictures may differ in minor details from the devices supplied.

Please also observe notes labeled as follows:

Note

A note contains important information on the product described in the documentation, on the handling of the product or on the section of the documentation to which particular attention should be paid.

Further support

If you have any questions relating to the products described in this manual, and do not find the answers in this documentation, please contact your Siemens partner at our local offices.

You will find your contact person on the Internet (http://www.automation.siemens.com/mcms/aspa-db/en/automation-technology/Pages/default.aspx).

A guide to the technical documentation for the various SIMATIC products and systems is available on the Internet (http://www.siemens.com/simatic-tech-doku-portal).

The online catalog and online ordering system are available on the Internet (https://mall.industry.siemens.com).

Training center

We offer a range of courses to help get you started with the S7 programmable controller. Please contact your regional training center or the central Training center (http://support.automation.siemens.com/WW/view/en/24486113).

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit (https://www.siemens.com/industrialsecurity).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under (https://www.siemens.com/industrialsecurity).

Table of contents

Preface **3**

1 Introduction **9**

1.1 Motion functionality of the CPU S7-1200 9

1.2 Hardware components for motion control 10

2 Basics for working with S7-1200 Motion Control **13**

2.1 Stepper motor on the PTO 13
2.1.1 CPU outputs relevant for motion control 13
2.1.2 How the pulse interface works 16
2.1.3 Relationship between the signal type and the direction of travel 17

2.2 PROFIdrive drive/analog drive connection 21
2.2.1 Drive and encoder connection 21
2.2.2 Automatic transfer of drive and encoder parameters in the device 22
2.2.3 PROFIdrive 24
2.2.4 Closed loop control 25
2.2.5 Data connection PROFIdrive drive/PROFIdrive encoder 26
2.2.6 Data connection drive with analog drive connection 31
2.2.7 Process response 33
2.2.7.1 Organization Blocks for Motion Control 33
2.2.7.2 Process image partition "PIP OB Servo" 34
2.2.7.3 Operational Sequence and Timeouts 35
2.2.7.4 Operating modes 36

2.3 Hardware and software limit switches 38

2.4 Jerk limit 39

2.5 Homing 40

3 Guidelines on use of motion control **42**

4 Using versions **43**

4.1 Overview of versions 43

4.2 Changing a technology version 47

4.3 Compatibility list of variables V1...3 <-> V4...5 48

4.4 Compatibility list of variables V4...5 <-> V6 51

4.5 Status of limit switch 52

5 Positioning axis technology object **53**

5.1 Integration of the positioning axis technology object 53

5.2 Tools of the positioning axis technology object 56

5.3 Adding a positioning axis technology object 58

5.4 Configuring the positioning axis technology object ... 59
5.4.1 Working with the configuration dialog ... 59
5.4.2 Monitor values ... 60
5.4.3 Basic parameters ... 61
5.4.3.1 Configuration - General ... 61
5.4.3.2 Configuration - Drive ... 63
5.4.3.3 Configuration - Encoder ... 68
5.4.4 Extended parameters ... 73
5.4.4.1 Mechanics ... 73
5.4.4.2 Configuration - Modulo (PROFIdrive/analog drive connection only) ... 75
5.4.4.3 Position limits ... 76
5.4.4.4 Dynamics ... 81
5.4.4.5 Homing (positioning axis technology object as of V2) ... 88
5.4.4.6 Positioning monitoring ... 96
5.4.4.7 Configuration - Control loop (PROFIdrive and analog drive connection only) ... 98
5.4.5 Parameter view ... 99
5.4.5.1 Introduction to the parameter view ... 99
5.4.5.2 Structure of the parameter view ... 101
5.4.5.3 Opening the parameter view ... 104
5.4.5.4 Default setting of the parameter view ... 104
5.4.5.5 Working with the parameter view ... 107
5.4.6 Configuring technology modules for Motion Control ... 119
5.4.6.1 Overview ... 119
5.4.6.2 TM PosInput 1 / TM PosInput 2 ... 120
5.4.6.3 TM Count 1x24V / TM Count 2x24V ... 122

6 Technology object command table ... 123

6.1 Use of the command table technology object ... 123
6.2 Command table technology object tools ... 123
6.3 Adding the technological object command table ... 124
6.4 Configuring the command table technology object ... 125
6.4.1 Working with the configuration dialog ... 125
6.4.2 Monitor values ... 126
6.4.3 Basic parameters ... 127
6.4.3.1 Configuration - General ... 127
6.4.3.2 Configuration - Command table ... 127
6.4.3.3 Shortcut menu commands - Command table ... 131
6.4.3.4 Working with the trend diagram ... 133
6.4.3.5 Shortcut menu commands - Curve chart ... 137
6.4.3.6 Transition from "Complete command" to "Blend motion" ... 138
6.4.3.7 Changing the command table configuration in the user program ... 140
6.4.4 Extended parameters ... 141
6.4.4.1 Configuration - Extended parameters ... 141
6.4.4.2 Configuration - Dynamics ... 142
6.4.4.3 Configuration - Limit values ... 143

7 Download to CPU ... 144

8 Commissioning ... 146

8.1 Axis control panel ... 146
8.2 Tuning ... 150

9 Programming 154

9.1 Overview of the Motion Control statements 154
9.2 Creating a user program 155
9.3 Programming notes 158
9.4 Behavior of the Motion Control commands after POWER OFF and restart 160
9.5 Monitoring active commands 161
9.5.1 Monitoring active commands 161
9.5.2 Motion control instructions with "Done" output parameter 161
9.5.3 Motion control instruction MC_MoveVelocity 165
9.5.4 Motion control instruction MC_MoveJog 168
9.6 Error displays of the Motion Control statements 172
9.7 Restart of technology objects 173
9.8 Parameter transfer for function blocks 174

10 Axis - Diagnostics 176

10.1 Status and error bits (technology objects as of V4) 176
10.2 Motion status 179
10.3 Dynamics settings 180
10.4 PROFIdrive frame 180

11 Instructions 181

11.1 S7-1200 Motion Control as of V6 181
11.1.1 MC_Power 181
11.1.1.1 MC_Power: Enable, disable axis as of V6 181
11.1.1.2 MC_Power: Function chart as of V6 185
11.1.2 MC_Reset 186
11.1.2.1 MC_Reset: Acknowledge fault, restart technology object as of V6 186
11.1.3 MC_Home 188
11.1.3.1 MC_Home: Home axes, set reference point as of V6 188
11.1.4 MC_Halt 193
11.1.4.1 MC_Halt: Stop axis as of V6 193
11.1.4.2 MC_Halt: Function chart as of V6 195
11.1.5 MC_MoveAbsolute 196
11.1.5.1 MC_MoveAbsolute: Absolute positioning of axis as of V6 196
11.1.5.2 MC_MoveAbsolute: Function chart as of V6 199
11.1.6 MC_MoveRelative 200
11.1.6.1 MC_MoveRelative: Relative positioning of axis as of V6 200
11.1.6.2 MC_MoveRelative: Function chart as of V6 203
11.1.7 MC_MoveVelocity 204
11.1.7.1 MC_MoveVelocity: Move axis at set velocity as of V6 204
11.1.7.2 MC_MoveVelocity: Function chart as of V6 207
11.1.8 MC_MoveJog 208
11.1.8.1 MC_MoveJog: Move axis in jog mode as of V6 208
11.1.8.2 MC_MoveJog: Function chart as of V6 210
11.1.9 MC_CommandTable 211
11.1.9.1 MC_CommandTable: Run axis commands as motion sequence as of V6 211

11.1.10 MC_ChangeDynamic ... 213
11.1.10.1 MC_ChangeDynamic: Change dynamic settings of axis as of V6 ... 213
11.1.11 MC_ReadParam ... 216
11.1.11.1 MC_ReadParam: Continuously read motion data of a positioning axis as of V6 ... 216
11.1.12 MC_WriteParam ... 218
11.1.12.1 MC_WriteParam: Write tag of positioning axis as of V6 ... 218

12 Appendix ... 220

12.1 Using multiple axes with the same PTO ... 220
12.2 Using multiple drives with the same PTO ... 223
12.3 Tracking jobs from higher priority classes (execution levels) ... 224
12.4 Special cases when using software limit switches for drive connection via PTO ... 227
12.4.1 Software limit switches in conjunction with a homing operation ... 227
12.4.2 Software limit switches and software limit switch position changes. ... 231
12.4.3 Software limit switches in conjunction with dynamic changes ... 232
12.5 Reducing velocity for a short positioning duration ... 234
12.6 Dynamic adjustment of start/stop velocity ... 234
12.7 Move the axis without position control for servicing ... 235
12.8 List of ErrorIDs and ErrorInfos (technology objects as of V6) ... 236
12.9 Tags of the positioning axis technology object as of V6 ... 259
12.9.1 Legend ... 259
12.9.2 Tags for position values and velocity values as of V6 ... 260
12.9.3 Simulation tags as of V6 ... 261
12.9.4 Actuator tags as of V6 ... 262
12.9.5 Sensor[1] tags as of V6 ... 266
12.9.6 Units tag as of V6 ... 271
12.9.7 Mechanics tag as of V6 ... 272
12.9.8 Modulo tags as of V6 ... 272
12.9.9 DynamicLimits tags as of V6 ... 273
12.9.10 DynamicDefaults tags as of V6 ... 274
12.9.11 PositionLimits_SW variables as of V6 ... 275
12.9.12 PositionLimits_HW variables as of V6 ... 276
12.9.13 Homing tags as of V6 ... 277
12.9.14 PositionControl tag as of V6 ... 278
12.9.15 FollowingError tags as of V6 ... 279
12.9.16 PositionMonitoring tags as of V6 ... 280
12.9.17 StandstillSignal tags as of V6 ... 280
12.9.18 StatusPositioning tags as of V6 ... 281
12.9.19 StatusDrive tags as of V6 ... 282
12.9.20 StatusSensor tags as of V6 ... 283
12.9.21 StatusBits tags as of V6 ... 284
12.9.22 ErrorBits tags as of V6 ... 288
12.9.23 ControlPanel tags as of V6 ... 289
12.9.24 Internal tags as of V6 ... 290
12.9.25 Update of the technology object tags ... 291
12.10 Tags of the command table V6 technology object ... 291

Index ... 293

Introduction

1

1.1 Motion functionality of the CPU S7-1200

The TIA Portal, together with the motion control functionality of the CPU S7-1200, supports you in controlling stepper motors and servo motors:

- You configure the positioning axis and command table technology objects in the TIA Portal. The CPU S7-1200 uses these technology objects to control the outputs that control the drives.
- In the user program you control the axis by means of Motion Control instructions and initiate motion commands of your drive.

See also

http://www.automation.siemens.com/mcms/topics/en/simatic/simatic-technology/integrated-functions/simatic-s7-1200/Pages/Default.aspx (http://www.automation.siemens.com/mcms/topics/en/simatic/simatic-technology/integrated-functions/simatic-s7-1200/Pages/Default.aspx)

Hardware components for motion control (Page 10)

Integration of the positioning axis technology object (Page 53)

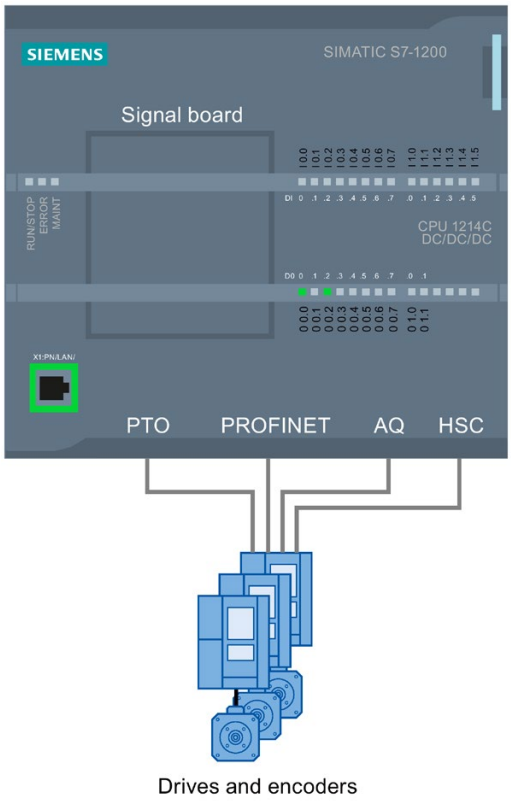
Tools of the positioning axis technology object (Page 56)

Use of the command table technology object (Page 123)

Command table technology object tools (Page 123)

1.2 Hardware components for motion control

The representation below shows the basic hardware configuration for a motion control application with the CPU S7-1200.



CPU S7-1200

CPU S7-1200 combines the functionality of a programmable logic controller with motion control functionality for operation of drives. The motion control functionality takes over the control and monitoring of the drives.

Signal board

You add further inputs and outputs to the CPU with the signal boards.

You can use the digital outputs as pulse generator outputs for controlling drives as required. In CPUs with relay outputs, the pulse signal cannot be output on the onboard outputs because the relays do not support the necessary switching frequencies. To be able to work with the PTO (Pulse Train Output) on these CPUs, you must use a signal board with digital outputs.

You can use the analog outputs for controlling connected analog drives as required.

PROFINET

Use the PROFINET interface to establish the online connection between the CPU S7-1200 and the programming device. In addition to the online functions of the CPU, additional commissioning and diagnostic functions are available for motion control.

PROFINET continues to support the PROFIdrive profile for connecting PROFIdrive capable drives and encoders.

Drives and encoders

Drives permit the movement of the axis. Encoders provide the actual position for the closed loop position control of the axis.

The table below shows the connection possibilities for drives and encoders:

Drive connection	Closed/open loop control of axis	Encoder connection
PTO (Pulse Train Output) (Stepper motors and servo motors with pulse interface)	Position-controlled	-
Analog output (AQ)	Position-controlled	• Encoder on high-speed counter (HSC) • Encoder on technology module (TM) • Encoder on PROFINET
PROFINET	Position-controlled	• Encoder on the drive • Encoder on high-speed counter (HSC) • Encoder on technology module (TM) • Encoder on PROFINET

Name	Article number
CPU 1211C DC/DC/DC	6ES7211-1AE40-0XB0
CPU 1211C AC/DC/RLY	6ES7211-1BE40-0XB0
CPU 1211C DC/DC/RLY	6ES7211-1HE40-0XB0
CPU 1212C DC/DC/DC	6ES7212-1AE40-0XB0
CPU 1212C AC/DC/RLY	6ES7212-1BE40-0XB0
CPU 1212C DC/DC/RLY	6ES7212-1HE40-0XB0
CPU 1214C DC/DC/DC	6ES7214-1AG40-0XB0
CPU 1214C AC/DC/RLY	6ES7214-1BG40-0XB0
CPU 1214C DC/DC/RLY	6ES7214-1HG40-0XB0
CPU 1214FC DC/DC/DC	6ES7214-1AF40-0XB0
CPU 1214FC DC/DC/RLY	6ES7214-1HF40-0XB0
CPU 1215C DC/DC/DC	6ES7215-1AG40-0XB0
CPU 1215C AC/DC/RLY	6ES7215-1BG40-0XB0
CPU 1215C DC/DC/RLY	6ES7215-1HG40-0XB0
CPU 1215FC DC/DC/DC	6ES7215-1AF40-0XB0
CPU 1215FC DC/DC/RLY	6ES7215-1HF40-0XB0

Name	Article number
CPU 1217C DC/DC/DC	6ES7217-1AG40-0XB0
Signal board DI4 x DC 24 V (200 kHz)	6ES7221-3BD30-0XB0
Signal board DI4 x DC 5 V (200 kHz)	6ES7 221-3AD30-0XB0
Signal board DQ4 x DC 24 V (200 kHz)	6ES7222-1BD30-0XB0
Signal board DQ4 x DC 5 V (200 kHz)	6ES7222-1AD30-0XB0
Signal board DI2/DQ2 x DC 24 V (20 kHz)	6ES7223-0BD30-0XB0
Signal board DI2/DQ2 x DC 24 V (200 kHz)	6ES7223-3BD30-0XB0
Signal board DI2/DQ2 x DC 5 V (200 kHz)	6ES7223-3AD30-0XB0
Signal board AQ1 x 12 bit (±10 V, 0 to 20 mA)	6ES7 232-4HA30-0XB0

Ordering information for CPU firmware V4.1

The order information listed below applies to the currently installed product phase (without any installed Hardware Support Packages) of the TIA Portal.

Use a Hardware Support Package (HSP) to install new hardware components. The hardware component will then be available in the hardware catalog.

See also

Motion functionality of the CPU S7-1200 (Page 9)

CPU outputs relevant for motion control (Page 13)

Basics for working with S7-1200 Motion Control 2

2.1 Stepper motor on the PTO

2.1.1 CPU outputs relevant for motion control

The number of usable drives depends on the number of PTOs (pulse train outputs) and the number of available pulse generator outputs.

The following tables provide information about the relevant dependencies:

Maximum number of PTOs

4 PTOs are available for each CPU with technology version V4. This means a maximum of 4 drives can be controlled.

Signal type of the PTO

Depending on the signal type of the PTO, 1-2 pulse generator outputs are required per PTO (drive):

Signal type	Number of pulse generator outputs
Pulse A and direction B (direction output disabled)[1]	1
Pulse A and direction B [1]	2
Clock up A and clock down B	2
A/B phase-shifted	2
A/B phase-shifted - quadruple	2

[1] The direction output must be on-board or on a signal board.

Usable pulse generator outputs and limit frequencies

The relay variants of the CPUs can only access the pulse generator outputs of a signal board.

Depending on the CPU the pulse generator outputs Q0.0 to Q1.1 can be used with the following limit frequencies:

CPU	Q0.0	Q0.1	Q0.2	Q0.3	Q0.4	Q0.5	Q0.6	Q0.7	Q1.0	Q1.1
1211 (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	-	-	-	-	-	-
1212 (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	20 kHz	20 kHz	-	-	-	-
1214(F) (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	20 kHz	20 kHz	20 kHz	20 kHz	20 kHz	20 kHz
1215(F) (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	20 kHz	20 kHz	20 kHz	20 kHz	20 kHz	20 kHz
1217 (DC/DC/DC)	1 MHz	1 MHz	1 MHz	1 MHz	100 kHz	100 kHz	100 kHz	100 kHz	100 kHz	100 kHz

Depending on the signal board, the pulse generator outputs Qx.0 to Qx.3 can be used with the following limit frequencies:

Signal board	Qx.0	Qx.1	Qx.2	Qx.3	-	-	-	-	-	-
DI2/DQ2 x DC24V 20kHz	20 kHz	20 kHz	-	-	-	-	-	-	-	-
DI2/DQ2 x DC24V 200kHz	200 kHz	200 kHz	-	-	-	-	-	-	-	-
DQ4 x DC24V 200kHz	200 kHz	200 kHz	200 kHz	200 kHz	-	-	-	-	-	-
DI2/DQ2 x DC5V 200kHz	200 kHz	200 kHz	-	-	-	-	-	-	-	-
DQ4 x DC5V 200kHz	200 kHz	200 kHz	200 kHz	200 kHz	-	-	-	-	-	-

The low limit frequency is always 1Hz.

The pulse generator outputs can be freely assigned to the PTOs.

Note

If pulse generator outputs with different limit frequencies are used in accordance with the signal type, the low limit frequency is used in each case.

Signal type "Pulse A and direction B" is an exception. With this type of signal, the limit frequency of the pulse generator output is always used.

Note

Access to pulse generator outputs via the process image

The firmware takes control via the corresponding pulse generator and direction outputs if the PTO (Pulse Train Output) is selected and assigned to an axis.

With this takeover of the control function, the connection between the process image and I/O output is also disconnected. Although the user has the option of writing the process image of pulse generator and direction outputs via the user program or watch table, this is not transferred to the I/O output. Accordingly, it is also not possible to monitor the I/O output via the user program or watch table. The information read reflects the value of the process image and does not match the real status of the I/O output.

For all other CPU outputs that are not used permanently by the CPU firmware, the status of the I/O output can be controlled or monitored via the process image, as usual.

Outputs for drive signals

For motion control, you can optionally parameterize a drive interface for "Drive enabled" and "Drive ready".

When using the drive interface the digital output for the drive enable and the digital input for "drive ready" can be freely selected.

Acceleration/deceleration limits

The following limits apply to acceleration and deceleration:

Acceleration/deceleration	Value
Minimum acceleration/deceleration	5.0E-3 pulses/s^2
Maximum acceleration/deceleration	9.5E+9 pulses/s^2

Jerk limits

The following limits apply to the jerk:

Jerk	Value
Minimum jerk	4.0E-3 pulses/s^3
Maximum jerk	1.0E+10 pulses/s^3

See also

How the pulse interface works (Page 16)

Relationship between the signal type and the direction of travel (Page 17)

Hardware and software limit switches (Page 38)

Jerk limit (Page 39)

Homing (Page 40)

Hardware components for motion control (Page 10)

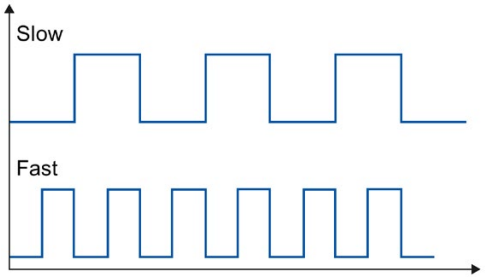
Integration of the positioning axis technology object (Page 53)

Tools of the positioning axis technology object (Page 56)

2.1.2 How the pulse interface works

Depending on the settings of the stepper motor, each pulse affects the movement of the stepper motor by a specific angle. If the stepper motor is set to 1000 pulses per revolution, for example, it moves 0.36° per pulse.

The speed of the stepper motor is determined by the number of pulses per time unit.



The statements made here also apply to servo motors with pulse interface.

See also

CPU outputs relevant for motion control (Page 13)

Relationship between the signal type and the direction of travel (Page 17)

Hardware and software limit switches (Page 38)

Jerk limit (Page 39)

Homing (Page 40)

Integration of the positioning axis technology object (Page 53)

Tools of the positioning axis technology object (Page 56)

2.1.3 Relationship between the signal type and the direction of travel

The CPU outputs the velocity and direction of travel via two outputs.

The relationships between the configuration and direction of travel differ depending on the selected signal type. You can configure the following signal types in the axis configuration under "Basic parameters > General":

- "PTO – pulse A and direction B"
- "PTO – clock up A and clock down B" (as of V4)
- "PTO – A/B phase-shifted" (as of V4)
- "PTO – A/B phase-shifted, quadruple" (as of V4)

You configure the direction under "Extended Parameters > Mechanics" in the axis configuration. If you select the "Invert direction" option, the direction logic described below for the respective signal type is inverted.

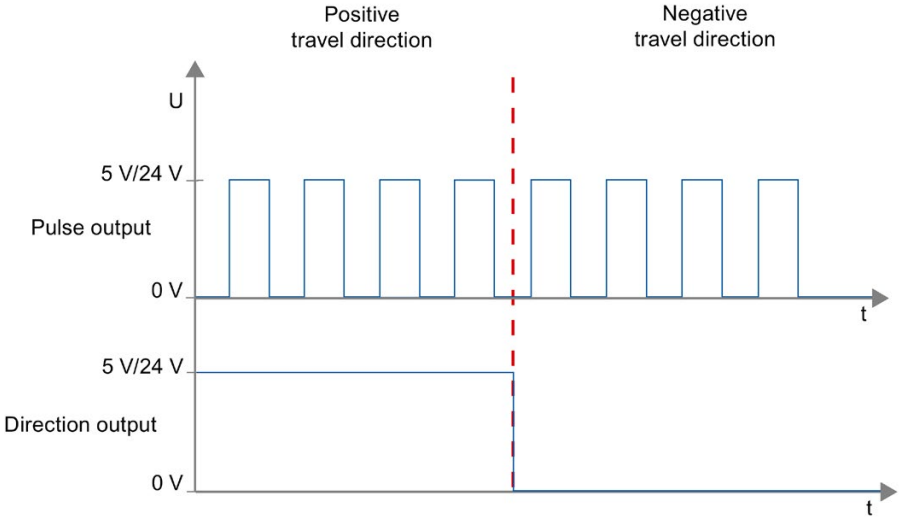
PTO – pulse (A) and direction (B)

The pulse output pulses and the direction output level are evaluated for this signal type.

The pulses are output via the pulse output of the CPU. The direction output of the CPU specifies the direction of rotation of the drive:

- 5 V/24 V at direction output ⇒ positive direction of rotation
- 0 V at direction output ⇒ negative direction of rotation

The specified voltage depends on the hardware used. The indicated values do not apply to the differential outputs of CPU 1217.

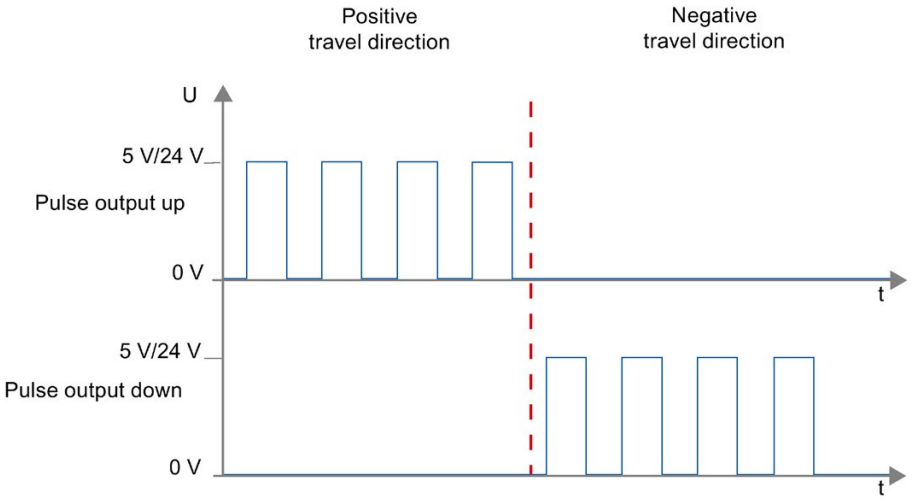


PTO – clock up A and clock down B (as of V4)

The pulses of one output are evaluated for this signal type.

The pulse for the positive direction is output via the "Pulse output up" The pulse for the negative direction is output via the "Pulse output down"

The specified voltage depends on the hardware used. The indicated values do not apply to the differential outputs of CPU 1217.



PTO – A/B phase-shifted (as of V4)

The positive edges of one output in each case are evaluated for this signal type.

The pulse is output via the "Signal A" output and phase-shifted via the "Signal B" output. The phase shifting between the outputs defines the direction of rotation:

- Signal A leads signal B by 90° ⇒ positive direction of rotation
- Signal B leads signal A by 90° ⇒ negative direction of rotation

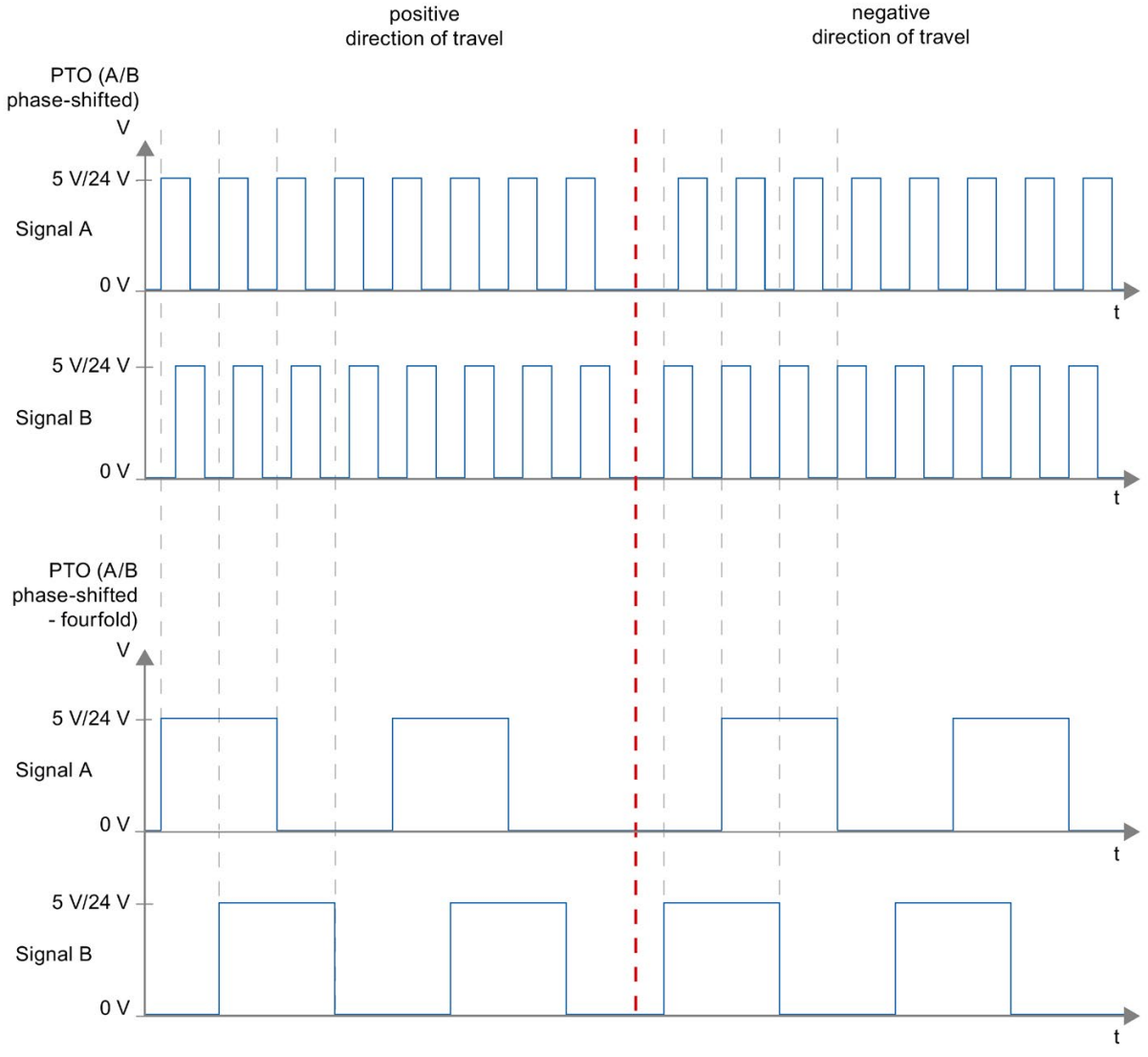
PTO – (A/B phase-shifted - quadruple) (as of V4)

The positive and negative edges of both outputs are evaluated for this signal type. A pulse period has four edges with two phases (A and B). The pulse frequency at the output is therefore reduced to a quarter.

The pulse is output via the "Signal A" output and phase-shifted via the "Signal B" output. The phase shifting between the outputs defines the direction of rotation:

- Signal A leads signal B by 90° ⇒ positive direction of rotation
- Signal B leads signal A by 90° ⇒ negative direction of rotation

The specified voltage depends on the hardware used. The indicated values do not apply to the differential outputs of CPU 1217.



Invert direction

If you select the "Invert rotation signal" option, the direction logic is inverted:

- PTO – pulse (A) and direction (B)
  - 0 V at direction output (low level) ⇒ positive direction of rotation
  - 5 V/24 V at direction output (high level) ⇒ negative direction of rotation

  The specified voltage depends on the hardware used. The voltages indicated do not apply to the differential outputs of CPU 1217.
- PTO – clock up A and clock down B

  The outputs "Pulse output down" and "Pulse output up" are swapped.
- PTO – A/B phase-shifted

  The "Signal A" and "Signal B" outputs are swapped.
- "PTO – A/B phase-shifted, quadruple

  The "Signal A" and "Signal B" outputs are swapped.

See also

CPU outputs relevant for motion control (Page 13)

How the pulse interface works (Page 16)

Hardware and software limit switches (Page 38)

Jerk limit (Page 39)

Homing (Page 40)

Integration of the positioning axis technology object (Page 53)

Tools of the positioning axis technology object (Page 56)

2.2 PROFIdrive drive/analog drive connection

2.2.1 Drive and encoder connection

A drive and an encoder are assigned to a positioning axis with drive connection via PROFIdrive/analog drive connection.

Drives with PROFIdrive capability are connected by means of PROFIdrive telegrams. The setpoint is specified via PROFIdrive telegrams.

Drives with analog setpoint interfaces are connected using an analog output and an optional enable signal. The setpoint is specified via an analog output.

Options for connecting

Drives with PROFIdrive capability are connected via the PROFINET interface of the CPU.

Drives with analog setpoint interface are connected with the CPU via one of the following connections:

- Analog output via signal board
- Analog output on-board
- Analog output via analog output module

The following connection options are available for an encoder:

- Encoder on the PROFIdrive drive
- Encoder on technology module
- PROFIdrive encoder directly on PROFINET IO

  (With these encoders, the encoder value is always transferred via PROFIdrive telegrams per PROFIBUS or PROFINET.)
- Encoder on high speed counter (HSC)

  With this connection option, the encoder signals are connected directly to an HSC, which forms the encoder values from them. Depending on the CPU in use, up to 6 HSC encoders can be used.

Maximum number of axes

You can control up to a maximum of 8 drives via PROFIdrive or the analog drive connection (the number is independent of the simulation status of the axis).

2.2.2 Automatic transfer of drive and encoder parameters in the device

For operation, identical reference values for the drive and encoder connections must be set in the controller and in the drive and encoder.

The speed setpoint NSET and the actual speed value NACT are transferred in the PROFIdrive telegram as a percentage of the reference speed. The reference value for the speed must be set identically in the controller and in the drive.

The resolution of the actual value in the PROFIdrive telegram must likewise be set identically in the controller and in the drive and encoder modules

Automatic transfer of parameters

For SINAMICS drives as of V4.x and PROFIdrive encoders as of product version A16, the drive and encoder parameters can be automatically transferred to the CPU.

The corresponding parameters are transferred after the (re)initialization of the technology object and (re)start of the drives and the CPU. Changes in the drive configuration are transferred after restart of the drive or technology object.

The successful transfer of the parameters can be verified in the controller in the variables of the <TO>.StatusDrive.AdaptionState = 1 and <TO>.StatusSensor[1].AdaptionState = 1 technology objects.

Parameters

The controller settings are made in the TIA Portal under "Technology object > Configuration > Basic parameters > Drive/encoder".

The drive and encoder settings are made in the configuration or the respective hardware.

The following table compares the settings in the TIA Portal and the corresponding drive/encoder parameters:

Setting in the TIA Portal	Controller variable in the technology data block	Drive parameter	Automatic transfer
Drive			
Telegram number	Input address telegram <TO>.Actor.Interface.AddressIn	Telegram number P922	-
	Output address telegram <TO>.Actor.Interface.AddressOut		
Reference speed in [1/min]	<TO>.Actor.DriveParameter.Reference Speed	SINAMICS drives: P2000	X
Maximum speed of motor in [1/min]	<TO>.Actor.DriveParameter.MaxSpeed	SINAMICS drives: P1082	X
Drive	<TO>.Actor.Type 0 = analog drive connection 1 = PROFIdrive 2 = PTO (pulse train output)	-	-

Setting in the TIA Portal	Controller variable in the technology data block	Drive parameter	Automatic transfer
Encoder			
Telegram	<TO>.Sensor[n].Interface.AddressIn	P922	-
	<TO>.Sensor[n].Interface.Addressout		
Encoder type • Linear incremental • Linear absolute • Rotary incremental • Rotary absolute	<TO>.Sensor[n].System 0: Rotary 1: Linear	P979[1] Bit0 Encoder 1 P979[11] Bit0 Encoder 2	X
	<TO>.Sensor[n].Type 0: Incremental 1: Absolute	P979[5] Encoder 1 P979[15] Encoder 2	-
Resolution, linear encoder The grid spacing is specified on the nameplate of the encoder as a separation distance of the marks on the linear measuring system.	<TO>.Sensor[n].Parameter.Resolution	P979[2] Encoder 1 P979[12] Encoder 2	X
Increments per revolution, rotary encoder	<TO>.Sensor[n].Parameter.StepsPerRevolution	P979[2] Encoder 1 P979[12] Encoder 2	X
Number of bits for fine resolution XIST1 Cyclic actual encoder value, linear or rotary encoder	<TO>.Sensor[n].Parameter.FineResolutionXist1	P979[3] Encoder 1 P979[13] Encoder 2	X
Number of bits for fine resolution XIST2 Absolute value of the encoder, linear or rotary encoder	<TO>.Sensor[n].Parameter.FineResolutionXist2	P979[4] Encoder 1 P979[14] Encoder 2	X
Differentiable encoder revolutions, rotary absolute encoder	<TO>.Sensor[n].Parameter.DeterminableRevolutions	P979[5] Encoder 1 P979[15] Encoder 2	X

See also

Configuration - Drive - PTO (Pulse Train Output) (Page 63)
Configuration - Drive - Analog drive connection (Page 65)
Configuration - Drive - PROFIdrive (Page 66)
Configuration - Encoder - Encoder on PROFINET/PROFIBUS (Page 68)
Configuration - Encoder - Encoder on high-speed counter (HSC) (Page 71)

2.2.3 PROFIdrive

PROFIdrive is the standardized standard profile for drive technology in the connection of drives and encoders via PROFINET IO. Drives and encoders that support the PROFIdrive profile are connected according to the PROFIdrive standard.

You can find the current PROFIdrive specification at:

http://www.profibus.com (http://www.profibus.com)

Communication between controller and drive/encoder is by means of various PROFIdrive telegrams. Each of the telegrams has a standardized structure. Depending on the application, you can select the applicable telegram. Control words and status words as well as setpoints and actual values are transmitted in the PROFIdrive telegrams.

Telegrams for PROFIdrive

The setpoint of a positioning axis is transferred to a drive via PROFIdrive telegram 1, 2 3 or 4.

The encoder value is transmitted either in a telegram together with the setpoint (telegram 3 and 4), or in a separate encoder telegram (telegram 81 or telegram 83).

The following table shows the supported PROFIdrive telegrams for the assignment of drives and encoders:

Standard telegrams	Brief description
1	• 16 bit speed setpoint (NSET) • 16 bit actual speed (NACT)
2	• 32 bit speed setpoint (NSET) • 32-bit actual speed (NACT) • Signs of life
3	• 32 bit speed setpoint (NSET) • 32-bit actual speed (NACT) • 1 encoder • Signs of life
4	• 32 bit speed setpoint (NSET) • 32-bit actual speed (NACT) • 2 encoders • Signs of life

Standard telegrams for encoder	Brief description
81	• 1 encoder • Signs of life
83	• 32-bit actual speed (NACT) • 1 encoder • Signs of life

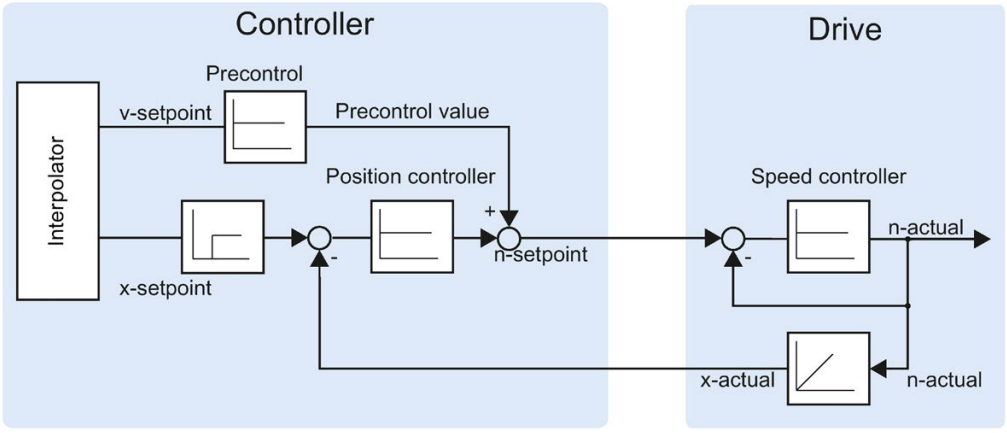
2.2.4 Closed loop control

Drives which are connected via PROFIdrive or an analog drive interface are generally operated under position control. If service is required, the axis can also be operated without position control.

The position controller is a P controller with precontrol of velocity.

Controller structure

The following figure shows the controller structure of an S7-1200 Motion Control:



The MC-Interpolator [OB92] calculates the setpoint position for the axis. The difference between the setpoint and actual position is multiplied by the gain factor of the position controller. The resulting value is added to the precontrol value and output as setpoint speed to the drive via PROFIdrive or analog output.

The encoder records the actual position of the axis and returns it to the controller via a PROFIdrive telegram or an HSC (high speed counter).

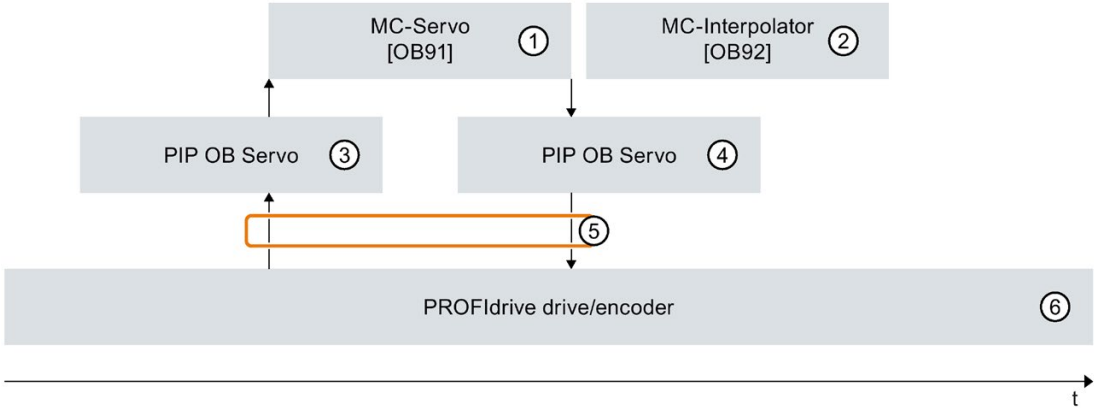
2.2.5 Data connection PROFIdrive drive/PROFIdrive encoder

The data connection of PROFIdrive drives and PROFIdrive encoders occurs either directly via the PROFIdrive telegram or via a data block.

Use the connection via data block if you want to influence or evaluate telegram contents in the user program for process-specific reasons.

Principle of data connection directly to the drive/encoder

The following simplified function chart shows the direct data connection to PROFIdrive drives and PROFIdrive encoders by means of telegrams:



① The organization block "MC-Servo" calculates the position controller.

At the start of "MC-Servo", the input telegram of the drive or encoder is read (⑥ -> ⑤ -> ③). If an organization block "MC-PreServo" was added, the telegram is read at the beginning of "MC-PreServo".

At the end of "MC-Servo", the output telegram is written to the drive or encoder (⑥ -> ⑤ -> ③). If an organization block "MC-PostServo" was added, the telegram is written at the end of "MC-PostServo".

② In every Motion application cycle, the organization block "MC-Interpolator" is called after the "MC-Servo".

In "MC-Interpolator", the Motion Control instructions are evaluated, setpoints are generated for the next Motion application cycle and the technology object is monitored.

③ The process image partition "PIP OB servo" of the inputs is updated in the Motion application cycle.

④ The process image partition "PIP OB servo" of the outputs is updated in the Motion application cycle.

⑤ Telegram exchange via the I/O addresses of the controller and the drive or the encoder.

⑥ PROFIdrive drive or PROFIdrive encoder

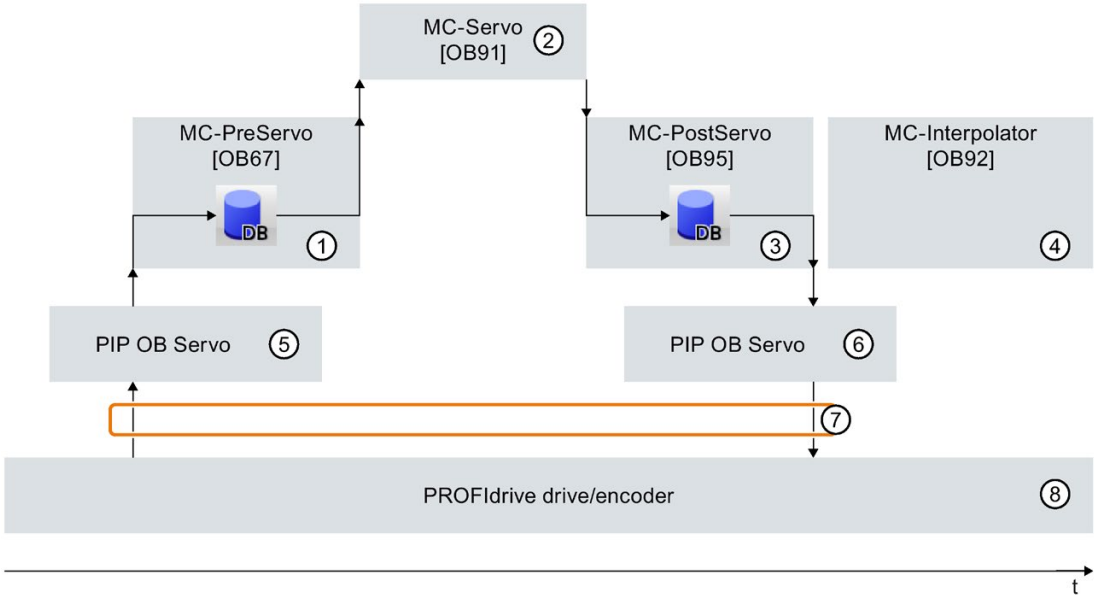
Principle of data connection via data block

The following simplified function chart shows the data connection to PROFIdrive drives and PROFIdrive encoders via data block: The following sections provide details on the execution.

To influence or evaluate telegram contents for process-specific reasons, a data block must be connected in between as a data interface (see ① and ③).

To do this, use the organization blocks "MC-PreServo" and "MC-PostServo" to achieve a high quality of position control.

The organization blocks "MC-PreServo" and "MC-PostServo" can be added in the project tree with the command "Add new block".



① The "MC-PreServo" is called before the "MC-Servo".

In the user program of the "MC-PreServo", transfer the content of the input telegram from the process image partition "PIP OB servo" ⑤ to the data block of the data interface.

In the other user program of the "MC-PreServo", the input area of the telegram can be processed or evaluated.

② The organization block "MC-Servo" calculates the position controller.

At the start of "MC-Servo", the input telegram of the drive or encoder is read from the data block of the data interface (① -> ②).

At the end of "MC-Servo", the output telegram of the drive or encoder is written to the data block of the data interface (② -> ③).

③ The "MC-PostServo" is called after the "MC-Servo".

In the user program of the "MC-PostServo", the output area of the telegram can be processed or evaluated.

At the end of the user program of the "MC-PostServo", transfer the content of the output telegram from the data interface of the data block to the process image partition "PIP OB servo" ⑥.

④ In every Motion application cycle, the organization block "MC-Interpolator" is called after the "MC-PostServo".

In "MC-Interpolator", the Motion Control instructions are evaluated, setpoints are generated for the next Motion application cycle and the technology object is monitored.

⑤ The process image partition "PIP OB servo" of the inputs is updated in the Motion application cycle.

⑥ The process image partition "PIP OB servo" of the outputs is updated in the Motion application cycle.

⑦ Telegram exchange via the I/O addresses of the controller and the drive or the encoder.

⑧ PROFIdrive drive or PROFIdrive encoder

Basic procedure for the data connection via a data block

To use the data connection via the data block, follow the steps described below. The data connection can be configured separately for a PROFIdrive drive and encoder.

Creating the data block for data connection

The data block for the data connection must be created by the user. The data block must contain a data structure of data type "PD_TELx" for the data connection. Here, "x" stands for the telegram number of the drive or encoder configured in the device configuration.

To create the data block, follow the steps described below:

1. Create a new data block of type "Global DB".
2. Select the data block in the project tree and select the shortcut menu command "Properties".
3. Under Attributes, disable the following attributes and confirm with OK
   "Only store in load memory"
   "Data block write-protected in the device"
   "Optimized block access"
4. Open the data block in the block editor.
5. Insert a "PD_TELx" type variable in the block editor.
6. Compile the data block for the data connection before you use it in the configuration of the axes.

This variable contains the "Input" variable structure for the input area of the telegram and the "Output" variable structure for the output area of the telegram.

Note

"Input" and "Output" relate to the view of the closed loop position control. For example, the input area contains the actual values of the drive and the output area contains the setpoints for the drive.

The data block may contain the data structures of multiple axes and encoders and other contents.

Configuring data connection via a data block

Proceed as decribed below for the configuration of the axis:

1. Open the configuration window "Hardware interface > Drive" or "Hardware interface > Encoder".
2. In the Data block drop-down list, select "Data block".
3. In the "Data block" field, select the previously created data block.
   Open this data block and select the tag name defined for the drive or encoder.

Adding a PLC variable for telegram access

To enable access to the input and output areas of the telegram, create the following PLC variable.

For the PLC variable of the input area, follow the steps described below:

1. Open the "PLC variables" folder in the project tree and have all variables displayed.
2. Add a new variable and assign a unique name, made up for example from the name of the axis or the encoder, the telegram type and the address area.
3. Enter the type "PD_TELx_IN" textually in the "Data type" column.
4. Enter the telegram input address of the drive/encoder in the "Address" column.
   You can find the address in the device configuration of the drive or encoder.

Follow the same procedure for the PLC variable of the output area and select "PD_TELx_OUT" as the data type and the telegram output address of the drive/encoder as the address.

Programming MC-PreServo and MC-PostServo

MC-PreServo

The user program of the "MC-PreServo" must read the input area of the telegram and transfer it to the data block of the data connection.

Assign the previously defined PLC tag of the input area to the tag structure "Input" of the data block in the "MC-PreServo" user program.

With further instructions, you can edit the data of the tag structure "Input" of the data block before it is then transferred to "MC-Servo" and processed in "MC_Servo" .

MC-PostServo

After it has been processed, "MC-Servo" transfers the output area of the telegram to the tag structure "Output" of the data block.

The content of the tag structure "Output" of the data block must be written to the telegram output address in the "MC-PostServo" user program.

Assign the "Output" tag structure of the data block to the previously defined PLC tag of the output area in the "MC-PostServo" user program.

If the output area is to be modified, this must be done before the assignment instruction.

NOTICE
Machine damage Improper manipulation of drive and encoder telegrams may result in unwanted movements of the drive. Check your user program in regard to consistency in the drive and encoder connection.

You can find an application example for the use of MC-PreServo and MC-PostServo at:

https://support.industry.siemens.com/cs/document/109741575
(https://support.industry.siemens.com/cs/ww/en/view/109741575)

See also

PROFIdrive frame (Page 180)

Configuration - Drive - PROFIdrive (Page 66)

Configuration - Encoder - Encoder on PROFINET/PROFIBUS (Page 68)

Data connection drive with analog drive connection (Page 31)

Organization Blocks for Motion Control (Page 33)

2.2.6 Data connection drive with analog drive connection

The data connection of drives with analog drive interface can alternatively be made via a data block.

Use the connection via a data block if you want to adapt the analog setpoint in the user program for process-related reasons.

Principle of data connection via data block

At the end of position control by MC-Servo [OB91], the setpoint of the analog drive is written to the assigned analog output.

To adapt the analog setpoint for process-related reasons, a data interface via a data block must be connected in between.

The setpoint of the analog drive can be edited via the MC-PostServo [OB95] organization block in the data block and can then be written to the I/O address.

The MC-PostServo is called after the MC-Servo. The organization block MC-PostServo can be programmed by the user and must be added with the command "Add new block".

The procedure in principle

To use the data connection via the data block, follow the steps described below. The data connection can be configured separately for drives with analog drive interface and PROFIdrive encoder. You can find information on data connection of the PROFIdrive encoders in the section Data connection PROFIdrive drive/PROFIdrive encoder (Page 26).

Creating the data block for data connection

The data block must be created on the user side.

To create the data block, follow the steps described below:

1. Create a new data block of type "Global DB".
2. Select the data block in the project tree and select "Properties" from the shortcut menu.
3. Under Attributes, disable the following attributes and confirm with OK
   "Only store in load memory"
   "Data block write-protected in the device"
   "Optimized block access"
4. Open the data block in the block editor.
5. Insert a variable of the "WORD" data type in the block editor.
6. Compile the data block for the data connection before you use it in the configuration of the axes.

Configuring data connection via a data block

Proceed in the configuration as described below ("Analog drive connection" must be selected in the Basic parameters > General configuration window):

1. Open the configuration window Basic parameters > Drive.
2. Select the previously defined variable of the data block in the "Analog output" box.

Set analog output address in the TPA OB Servo process image

To achieve a good level of control quality, the address area of the analog output must be within the process image "TPA OB Servo".

Proceed as described below:

1. Open the module of the analog output in the device configuration.
2. Open the "General" tab
3. Select the "I/O addresses"
4. Select the organization block "MC-Servo". "TPA OB Servo" is automatically selected as the process image.
5. Assign a variable name to the analog output in the "I/O variables" tab.

Program MC-PostServo

Assign the variable of the data block to the variable of the analog output in the MC-PostServo user program.

At the end of MC-PostServo, the output area of "TPA OB Servo" is written to the I/O.

⚠ WARNING
Improper manipulation of the drive setpoint may endanger humans and machines.
Take adequate precautions to prevent danger to humans and machines.

2.2.7 Process response

2.2.7.1 Organization Blocks for Motion Control

Description

When you create a "Positioning axis" technology object with a PROFIdrive drive or with an analog drive interface, organization blocks for processing the technology objects are created automatically. The Motion Control functionality of the technology objects creates its own execution level, and is called according to the Motion Control application cycle.

The following blocks are created:

- MC-Servo [OB91]

  Calculation of the Position Controller

- MC-Interpolator [OB92]

  Evaluation of the motion control instructions, generation of setpoints and monitoring functionality

The organization blocks are protected (know-how protection). The program code cannot be viewed or changed.

The frequency relationship of the two organization blocks to one another is always 1:1. MC-Servo [OB91] is always executed before MC-Interpolator [OB92].

You can set the Motion Control application cycle and the priority of the organization blocks according to your requirements for control quality and system load.

Motion Control application cycle

You can set the Motion Control application cycle, in which the MC-Servo [OB91] is called, in the properties of the organization block in "General > Cycle Time".

The MC-Servo [OB91] is called cyclically with the specified "application cycle".

The selected Motion Control application cycle must be long enough to be able to process all technology objects in one cycle. If the processing time of the technology objects is longer than the application cycle, overflows (Page 35) will occur.

To avoid disruptions in the program execution on the CPU, set the Motion Control application cycle depending on the number of axes used as follows:

Motion Control application cycle = (number of position-controlled axes × 2 ms) + 2 ms

The following table shows the resulting Motion Control application cycle as an example according to the number of position-controlled axes:

Number of axes	Motion Control application cycle
1	4 ms
2	6 ms
4	10 ms
8	18 ms

For SINAMICS, the following should continue to apply:

- Motion Control application cycle (MC-Servo) ≥ SINAMICS drive process image (parameter P2048) ≥ bus clock cycle

All times should be selected as integral multiples of one another.

Priority

You can configure the priority of the organization blocks as needed in their properties under "General > Properties > Priority":

- MC-Servo [OB91]

  Priority 17 to 26 (default value 25)

- MC-Interpolator [OB92]

  Priority 17 to 26 (default value 24)

The priority of MC-Servo [OB91] must be at least one higher than the priority of the MC-Interpolator [OB92].

MC-PreServo [OB67] and MC-PostServo [OB95]

Organization blocks MC-PreServo [OB67] and MC-PostServo [OB95] are programmable and are called in the configured application cycle. MC-PreServo [OB67] is called directly before MC-Servo [OB91]. MC-PostServo [OB95] is called directly after MC-Servo [OB91].

See also

Data connection PROFIdrive drive/PROFIdrive encoder (Page 26)

2.2.7.2 Process image partition "PIP OB Servo"

For optimal control, assign all I/O modules used by Motion Control (e.g. drives, technology modules, digital and analog input/output modules) to the process image partition "PIP OB servo". The assignment causes the I/O modules to be processed simultaneously with the technology object.

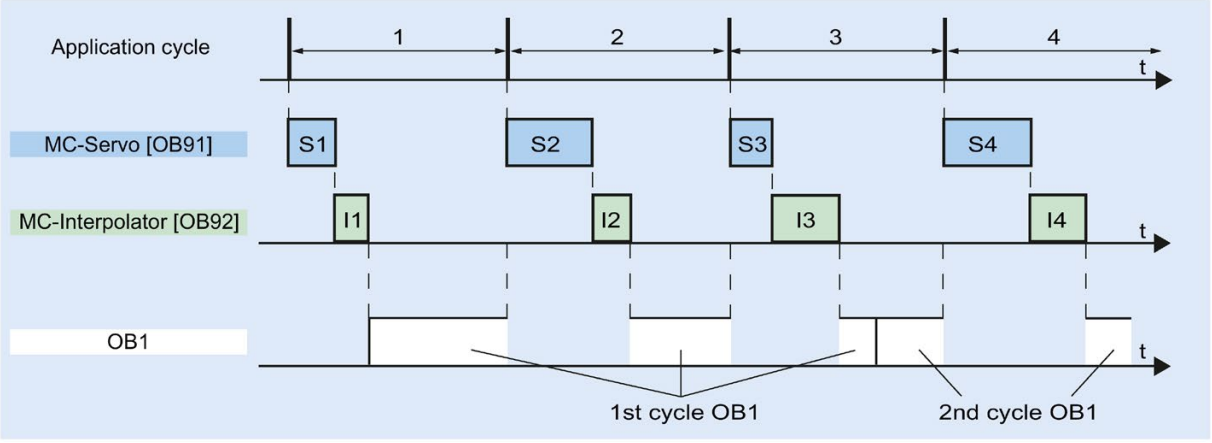
2.2.7.3 Operational Sequence and Timeouts

When processing the Motion Control functionality, the organization blocks MC-Servo [OB91] and MC-Interpolator [OB92] are called and processed in each application cycle (processing also occurs in the STOP operating state of the CPU). The remaining cycle time is available for the processing of your user program.

For error-free program execution, keep to the following rules:

- In each application cycle, MC-Servo [OB91] must be started and executed completely.
- In every application cycle, the relevant MC-Interpolator [OB92] must at least be started.

The following figure shows an example of the error-free operational sequence for the processing of organization block OB1:



Overflows

If the set application cycle is not adhered to, for example because the application cycle is too short, overflows can occur.

An overflow of the MC-Servo [OB91], MC-Interpolator [OB92], MC_PreServo [OB67] and MC_PostServo [OB95] is entered in the diagnostic buffer of the CPU and results in setting the CPU to STOP.

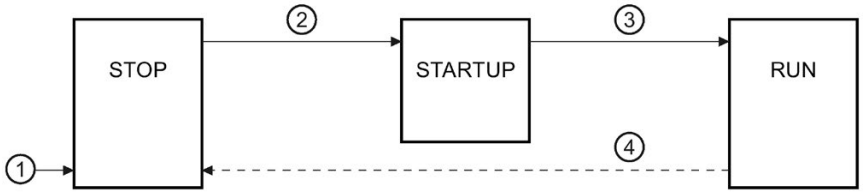
MC-PreServo, MC-Servo, MC-PostServo and MC-Interpolator are stopped. If necessary, you can evaluate the entry in the diagnostic buffer via a time error OB (OB80).

2.2.7.4 Operating modes

This section examines the behavior of Motion Control in each operating mode, and in the transitions between operating modes. A general description of the operating modes can be found in system manual S7-1200.

Operating modes and transitions

The CPU has three operating modes: STOP, STARTUP and RUN. The following figure shows the operating modes and the operating mode transitions:



① POWER ON → STOP
② STOP → STARTUP
③ STARTUP → RUN
④ RUN → STOP

STOP mode

In STOP mode the user program is not processed and all process outputs are disabled. Thus no Motion Control jobs are executed.

The technology data blocks of the position-controlled axes are updated.

STARTUP mode

Before the CPU starts processing of the cyclical user program, the startup OBs are processed one time.

In STARTUP mode, the process outputs are disabled. Motion Control jobs are rejected.

The technology data blocks of the position-controlled axes are updated.

RUN mode

The user program is processed in RUN mode.

In RUN mode, the Motion Control commands programmed in OB1 are called and processed cyclically (other execution levels are possible).

The technology data blocks are updated.

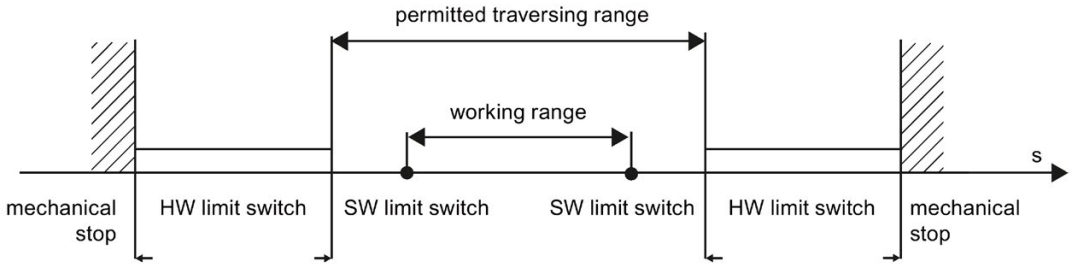
Operating mode transitions

The following table shows the behavior of Motion Control in the transitions between the operating modes:

Operating mode transition	Behavior
POWER ON → STOP	The CPU performs a restart of the technology objects. The technology objects are reinitialized with the values from the load memory.
STOP → STARTUP	The technology objects are initialized with the start values of the CPU.
STARTUP → RUN	The process outputs are enabled.
RUN → STOP	When the CPU changes to RUN mode after STOP mode, all technology objects are disabled in accordance with the error response "remove enablement". Running Motion Control jobs are terminated.

2.3 Hardware and software limit switches

Use the hardware and software limit switches to limit the "permitted traversing range" and the "working range" of your positioning axis technology object. The relationships are shown in the following diagram:



Hardware limit switches are limit switches that limit the maximum "permitted traversing range" of the axis. Hardware limit switches are physical switching elements that must be connected to interrupt-capable inputs of the CPU.

Software limit switches limit the "working range" of the axis. They should fall inside the hardware limit switches relative to the traversing range. Since the positions of the software limit switches can be flexibly set, the working range of the axis can be adapted on an individual basis, depending on the current traversing profile. In contrast to hardware limit switches, software limit switches are implemented exclusively via the software and do not require their own switching elements.

Hardware and software limit switches must be activated prior to use in the configuration or in the user program. Software limit switches are only active after homing the axis.

See also

CPU outputs relevant for motion control (Page 13)

How the pulse interface works (Page 16)

Relationship between the signal type and the direction of travel (Page 17)

Jerk limit (Page 39)

Homing (Page 40)

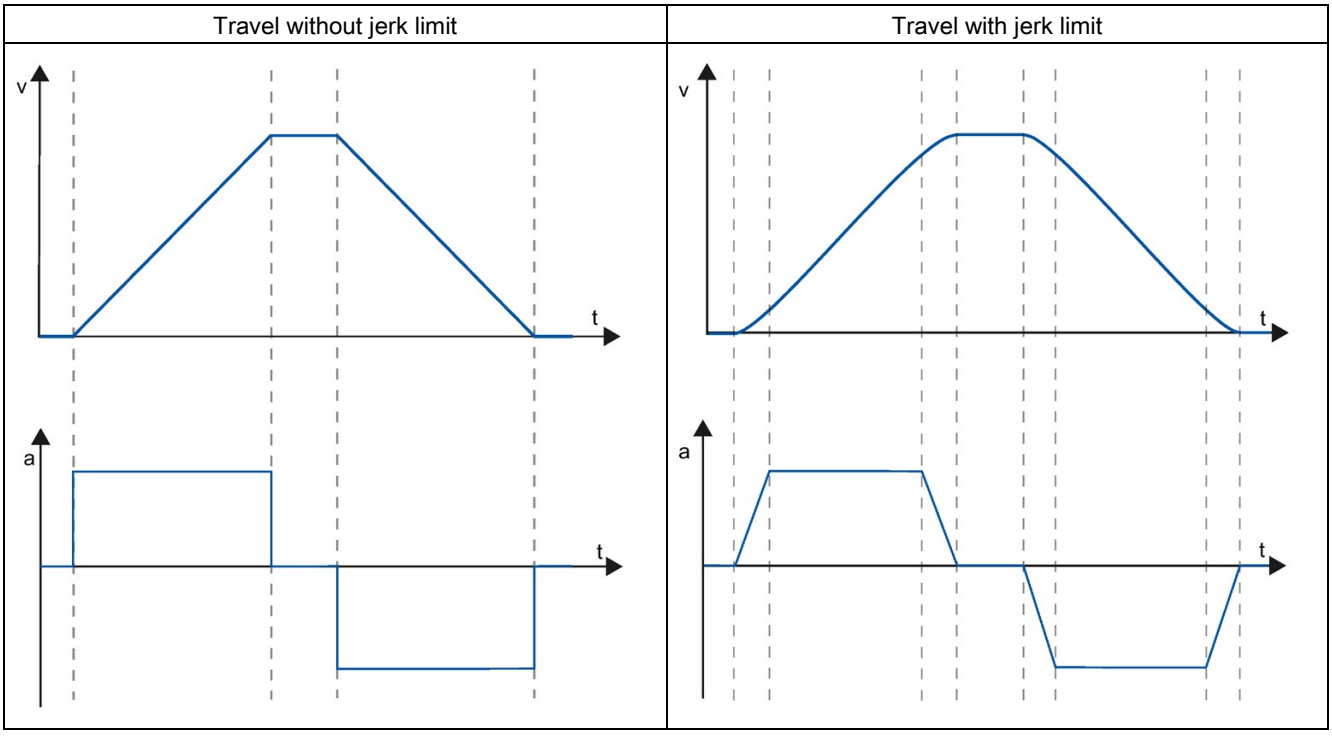
Integration of the positioning axis technology object (Page 53)

Tools of the positioning axis technology object (Page 56)

Position limits (Page 76)

2.4 Jerk limit

With the jerk limit you can reduce the stresses on your mechanics during an acceleration and deceleration ramp. The acceleration and deceleration value is not changed abruptly when the jerk limiter is active; it is gradually increased and decreased. The figure below shows the velocity and acceleration curve without and with jerk limit.



The jerk limit gives a "smoothed" velocity profile of the axis motion. This ensures, for example, soft starting and braking of a conveyor belt.

See also

Behavior of the axis when using the jerk limit (Page 85)

CPU outputs relevant for motion control (Page 13)

How the pulse interface works (Page 16)

Relationship between the signal type and the direction of travel (Page 17)

Hardware and software limit switches (Page 38)

Homing (Page 40)

Integration of the positioning axis technology object (Page 53)

Tools of the positioning axis technology object (Page 56)

2.5 Homing

Homing means matching the axis coordinates of the technology object to the real, physical location of the drive. For position-controlled axes the entries and displays for the position refer exactly to these axis coordinates. Therefore, agreement between the axis coordinates and the real situation is extremely important. This step is necessary to ensure that the absolute target position of the axis is also achieved exactly with the drive.

In the S7-1200 CPU, axis homing is implemented with the motion control instruction, "MC_Home". The "Homed" status is displayed in the tags of the technology object <Axis name>.StatusBits.HomingDone . The following homing modes exist:

Homing modes

- Active homing

  In active homing mode, the motion control instruction "MC_Home" performs the required reference point approach. When the homing switch is detected, the axis is homed according to the configuration. Active traversing motions are aborted.

- Passive homing

  During passive homing, the "MC_Home" Motion Control instruction does not carry out any homing motion. The traversing motion required for this must be implemented by the user via other Motion Control instructions. When the homing switch is detected, the axis is homed according to the configuration. Active traversing motions are not aborted upon start of passive homing.

- Direct homing absolute

  The axis position is set regardless of the homing switch. Active traversing motions are not aborted. The value of input parameter "Position" of motion control instruction "MC_Home" is set immediately as the reference point of the axis.

- Direct homing relative

  The axis position is set regardless of the homing switch. Active traversing motions are not aborted. The following statement applies to the axis position after homing:

  New axis position = current axis position + value of parameter "Position" of instruction "MC_Home".

Resetting the "Homed" status

The "Homed" status of a technology object (<Axis name>.StatusBits.HomingDone) is reset under the following conditions:

- Drive connection via PTO (Pulse Train Output):
  - Start an "MC_Home" command for active homing

    (After successful completion of the homing operation, the "Homed" status is set again.)
  - Disabling of axis by the "MC_Power" Motion Control instruction
  - Changeover between automatic mode and manual control
  - After POWER OFF -> POWER ON of the CPU
  - After CPU restart (RUN-STOP -> STOP-RUN)
- Technology objects with incremental actual values:
  - Start an "MC_Home" command for active homing

    (After successful completion of the homing operation, the "Homed" status is set again.)
  - Error in the encoder system, or encoder failure
  - Restart of the technology object
  - After POWER OFF → POWER ON of the CPU
  - Memory reset
  - Modification of the encoder configuration
- Technology objects with absolute actual values:
  - Errors in the sensor system/encoder failure
  - Replacement of the CPU
  - Modification of the encoder configuration
  - Restoration of the CPU factory settings
  - Transfer of a different project to the controller

See also

CPU outputs relevant for motion control (Page 13)

How the pulse interface works (Page 16)

Relationship between the signal type and the direction of travel (Page 17)

Hardware and software limit switches (Page 38)

Jerk limit (Page 39)

Integration of the positioning axis technology object (Page 53)

Tools of the positioning axis technology object (Page 56)

Homing (positioning axis technology object as of V2) (Page 88)

Guidelines on use of motion control

3

The guidelines described here present the basic procedure for using motion control with the CPU S7-1200.

Requirements

To use the positioning axis technology object, a project with a CPU S7-1200 must be created.

Procedure

Follow the steps below in the order given to use motion control with the CPU S7-1200. Use the following links for this purpose:

1. Adding a positioning axis technology object (Page 58)
2. Working with the configuration dialog (Page 59)
3. Download to CPU (Page 144)
4. Function test of the axis in the commissioning window (Page 146)
5. Programming (Page 154)
6. Diagnostics of the axis control (Page 176)

Using versions

4

4.1 Overview of versions

The relationship between the relevant versions for S7-1200 Motion Control can be found in the following table:

Technology version

You can check the currently selected technology version in the "Instructions" task card in the folder "Technology > Motion Control" and in the "Technology object > Add new object" dialog.

Select the technology version in the "Instructions" task card in the folder "Technology > Motion Control".

If a technology object with an alternative version is added in the "Add new object" dialog, the technology version will also be changed.

Note

The selection of an alternative technology version will also affect the Motion Control Instructions version (task card).

The technology objects and Motion Control instructions will only be converted to the selected version upon compilation or "Download to device".

Version of the technology object

The version of a technology object can be checked in the inspector window under "Properties > General > Information" in the "Version" box.

Motion Control instruction version

The Motion Control instruction version can be checked in the inspector window under "Properties > General > Information" in the "Version" box.

If the Motion Control instruction version used is not in line with the following compatibility list, the relevant Motion Control instructions will be highlighted in the program editor.

Compatibility list

Technology		CPU	Technology object	Motion Control instruction
V6.0	Innovations: • MC-PreServo • MC-PostServo • Data connection directly to the SINAMICS drive or via data block • Data connection directly to the analog output of a drive with analog drive connection or via a data block • Transfer of drive and encoder parameters in the device for PROFIdrive drives and encoders. • Move position-controlled drives without position control for servicing purposes • Simulation of position-controlled drives without configured or existing hardware • Level selection when configuring the hardware limit switch • Support PROFIdrive Telegram 4	V4.2	Positioning axis V6.0 Command table V6.0	MC_Power V6.0 MC_Reset V6.0 MC_Home V6.0 MC_Halt V6.0 MC_MoveAbsolute V6.0 MC_MoveRelative V6.0 MC_MoveVelocity V6.0 MC_MoveJog V6.0 MC_CommandTable V6.0 MC_ChangeDynamic V6.0 MC_ReadParam V6.0 MC_WriteParam V6.0
V5.0	Innovations: • Drive connection via PROFIdrive • Analog drive connection • Position control for PROFIdrive / analog drive connection • Position monitoring for PROFIdrive / analog drive connection • MC-Servo [OB91] • MC-Interpolator [OB92]	V4.1	Positioning axis V5.0 Command table V5.0	MC_Power V5.0 MC_Reset V5.0 MC_Home V5.0 MC_Halt V5.0 MC_MoveAbsolute V5.0 MC_MoveRelative V5.0 MC_MoveVelocity V5.0 MC_MoveJog V5.0 MC_CommandTable V5.0 MC_ChangeDynamic V5.0 MC_ReadParam V5.0 MC_WriteParam V5.0

Technology		CPU	Technology object	Motion Control instruction
V4.0	Innovations: • MC_ReadParam • MC_WriteParam • Standardization of S7-1200 and S7-1500 Motion Control technology data blocks.	V4.0	Positioning Axis V4.0 Command table V4.0	MC_Power V4.0 MC_Reset V4.0 MC_Home V4.0 MC_Halt V4.0 MC_MoveAbsolute V4.0 MC_MoveRelative V4.0 MC_MoveVelocity V4.0 MC_MoveJog V4.0 MC_CommandTable V4.0 MC_ChangeDynamic V4.0 MC_ReadParam V4.0 MC_WriteParam V4.0
V3.0	Innovation: Load in RUN mode	V2.2 V3.0 V4.0	Axis V3.0 Command table V3.0	MC_Power V3.0 MC_Reset V3.0 MC_Home V3.0 MC_Stop V3.0 MC_MoveAbsolute V3.0 MC_MoveRelative V3.0 MC_MoveVelocity V3.0 MC_MoveJog V3.0 MC_CommandTable V3.0 MC_ChangeDynamic V3.0

Technology		CPU	Technology object	Motion Control instruction
V2.0	Innovations: • Jerk limit • Command table • MC_ChangeDynamic	V2.1 V2.2 V3.0	Axis V2.0 Command table V2.0	MC_Power V2.0 MC_Reset V2.0 MC_Home V2.0 MC_Halt V2.0 MC_MoveAbsolute V2.0 MC_MoveRelative V2.0 MC_MoveVelocity V2.0 MC_MoveJog V2.0 MC_CommandTable V2.0 MC_ChangeDynamic V2.0
V1.0		V1.0 V2.0 V2.1 V2.2 V3.0	Axis V1.0	MC_Power V1.0 MC_Reset V1.0 MC_Home V1.0 MC_Halt V1.0 MC_MoveAbsolute V1.0 MC_MoveRelative V1.0 MC_MoveVelocity V1.0 MC_MoveJog V1.0

See also

Changing a technology version (Page 47)
Compatibility list of variables V1...3 <-> V4...5 (Page 48)
Status of limit switch (Page 52)

4.2 Changing a technology version

Before you can access all the benefits of a new technology version, you may need to set / modify the technology version for existing projects.

Note

Compatibility of the technology object tags

When switching between V1...3 and ≥ V4, please see the compatibility list (Page 48) when using tags of the technology object in the user program, monitoring tables, etc .

Setting/changing a technology version

To set or change the technology version, follow these steps:

1. Open the program editor (e.g., by opening the OB1).
2. In the "Instructions" task card, select the desired technology version in the "Technology > Motion Control" folder.
3. Save and compile the project. Pay attention to any error information that is displayed during compilation. Deal with the causes of the errors indicated.
4. Check the configuration of the technology objects.
5. If necessary, adapt the tag names in the following objects in line with the compatibility list.

- User program
- Watch tables
- Force tables
- HMI configuration
- Trace configuration

See also

Overview of versions (Page 43)

Status of limit switch (Page 52)

4.3 Compatibility list of variables V1...3 <-> V4...5

The technology data blocks for S7-1200 Motion Control and S7-1500 Motion Control have been standardized within the framework of the V4 technology. As of V4, this has resulted in new tags and tag names for the positioning axis and command table technology objects.

Observe the information in the following tables if you have used tags of the technology objects in the user program and you want to convert the project from V1...3 to V4 or higher (or vice versa).

The tags listed in the "Automatic conversion V1...3 to ≥ V4" column are converted automatically when the project is compiled. Tag names in monitoring and force tables or the HMI or trace configuration are not converted.

The following tags are new or have been adapted and may have to be corrected in the user program, watch tables, etc.:

Config tags (positioning axis)

Tag name V1.0 to V3.0	Tag name as of V4.0	Automatic conversion V1..3 to ≥ V4
<Axis name>.Config.DynamicDefaults.Acceleration	<Axis name>.DynamicDefaults.Acceleration	Yes
<Axis name>.Config.DynamicDefaults.Deceleration	<Axis name>.DynamicDefaults.Deceleration	Yes
<Axis name>.Config.DynamicDefaults.Emergency Deceleration	<Axis name>.DynamicDefaults.EmergencyDeceleration	Yes
<Axis name>.Config.DynamicDefaults.Jerk	<Axis name>.DynamicDefaults.Jerk	Yes
<Axis name>.Config.DynamicDefaults.JerkActive	Not available The jerk is activated if the configured jerk is > 0.004 pulse/s^3.	No
<Axis name>.Config.DynamicLimits.MaxVelocity	<Axis name>.DynamicLimits.MaxVelocity	Yes
<Axis name>.Config.DynamicLimits.MinVelocity	<Axis name>.DynamicLimits.MinVelocity	Yes
<Axis name>.Config.General.LengthUnit	<Axis name>.Units.LengthUnit	Yes
<Axis name>.Config.Homing.AutoReversal	<Axis name>.Homing.AutoReversal	Yes
<Axis name>.Config.Homing.Direction	<Axis name>.Homing.ApproachDirection	Yes
<Axis name>.Config.Homing.FastVelocity	<Axis name>.Homing.ApproachVelocity	Yes
<Axis name>.Config.Homing.Offset	<Axis name>.Sensor[1].ActiveHoming.HomePositionOffset	Yes
<Axis name>.Config.Homing.SideActiveHoming	<Axis name>.Sensor[1].ActiveHoming.SideInput	Yes
<Axis name>.Config.Homing.SidePassiveHoming	<Axis name>.Sensor[1].PassiveHoming.SideInput	Yes
<Axis name>.Config.Homing.SlowVelocity	<Axis name>.Homing.ReferencingVelocity	Yes
<Axis name>.Config.Homing.SwitchedLevel	<Axis name>.Sensor[1].ActiveHoming.SwitchLevel <Axis name>.Sensor[1].PassiveHoming.SwitchLevel	No

Tag name V1.0 to V3.0	Tag name as of V4.0	Automatic conversion V1..3 to ≥ V4
<Axis name>.Config.Mechanics.InverseDirection	<Axis name>.Actor.InverseDirection	Yes
<Axis name>.Config.Mechanics.LeadScrew	<Axis name>.Mechanics.LeadScrew	Yes
<Axis name>.Config.Mechanics.PulsesPerDrive Revolution	<Axis name>.Actor.DriveParameter.PulsesPerDriveRevolution	Yes
<Axis name>.Config.PositionLimits_HW.Active	<Axis name>.PositionLimitsHW.Active	Yes
<Axis name>.Config.PositionLimits_HW.Max SwitchedLevel	<Axis name>.PositionLimitsHW.MaxSwitchLevel	Yes
<Axis name>.Config.PositionLimits_HW.Min SwitchedLevel	<Axis name>.PositionLimitsHW.MinSwitchLevel	Yes
<Axis name>.Config.PositionLimits_SW.Active	<Axis name>.PositionLimitsSW.Active	Yes
<Axis name>.Config.PositionLimits_SW.Max Position	<Axis name>.PositionLimitsSW.MaxPosition	Yes
<Axis name>.Config.PositionLimits_SW.Min Position	<Axis name>.PositionLimitsSW.MinPosition	Yes
Not available	<Axis name>.Actor.DirectionMode	No
Not available	<Axis name>.Actor.Type	No
Not available	<Axis name>.Sensor[1].ActiveHoming.Mode	No
Not available	<Axis name>.Sensor[1].PassiveHoming.Mode	No

ErrorBits tags (positioning axis)

Tag name V1.0 to V3.0	Tag name as of V4.0	Automatic conversion V1..3 to ≥ V4
<Axis name>.ErrorBits.HwLimitMax	<Axis name>.ErrorBits.HWLimit (Note the new status bits and the section Status of the limit switch (Page 52).)	No
<Axis name>.ErrorBits.HwLimitMin		
<Axis name>.ErrorBits.SwLimitMaxExceeded	<Axis name>.ErrorBits.SWLimit (Note the new status bits and the section Status of the limit switch (Page 52).)	No
<Axis name>.ErrorBits.SwLimitMaxReached		
<Axis name>.ErrorBits.SwLimitMinExceeded		
<Axis name>.ErrorBits.SwLimitMinReached		
Not available	<Axis name>.ErrorBits.DirectionFault	No

MotionStatus tags (positioning axis)

Tag name V1.0 to V3.0	Tag name as of V4.0	Automatic conversion V1..3 to ≥ V4
<Axis name>.MotionStatus.Distance	<Axis name>.StatusPositioning.Distance	Yes
<Axis name>.MotionStatus.Position	<Axis name>.Position	Yes
<Axis name>.MotionStatus.TargetPosition	<Axis name>.StatusPositioning.TargetPosition	Yes
<Axis name>.MotionStatus.Velocity	<Axis name>.Velocity	Yes

StatusBits tags (positioning axis)

Tag name V1.0 to V3.0	Tag name as of V4.0	Automatic conversion V1..3 to ≥ V4
<Axis name>.StatusBits.Homing	<Axis name>.StatusBits.HomingCommand	Yes
<Axis name>.StatusBits.SpeedCommand	<Axis name>.StatusBits.VelocityCommand	Yes
Not available	<Axis name>.StatusBits.HWLimitMaxActive	No
Not available	<Axis name>.StatusBits.HWLimitMinActive	No
Not available	<Axis name>.StatusBits.SWLimitMaxActive	No
Not available	<Axis name>.StatusBits.SWLimitMinActive	No

Tags (command table)

Tag name V1.0 to V3.0	Tag name as of V4.0	Automatic conversion V1..3 to ≥ V4
<Command table>.Config.Command[n].Position	<Command table>.Command[n].Position	Yes
<Command table>.Config.Command[n].Velocity	<Command table>.Command[n].Velocity	Yes
<Command table>.Config.Command[n].Duration	<Command table>.Command[n].Duration	Yes
<Command table>.Config.Command[n].NextStep	<Command table>.Command[n].NextStep	Yes
<Command table>.Config.Command[n].StepCode	<Command table>.Command[n].StepCode	Yes

See also

Overview of versions (Page 43)

Changing a technology version (Page 47)

4.4 Compatibility list of variables V4...5 <-> V6

The technology data blocks for S7-1200 Motion Control and S7-1500 Motion Control have continued to be standardized within the framework of the V6 technology. As of V6, this has resulted in new tag names for the technology object positioning axis.

Observe the information in the following tables if you have used tags of the technology objects in the user program and you want to convert the project from V4...5 to ≥ V6 or higher (or vice versa).

The tags listed in the "Automatic conversion V4...5 to ≥ V6" column are converted automatically when the project is compiled. Tag names in monitoring and force tables, HMI and trace configurations are not converted.

The following tags are new or have been adapted and may need to be corrected in the user program, monitoring tables, etc.:

Config variables (positioning axis)

Variable name V4.0 to V5.0	Variable name as of V6.0	Automatic conversion V4...5 to ≥ V6
<Axis name>.PositionLimitsSW.Active	<Axis name>.PositionLimits_SW.Active	Yes
<Axis name>.PositionLimitsSW.MinPosition	<Axis name>.PositionLimits_SW.MinPosition	Yes
<Axis name>.PositionLimitsSW.MaxPosition	<Axis name>.PositionLimits_SW.MaxPosition	Yes
<Axis name>.PositionLimitsHW.Active	<Axis name>.PositionLimits_HW.Active	Yes
<Axis name>.PositionLimitsHW.MinSwitchLevel	<Axis name>.PositionLimits_HW.MinSwitchLevel	Yes
<Axis name>.PositionLimitsHW.MinSwitchAddress	<Axis name>.PositionLimits_HW.MinSwitchAddress	Yes

4.5 Status of limit switch

The status and error bits for the display of the reached limit switch have been adapted in version V4.

In order to replicate the behavior of the error bits of versions V1...3, use the following logical operators:

V1...3	V4 or higher
<Axis name>.ErrorBits.HwLimitMin	<Axis name>.ErrorBits.HWLimit AND <Axis name>.StatusBits.HWLimitMinActive
<Axis name>.ErrorBits.HwLimitMax	<Axis name>.ErrorBits.HWLimit AND <Axis name>.StatusBits.HWLimitMaxActive
<Axis name>.ErrorBits.SwLimitMinReached	<Axis name>.ErrorBits.SWLimit AND (<Axis name>.Position = <Axis name>.PositioningLimits_SW.MinPosition)
<Axis name>.ErrorBits.SwLimitMinExceeded	<Axis name>.ErrorBits.SWLimit AND (<Axis name>.Position < <Axis name>.PositioningLimits_SW.MinPosition)
<Axis name>.ErrorBits.SwLimitMaxReached	<Axis name>.ErrorBits.SWLimit AND (<Axis name>.Position = <Axis name>.PositioningLimits_SW.MaxPosition)
<Axis name>.ErrorBits.SwLimitMaxExceeded	<Axis name>.ErrorBits.SWLimit AND (<Axis name>.Position > <Axis name>.PositioningLimits_SW.MaxPosition)

See also

Overview of versions (Page 43)

Changing a technology version (Page 47)

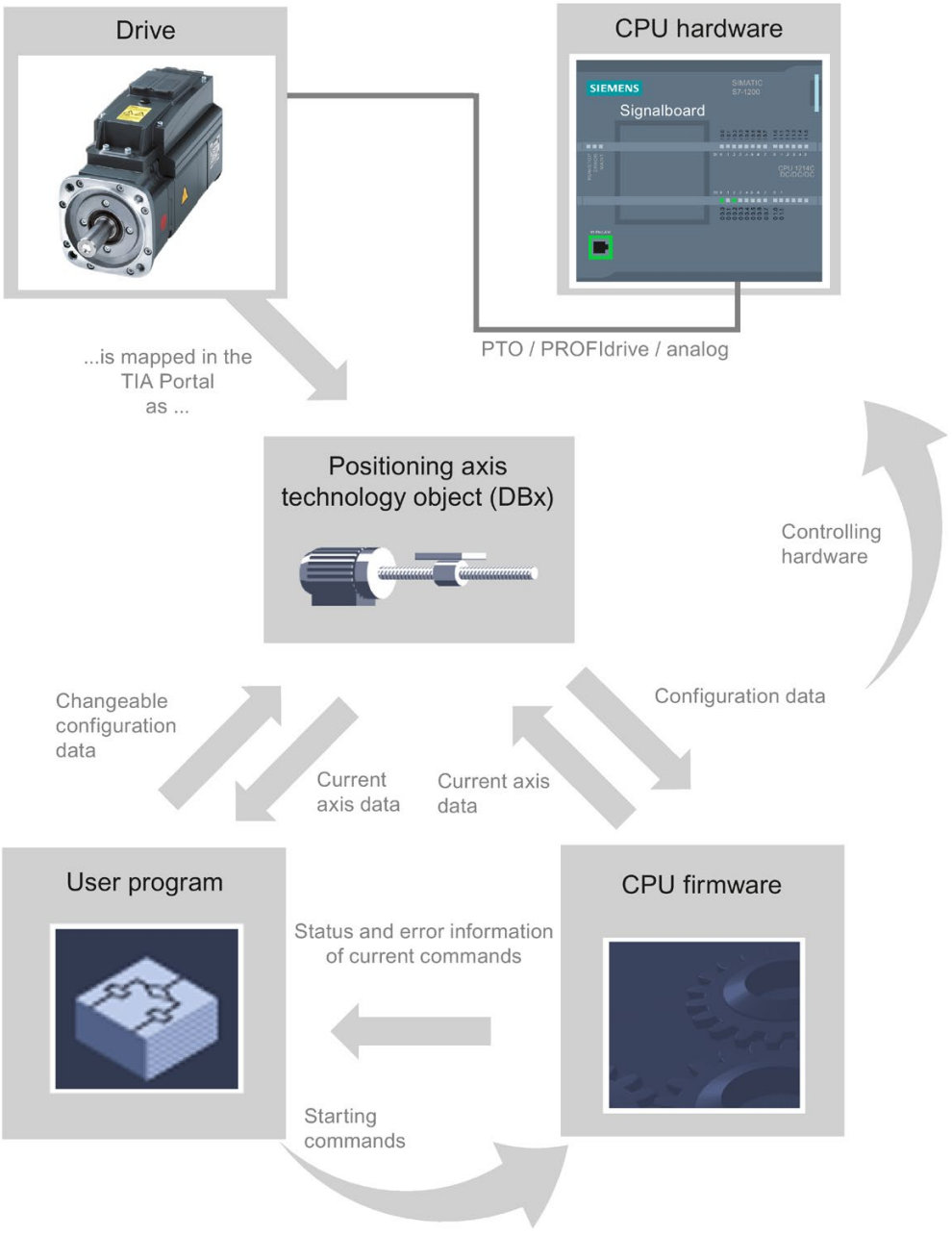
Compatibility list of variables V1...3 <-> V4...5 (Page 48)

Positioning axis technology object

5

5.1 Integration of the positioning axis technology object

The following representation shows the relationships between the hardware and software components that are implemented when using the positioning axis technology object:



CPU hardware

The physical drive is controlled and monitored by the CPU hardware.

Drive

The drive represents the unit of power unit and motor. You can use stepper motors and servo motors with pulse, PROFIdrive or analog interfaces.

Positioning axis technology object

The physical drive including mechanics is mapped in the TIA Portal as a positioning axis technology object. To do this, configure the positioning axis technology object with the following parameters:

- Selection of the PTOs (Pulse Train Output)/PROFIdrive drives/analog outputs to be used and configuration of the drive interface
- Parameter for mechanics and gear transmission of the drive (or the machine or system)
- Parameters for position limits and position monitoring
- Parameters for dynamics and homing
- Parameters for the control loop

The configuration of the positioning axis technology object is saved in the technology object (data block). This data block also forms the interface between the user program and the CPU firmware. The current axis data is saved in the data block of the technology object at the runtime of the user program.

User program

You start Motion Control instructions jobs in the CPU firmware with the user program. The following jobs for controlling the axis are possible:

- Enable and disable axis
- Position axis absolutely
- Position axis relatively
- Move axis with velocity set point
- Run axis commands as movement sequence (technology as of V2, PTO only)
- Moving axes in jog mode
- Stop axis
- Reference axis; set reference point
- Change dynamic settings of axis
- Continuously read motion data of the axis
- Read and write variable of the axis
- Acknowledge error

You determine the command parameters with the input parameters of the Motion Control instructions and the axis configuration. The output parameters of the instruction give you up to date information about the status and any errors of the command.

Before starting a command for the axis, you must enable the axis with the Motion Control instruction "MC_Power".

You can read out configuration data and current axis data with the variables of the technology object. You can change individual, changeable variables of the technology object (e.g. the current acceleration) from the user program.

You can also change the dynamic settings of the axis with the Motion Control instruction "MC_ChangeDynamic" and write additional configuration data with "MC_WriteParam". You can read the current motion status of the axis with the Motion Control instruction "MC_ReadParam".

CPU firmware

The motion control jobs started in the user program are processed in the CPU firmware. When using the axis control panel, Motion Control jobs are triggered by operating the axis control panel. The CPU firmware performs the following jobs depending on the configuration:

- Calculate the exact motion profile for motion jobs and emergency stop situations
- Position control for drive connection via PROFIdrive/analog drive connection
- Control of the pulse and direction signal for drive connection via PTO
- Control of the drive enable
- Monitoring of the drive and the hardware and software limit switches
- Up to date feedback of status and error information to the Motion Control instructions in the user program
- Writing of current axis data into the data block of the technology object

See also

Tags of the positioning axis technology object as of V6 (Page 259)

CPU outputs relevant for motion control (Page 13)

Relationship between the signal type and the direction of travel (Page 17)

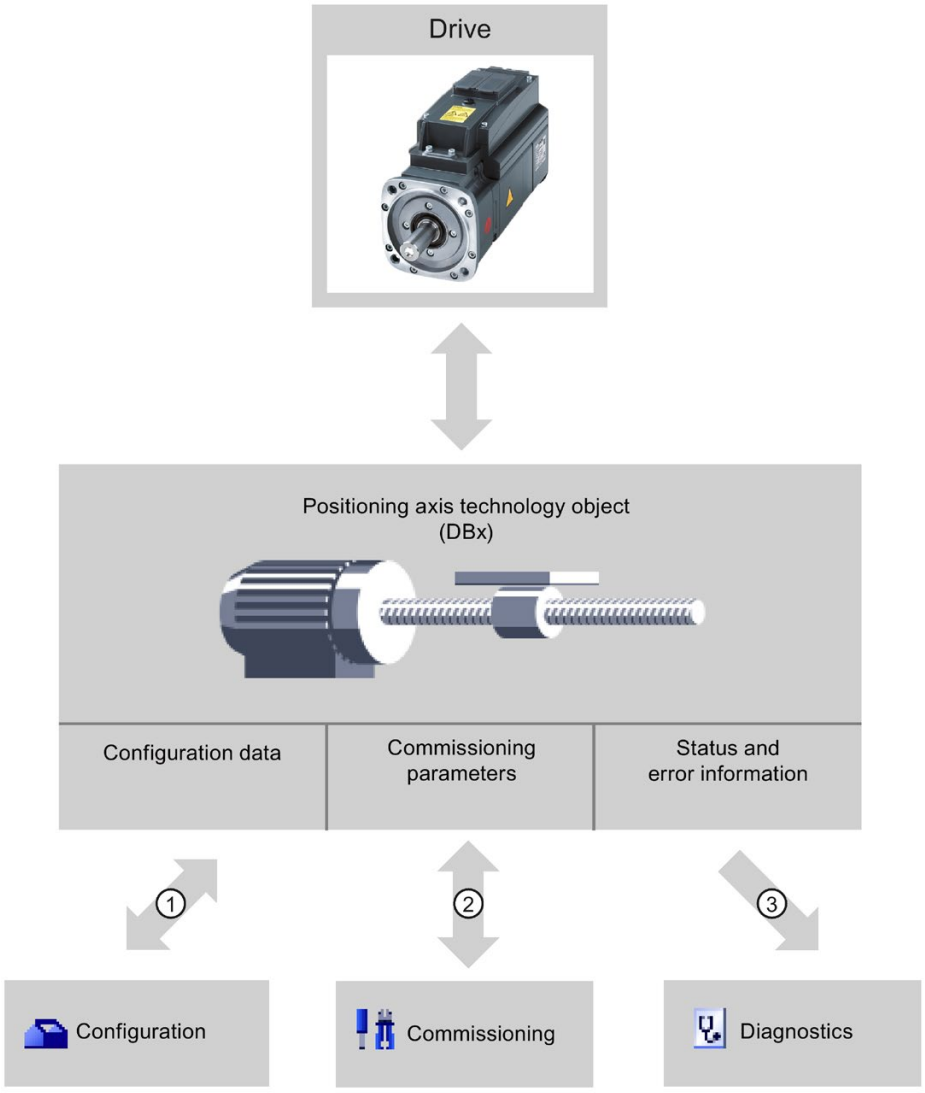
Tools of the positioning axis technology object (Page 56)

Hardware and software limit switches (Page 38)

Homing (Page 40)

5.2 Tools of the positioning axis technology object

The TIA Portal provides the "Configuration", "Commissioning", and "Diagnostics" tools for the positioning axis technology object. The following representation shows the interaction of the three tools with the technology object and the drive:



①	Reading and writing of configuration data of the technology object
②	Drive control via the technology object. Read the axis status for display in the axis control panel. Optimization of the position control
③	Readout of the current status and error information of the technology object Additional telegram information is displayed for PROFIdrive drives.

Configuration

Use the "Configuration" tool to configure the following properties of the positioning axis technology object:

- Selection of the PTOs (Pulse Train Output)/PROFIdrive drives/analog outputs to be used and configuration of the drive interface
- Properties of the mechanics and the transmission ratio of the drive (or machine/plant)
- Properties of the position limits and the position monitoring
- Properties of the dynamics and the homing
- Parameters of the control loop

Save the configuration in the data block of the technology object.

Commissioning

Use the "Commissioning" tool to test the function of your axis without having to create a user program. When the tool is started, the axis control panel will be displayed. The following commands are available on the axis control panel:

- Enabling and disabling the axis
- Move axis in jog mode
- Position axis in absolute and relative terms
- Home axis
- Acknowledge errors

The dynamic values can be adjusted accordingly for the motion commands. The axis control panel also shows the current axis status.

With drive connection via PROFIdrive / analog output, tuning supports you in determining the optimal gain for the control loop.

Diagnostics

Use the "Diagnostics" tool to keep track of the current status and error information for the axis and drive.

See also

CPU outputs relevant for motion control (Page 13)

Relationship between the signal type and the direction of travel (Page 17)

Integration of the positioning axis technology object (Page 53)

Hardware and software limit switches (Page 38)

Homing (Page 40)

Configuring the positioning axis technology object (Page 59)

Axis control panel (Page 146)

Axis - Diagnostics (Page 176)

5.3 Adding a positioning axis technology object

Requirements

A project with a CPU S7-1200 has been created.

Procedure

To add a positioning axis technology object in the project tree, follow these steps:

1. Open the "CPU > Technology objects" folder in the project tree.
2. Double-click the "Add new object" command.

   The "Add new object" dialog opens.
3. Select the "Motion Control" technology.
4. Open the "Motion Control" folder.
5. Select the desired technology version in the "Version" column.
6. Select the "TO_PositioningAxis" object.
7. Enter the name of the axis in the "Name" input box.
8. To change the automatically assigned data block number, select the "Manual" option.
9. To display additional information about the technology object, click "Additional information".
10. Confirm your entry with "OK".

Result

The new technology object is created and saved to the "Technology objects" folder in the project tree.

The organization blocks MC-Servo [OB91] and MC-Interpolator [OB92] are automatically created in the "Program blocks" folder. The technology objects are processed in these organization blocks. The position controller is calculated in the MC-Servo [OB91]. The MC-Interpolator [OB92] takes over the evaluation of the Motion Control instructions, the setpoint generation and the monitoring functionality.

See also

Guidelines on use of motion control (Page 42)

5.4 Configuring the positioning axis technology object

5.4.1 Working with the configuration dialog

You configure the properties of the technology object in the configuration window. Proceed as follows to open the configuration window of the technology object:

1. Open the group of the required technology object in the project tree.
2. Double-click the "Configuration" object.

The configuration is divided into the following categories:

- **Basic parameters**

  The basic parameters contain all the parameters which must be configured for a functioning axis.

- **Extended parameters**

  The advanced parameters include parameters to adapt to your drive or your plant.

Configuration window icons

Icons in the area navigation of the configuration show additional details about the status of the configuration:

	**The configuration contains default values and is complete.** The configuration contains only default values. With these default values you can use the technology object without additional changes.
	**The configuration contains user-defined or automatically adapted values and is complete.** All input fields of the configuration contain valid values and at least one preset value has changed.
	**The configuration is incomplete or incorrect** At least one input field or drop-down list contains an invalid value. The corresponding field or the drop-down list is displayed on a red background. Click the roll-out error message to display the cause of the error.
	**The configuration is valid but contains warnings** For example, only one hardware limit switch is configured. Depending on the application, the lacking configuration of a hardware limit switch may result in a hazard. The corresponding field or the drop-down list is displayed on a yellow background.

See also

Guidelines on use of motion control (Page 42)

Basic parameters (Page 61)

Extended parameters (Page 73)

5.4.2 Monitor values

If there is an online connection to the CPU, the icon "Monitor all"  is displayed in the configuration dialog of the technology object.

The "Monitor all" function provides the following options:

- Comparison of configured start values of the project with the start values in the CPU and the actual values
- Direct editing of actual values and the start values of the project
- Immediate detection and display of input errors with suggested corrections
- Backup of the actual values in the project by manual transfer to the start value of the project

Icons and operator controls

If there is an online connection to the CPU, the actual values are displayed at the parameters.

In addition to the actual values of the parameters, the following symbols appear:

Icon	Description
	Start value in CPU matches the configured Start value in the project
	Start value in CPU does not match the configured Start value in the project
	A comparison of the start value in the CPU with the configured start value in the project cannot be performed because the selected CPU module does not support this comparison.
	The value is not comparable with any significance since it is not relevant in one of the configurations.
	Use the button to show the start value of the CPU and the start value of the project for the respective parameter.

The actual value and the start value in the project can be changed directly and then downloaded to the CPU. The change of the actual value is transferred directly to the CPU for directly modifiable parameters.

5.4.3 Basic parameters

5.4.3.1 Configuration - General

Configure the basic properties of the positioning axis technology object in the "General" configuration window.

Axis name

Define the name of the axis or the name of the positioning axis technology object in this field. The technology object is listed under this name in the project tree.

Drive

Select the type of drive connection:

- **PTO (Pulse Train Output)**

  The drive is connected via a pulse generator output, an optional enable output and an optional ready input.

- **Analog drive connection**

  The drive is connected via an analog output, an encoder, an optional enable output and an optional ready input.

  All movements of the axis are position-controlled.

- **PROFIdrive**

  The drive is connected via PROFINET/PROFIBUS. Communication between controller and drive is by means of PROFIdrive telegrams.

  All movements of the axis are position-controlled.

If you select the "Analog drive connection" or "PROFIdrive", additional elements are added to the navigation of the configuration:

- Encoder
- Modulo
- Position supervisions (positioning supervision, following error and standstill signal)
- Control loop

In the additional configuration windows, you configure the encoders that are to be connected and the resulting options for position control and position monitoring.

Unit of measurement position

In the drop-down list, select the desired measurement unit for the dimension system of the axis. The selected measurement unit is used for further configuration of the positioning axis technology object and for displaying the current axis data.

The values at the input parameters (Position, Distance, Velocity, ...) of the Motion Control instructions also refer to this measurement unit.

Note

Select the drive connection and the measurement unit of the position at the beginning of the axis configuration.

With a subsequent change, the parameters are reset or re-initialized, which requires you to check the parameters of the configuration dialogs once again.

You may have to adapt the values of the input parameters of the Motion Control instructions to the new unit of measurement in the user program.

Simulation

In the drop-down list, select whether or not the drive and the encoder are to be simulated. The simulation can be selected for the analog drive connection or for a PROFIdrive drive. The configuration of the drive and encoder hardware is not required for simulation mode (potential errors in the drive and encoder configuration are ignored).

Application: The drive is simulated, for example, for commissioning and later operated with the hardware that may be configured.

The "Simulation" operating mode can be changed during runtime of the user program with a download and then MC_Reset with parameter "Restart" = TRUE.

In simulation mode, setpoints are not output to the drive and actual values are not read in from the drive/encoder. Hardware limit switches and homing switches have no effect.

The following table shows Motion Control instructions with adapted behavior in simulation mode.

Motion Control instruction	Behavior in simulation mode
MC_Power	The axis is enabled immediately without waiting for feedback from the drive.
MC_Home	Homing jobs are executed immediately without simulated axis motion.

PTO drives work without control loop. No separate simulation function is required in order to simulate a PTO drive when the PTO drive is not connected.

See also

CPU outputs relevant for motion control (Page 13)

Relationship between the signal type and the direction of travel (Page 17)

5.4.3.2 Configuration - Drive

Configuration - Drive - PTO (Pulse Train Output)

In the "Drive" configuration window, configure the pulse generator and the enable and feedback of the drive.

Hardware interface

The pulses are output to the power unit of the drive by fixed assigned digital outputs.

In CPUs with relay outputs, the pulse signal cannot be output at these outputs because the relays do not support the necessary switching frequencies. To be able to work with the PTO (Pulse Train Output) on these CPUs, you must use a signal board with digital outputs.

Note

The PTO requires the functionality of a high-speed counter (HSC). An internal HSC is used for this, the count of which cannot be evaluated.

Pulse generator

In the drop-down list, select the PTO (Pulse Train Output) to control the stepper motor or servo motor by means of pulse interface. If you have not used the pulse generators and high-speed counters elsewhere in the device configuration, the hardware interface can be configured automatically. In this case, the PTO selected in the drop-down list is displayed with a white background.

If PTO (PulseTrain Output) is selected, the "Device configuration" button takes you to the parameter assignment of the pulse options in the device configuration of the CPU. This may be necessary if there is a conflict because the PTO is being used at the other end or the parameters have been changed by the user.

Signal type

Select the signal type in the drop-down list. The following signal types are available:

- **PTO (pulse A and direction B)**

  A pulse output and a direction output are used for controlling the stepper motor.

- **PTO (clock up A and clock down B)**

  One pulse output each for motion in positive direction and negative direction is used for controlling the stepper motor.

- **PTO (A/B phase-shifted)**

  Both pulse outputs for Phase A and for Phase B run at the same frequency.
  The period of the pulse outputs is evaluated at the drive end as a step.
  The phase offset between Phase A and Phase B determines the direction of the motion.

- **PTO (A/B phase offset - quadruple)**

  Both pulse outputs for Phase A and for Phase B run at the same frequency.
  All positive edges and all negative edges of Phase A and Phase B are evaluated as a step at the drive end.
  The phase offset between Phase A and Phase B determines the direction of the motion.

The following table shows the parameters to be configured depending on the signal type:

Signal type/parameter		Description
PTO (pulse A and direction B)		
	Pulse output	Select the pulse output for motion in positive direction in this field. You can select the output using a symbolic address or assign it to an absolute address.
	Activate direction output	With this option, you enable or disable the direction output. The motion direction is restricted when you disable the direction output.
	Direction output	Select the output for the direction output in this field. You can select the output using a symbolic address or assign it to an absolute address.
PTO (clock up A and clock down B)		
	Pulse output forward	Select the pulse output for motion in positive direction in this field. You can select the output using a symbolic address or assign it to an absolute address.
	Pulse output backward	Select the pulse output for motion in negative direction in this field. You can select the output using a symbolic address or assign it to an absolute address.
PTO (A/B phase offset)/PTO (A/B phase offset - quadruple)		
	Signal A	Select the pulse output for Phase A signals in this field. You can select the output using a symbolic address or assign it to an absolute address.
	Signal B	Select the pulse output for Phase B signals in this field. You can select the output using a symbolic address or assign it to an absolute address.

Drive enable and feedback

In this area, you configure the output for drive enable and the input for the "Drive ready" feedback of the drive:

- **Enable output**

  Select the enable output for the drive enable in this field.

- **Ready input**

  Select the ready input for the "Drive ready" feedback of the drive in this field

Drive enable is controlled by Motion Control instruction "MC_Power" and enables power to the drive. The drive signals "Drive ready" to the CPU if it is ready to start executing movement after receiving the drive enable.

If the drive does not have any interfaces of this type, you do not have to configure the parameters. In this case, select the value TRUE for the ready input.

Configuration - Drive - Analog drive connection

In the "Drive" configuration window, configure the analog output and the enable and feedback of the drive.

Hardware interface

The speed setpoint is output to the power unit of the drive by means of permanently assigned analog outputs.

Configure the inputs and outputs for the control of the drive in this area:

- **Analog output**

  In this field, select the PLC variable of the analog output via which the drive is controlled.

  When you open the autocompletion, all output addresses are displayed with 16 bits (WORD, INT, UINT). The variable of a data block with the WORD data type can be selected for data connection via a data block.

  You can also enter an address, for example QW20. If the address is valid, the name "Axis_1_AnalogOutput" is generated for this address and inserted in the variable table. For the address to be valid, it needs to be occupied by the appropriate data type and a HW module.

- **Selection of enable output**

  Select an available output as the enable output for the drive enable in this field.

- **Selection of ready input**

  Select the ready input for the "Drive ready" feedback of the drive in this field

Drive enable is controlled by Motion Control instruction "MC_Power" and enables power to the drive. The drive signals "Drive ready" to the CPU if it is ready to start executing movement after receiving the drive enable. If the drive does not have any interfaces of this type, you do not have to configure the parameters. In this case, select the value TRUE for the ready input.

Data exchange with the drive

In this area, you can configure the scaling of the setpoint speed:

- **Reference speed**

  The reference speed of the drive is the speed, with which the drive spins when there is an output of 100% at the analog output. The reference speed must be configured in the drive, and transferred into the configuration of the technology object.

  The analog value that is output at 100% depends on the type of the analog output. As an example, for an analog output with ±10 V, the value 10 V is output at 100%.

- **Maximum speed**

  In this field, specify the maximum speed of the drive.

  The maximum speed is limited by the drive and by the value range of the analog output. In the simplest situation, the reference speed and maximum speed are identical.

  Analog outputs can be overloaded by approximately 17%. If the drive permits overloading, you can use this to operate an analog output as a limit in the -117% to 117% range.

- **Invert drive direction**

  To invert the rotation direction of the drive, select the check box.

Configuration - Drive - PROFIdrive

In the "Drive" configuration window, configure the data connection and the parameters of the PROFIdrive drive.

PROFIdrive drive (as of V6)

- **Data connection**

  In the drop-down list, select whether the data connection is to be made directly with the drive device or via an editable data block in the user program.

- **Drive** (for data connection: "Drive")

  In the "Drive" field, select an already configured PROFIdrive drive.

- **Data block** (for data connection: "Data block")

  Select a previously created data block which contains a variable structure of the data type "PD_TELx" ("x" stands for the telegram number to be used).

Data exchange with the drive

In this area, you can configure the data exchange between the drive and controller:

- **Drive telegram** (for data connection: "Data block" not selectable)

  In the drop-down list, check or select the telegram of the drive. The specification must match the device configuration of the drive.

- **Input/output address**

  The fields show the symbolic and absolute input and output address of the telegram.

- **Invert drive direction**

  To invert the rotation direction of the drive, select the check box.

- **Automatic transfer of drive parameters in the device**

  Select the check box if you want the drive parameters "Reference speed" and "Maximum speed" to be transferred as values from the drive configuration to the CPU. The drive parameters are transferred from the bus after the (re-)initialization of the technology object and (re)start of the drives and the CPU.

  Alternatively, you must synchronize the following parameters manually:

  – **Reference speed**

    Configure the reference speed to match the one in the configuration of the drive.

    On the bus, the value 16#4000 is transferred, for example, which corresponds to 100% of the reference speed.

  – **Maximum speed**

    Configure the maximum speed of the drive in this field.

    The maximum speed is obtained from the configuration of the drive. A maximum of -200% to +200% of the reference speed can be transmitted over the bus. The maximum speed can thus be twice the reference speed at maximum.

Note

Automatic transfer of drive parameters is only possible with SINAMICS drives as of V4.x. For this, "Drive" must be selected as the data connection in the configuration window.

See also

Data connection PROFIdrive drive/PROFIdrive encoder (Page 26)

Automatic transfer of drive and encoder parameters in the device (Page 22)

5.4.3.3 Configuration - Encoder

Encoder connection

Depending on the selection of the encoder connection, you configure various parameters in the "Encoder" configuration window. The following encoder connections are possible:

- Encoder on high-speed counter (HSC) (Page 71)
- PROFIdrive encoder on PROFINET/PROFIBUS (encoder on drive, encoder on technology module, PROFIdrive encoder) (Page 68)

Configuration - Encoder - Encoder on PROFINET/PROFIBUS

Encoder selection

In the "PROFIdrive encoder" box, select a PROFIdrive encoder on PROFINET.

- **Data connection**

  In the drop-down list, select whether the data connection should be established directly with the encoder or via a data block that can be edited in the user program.

- **PROFIdrive encoder/data block**

  Select a previously configured PROFIdrive encoder in this configuration field.

  The following encoders can be selected:

  – **Connection to drive (not with analog drive connection)**

    The encoder is connected to the drive. The encoder signals are evaluated by the drive and transmitted to the controller as part of the drive telegram (telegram 3 or 4) (the encoders of the telegrams from other drives cannot be used).

    The encoder is configured using the configuration of the PROFIdrive drive.

  – **Encoder on technology module (TM)**

    Select a previously configured technology module and the channel to be used. Only technology modules set to the "Position input for Motion Control" mode are displayed for selection.

    If no technology module is available for selection, change to the device configuration and add a technology module.

    You can identify the technology modules suitable for position detection for Motion Control in the documentation for the technology module and the catalog data.

  – **PROFIdrive encoder on PROFINET/PROFIBUS (PROFIdrive)**

    In the "PROFIdrive encoder" field, select a previously configured encoder on PROFINET/PROFIBUS. Switch to the device configuration in the network view and add an encoder, in the event that no encoder is offered for selection.

If "Data block" was selected for the data connection, a previously created data block containing a tag structure of data type "PD_TELx" must be selected here ("x" stands for the telegram number to be used by which the encoder is connected). The encoder of the selected drive telegram (Tel 3 or 4) or a separate encoder (Tel 81 or 83) can be used.

Data exchange with encoder

In this area, you can configure the data exchange between the encoder and controller:

- **Encoder telegram** (for data connection: "Data block" not selectable)

  In the drop-down list, select the telegram of the encoder. The specification must match the device configuration.

- **Input/output address**

  The fields show the symbolic and absolute input and output address of the telegram.

- **Invert encoder direction**

  To invert the actual value of the encoder, select the check box.

- **Automatic transfer of encoder parameters in the device**

  Select the check box if you want to transfer the encoder parameters from the encoder configuration to the CPU. The encoder parameters are transferred from the bus after the (re-)initialization of the technology object and (re)start of the encoder and the CPU. The encoder type must be the same in the configuration of the axis and in the configuration of the drive.

  Note

  Automatic transfer of encoder parameters is only possible with PROFIdrive encoders as of product version A16. For this, "Encoder" must be selected as the data connection in the configuration window.

  A product version > V4.x is required to use an encoder on the SINAMICS drive.

  The parameters must be adjusted manually if there is no automatic transfer of encoder parameters. You can find the parameters to be synchronized in the section Automatic transfer of drive and encoder parameters in the device (Page 22).

Encoder type

Set the employed encoder type in the "Encoder type" box. The following encoder types can be selected:

- **Linear incremental**
- **Linear absolute**
- **Rotary incremental**
- **Rotary absolute**

Configure the various parameters depending on the selected encoder type. Depending on the selected encoder type, configure the following parameters:

Encoder type/parameter		Description
Linear incremental		
	Distance between two increments	In this field, you configure the distance between two steps of the encoder.
	Fine resolution - Bits in incr. actual value (Gn_XIST1)	In this field, configure the number of bits for fine resolution within the incremental actual value (Gn_XIST1).
Linear absolute		
	Distance between two increments	In this field, you configure the distance between two steps of the encoder.
	Fine resolution - Bits in incr. actual value (Gn_XIST1)	In this field, configure the number of bits for fine resolution within the incremental actual value (Gn_XIST1).
	Fine resolution - Bits in abs. actual value (Gn_XIST2)	In this field, configure the number of reserved bits for the multiplication factor of the absolute value of the fine resolution (Gn_XIST2).
Rotary incremental		
	Steps per revolution	In this field, configure the number of steps that the encoder resolves per revolution.
	Fine resolution - Bits in incr. actual value (Gn_XIST1)	In this field, configure the number of bits for fine resolution within the incremental actual value (Gn_XIST1).
Rotary absolute		
	Steps per revolution	In this field, configure the number of steps that the encoder resolves per revolution.
	Number of revolutions	In this field, configure the number of revolutions that the absolute value encoder can detect.
	Fine resolution - Bits in incr. actual value (Gn_XIST1)	In this field, configure the number of bits for fine resolution within the incremental actual value (Gn_XIST1).
	Fine resolution - Bits in abs. actual value (Gn_XIST2)	In this field, configure the number of reserved bits for the multiplication factor of the absolute value of the fine resolution (Gn_XIST2).

See also

Data connection PROFIdrive drive/PROFIdrive encoder (Page 26)

Configuring technology modules for Motion Control (Page 119)

Configuration - Encoder - Encoder on high-speed counter (HSC)

Selection of high-speed counter (HSC)

In the high-speed counter box, select the high-speed counter to which the encoder transfers the actual value.

Check the filter times of the two high-speed counter digital inputs that are used. The filter times should be short enough to ensure reliable recording of the pulses.

HSC interface

Select the operating mode of the high-speed counter in the "Operating mode" box.

Depending on the operating mode, configure the various inputs:

Operating mode/parameter		Description
Two-phase		
	Clock generator forward	In this field, select the input for counting up. You can select the input using a symbolic address or assign it to an absolute address. The frequency and the location (on-board, signal board) of the input are displayed next to the address box.
	Clock generator backward	In this field, select the input for counting down. You can select the input using a symbolic address or assign it to an absolute address. The frequency and the location (on-board, signal board) of the input are displayed next to the address box.
A/B counter / A/B counter quadruple		
	Clock generator A	In this field, select the input for Phase A signals. You can select the input using a symbolic address or assign it to an absolute address. The frequency and the location (on-board, signal board) of the input are displayed next to the address box.
	Clock generator B	In this field, select the input for Phase B signals. You can select the input using a symbolic address or assign it to an absolute address. The frequency and the location (on-board, signal board) of the input are displayed next to the address box.

Invert encoder direction

To invert the actual value of the encoder, select the check box.

Automatic transfer of encoder parameters in the device

This selection is not possible when using encoders on the high-speed counter (HSC).

Encoder type

Select the encoder type in the "Encoder type" box. The following encoder types can be selected:

- **Linear incremental**
- **Rotary incremental**

Configure the various parameters depending on the selected encoder type. Depending on the selected encoder type, configure the following parameters:

Encoder type/parameter		Description
Linear incremental		
	Distance between two increments	In this field, you configure the distance between two steps of the encoder.
	Fine resolution - Bits in incr. actual value (Gn_XIST1)	In this field, configure the number of bits for fine resolution within the incremental actual value (Gn_XIST1).
Rotary incremental		
	Steps per revolution	In this field, configure the number of steps that the encoder resolves per revolution.
	Fine resolution - Bits in incr. actual value (Gn_XIST1)	In this field, configure the number of bits for fine resolution within the incremental actual value (Gn_XIST1).

5.4.4 Extended parameters

5.4.4.1 Mechanics

Configuration - Mechanics - PTO (Pulse Train Output)

Configure the mechanical properties of the drive in the "Mechanics" configuration window.

Pulses per motor revolution

Configure the number of pulses required for one revolution of the motor in this box.

Limits (independent of the selected unit of measurement):

- 0 < Pulse per motor revolution ≤ 2147483647

Load motion per motor revolution

In this box, configure the load distance per motor revolution covered by the mechanical system of your unit.

Limits (independent of the selected unit of measurement):

- 0.0 < Distance per revolution ≤ 1.0e12

Permitted direction of rotation (technology version as of V4)

Configure this box to determine whether the mechanics of your system are to move in both directions or only in the positive or negative direction.

If you have not activated the direction output of the pulse generator in the "PTO (pulse A and direction B)" mode, the selection is limited to the positive or negative direction.

Invert direction

You can use the "Invert direction" check box to adapt the control system to the direction logic of the drive.

The direction logic is inverted according to the selected mode of the pulse generator:

- **PTO (pulse A and direction B)**
  - 0 V at direction output ⇒ positive direction of rotation
  - 5 V/24 V at direction output ⇒ negative direction of rotation

  The specified voltage depends on the hardware used. The indicated values do not apply to the differential outputs of CPU 1217.
- **PTO (clock up A, clock down B)**

  The outputs "Pulse output down" and "Pulse output up" are swapped.
- **PTO (A/B phase-shifted)**

  The "Phase A" and "Phase B" outputs are swapped.
- **"PTO (A/B phase-shifted - quadruple)**

  The "Phase A" and "Phase B" outputs are swapped.

Configuration - Mechanics - PROFIdrive/analog drive connection

Configure the mechanical properties of the drive and its encoder in the "Mechanics" configuration window.

Encoder mounting type

In the drop-down list, select how the encoder is mounted on the mechanism. The following encoder installation types are possible:

- **On motor shaft**
- **External measuring system** (rotary encoders only)

Position parameters

Depending on the selected encoder installation type, configure the following position parameters:

Encoder installation type/position parameter		Description
On the motor shaft		
	Load motion per motor revolution	In this field, configure the load distance for one motor revolution.
External measuring system		
	Load motion per motor revolution	In this field, configure the load distance for one motor revolution.
	Distance per encoder revolution	In this field, configure the distance recorded by the external measuring system per encoder revolution.

5.4.4.2 Configuration - Modulo (PROFIdrive/analog drive connection only)

Use the "Modulo" setting if you want to limit the traversing range to a recurring distance based on the product length / product cycle. The modulo function is only possible in position-controlled operation of the axis.

When "modulo" is enabled, the position value of the technology object is represented by means of a recurring modulo range. The modulo range is defined by the start value and the length.

For example, to limit the position value of an axis to one full rotation, the modulo range can be defined with start value = 0° and length = 360°. With an encoder resolution of 0.1°/encoder step, the position value is represented in the modulo range 0.0° to 359.9°. If the axis in this example moves to the position 400°, the actual position 40° (400° to 360°) is reached.

When "Modulo" is activated, specify the traversing direction at the Motion Control instruction "MC_MoveAbsolute" with the "Direction" input parameter. The following parameter values are available:

- 0: The sign for the velocity ("Velocity" parameter) determines the motion direction.
- 1: Target position is approached in a positive direction.
- 2: Target position is approached in a negative direction.
- 3: Starting from the current position, the technology selects the shortest distance to the target position.

Enable modulo

Select the "Enable modulo" check box to use a recurring reference system for the axis (for example, 0.0° to 359.9°).

Modulo start value

In this field, define the position at which the modulo range should begin (for example, 0°).

Modulo length

In this field, define the length of the modulo range (for example, 360°).

See also

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

5.4.4.3 Position limits

Requirements for hardware limit switches

Use only hardware limit switches that remain permanently switched after being approached. This switching status may only be revoked after a return to the valid travel range.

See also

Configuration - Position limits (Page 76)

Response of the axis when position limits are triggered (Page 78)

Changing the configuration of the position limits in the user program (Page 80)

Configuration - Position limits

Configure the hardware and software limit switches of the axis in the "Position limits" configuration window.

Enable HW limit switches

Activate the function of the low and high hardware limit switch with this check box. The hardware limit switches can be used for purposes of direction reversal during a homing procedure. For details, refer to the configuration description for homing.

Enable SW limit switch

Activate the function of the low and high software limit switch with this check box.

Note

Activated software limit switches act only on a homed axis.

Input HW low / high limit switch

Select the digital input for the low or hardware high limit switch from the drop-down list.

The input of PTO axes must be interrupt-capable. You achieve the shortest response time with interrupt-capable inputs with a drive connection via PROFIdrive / analog drive connection. Alternatively, you can assign the input to the "TPA OB Servo" process image and then receive a response time in the cycle time of the "TPA OB Servo". The assignment of the standard process image of the organization block OB1 is not recommended, as the longest response times occur here.

The digital onboard CPU inputs and the digital inputs of a plugged signal board can be selected as interrupt-capable inputs for the HW limit switches.

Note

The digital inputs are set to a filter time of 6.4 ms by default. If these are used as hardware limit switches, undesired decelerations may occur. If this occurs, reduce the filter time for the relevant digital inputs.

The filter time can be set under "Input filter" in the device configuration of the digital inputs.

Select level

In the drop-down list, select the signal level available at the CPU when the hardware limit switch is approached.

- Selection of "Low level" (normally closed contact)

  0 V (FALSE) at CPU input corresponds to hardware limit switch approached
- Selection of "High level" (normally open contact)

  5 V / 24 V (TRUE) at the CPU input = hardware limit switch approached (the actual voltage depends on the hardware used)

Software high / low limit switch

Enter the position value of the low and high software limit switch in these boxes.

Limits (independent of the selected unit of measurement):

- -1.0e12 ≤ software low limit switch ≤ 1.0e12
- -1.0e12 ≤ software high limit switch ≤ 1.0e12

The value of the software high limit switch must be greater than or equal to the value of the software low limit switch.

See also

Requirements for hardware limit switches (Page 76)

Response of the axis when position limits are triggered (Page 78)

Changing the configuration of the position limits in the user program (Page 80)

Configuration - Homing - Active (Page 88)

Response of the axis when position limits are triggered

Behavior of axis when hardware limit switches are approached

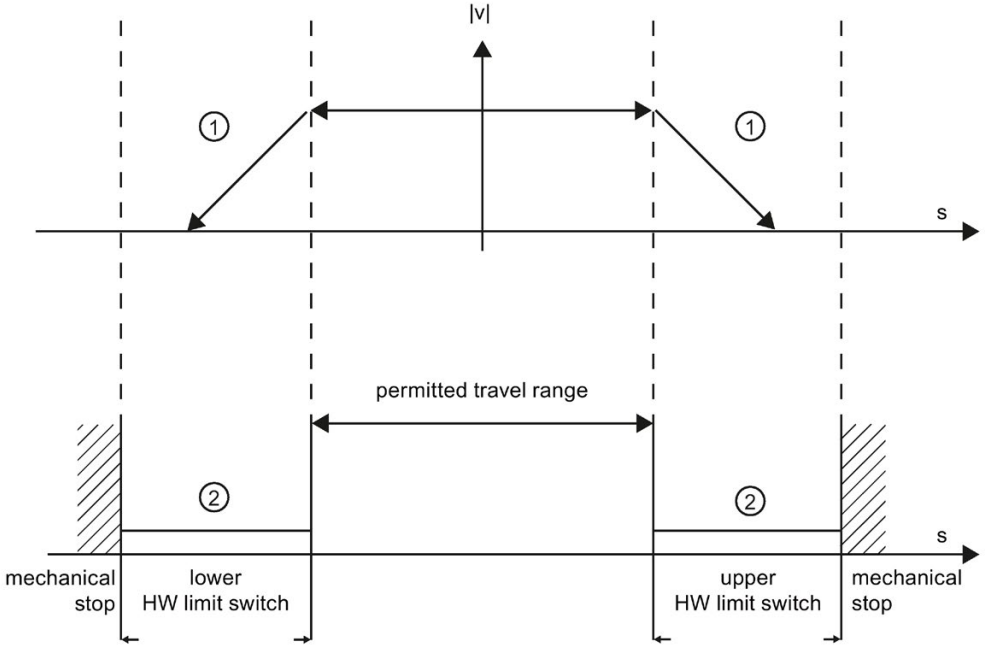
When a hardware limit switch is approached, the axis behaves differently depending on the drive connection:

- Drive connection via PROFIdrive / analog output

  When a hardware limit switch is approached, the axis is disabled and, depending on the configuration, braked at the drive and brought to a standstill. You must select the deceleration sufficiently large in the drive so that the axis stops reliably before the mechanical stop.

- Drive connection via PTO (Pulse Train Output)

  When the hardware limit switches are approached, the axis brakes to a standstill at the configured emergency deceleration. You must select the emergency deceleration sufficiently large so that the axis stops reliably before the mechanical stop. The following diagram presents the behavior of the axis after it approaches the hardware limit switches:



① The axis brakes at the configured emergency stop deceleration (PTO) or the deceleration configured in the drive (PROFIdrive or analog drive interface) until standstill occurs.

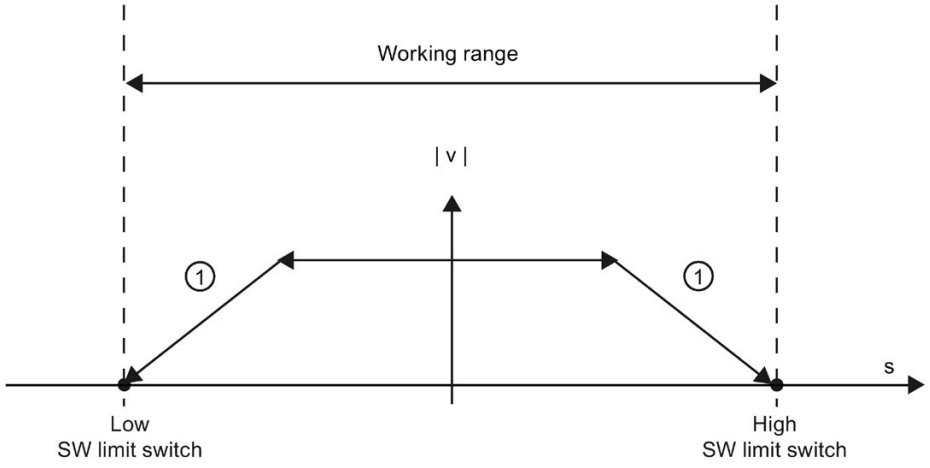
② Range in which the HW limit switches signal the status "approached".

The "HW limit switch approached" error is displayed in the initiating Motion Control instruction, at "MC_Power", and in the technology object variables. Instructions for eliminating errors can be found in the Appendix under "List of ErrorIDs and ErrorInfos".

Behavior of axis when software limit switches are reached

If software limit switches are activated, an active motion is stopped at the position of the software limit switch. The axis is braked at the configured deceleration.

The following diagram presents the behavior of the axis until it reaches the software limit switches:



① The axis brakes to a standstill at the configured deceleration.

The "SW limit switch approached" error is displayed in the initiating Motion Control instruction, at "MC_Power", and in the technology object variables. Instructions for eliminating errors can be found in the Appendix under "List of ErrorIDs and ErrorInfos".

When a software limit switch is overtraveled, the axis behaves differently depending on the drive connection:

- Drive connection via PROFIdrive / analog output

  When a software limit switch is overtraveled, the axis is disabled and, depending on the configuration, braked at the drive and brought to a standstill.

- Drive connection via PTO (Pulse Train Output)

  You can learn about the behavior of the axis when a software limit switch is overtraveled in the sections "Software limit switches in conjunction with a homing operation (Page 227)" and "Software limit switches in conjunction with dynamic changes (Page 232)".

Use additional hardware limit switches if a mechanical endstop is located after the software limit switches and there is a risk of mechanical damage.

See also

Requirements for hardware limit switches (Page 76)

Configuration - Position limits (Page 76)

Changing the configuration of the position limits in the user program (Page 80)

Changing the configuration of the position limits in the user program

You can change the following configuration parameters during runtime of the user program in the CPU:

Hardware limit switches

You can also activate and deactivate the hardware limit switches during runtime of the user program. Use the following technology object variable for this purpose:

- <axis name>.PositionLimits_HW.Active

Refer to the description of the technology object variables (Page 259) in the appendix for information on when changes to the configuration parameter take effect.

Software limit switches

You can also activate and deactivate the software limit switches and change their position values during runtime of the user program. Use the following technology object variables for this purpose:

- <axis name>.PositionLimits_SW.Active

  for activating and deactivating the software limit switches
- <axis name>.PositionLimits_SW.MinPosition

  for changing the position of the low software limit switch
- <axis name>.PositionLimits_SW.MaxPosition

  for changing the position of the high software limit switch

Refer to the description of the technology object variables (Page 259) in the appendix for information on when changes to the configuration parameters take effect.

See also

Compatibility list of variables V1...3 <-> V4...5 (Page 48)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

Requirements for hardware limit switches (Page 76)

Configuration - Position limits (Page 76)

Response of the axis when position limits are triggered (Page 78)

5.4.4.4 Dynamics

Configuration - Dynamics - General

Configure the maximum velocity, the start/stop velocity, the acceleration and deceleration, and the jerk limit (positioning axis technology object as of V2) of the axis in the "General dynamics" configuration window.

Unit of velocity limits

Select the unit of measurement with which you want to set the velocity limits in the drop-down list. The unit set here depends on the unit of measurement set under "Configuration > Basic parameters > General" and only serves to simplify input. This provides the possibility, for example, to enter the maximum velocity as a speed value of the motor in rpm.

Note

Rounding error

If you select a different unit in the "Unit of velocity limitation" drop-down list than in "Configuration > Basic parameters > General", note that a rounding error may occur.

Maximum velocity / Start/stop velocity

Define the maximum permissible velocity and the start/stop velocity of the axis in these boxes. The start/stop velocity is the minimum permissible velocity of the axis and can only be configured for drive connection via PTO (Pulse Train Output).

For drive connection via PROFIdrive or analog output, the start/stop velocity is fixed at zero. The maximum velocity when connecting via PROFdrive or analog output amounts to 1.0E12 of the selected measurement unit (e.g. mm/s, °/s, ...).

Positioning axis technology object (PTO) as of V4

Signal board	Velocity [pulse/s]
20 kHz	1 ≤ start/stop velocity ≤ 20 000 1 ≤ maximum velocity ≤ 20 000
200 kHz	1 ≤ start/stop velocity ≤ 200 000 1 ≤ maximum velocity ≤ 200 000

On-board CPU output	Velocity [pulse/s]
100 kHz	1 ≤ start/stop velocity ≤ 100 000 1 ≤ maximum velocity ≤ 100 000
20 kHz	1 ≤ start/stop velocity ≤ 20 000 1 ≤ maximum velocity ≤ 20 000
1 MHz CPU 1217	1 ≤ start/stop velocity ≤ 1000 000 1 ≤ maximum velocity ≤ 1000 000

You can learn about the limits for the technology object positioning axis < V4 in the appendix Outputs of the CPU relevant for Motion Control (technology version V1...3).

The value of the maximum velocity must be greater or equal to the value of the start/stop velocity.

The limits for other units of measurement must be converted by the user to conform to the given mechanics.

Acceleration / Deceleration - Ramp-up time / Ramp-down time

Set the desired acceleration in the "Ramp-up time" or "Acceleration" boxes. The desired deceleration can be set in the "Ramp-down time" or "Deceleration" boxes.

The relation between the ramp-up time and acceleration and the ramp-down time and deceleration is shown in the following equations:

$$\text{Ramp-up time} = \frac{\text{Maximum velocity} - \text{Start/stop velocity}}{\text{Acceleration}}$$

$$\text{Ramp-down time} = \frac{\text{Maximum velocity} - \text{Start/stop velocity}}{\text{Deceleration}}$$

Motion jobs started in the user program are performed with the selected acceleration / deceleration.

The limits for acceleration and deceleration with drive connection via PTO (Pulse Train Output) can be found in section CPU outputs relevant for motion control (Page 13).

Note

Changes to the velocity limits ("start/stop velocity" and "maximum velocity") influence the acceleration and deceleration values of the axis. The ramp-up and ramp-down times are retained.

Enable jerk limit, positioning axis technology object (as of V2)

Enable the jerk limit with this check box.

Axis acceleration and deceleration is not stopped abruptly when the jerk limit is activated; it is adjusted gently according to the set step or smoothing time.

Note

The check box is no longer displayed as a parameter in the technology data block as of V4. By disabling the check box, the jerk value is set to 0.0.

Rounding time/jerk, positioning axis technology object (as of V2)

You can input the parameters of the jerk limit in the "Smoothing time" box or alternatively in the "Jerk" box.

- Set the desired jerk for acceleration and deceleration ramp in the "Jerk" box.
- Set the desired smoothing time for the acceleration ramp in the "Smoothing time" box.

Note

Smoothing time V2...3

The set smoothing time visible in the configuration only applies to the acceleration ramp.

If the values for acceleration and deceleration differ, the smoothing time of the deceleration ramp is calculated according to the jerk of the acceleration ramp and used. (See also Behavior of the axis when using the jerk limit (Page 85))

The smoothing time of the deceleration is adapted as follows:

- Acceleration > deceleration

  The smoothing time used for the deceleration ramp is shorter than that for the acceleration ramp.
- Acceleration < deceleration

  The smoothing time used for the deceleration ramp is greater than that for the acceleration ramp.
- Acceleration = deceleration

  The smoothing times of the acceleration and deceleration ramp are equal.

The relation between smoothing times and jerk is shown in the following equation:

$$\text{Smoothing time (acceleration ramp)} = \frac{\text{Acceleration}}{\text{Jerk}}$$

$$\text{Smoothing time (deceleration ramp)} = \frac{\text{Deceleration}}{\text{Jerk}}$$

Motion jobs started in the user program are performed with the selected jerk.

The limits for jerk with drive connection via PTO (Pulse Train Output) can be found in section CPU outputs relevant for motion control (Page 13).

For PROFIdrive drives and drives with analog drive interface, the limit is 1E12.

See also

Behavior of the axis when using the jerk limit (Page 85)

Hardware components for motion control (Page 10)

CPU outputs relevant for motion control (Page 13)

Configuration - Dynamics - Emergency stop (Page 84)

Changing the configuration of dynamics in the user program (Page 86)

Configuration - Dynamics - Emergency stop

Configure the emergency stop deceleration of the axis in the "Dynamics emergency stop" configuration window. In the event of an error, and when disabling the axis, the axis is brought to a standstill with this deceleration using the Motion Control instruction "MC_Power" (input parameter StopMode = 0 or 2).

Velocity

The velocity values configured in the "General dynamics" configuration window are once again displayed in this information area.

Deceleration

Set the deceleration value for emergency stop in the "Emergency deceleration" or "Emergency stop ramp-down time" field.

The relation between emergency stop ramp-down time and emergency deceleration is shown in the following equation:

$$\textit{Emergency stop ramp-down time} = \frac{\textit{Maximum velocity - Start/stop velocity}}{\textit{Emergency deceleration}}$$

The specified emergency deceleration must be sufficient to bring the axis to a standstill in a timely manner in the event of an emergency (for example, when the hardware limit switch is approached prior to reaching the mechanical endstop).

The configured maximum velocity of the axis must be used as a basis for selecting the emergency deceleration.

Limit values:

The limits indicated below refer to the "Pulses/s^2" unit of measurement.

- As of CPU firmware V3

  0.005 ≤ emergency deceleration ≤ 9.5E9

- CPU Firmware V1...2

  0.28 ≤ emergency deceleration ≤ 9.5E9

The limits for other units of measurement must be converted to conform to the given mechanics.

The limits for jerk with drive connection via PTO (Pulse Train Output) can be found in section CPU outputs relevant for motion control (Page 13).

For PROFIdrive drives and drives with analog drive interface, the limit is 1.0E12.

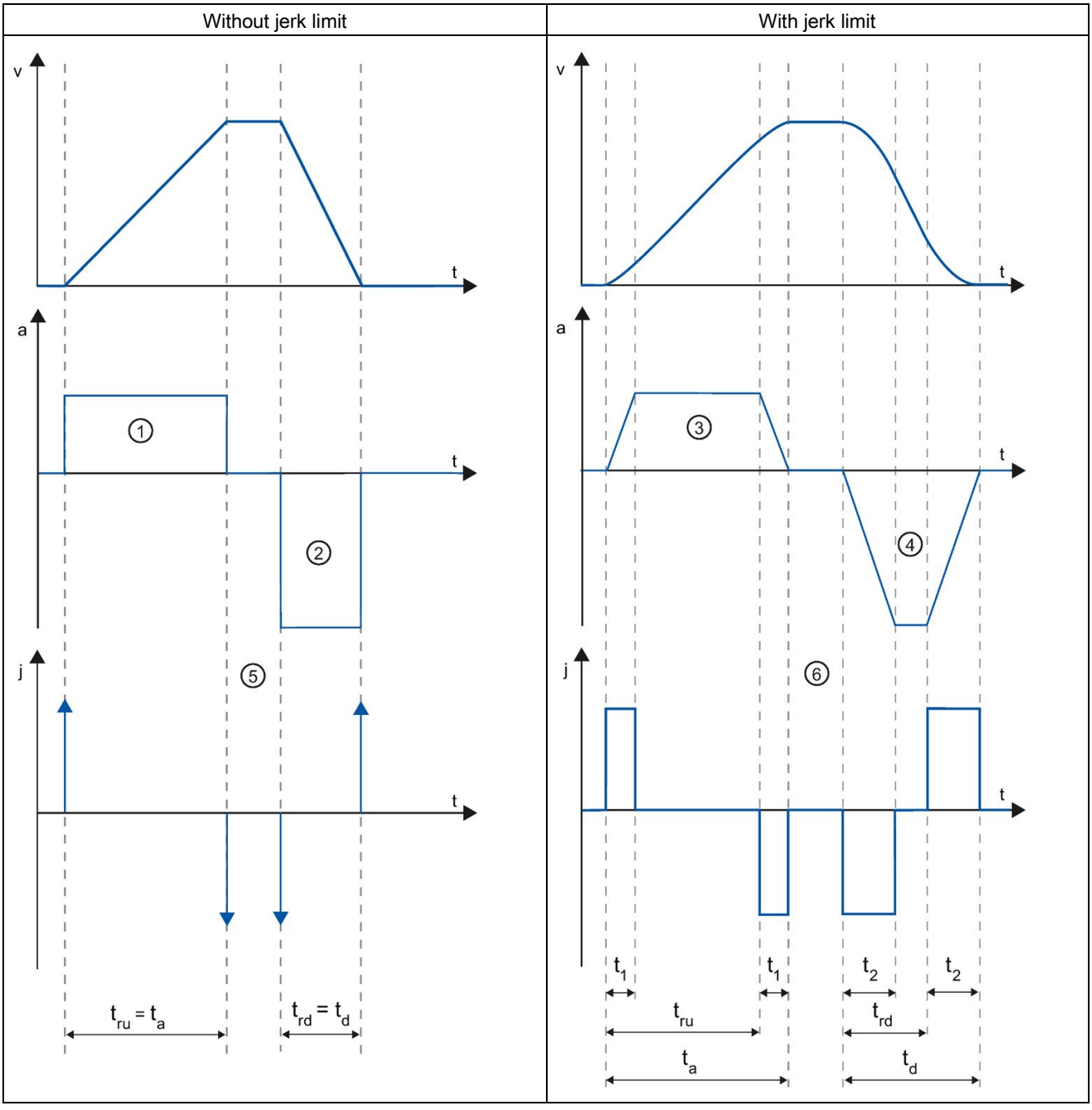
See also

Configuration - Dynamics - General (Page 81)

Changing the configuration of dynamics in the user program (Page 86)

Behavior of the axis when using the jerk limit

Axis acceleration and deceleration is not stopped abruptly when the jerk limit is activated; it is adjusted gently according to the set step or smoothing time. The diagram below details the behavior of the axis with and without activated jerk limit:



t	Time axis
v	Velocity
a	Acceleration
j	Jerk
t_{ru}	Rampup time
t_a	Time taken for the axis to accelerate
t_{rd}	Deceleration time
t_d	Time taken for the axis to decelerate
t_1	Smoothing time of the acceleration ramp
t_2	Smoothing time of the deceleration ramp

The example shows travel in which the deceleration value ② is twice the acceleration value ①. The resulting ramp-down time t_{rd} is therefore only half the length of the ramp-up time t_{ru}.

Acceleration ① and deceleration ② change abruptly without a jerk limit. Acceleration ① and deceleration ② change gradually with activated jerk limiter. As the jerk applies to entire motion, the rate is the same for the increase and decrease in acceleration and deceleration.

The step value j becomes infinitely high ⑤ as soon as the change is made without jerk limit. The step is limited to the configured value ⑥ when the jerk limit is activated.

The smoothing time t_1 given in the configuration applies to the acceleration ramp. The deceleration ramp smoothing time t_2 is calculated from the configured jerk value and the configured deceleration.

See also

Configuration - Dynamics - General (Page 81)

Changing the configuration of dynamics in the user program

You can change the following configuration parameters during runtime of the user program in the CPU:

Acceleration and deceleration

You can also change the values for acceleration and deceleration during runtime of the user program. Use the following technology object variables for this purpose:

- <Axis name>.DynamicDefaults.Acceleration

  for changing acceleration

- <Axis name>.DynamicDefaults.Deceleration

  for changing deceleration

Refer to the description of the technology object variables (Page 259) in the appendix for information on when changes to the configuration parameters take effect.

Emergency stop deceleration

You can also change the value for the emergency stop deceleration during runtime of the user program. Use the following technology object variable for this purpose:

- <Axis name>.DynamicDefaults.EmergencyDeceleration

Refer to the description of the technology object variables in the appendix for information on when changes to the configuration parameter take effect.

Note

After changes to this parameter, it may be necessary to adapt the positions of the hardware limit switches and other safety-relevant settings.

Jerk limit

You can also activate and deactivate the jerk limit at runtime of the user program and change the value for the jerk. To do this, use the technology object tag <axis name>.DynamicDefaults.Jerk For technology objects < V4, the tag <axis name>.Config.DynamicDefaults.JerkActive must be set to TRUE in order to activate the jerk limitation and in order that a value change is visible/effective at the jerk.

The following applies to PTO axes:

- If you enter a value ≥ 0.004 pulses/s^3 for the jerk, the jerk limit is enabled with the specified value.
- If you enter a value < 0.004 pulses/s^3 for the jerk, the jerk limit is disabled.

For position-controlled axes, the jerk limit is disabled for a value of 0.0, and activated for values > 0.0.

Refer to the description of the technology object variables in the appendix for information on when changes to the configuration parameter take effect.

See also

Compatibility list of variables V1...3 <-> V4...5 (Page 48)

Configuration - Dynamics - General (Page 81)

Configuration - Dynamics - Emergency stop (Page 84)

5.4.4.5 Homing (positioning axis technology object as of V2)

Configuration - Homing - Active

Configure the necessary parameters for active homing in the "Active homing" configuration window. Active homing is started using Motion Control instruction "MC_Home" with input parameter "Mode" = 3.

Select homing mode (drive connection via PROFIdrive V5 or higher only)

Select one of the following homing modes:

- Use zero mark via PROFIdrive telegram and proximity switch
- Use zero mark via PROFIdrive telegram
- Use homing mark via digital input

If you have selected drive connection via PTO (Pulse Train Output) or analog output with HSC as the encoder, the only homing mode available is "Use homing mark via digital input".

Digital inputs

In this area, you configure the homing switch:

- **Input homing switch**

  Select the digital input for the homing switch in this field.

  Note

  The digital inputs are set to a filter time of 6.4 ms by default.

  When the digital inputs are used as a homing switch, this can result in undesired decelerations and thus inaccuracies. Depending on the homing velocity and extent of the homing switch, the home position may not be detected. The filter time can be set under "Input filter" in the device configuration of the digital inputs.

  The specified filter time must be less than the duration of the input signal at the homing switch.

  For drive connection via PTO (Pulse Train Output):

  The input must be interrupt-capable. The onboard CPU inputs and the inputs of an inserted signal board can be selected as inputs for the homing switch.

- **Select level**

  In the drop-down list, select the level of the homing switch that is to be used for homing.

- **Permit auto reverse at HW limit switch**

  Activate the check box to use the hardware limit switch as a reversing cam for the homing procedure. The hardware limit switches must be enabled for the reversal of direction (at least the hardware limit switch in the direction of approach must be configured).

If the hardware limit switch is reached during active homing, the axis brakes at the configured deceleration (not with the emergency stop deceleration) and reverses direction. The homing switch is then sensed in reverse direction.

If the direction reversal is not active and the axis reaches the hardware limit switch during active homing, the homing procedure is aborted with an error and the axis is braked at the emergency stop deceleration.

Note

If possible, use one of the following measures to ensure that the machine does not travel to a mechanical endstop in the event of a direction reversal:

- Keep the approach velocity low.
- Increase the configured acceleration/deceleration.
- Increase the distance between the hardware limit switch and the mechanical endstop.

Approach/homing direction

With the direction selection, you determine the approach direction used during active homing to search for the homing switch, as well as the homing direction. The homing direction specifies the travel direction the axis uses to approach the configured end of the homing switch to carry out the homing operation.

Side of homing switch

This is where you select whether the axis is to be homed on the top or bottom side of the homing switch.

Approach velocity

In this field, specify the velocity at which the homing switch is to be searched for during the homing procedure.

Limits (independent of the selected unit of measurement):

- Start/stop velocity ≤ approach velocity ≤ maximum velocity

Homing velocity

In this field, specify the velocity at which the homing switch is to be approached for homing.

Limits (independent of the selected unit of measurement):

- Start/stop velocity ≤ Homing velocity ≤ Maximum velocity

Home position offset

If the desired home position deviates from the position of the homing switch, the home position offset can be specified in this field.

If the value does not equal 0, the axis executes the following actions following homing at the homing switch:

1. Move the axis at the homing velocity by the value of the home position offset
2. Upon reaching the "home position offset", the axis is at the home position that was specified in input parameter "Position" of the "MC_Home" Motion Control instruction.

Limits (independent of the selected unit of measurement):

- -1.0e12 ≤ home position offset ≤ 1.0e12

Home position

The position configured in the Motion Control instruction "MC_Home" is used as the home position.

Configuration - Homing - Passive

Configure the necessary parameters for passive homing in the "Homing - Passive" configuration window.

The movement for passive homing must be triggered by the user (e.g. using an axis motion command). Passive homing is started using Motion Control instruction "MC_Home" with input parameter "Mode" = 2.

Select homing mode (drive connection via PROFIdrive V5 or higher only)

Select one of the following homing modes:

- **Use zero mark via PROFIdrive telegram and proximity switch**

  The system checks for when the proximity switch is reached. After the proximity switch is reached and is left again in the assigned homing direction, zero mark detection is enabled via the PROFIdrive telegram. When the zero mark is reached in the pre-selected direction, then the actual position of the technology object is set to the homing mark position.

- **Use zero mark via PROFIdrive telegram**

  The system enables zero mark detection as soon as the actual value of the technology object moves in the assigned homing direction. When the zero mark is reached in the specified homing direction, the actual position of the technology object is set to the homing mark position.

- **Use homing mark via digital input**

  The system checks the state of the digital input as soon as the actual value of the axis or encoder moves in the assigned homing direction. When the homing mark is reached (setting of the digital input) in the specified homing direction, the actual position of the technology object is set to the homing mark position.

If you have selected drive connection via PTO (Pulse Train Output), a homing mark via a digital input is used by default.

Digital inputs

In this area, you configure the homing switch:

- **Input homing switch**

  Select the digital input for the homing switch in this field. The input must be interrupt-capable. The onboard CPU inputs and the inputs of an inserted signal board can be selected as inputs for the homing switch.

  Note

  The digital inputs are set to a filter time of 6.4 ms by default.

  When the digital inputs are used as a homing switch, this can result in undesired decelerations and thus inaccuracies. Depending on the homing velocity and extent of the homing switch, the home position may not be detected. The filter time can be set under "Input filter" in the device configuration of the digital inputs.

  The specified filter time must be less than the duration of the input signal at the homing switch.

- **Select level**

  In the drop-down list, select the level of the homing switch that is to be used for homing.

Side of homing switch

This is where you select whether the axis is to be homed on the top or bottom side of the homing switch.

Home position

The position configured in the Motion Control instruction "MC_Home" is used as the home position.

Note

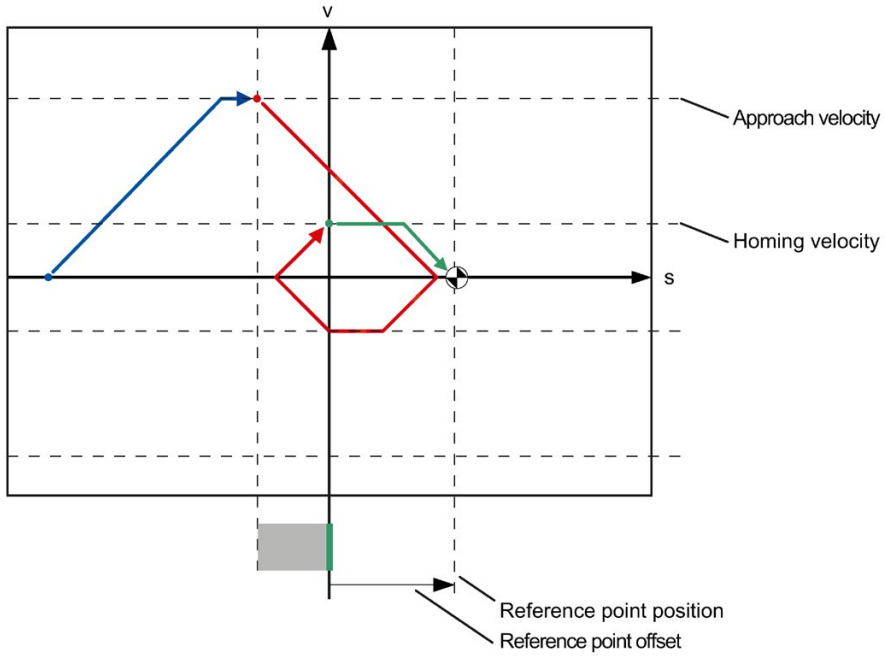
If passive homing is carried out without an axis motion command (axis at a standstill), homing will be executed upon the next rising or falling edge at the homing switch.

Sequence - Active homing

You start active homing with motion control instruction "MC_Home" (input parameter Mode = 3). The "Position" input parameter specifies the absolute home position. Alternatively, you can start active homing on the axis control panel for test purposes.

The diagram below shows an example of a characteristic curve for an active home position approach with the following configuration parameters:

- "Homing mode" = "Use homing mark via digital input"
- "Approach/homing direction" = "Positive direction"
- "Side of homing switch" = "Top side"
- Value of "home position offset" > 0



Search for homing switch (blue curve section)

When active homing starts, the axis accelerates to the configured "approach velocity" and searches at this velocity for the homing switch. The tag <axis name>.StatusBits.HomingDone is set to FALSE.

Reference point approach (red curve section)

When the homing switch is detected, the axis in this example brakes and reverses, to be homed to the configured side of the homing switch at the configured homing velocity. Homing causes the tag <axis name>.StatusBits.HomingDone to change to TRUE.

Travel to home position offset (green curve segment)

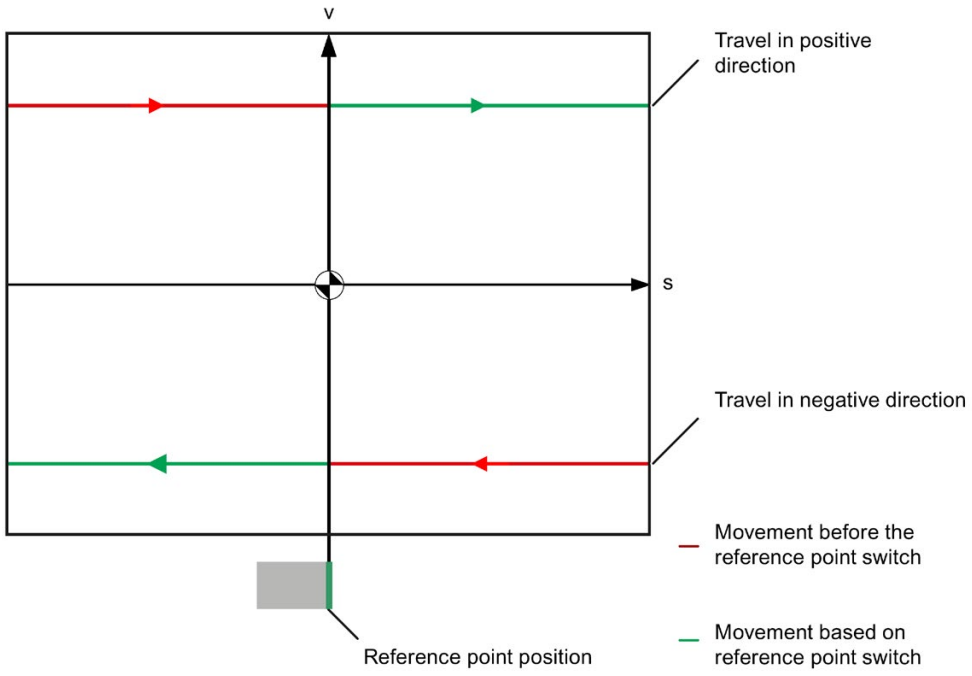
After homing, the axis moves at the homing velocity along the path to the home position offset. There the axis is at the homing point position that was specified in input parameter "Position" of the "MC_Home" Motion Control instruction.

Sequence - Passive homing

Passive homing is started with Motion Control instruction "MC_Home" (input parameter Mode = 2). Input parameter "Position" specifies the absolute reference point position.

The diagram below shows an example of a characteristic curve for passive homing with the following configuration parameters:

- "Side of homing switch" = "Top side"
- "Homing mode" = "Use homing mark via digital input"



Movement towards homing switch (red section of curve)

The Motion Control instruction "MC_Home" does not itself carry out any homing motion when passive homing is started. The travel required for reaching the homing switch must be implemented by the user via other motion control instructions such as "MC_MoveRelative". The tag <axis name>.StatusBits.HomingDone remains TRUE during passive homing if the axis has already been homed.

Axis homing (transition from red to green section of curve)

The axis is homed when the configured side of the homing switch is reached. The current position of the axis is set to the home position. This is specified at the "Position" parameter of the "MC_Home" Motion Control instruction. The tag <axis name>.StatusBits.HomingDone will be set to "TRUE" if the axis has not been homed before. The travel previously started is not canceled.

Movement beyond homing switch (green section of curve)

Following homing at the homing switch, the axis continues and completes the previously started travel with the corrected axis position.

Changing the homing configuration in the user program

With positioning axis technology object as of V2, you can change the following configuration parameters during runtime of the user program in the CPU:

Passive homing

You can change the end of the homing switch for passive homing during the user program runtime. Use the following technology object tag for this purpose:

- <Axis name>.Sensor[1].PassiveHoming.SideInput

  for changing the side of the homing switch

- <Axis name>.Sensor[1].PassiveHoming.Mode

  for changing the homing mode

Refer to the description of the technology object tags (Page 259) in the appendix for information on when changes to the configuration parameter take effect.

Active homing

You can change the direction of approach, the side of the homing switch, the approach velocity, the homing velocity, and the home position offset for active homing during the program runtime of the user program. Use the following technology object tags for this purpose:

- <Axis name>.Homing.AutoReversal

  for changing the auto reverse at the HW limit switch
- <Axis name>.Homing.ApproachDirection

  for changing the approach/homing direction
- <Axis name>.Sensor[1].ActiveHoming.SideInput

  for changing the side of the homing switch
- <Axis name>.Homing.ApproachVelocity

  for changing the approach velocity
- <Axis name>.Homing.ReferencingVelocity

  for changing the homing velocity
- <Axis name>.Sensor[1].ActiveHoming.HomePositionOffset

  for changing the home position offset
- <Axis name>.Sensor[1].ActiveHoming.Mode

  for changing the homing mode

Refer to the description of the technology object tags in the appendix for information on when changes to the configuration parameter take effect.

See also

Compatibility list of variables V1...3 <-> V4...5 (Page 48)

5.4.4.6 Positioning monitoring

Configuration - Position monitoring (PROFIdrive and analog drive connection only)

In the "Position monitoring" configuration window, configure the criteria for monitoring the target position.

Position monitoring monitors the behavior of the actual position at the end of the setpoint calculation. As soon as the setpoint velocity reaches the value zero, the actual position value must be located within a tolerance time in the positioning window. The actual value must not exit the positioning window during the minimum dwell time.

If the actual position reaches the positioning window within the tolerance time and remains in the positioning window for the minimum dwell time, the status bit <axis name>.StatusBits.Done is set. This completes a motion command.

Position monitoring does not make any distinction between how the setpoint interpolation was completed. The end of setpoint interpolation can, for example, be reached as follows:

- By the setpoint reaching the target position
- By position-controlled stopping during the motion through the Motion Control instruction "MC_Halt"

In the following cases, the axis is stopped by the position monitoring and a positioning error (ErrorID 16#800F) is displayed at the Motion Control instruction:

- The actual value does not reach the positioning window within the tolerance time.
- The actual value exits the positioning window during the minimum dwell time.

Positioning window

In this field, configure the size of the positioning window.

Tolerance time

In this field, configure the tolerance time within which the position value must reach the positioning window.

Minimum dwell time in positioning window

In this field, configure the minimum dwell time for which the actual position value must be located in the positioning window.

Configuration - Following error (PROFIdrive and analog drive connection only)

In the "Following error" configuration window, you configure the permissible deviation of the actual position of the axis from the setpoint position.

The following error is the difference between the setpoint position and the actual position value of the axis. The transmission times of the setpoint to the drive and of the actual value to the controller are taken into account in the calculation of the following error.

The following error is monitored based on a velocity-dependent following error limit. The permissible following error depends on the setpoint velocity.

A constant permissible following error can be specified for velocities lower than an adjustable velocity low limit. Above this low velocity limit, the permissible following error increases in proportion to the setpoint velocity. The maximum following error is permitted at the maximum velocity.

If the permitted following error is exceeded, the axis is stopped and an error (ErrorID 16#800D) is displayed at the Motion Control instruction.

Enable following error monitoring

Select the check box to enable following error monitoring.

When following error monitoring is enabled, the axis is stopped in the error range (orange).

Maximum following error

In this field, configure the following error that is permissible at maximum velocity.

Following error

In this field, configure the permissible following error for low velocities (without dynamic adaptation).

Start dynamic adjustment

In this field, configure the velocity above which the following error should be dynamically adapted. Above this velocity, the following error up to the maximum velocity will be adapted to the maximum following error.

Maximum velocity

This box shows the maximum velocity configured under "Dynamics > General".

Configuration - Standstill signal (PROFIdrive and analog drive connection only)

In the "Standstill signal" configuration window, configure the criteria for standstill detection.

To display the standstill (<Axis name>.StatusBits.StandStill), the velocity of the axis must remain in the standstill window for the minimum dwell time.

Standstill window

In this field, configure the size of the standstill window.

Minimum dwell time in standstill window

In this field, configure the minimum dwell time in the standstill window.

5.4.4.7 Configuration - Control loop (PROFIdrive and analog drive connection only)

In the "Control loop" configuration window, configure the precontrol and the gain Kv of the position control loop.

The Kv factor affects the following parameters:

- Positioning accuracy and stop control
- Uniformity of motion
- Positioning time

The better the mechanical conditions of the axis are (high stiffness), the higher you can configure the Kv factor. This reduces the following error, and a higher dynamic response is achieved.

The "Tuning (Page 150)" function supports you in determining the optimum gain for the position control of the axis.

Precontrol

In this field, configure the velocity precontrol of the position control loop as a percentage.

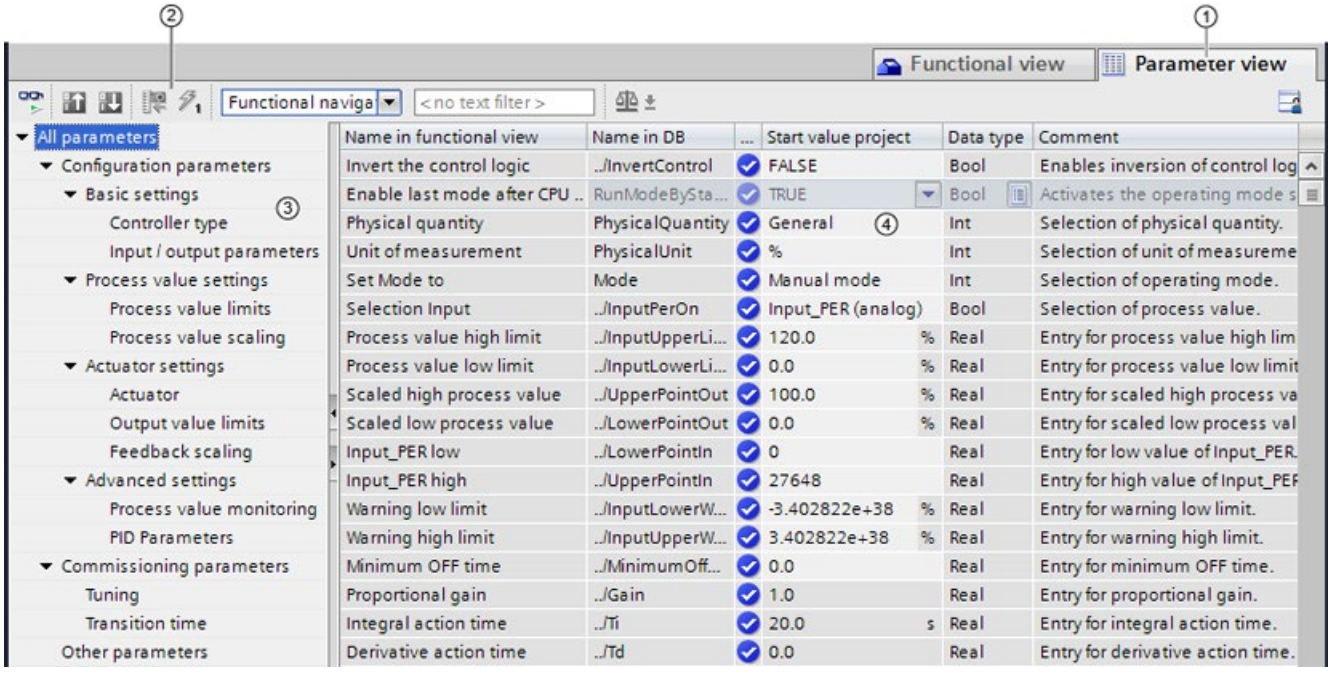
Gain (Kv factor)

In this field, you configure the gain Kv of the position control loop.

5.4.5 Parameter view

5.4.5.1 Introduction to the parameter view

The Parameter view provides you with a general overview of all relevant parameters of a technology object. You obtain an overview of the parameter settings and can easily change them in offline and online mode.



① "Parameter view" tab

② Toolbar (Page 101)

③ Navigation (Page 102)

④ Parameter table (Page 102)

Function scope

The following functions are available for analyzing the parameters of the technology objects and for enabling targeted monitoring and modification.

Display functions:

- Display of parameter values in offline and online mode
- Display of status information of the parameters
- Display of value deviations and option for direct correction
- Display of configuration errors
- Display of value changes as a result of parameter dependencies
- Display of all memory values of a parameter: Start value PLC, Start value project, Monitor value
- Display of the parameter comparison of the memory values of a parameter

Operator control functions:

- Navigation for quickly changing between the parameters and parameter structures.
- Text filter for faster searches for particular parameters.
- Sorting function for customizing the order of parameters and parameter groups to requirements.
- Memory function for backing up structural settings of the Parameter view.
- Monitoring and modifying of parameter values online.
- Change display format of value.
- Function for saving a snapshot of parameter values of the CPU in order to capture momentary situations and to respond to them.
- Function for applying a snapshot of parameter values as start values.
- Download of modified start values to the CPU.
- Comparison functions for comparing parameter values with one another.

Validity

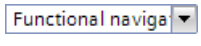
The Parameter view described here is available for the following technology objects:

- PID_Compact
- PID_3Step
- PID_Temp
- CONT_C (S7-1500 only)
- CONT_S (S7-1500 only)
- TCONT_CP (S7-1500 only)
- TCONT_S (S7-1500 only)
- TO_Axis_PTO (S7-1200 Motion Control)
- TO_Positioning_Axis (S7-1200 Motion Control)
- TO_CommandTable_PTO (S7-1200 Motion Control)
- TO_CommandTable (S7-1200 Motion Control)

5.4.5.2 Structure of the parameter view

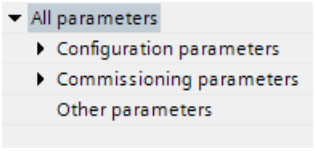
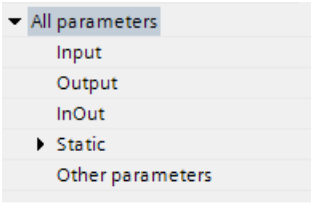
Toolbar

The following functions can be selected in the toolbar of the parameter view.

Icon	Function	Explanation
	Monitor all	Starts the monitoring of visible parameters in the active Parameter view (online mode).
	Create snapshot of monitor values and accept set-points of this snapshot as start values	Applies the current monitor values to the “Snapshot” column and updates the start values in the project. Only in online mode for PID_Compact, PID_3Step and PID_Temp.
	Initialize setpoints	Transfers the start values updated in the project to the CPU. Only in online mode for PID_Compact, PID_3Step and PID_Temp.
	Create snapshot of monitor values	Applies the current monitor values to the “Snapshot” column. Only in online mode.
	Modify all selected parameters immediately and once	This command is executed once and as quickly as possible without reference to any particular point in the user program. Only in online mode.
	Select navigation structure	Toggles between functional navigation and data navigation.
	Text filter...	After entry of a character string: Display of all parameters containing the specified string in one of the currently visible columns.
	Selection of compare values	Selection of parameter values that are to be compared with one another in online mode (Start value project, Start value PLC, Snapshot) Only in online mode.
	Save window settings	Saves your display settings for the Parameter view (e.g., selected navigation structure, activated table columns, etc.)

Navigation

Within the "Parameter view" tab, the following alternative navigation structures can be selected.

Navigation		Explanation
Functional navigation		In the functional navigation, the structure of the parameters is based on the structure in the configuration dialog ("Functional view" tab), commissioning dialog, and diagnostics dialog. The last group "Other parameters" contains all other parameters of the technology object.
Data navigation		In the data navigation, the structure of the parameters is based on the structure in the instance DB / technology DB. The last group "Other parameters" contains the parameters that are not contained in the instance DB / technology DB.

You can use the "Select navigation structure" drop-down list to toggle the navigation structure.

Parameter table

The table below shows the meaning of the individual columns of the parameter table. You can show or hide the columns as required.

- Column "Offline" = X: Column is visible in offline mode.
- Column "Online" = X: Column is visible in online mode (online connection to the CPU).

Column	Explanation	Offline	Online
Name in functional view	Name of the parameter in the functional view. The display field is empty for parameters that are not configured via the technology object.	X	X
Full name in DB	Complete path of the parameter in the instance DB / technology DB. The display field is empty for parameters that are not contained in the instance DB / technology DB.	X	X
Name in DB	Name of the parameter in the instance DB / technology DB. If the parameter is part of a structure or UDT, the prefix ". ./" is added. The display field is empty for parameters that are not contained in the instance DB / technology DB.	X	X
Status of configuration	Display of the completeness of the configuration using status symbols. see Status of configuration (offline) (Page 111)	X	
Compare result	Result of the "Compare values" function. This column is shown if there is an online connection and the "Monitor all" button is selected.		X

Column	Explanation	Offline	Online
Start value project	Configured start value in the project. Error indication if entered values have a syntax or process-related error.	X	X
Default value	Value that is pre-assigned to the parameter. The display field is empty for parameters that are not contained in the instance DB / technology DB.	X	X
Snapshot	Snapshot of the current values in the CPU (monitor values). Error indication if values have a process-related error.	X	X
Start value PLC	Start value in the CPU. This column is shown if there is an online connection and the "Monitor all" button is selected. Error indication if values have a process-related error.		X
Monitor value	Current value in the CPU. This column is shown if there is an online connection and the "Monitor all" button is selected. Error indication if values have a process-related error.		X
Modify value	Value that is to be used to change the monitor valuet. This column is shown if there is an online connection and the "Monitor all" button is selected. Error indication if entered values have a syntax or process-related error.		X
Selection for transmission	Selection of the Modify values that are to be transmitted using the "Modify all selected parameters immediately and once" button. This column is displayed together with the "Modify value" column.		X
Minimum value	Minimum process-related value of the parameter. If the minimum value is dependent on other parameters, it is defined: • Offline: By the Start value project. • Online: By the Monitor values.	X	X
Maximum value	Maximum process-related value of the parameter. If the maximum value is dependent on other parameters, it is defined: • Offline: By the Start value project. • Online: By the Monitor values.	X	X
Setpoint	Designates the parameter as a setpoint. These parameters can be initialized online.	X	X
Data type	Data type of the parameter. The display field is empty for parameters that are not contained in the instance DB / technology DB.	X	X
Retain	Designates the value as a retentive value. The values of retentive parameters are retained even after the voltage supply is switched off.	X	X
Accessible from HMI	Indicates whether the HMI can access this parameter during runtime.	X	X
Visible in HMI	Indicates whether the parameter is visible in the selection list of the HMI by default.	X	X
Comment	Brief description of the parameter.	X	X

5.4.5.3 Opening the parameter view

Requirement

The technology object has been added in the project tree, i.e., the associated instance DB / technology DB of the instruction has been created.

Procedure

1. Open the "Technology objects" folder in the project tree.
2. Open the technology object in the project tree.
3. Double-click the "Configuration" object.
4. Select the "Parameter view" tab in the top right corner.

Result

The Parameter view opens. Each displayed parameter is represented by one row in the parameter table.

The displayable parameter properties (table columns) vary depending on whether you are working with the Parameter view in offline or online mode.

In addition, you can selectively display and hide individual table columns.

See also

Default setting of the parameter view (Page 104)

5.4.5.4 Default setting of the parameter view

Default settings

To enable you to work efficiently with the Parameter view, you can customize the parameter display and save your settings.

The following customizations are possible and can be saved:

- Show and hide columns
- Change column width
- Change order of the columns
- Toggle navigation
- Select parameter group in the navigation
- Selection of compare values

Show and hide columns

To show or hide columns in the parameter table, follow these steps:

1. Position the cursor in the header of the parameter table.
2. Select the "Show/Hide" command in the shortcut menu.
   The selection of available columns is displayed.
3. To show a column, select the check box for the column.
4. To hide a column, clear the check box for the column.

or

1. Position the cursor in the header of the parameter table.
2. Select the "Show all columns" command in the shortcut menu if all columns of the offline or online mode are to be displayed.

Some columns can only be displayed in online mode: see Parameter table (Page 102).

Change column width

To customize the width of a column so that all texts in the rows can be read, follow these steps:

1. Position the cursor in the header of the parameter table to the right of the column to be customized until the shape of the cursor changes to a cross.
2. Then double-click this location.

or

1. Open the shortcut menu on the header of the parameter table.
2. Click
   - "Optimize column width" or
   - "Optimize width of all columns".

If the column width setting is too narrow, the complete content of individual fields are shown if you hover the cursor briefly over the relevant field.

Change order of the columns

The columns of the parameter table can be arranged in any way.

To change the order of the columns, follow these steps:

1. Click on the column header and use a drag-and-drop operation to move it to the desired location.

   When you release the mouse button, the column is anchored to the new position.

Toggle navigation

To toggle the display form of the parameters, follow these steps:

1. Select the desired navigation in the “Select navigation structure” drop-down list.
   - Data navigation
   - Functional navigation

See also Navigation (Page 102).

Select parameter group in the navigation

Within the selected navigation, you choose between the “All parameters” display or the display of a subordinate parameter group of your choice.

1. Click the desired parameter group in the navigation.

   The parameter table only displays the parameters of the parameter group.

Selection of compare values (online)

To set the compare values for the “Compare values” function, follow these steps:

1. Select the desired compare values in the “Selection of compare values” drop-down list.
   - Start value project / Start value PLC
   - Start value project / Snapshot
   - Start value PLC / Snapshot

The “Start value project / Start value PLC” option is set by default.

Saving the default setting of the Parameter view

To save the above customizations of the Parameter view, follow these steps:

1. Customize the Parameter view according to your requirements.
2. Click the “Save window settings” button  at the top right of the Parameter view.

5.4.5.5 Working with the parameter view

Overview

The following table provides an overview of the functions of the Parameter view in online and offline mode described in the following.

- Column "Offline" = X: This function is possible in offline mode.
- Column "Online" = X: This function is possible in online mode.

Function/action	Offline	Online
Filtering the parameter table (Page 107)	X	X
Sorting the parameter table (Page 108)	X	X
Transferring parameter data to other editors (Page 108)	X	X
Indicating errors (Page 109)	X	X
Editing start values in the project (Page 109)	X	X
Status of configuration (offline) (Page 111)	X	
Monitoring values online in the parameter view (Page 111)		X
Create snapshot of monitor values (Page 113)		X
Modifying values (Page 114)		X
Comparing values (Page 115)		X
Applying values from the online program as start values (Page 117)		X
Initializing setpoints in the online program (Page 118)		X

Filtering the parameter table

You can filter the parameters in the parameter table in the following ways:

- With the text filter
- With the subgroups of the navigation

Both filter methods can be used simultaneously.

With the text filter

Texts that are visible in the parameter table can be filtered. This means only texts in displayed parameter rows and columns can be filtered.

1. Enter the desired character string for filtering in the “Text filter...” input box.

   The parameter table displays only the parameters containing the character string.

The text filtering is reset.

- When another parameter group is selected in the navigation.
- When navigation is changed from data navigation to functional navigation, or vice versa.

With the subgroups of the navigation

1. Click the desired parameter group in the navigation, e.g., "Static".

   The parameter table only shows the static parameters. You can select further subgroups for some groups of the navigation.

2. Click “All parameters” in the navigation if all parameters are to be shown again.

Sorting the parameter table

The values of the parameters are arranged in rows. The parameter table can be sorted by any displayed column.

- In columns containing numerical values, sorting is based on the magnitude of the numerical value.
- In text columns, sorting is alphabetical.

Sorting by column

1. Position the cursor in the header cell of the desired column.

   The background of this cell turns blue.

2. Click the column header.

Result

The entire parameter table is sorted by the selected column. A triangle with tip facing up appears in the column header.

Clicking the column header again changes the sorting as follows:

- Symbol “▲”: Parameter table is sorted in ascending order.
- Symbol “▼”: Parameter table is sorted in descending order.
- No symbol: The sorting is removed again. The parameter table assumes the default display.

The “../“ prefix in the “Name in DB” column is ignored when sorting.

Transferring parameter data to other editors

After selecting an entire parameter row of the parameter table, you can use the following:

- Drag-and-drop
- <Ctrl+C>/<Ctrl+V>
- Copy/Paste via shortcut menu

Transfer parameters to the following editors of the TIA Portal:

- Program editor
- Watch table
- Signal table for trace function

The parameter is inserted with its full name: See information in “Full name in DB” column.

Indicating errors

Error indication

Parameter assignment errors that result in compilation errors (e.g. limit violation) are indicated in the Parameter view.

Every time a value is input in the Parameter view, a check is made for process-related and syntax errors and the result is indicated.

Bad values are indicated by:

- Red error symbol in the "Status of configuration" (offline mode) or "Compare result" (online mode, depending on the selected comparison type) columns

and/or

- Table field with red background

  If you click the bad field, a roll-out error message appears with information of the permissible value range or the required syntax (format)

Compilation error

From the error message of the compiler, you can directly open the Parameter view (functional navigation) containing the parameter causing the error in situations where the parameter is not displayed in the configuration dialog.

Editing start values in the project

With the Parameter view, you can edit the start values in the project in offline mode and online mode.

- You make value changes in the “Start value project” column of the parameter table.
- In the “Status of configuration” column of the parameter table, the progress of the configuration is indicated by the familiar status symbols from the configuration dialog of the technology object.

Boundary conditions

- If other parameters depend on the parameter whose start value was changed, the start value of the dependent parameters are also adapted.
- If a parameter of a technology object is not editable, it is also not editable in the parameter view. The ability to edit a parameter can also depend on the values of other parameters.

Defining new start values

To define start values for parameters in the Parameter view, follow these steps:

1. Open the Parameter view of the technology object.
2. Enter the desired start values in the "Start value project" column. The value must match the data type of the parameter and must not exceed the value range of the parameter. The limits of the value range can be seen in the “Maximum value” and “Minimum value” columns.

The "Status of configuration" column indicates the progress of the configuration with colored symbols.

See also Status of configuration (offline) (Page 111)

Following adaptation of the start values and downloading of the technology object to the CPU, the parameters take the defined value at startup if they are not declared as retentive (“Retain” column).

Error indication

When a start value is input, a check is made for process-related and syntax errors and the result is indicated.

Bad start values are indicated by:

- Red error symbol in the "Status of configuration" (offline mode) or "Compare result" (online mode, depending on the selected comparison type) columns

and/or

- Red background in the “Start value project” field
  If you click on the bad field, a roll-out error message appears with information of the permissible value range or the necessary syntax (format)

Correcting bad start values

1. Correct bad start values using information from the roll-out error message.

   Red error symbol, red field background, and roll-out error message are no longer displayed.

The project cannot be successfully compiled unless the start values are error-free.

Status of configuration (offline)

The status of the configuration is indicated by icons:

- In the “Status of configuration” column in the parameter table
- In the navigation structure of the functional navigation and data navigation

Symbol in “Status of configuration” column

Symbol	Meaning
	The start value of the parameter corresponds to the default value and is valid. A start value has not yet been defined by the user.
	The start value of the parameter contains a value defined by the user or an automatically adjusted value. The start value is different than the default value. The start value is error-free and valid.
	The start value of the parameter is invalid (syntax or process-related error). The input box has a red background. When clicked, the roll-out error message indicates the cause of the error.
	Only for S7-1200 Motion Control: The start value of the parameter is valid but contains warnings. The input box has a yellow background.
	The parameter is not relevant in the current configuration.

Symbol in the navigation

The symbols in the navigation indicate the progress of the configuration in the same way as in the configuration dialog of the technology object.

Monitoring values online in the parameter view

You can monitor the values currently taken by the parameters of the technology object in the CPU (monitor values) directly in the Parameter view.

Requirements

- There is an online connection.
- The technology object is downloaded to the CPU.
- The program execution is active (CPU in "RUN").
- The Parameter view of the technology object is open.

Procedure

1. Start the monitoring by clicking  .

   As soon as the Parameter view is online, the following columns are additionally displayed:

   – Compare result
   – Start value PLC
   – Monitor value
   – Modify value
   – Selection for transmission

   The "Monitor value" column shows the current parameter values on the CPU.

   Meaning of the additional columns: see Parameter table (Page 102)

2. Stop the monitoring by clicking  again.

Display

All columns that are only available online have an orange background:

- Values in light-orange cells can be changed.
- Values in cells with a dark orange background cannot be changed.

Change display format of value

The display format of the value can be selected via the shortcut menu of a table row in the Parameter view of the technology object.

The display format of the following values can be changed both in online mode and in offline mode:

- Start value project
- Start value PLC
- Maximum value
- Minimum value
- Snapshot
- Monitor value
- Default value
- Modify value

The set display format applies to all values of the table row.

The following display formats of the value can be changed:

- Default
- Hex
- Octal
- Bin
- Dec (+/-)
- DEC

Depending on the parameter selected in the parameter view, only the supported display formats can be selected.

Requirements

- The Parameter view of the technology object is open.

Procedure

To change the display format of the value, proceed as follows:

1. Select one or more table rows in which you want to change the display format.
2. Select the "Display format" command in the shortcut menu.
3. Select the desired display format.

Note

To change the display format of a certain data type in multiple table rows, sort the Parameter view by this data type. Then select the first and last table row with this data type while keeping the <Shift> key pressed and change the display format for the selected table rows.

Create snapshot of monitor values

You can back up the current values of the technology object on the CPU (monitor values) and display them in the Parameter view.

Requirements

- There is an online connection.
- The technology object is downloaded to the CPU.
- The program execution is active (CPU in "RUN").
- The Parameter view of the technology object is open.
- The “Monitor all” button  is selected.

Procedure

To show the current parameter values, follow these steps:

1. In the Parameter view, click the “Create snapshot of monitor values" icon .

Result

The current monitor values are transferred once to the "Snapshot" column of the parameter table.

You can analyze the values "frozen" in this way while the monitor values continue to be updated in the "Monitor values" column.

Modifying values

With the Parameter view, you can modify values of the technology object in the CPU.

You can assign values to the parameter once (Modify value) and modify them immediately. The modify request is executed as quickly as possible without reference to any particular point in the user program.

DANGER

Danger when modifying:

Changing the parameter values while the plant is operating may result in severe damage to property and personal injury in the event of malfunctions or program errors.

Make sure that dangerous states cannot occur before you use the "Modify" function.

Requirements

- There is an online connection.
- The technology object is downloaded to the CPU.
- The program execution is active (CPU in "RUN").
- The Parameter view of the technology object is open.
- The “Monitor all” button is selected.
- The parameter can be modified (associated field in the "Modify value" column has a light-orange background).

Procedure

To modify parameters immediately, follow these steps:

1. Enter the desired modify values in the “Modify values” column of the parameter table.
2. Check whether the check box for modifying is selected in the "Select for transmission" column.

   The modify values and associated check boxes of dependent parameters are automatically adapted at the same time.
3. Click the “Modify all selected parameters immediately and once” icon  .

The selected parameters are modified once and immediately with the specified values and can be monitored in the "Modify values" column. The check boxes for modifying in the "Selection for transmission" column are automatically cleared after the modify request is complete.

Error indication

When a start value is input, a check is made immediately for process-related and syntax errors and the result is indicated.

Bad start values are indicated by:

- Red background in the “Modify value” field

and

- If you click the bad field, a roll-out error message appears with information of the permissible value range or the necessary syntax (format)

Bad modify values

- Modify values with process-related errors can be transmitted.
- Modify values with syntax errors **cannot** be transmitted.

Comparing values

You can use comparison functions to compare the following memory values of a parameter:

- Start value project
- Start value PLC
- Snapshot

Requirements

- There is an online connection.
- The technology object is downloaded to the CPU.
- The program execution is active (CPU in "RUN").
- The Parameter view of the technology object is open.
- The “Monitor all” button is selected.

Procedure

To compare the start values on the various target systems, follow these steps:

1. Click the "Selection of compare values" icon .

   A selection list containing the comparison options opens:

   – Start value project - Start value PLC (default setting)

   – Start value project - Snapshot

   – Start value PLC - Snapshot

2. Select the desired comparison option.

   The selected comparison option is executed as follows:

   – A scales symbol appears in the header cells of the two columns selected for comparison.

   – Symbols are used in the "Compare result" column to indicate the result of the comparison of the selected columns.

Symbol in "Compare result" column

Symbol	Meaning
	The compare values are equal and error-free.
	The compare values are not equal and error-free.
	At least one of the two compare values has a process-related or syntax error.
	The comparison cannot be performed. At least one of the two comparison values is not available (e.g. snapshot).
	Comparison of the value is inappropriate since it is not relevant in one of the configurations.

Symbol in the navigation

The symbols are shown in the same way in the navigation if the comparison result applies to at least one of the parameters below the displayed navigation structure.

Applying values from the online program as start values

In order to apply optimized values from the CPU to the project as start values, you create a snapshot of the monitor values. Values of the snapshot marked as a "Setpoint" are then applied to the project as start values.

Requirements

- The technology object is of type "PID_Compact" or "PID_3Step".
- There is an online connection.
- The technology object is downloaded to the CPU.
- The program execution is active (CPU in "RUN").
- The Parameter view of the technology object is open.
- The “Monitor all” button is selected.

Procedure

To apply optimized values from the CPU, follow these steps:

1. Click the "Create snapshot of monitor values and accept setpoints of this snapshot as start values" icon .

Result

The current monitor values are applied to the "Snapshot" column and their setpoints are copied to the "Start value project" column as new start values.

Note

Applying values of individual parameters

You can also apply the values of individual parameters that are not marked as a setpoint from the "Snapshot" column to the "Start values project" column. To do so, copy the values and insert them into the "Start value project" column using the "Copy" and "Paste" commands in the shortcut menu.

Initializing setpoints in the online program

You can initialize all parameters that are marked as a "Setpoint" in the Parameter view with new values in the CPU in one step. In so doing, the start values are downloaded from the project to the CPU. The CPU remains in "RUN" mode.

To avoid data loss on the CPU during a cold restart or warm restart, you must also download the technology object to the CPU.

DANGER

Danger when changing parameter values

Changing the parameter values while the plant is operating may result in severe damage to property and personal injury in the event of malfunctions or program errors.

Make sure that dangerous states cannot occur before you reinitialize the setpoints.

Requirements

- The technology object is of type "PID_Compact" or "PID_3Step".
- There is an online connection.
- The technology object is downloaded to the CPU.
- The program execution is active (CPU in "RUN").
- The Parameter view of the technology object is open.
- The “Monitor all” button is selected.
- The parameters marked as a "Setpoint" have a "Start value project" that is free of process-related and syntax errors

Procedure

To initialize all setpoints, follow these steps:

1. Enter the desired values in the "Start value project" column.

   Ensure that the start values are free of process-related and syntax errors.
2. Click the "Initialize setpoints" icon .

Result

The setpoints in the CPU are initialized with the start values from the project.

5.4.6 Configuring technology modules for Motion Control

5.4.6.1 Overview

The following technology modules can be used as the encoder connection in S7-1200 Motion Control.

ET 200MP	ET 200 SP
TM Count 2x24V (Page 122)	TM Count 1x24V (Page 122)
TM PosInput 2 (Page 120)	TM PosInput 1 (Page 120)

Technology modules can be used centrally or distributed in the system. However, isochronous mode is supported only with decentralized use.

The following section describes how to configure the technology modules as encoder:

See also

TM Count 1x24V / TM Count 2x24V (Page 122)

TM PosInput 1 / TM PosInput 2 (Page 120)

5.4.6.2 TM PosInput 1 / TM PosInput 2

For use with S7-1200 Motion Control, the following parameters must be configured:

Configuration	
Technology module **TM PosInput 1 / TM PosInput 2**	**Technology object** **Axis**
Basic parameters > Channel 0/1 > Operating mode Select "Position input for Motion Control" mode	–
Basic parameters > Channel 0/1 > Module parameters	**Basic parameters > Encoder**
–	Select encoder and corresponding channel
Signal type • Incremental encoder • Absolute encoder	Select encoder type corresponding to configuration for technology module • Incremental • Absolute
–	**Data exchange with encoder**
	Telegram "DP_TEL83_STANDARD" is automatically selected after the selection of the encoder.
	Disable "Automatic transfer of encoder parameter values in the device"
	Select rotary or linear measuring system type
Signal evaluation • Single • Double • Quadruple	Select fine resolution corresponding to configuration for technology module • Incremental encoder: – 0 = Single – 1 = Double *or* – 2 = Quadruple • Absolute encoder: – 0 (= single)
Enter increments per revolution	• Rotary type: Enter increments per revolution corresponding to configuration at technology module (1:1) • Linear type: Enter distance per revolution
–	**Hardware interface > Data exchange with drive**
• Rotary type: Enter reference speed corresponding to configuration for technology object (1:1) • Linear type: Configuration not relevant	Enter reference speed

Configuration	
Technology module **TM PosInput 1 / TM PosInput 2**	**Technology object** **Axis**
I/O addresses	–
Organization block: MC-Servo	
Process image: PIP OB servo	
Select "Isochronous mode"	

"–" No configuration for technology module/technology object is required for these parameters

See also

Overview (Page 119)

TM Count 1x24V / TM Count 2x24V (Page 122)

5.4.6.3 TM Count 1x24V / TM Count 2x24V

For use with S7-1200 Motion Control, the following parameters must be configured:

Configuration	
Technology module **TM Count 1x24V / TM Count 2x24V**	**Technology object** **Axis**
Basic parameters > Channel 0/1 > Operating mode	–
Select "Position input for Motion Control" mode	
Basic parameters > Channel 0/1 > Module parameters	**Hardware interface > Encoder**
–	Select encoder and corresponding channel
Signal type • Incremental encoder	Select encoder type corresponding to configuration for technology module • Incremental
–	**Hardware interface > Data exchange with encoder**
	Telegram "DP_TEL83_STANDARD" is automatically selected after the selection of the encoder.
	Disable "Automatic transfer of encoder parameter values in the device"
	Select rotary or linear measuring system type
Signal evaluation • Single • Double • Quadruple	Select fine resolution corresponding to configuration for technology module • 0 = Single • 1 = Double • 2 = Quadruple
Enter increments per revolution	• Rotary type: Enter increments per revolution corresponding to configuration at technology module (1:1) • Linear type: Enter distance per increment
–	**Hardware interface > Data exchange with drive**
• Rotary type: Enter reference speed corresponding to configuration for technology object (1:1) • Linear type: Configuration not relevant	Enter reference speed
I/O addresses	–
Organization block: MC-Servo	
Process image: PIP OB servo	
Select "Isochronous mode"	

See also

Overview (Page 119)

TM PosInput 1 / TM PosInput 2 (Page 120)

Technology object command table

6

6.1 Use of the command table technology object

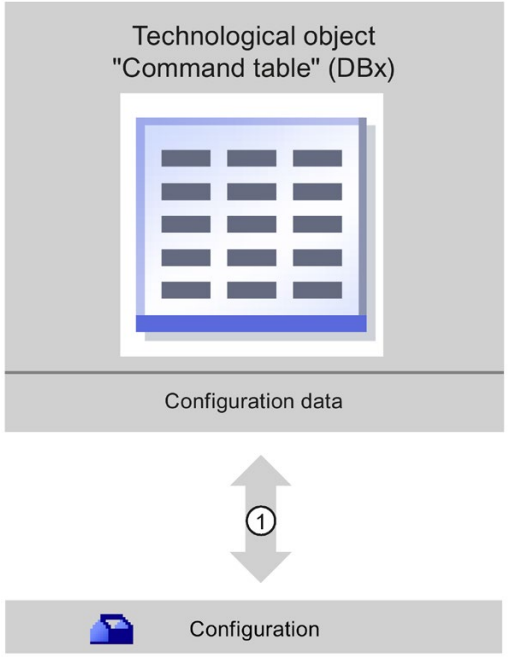
The technology object "Command table" allows you to combine multiple individual axis control commands in one movement sequence. The technology object can be used as of technology version V2 for axes with drive connection via PTO (Pulse Train Output).

You configure the movement sequence as a table in a configuration dialog.

The motion profile of the movement sequences can be checked on a graph before the project is loaded to the CPU. The command tables created are then linked to an axis and used in the user program with the "MC_CommandTable" Motion Control instruction. You can process part or all of the command table.

6.2 Command table technology object tools

The "Configuration" tool is provided in the TIA Portal for the "Command Table" technology object. The representation below shows the interaction of the tool with the technology object:



①	Writing and reading the configuration of the technology object

Configuration

Configure the following properties of the "Command Table" technology object with the "Configuration" tool:

- You can create one or more movement sequences by configuring individual jobs.
- You can configure the graphic display to check your movement sequence using an axis already configured or a configurable default axis.

The movement sequence data are saved in the data block of the technology object.

6.3 Adding the technological object command table

Requirements

- A project with a CPU S7-1200 has been created.
- The CPU firmware version is V2.1 or higher

Procedure

Proceed as follows to add a "Command table" technology object in the project tree:

1. Open the "CPU > Technology objects" folder in the project tree.
2. Double-click the "Add new object" command.

   The "Add new object" dialog opens.
3. Select the "Motion Control" technology.
4. Open the "Motion Control" folder.
5. Select the desired technology version in the "Version" column.
6. Select the "TO_CommandTable" object.
7. Enter the name of the command table in the "Name" input box.
8. To change the automatically assigned data block number, select the "Manual" option.
9. To display additional information about the technology object, click "Additional information".
10. Confirm your entry with "OK".

Result

The new technology object is created and saved to the "Technology objects" folder in the project tree.

6.4 Configuring the command table technology object

6.4.1 Working with the configuration dialog

You configure the properties of the technology object in the configuration window. Proceed as follows to open the configuration window of the technology object:

1. Open the group of the required technology object in the project tree.
2. Double-click the "Configuration" object.

The configuration is divided into the following categories:

- **Basic parameters**

  The basic parameters contain all parameters which must be configured for a functional command table.

- **Extended parameters**

  The extended parameters contain the parameters of the default axis or display the parameter values of the axis selected.

Configuration window icons

Icons in the area navigation of the configuration show additional details about the status of the configuration:

	**The configuration contains default values and is complete**. The configuration contains only default values. With these default values you can use the technology object without additional changes.
	**The configuration contains user-defined or automatically adapted values and is complete.** All input fields of the configuration contain valid values and at least one preset value has changed.
	The configuration is incomplete or incorrect At least one input field or drop-down list contains an invalid value. The corresponding field or the drop-down list is displayed on a red background. Click the roll-out error message to display the cause of the error.
	**The configuration contains mutually incompatible parameter values** The configuration contains parameter values that contradict each other either in size or logic. The corresponding field or the drop-down list is displayed on a yellow background.

See also

Guidelines on use of motion control (Page 42)

Basic parameters (Page 127)

Extended parameters (Page 141)

6.4.2 Monitor values

If there is an online connection to the CPU, the icon "Monitor all"  is displayed in the configuration dialog of the technology object.

The "Monitor all" function provides the following options:

- Comparison of configured start values of the project with the start values in the CPU and the actual values
- Direct editing of actual values and the start values of the project
- Immediate detection and display of input errors with suggested corrections
- Backup of the actual values in the project by manual transfer to the start value of the project

Icons and operator controls

If there is an online connection to the CPU, the actual values are displayed at the parameters.

In addition to the actual values of the parameters, the following symbols appear:

Icon	Description
	Start value in CPU matches the configured Start value in the project
	Start value in CPU does not match the configured Start value in the project
	A comparison of the start value in the CPU with the configured start value in the project cannot be performed because the selected CPU module does not support this comparison.
	The value is not comparable with any significance since it is not relevant in one of the configurations.
	Use the button to show the start value of the CPU and the start value of the project for the respective parameter.

The actual value and the start value in the project can be changed directly and then downloaded to the CPU. The change of the actual value is transferred directly to the CPU for directly modifiable parameters.

6.4.3 Basic parameters

6.4.3.1 Configuration - General

Configure the name of the technology object in the "General" configuration window.

Name

Define the name of the command table or the name of the "Command table" technology object in this field. The technology object is listed under this name in the project tree.

See also

Configuration - Command table (Page 127)

Shortcut menu commands - Command table (Page 131)

Working with the trend diagram (Page 133)

Shortcut menu commands - Curve chart (Page 137)

Transition from "Complete command" to "Blend motion" (Page 138)

Changing the command table configuration in the user program (Page 140)

6.4.3.2 Configuration - Command table

Create the desired movement sequence in the "Command Table" configuration window and check the result against the graphic view in the trend diagram.

Note

Small deviations are possible between the time behavior and position in the trend shown and the real movement of the axis. Movements in response to software limit switches being reached are not shown.

Enable warnings

Activate the display of warnings in the command table with this check box.

Use axis parameters of

From the drop-down list, select which axis parameters are to be used for selecting the graphic view of and checking the movement sequence. Select "Default axis" if you have yet to add an axis to the "Technology object" folder or wish to use values which have not been configured in any of the available axes. You configure the properties of the default axis under "Advanced parameters".

The axis parameters of the axis selected at the "Axis" parameter are used to process the command table in the user program.

Column: Step

Shows the step number of the command.

Column: Command type

In this column, select the command types which are to be used for processing the command table. Up to 32 commands can be entered. The commands will be processed in sequence. You can choose between the following entries and command types:

- **Empty**

  The entry serves as a placeholder for any commands to be added. The empty entry is ignored when the command table is processed.

- **Halt**

  Stop axis
  (the command only takes effect after a "Velocity set point" command)

- **Positioning Relative**

  Position axis relatively

- **Positioning Absolute**

  Position axis absolutely (the axis must be homed for this)

- **Velocity set point**

  Move axis at set velocity

- **Wait**

  Waits until the given period is over. Wait does not stop active travel.

- **Separator**

  Adds a Separator line above the selected line. The Separator line acts as a range limit for the graphic display of the trend view.

  Use the Separator lines if you wish to process parts of the command table.

Column: Position/travel path

Enter the position or travel path for the selected command in this column:

- **Command "Positioning Relative"**

  The command will move the axis by the given travel path.

- **Command "Positioning Absolute"**

  The command will move the axis by the given position.

  The axis must be homed for this.

- **Separator**

  The value given specifies the start position for the graphic display.

Limit values (independent of the selected user unit):

- -1.0e12 ≤ position / distance ≤ -1.0e-12
- 1.0e-12 ≤ position / distance ≤ 1.0e12
- Position / travel path = 0.0

Column: Velocity

In this column, you enter the velocity for the selected command:

- **Command "Positioning Relative"**

  The command will move the axis at the given velocity.

  The given velocity will not be reached if the travel path selected is not large enough.

- **Command "Positioning Absolute"**

  The command will move the axis at the given velocity.

  The given velocity will not be reached if the target position is too close to the starting position.

- **Command " Velocity set point"**

  The command will move the axis at the given velocity.

  The given velocity will not be reached during the command if too short a runtime is selected.

Limit values (independent of the selected user unit):

- For the commands: "Positioning Relative" and "Positioning Absolute"
  - 1.0e-12 ≤ velocity ≤ 1.0e12
- For the command: "Velocity set point"
  - -1.0e12 ≤ velocity ≤ -1.0e-12
  - 1.0e-12 ≤ velocity ≤ 1.0e12
  - Velocity = 0.0

Column: Duration

Enter the duration of the selected command in this column:

- **Command " Velocity set point"**

  The command will move the axis for the specified duration. The duration includes both the acceleration phase and the constant travel phase. The next command will be processed once the duration is over.

- **Command "Wait"**

  Waits until the given duration is over.

Limit values (independent of the selected user unit):

- 0.001s ≤ duration ≤ 64800s

Column: Next step

Select the mode of transition to the next step from the drop-down list:

- **Complete command**

  The command will be completed. The next command will be processed immediately.

- **Blend motion**

  The motion of the current command will be blended with the motion of the following command. The transition mode "Blend motion" is available with command types "Positioning Relative" and "Positioning Absolute".

  Motion will be blended with motions of the following command types:

  – Positioning Relative

  – Positioning Absolute

  – Velocity set point

No blending occurs with other command types.

For a detailed description of the response of the axis when a command is appended or overlapped, see section Transition from "Complete command" to "Blend motion" (Page 138)

Column: Step code

Enter a numerical value / bit pattern in this column which is to be output at the "StepCode" output parameter of the "MC_CommandTable" Motion Control instruction while the command is being processed.

Limit values:

- 0 ≤ code number ≤ 65535

See also

Configuration - General (Page 127)

Shortcut menu commands - Command table (Page 131)

Working with the trend diagram (Page 133)

Shortcut menu commands - Curve chart (Page 137)

Transition from "Complete command" to "Blend motion" (Page 138)

Changing the command table configuration in the user program (Page 140)

6.4.3.3 Shortcut menu commands - Command table

The following shortcut menu commands are available in the command table:

Insert empty line

Adds an empty line above the selected line.

This shortcut menu command can only be executed if there are enough empty lines at the end of the command table.

Add empty line

Adds an empty line below the selected line.

This shortcut menu command can only be executed if there are enough empty lines at the end of the command table.

Insert separator line

Adds a separator line above the selected line.

You cannot have two consecutive separator lines.

Add separator line

Adds a separator line below the selected line.

You cannot have two consecutive separator lines, nor can you add a separator line at the end of the command table.

Cut

Removes the selected lines or content of the selected cell and saves them/it in the clipboard.

Selected lines will be deleted and the subsequent lines of the command table shifted up.

Copy

Copies the selected lines or content of the selected cell and saves them/it in the clipboard.

Paste

- Selected lines:

  Pastes the lines from the clipboard into the table above the selected line.
- Selected cell:

  Pastes the content of the clipboard into the selected line.

This shortcut menu command can only be executed if there are enough empty lines at the end of the command table.

Replace

Replaces the selected lines with the lines in the clipboard.

Delete

Deletes the selected lines. The lines below in the command table shift up.

See also

Configuration - General (Page 127)

Configuration - Command table (Page 127)

Working with the trend diagram (Page 133)

Shortcut menu commands - Curve chart (Page 137)

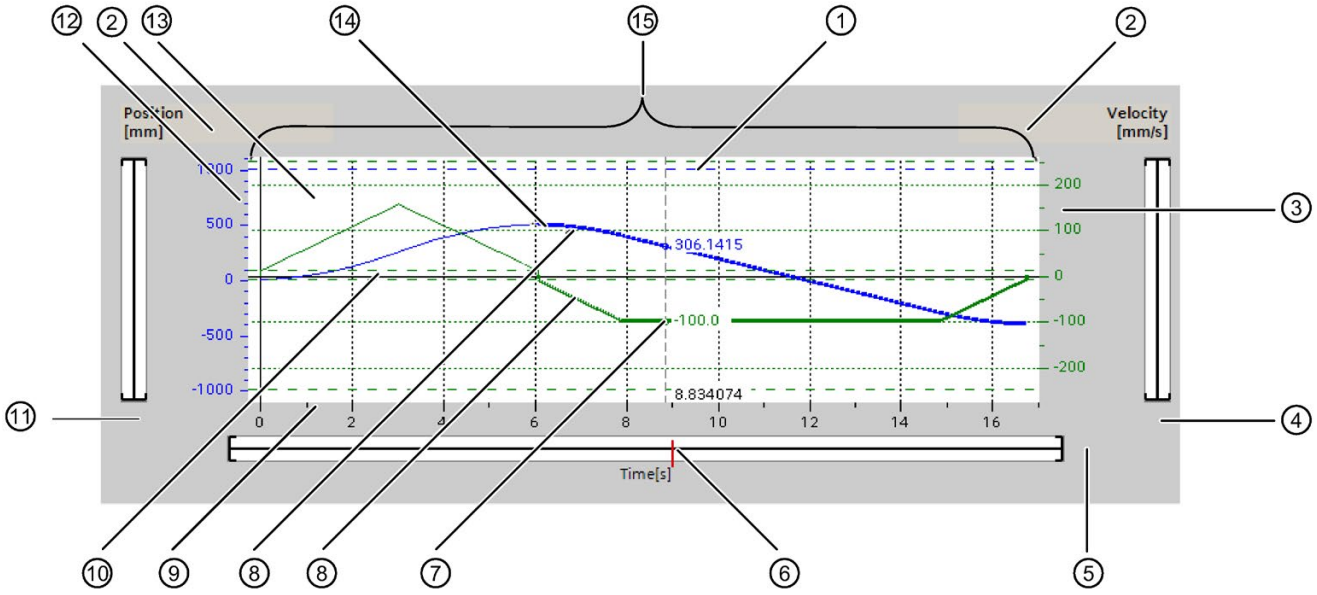
Transition from "Complete command" to "Blend motion" (Page 138)

Changing the command table configuration in the user program (Page 140)

6.4.3.4 Working with the trend diagram

The following tools and information are available in the trend view:

Trend view and components



①	Ruler
②	Selecting the grid
③	Velocity axis scale range
④	Scroll bar, velocity axis
⑤	Scroll bar time axis
⑥	Ruler position marking
⑦	Velocity curve
⑧	Curve section of a selected command
⑨	Time axis scale range
⑩	Start/stop velocity
⑪	Scroll bar, position axis
⑫	Position axis scale range
⑬	Software limit switch position
⑭	Position curve
⑮	Trend view

Selecting separator sections

If the command table consists of multiple sections separated by separators, you can select these sections in the trend view by selecting a command in the section.

Selecting commands

Commands can be selected in the trend view and in the command table:

- Click on a point on the velocity or position curve in the trend view. The corresponding command will be highlighted in the command table.
- Select a command in the command table.

  The corresponding section of curve will be highlighted.

Selecting the visible range of the trend view

Follow the steps below to adjust the section of the trend view to be displayed:

Select the scaling in the shortcut menu:

- Scale to curves:

  Scales the axes so the position and velocity curves are visible.

- Scale to curves and limits:

  Scales the axes so the position and velocity curves, the positions of the activated software limit switches and the minimum and maximum velocity limits are visible.

The view selected will be marked in the shortcut menu with a tick.

Selecting the section to be shown within the range:

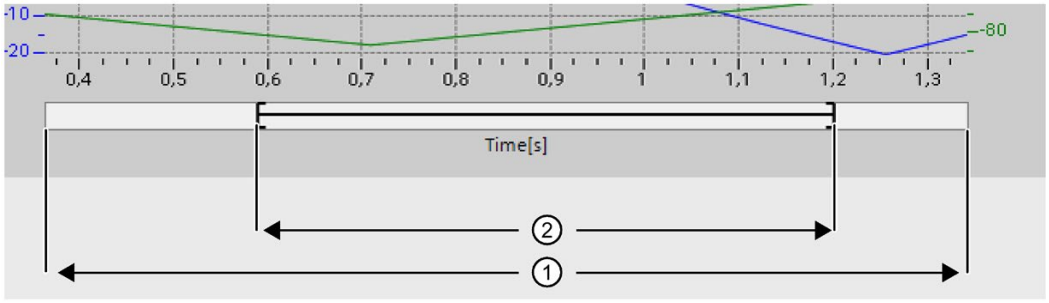
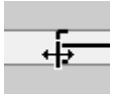
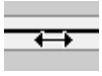
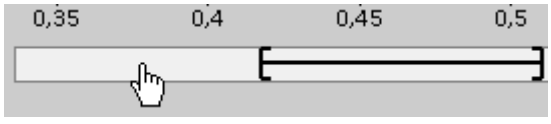
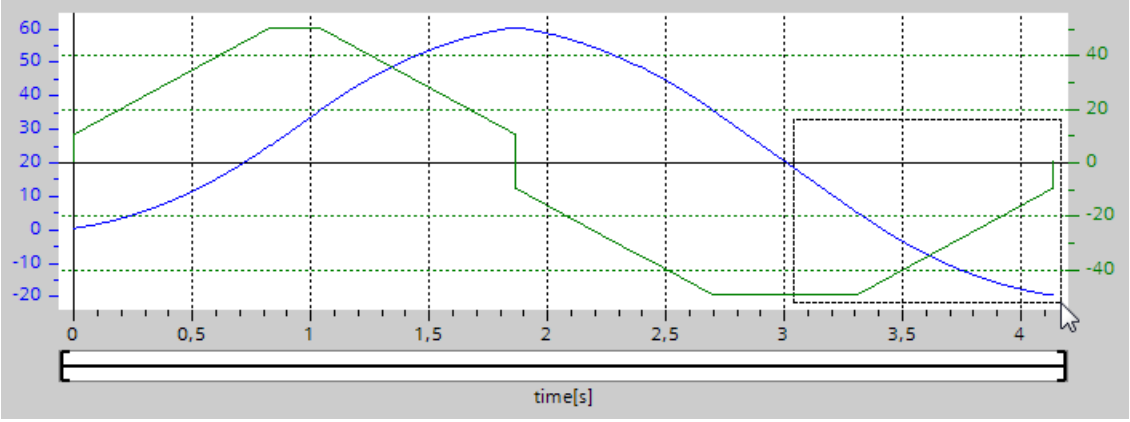


Figure 6-1 CmdTable_Scle01_new

①	Range which the curve values and / or limits are within. (see Selecting in the shortcut menu)
②	Selected range to be shown in the trend window. You set the range with the margin cursor at the right-hand and left-hand margin.  You set the position within range ① with the drag cursor.  You can also define the position by clicking in range ①. 

Selecting the section to be shown with the mouse:

Drag a section of the trend view by clicking and dragging with the mouse. The section of curve selected will be enlarged once you release the mouse.



Undoing the last change to the section:

Select the shortcut command "Undo zoom" to undo the last change to the section.

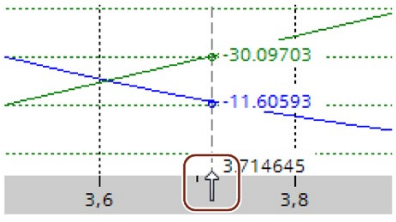
Synchronizing the grid

Click on the axis scales to select whether the grid is to be synchronized with the position axis or velocity axis.

Reading off curve values from the ruler

Activate the ruler using the shortcut menu command "Show ruler".

You can move the ruler to any point on the curves using the ruler cursor.



See also

Configuration - General (Page 127)

Configuration - Command table (Page 127)

Shortcut menu commands - Command table (Page 131)

Shortcut menu commands - Curve chart (Page 137)

Transition from "Complete command" to "Blend motion" (Page 138)

Changing the command table configuration in the user program (Page 140)

6.4.3.5 Shortcut menu commands - Curve chart

The following shortcut menu commands are available in the curve window:

Zoom 100%

Selects a zoom factor which will show 100% of the curve values and / or limits.

Undo zoom

Undoes the last zoom change.

Scaling on trends

Scales the axes so the position and velocity trends are visible.

Scaling on trends and limits

Scales the axes so the position and velocity trends, the positions of the activated software limit switches and the minimum and maximum velocity limits are visible.

Show velocity limits

Shows the lines of the velocity limits.

Show software limit switches

Shows the lines of the software limit switches.

Show measuring ruler

Fades the measuring ruler in / out

Use the measuring ruler when you want to see the individual values of the trends.

See also

Configuration - General (Page 127)

Configuration - Command table (Page 127)

Shortcut menu commands - Command table (Page 131)

Working with the trend diagram (Page 133)

Transition from "Complete command" to "Blend motion" (Page 138)

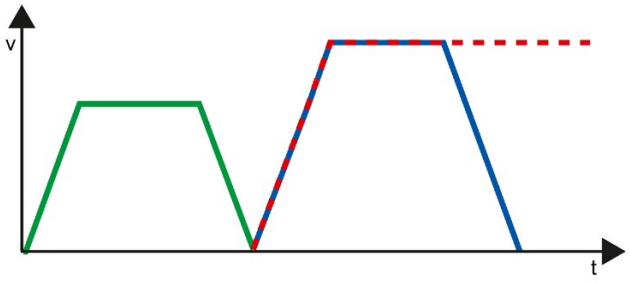
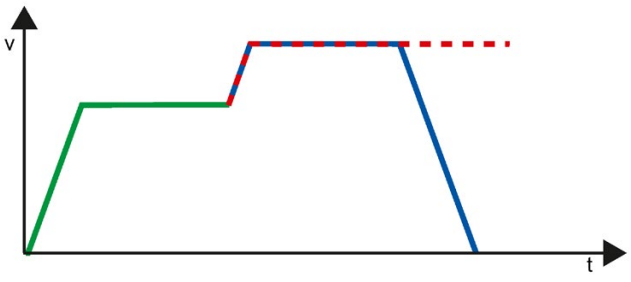
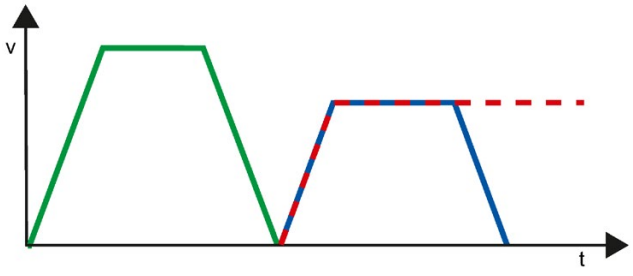
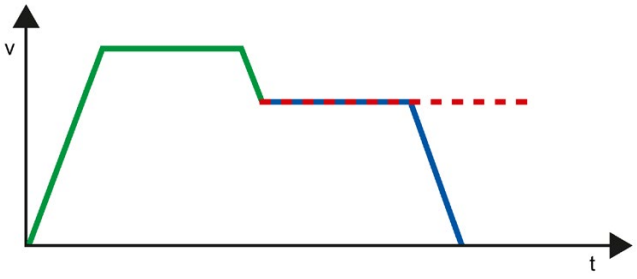
Changing the command table configuration in the user program (Page 140)

6.4.3.6 Transition from "Complete command" to "Blend motion"

The charts below show the transition between movements in various different transition modes in the "Next step" column:

Motion transition with preceding positioning jobs

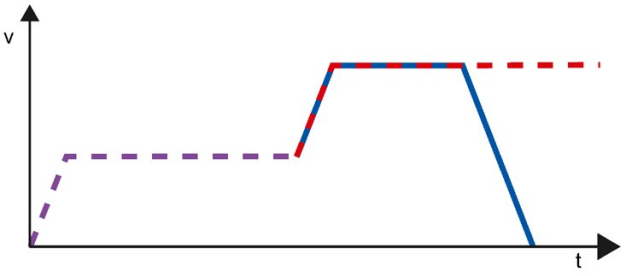
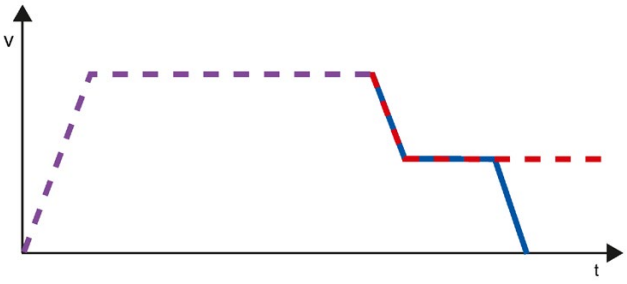
The following diagrams show a command sequence with two motion tasks. The first command is for positioning (green). The second command is for velocity (red) or positioning (blue):

Complete job	Blend motion
Transition from lower to higher velocity  A job with high velocity is appended to a previous positioning job. The positioning job terminates at its target position at velocity "0". The second job starts from standstill.	**Transition from lower to higher velocity**  A job with high velocity is overlapped with a previous positioning job. The first positioning job terminates without standstill at its target position. The second job starts with the new velocity.
Transition from higher to lower velocity  A job with low velocity is appended to a previous positioning job. The positioning job terminates at its target position at velocity "0". The second job starts from standstill.	**Transition from higher to lower velocity**  A job with low velocity is overlapped with a previous positioning job. The first positioning job terminates without standstill at its target position. The first job starts with the new velocity.

▬	1. Job "Positioning Relative" or "Positioning Absolute"
▬	2. Job "Velocity set point"
▬	2. Job "Positioning Relative" or "Positioning Absolute"

Motion transition with preceding velocity jobs

The following diagrams show a command sequence with two motion tasks. The first command is for velocity (violet). The second command is for velocity (red) or positioning (blue):

Transition from lower to higher velocity	Transition from higher to lower velocity
	
A job with a high velocity is appended to a previous velocity job. The first velocity job ends after the defined runtime. The second job starts with the new velocity.	A command with lower velocity is blended with a previous velocity command. The second job starts with the new velocity.

	1. Job "Velocity set point"
	2. Job "Velocity set point"
	2. Job "Positioning Relative" or "Positioning Absolute"

See also

Configuration - General (Page 127)

Configuration - Command table (Page 127)

Shortcut menu commands - Command table (Page 131)

Working with the trend diagram (Page 133)

Shortcut menu commands - Curve chart (Page 137)

Changing the command table configuration in the user program (Page 140)

6.4.3.7 Changing the command table configuration in the user program

You can change the following configuration parameters during runtime of the user program in the CPU:

Commands and corresponding values

You can also change the parameters of the command table during the runtime of the user program. Use the following technology object tags for this purpose:

- <Table name>.Command[1..32].Type

  for changing the command type
- <Table name>.Command[1..32].Position

  for changing the position/travel distance
- <Table name>.Command[1..32].Velocity

  for changing the velocity
- <Table name>.Command[1..32].Duration

  for changing the duration
- <Table name>.Command[1..32].NextStep

  for changing the parameter "Next step"
- <Table name>.Command[1..32].StepCode

  for changing the step code

Refer to the description of the technology object tags in the appendix for information on when changes to the configuration parameters take effect.

See also

Compatibility list of variables V1...3 <-> V4...5 (Page 48)

Configuration - General (Page 127)

Configuration - Command table (Page 127)

Shortcut menu commands - Command table (Page 131)

Working with the trend diagram (Page 133)

Shortcut menu commands - Curve chart (Page 137)

Transition from "Complete command" to "Blend motion" (Page 138)

6.4.4 Extended parameters

6.4.4.1 Configuration - Extended parameters

Configure the basic properties of the chart view of the "Command table" technology object in the "Extended parameters" configuration window.

Use axis parameters of

From the drop-down list, select which axis parameters are to be used for selecting the graphic view and checking the movement sequence. Select "Default axis" in the drop-down list if you have yet to add an axis to the "Technology object" folder or wish to use values which have not been configured in any of the available axes.

The axis parameters of the axis selected at the "Axis" parameter will be used to process the command table in the user program.

Unit of measurement position

If you select a default axis in the "Use axis parameter from" drop-down list, you can set the measurement unit.

If you select a configured axis in the drop-down list, the measurement unit of the configured axis is displayed.

Copy axis parameters

Select the direction of copy and the axis for copying the axis parameters. You can copy the axis parameters of the default axis to the selected axis or accept the axis parameters of the selected axis for the default axis. Use the "Apply configuration" button to copy the axis parameters according to your configuration.

6.4.4.2 Configuration - Dynamics

Configure the acceleration and deceleration and the jerk limit for the default axis in the "Dynamics" configuration window.

If you select a configured axis under "Configuration > Extended parameters > Extended parameters" in the drop-down list "Use axis parameters from", the values of the configured axis is displayed.

If you select the entry "Default axis" under "Configuration > Extended parameters > Extended parameters" in the drop-down list "Use axis parameters from", you can edit the fields described below.

Acceleration / deceleration

Set the desired acceleration of the default axis in the "Acceleration" field. The desired deceleration can be set in the "Deceleration" field.

Motion jobs configured in the command table will be calculated with the selected acceleration / deceleration.

Limit values:

- 1.0E-12 ≤ acceleration ≤ 1.0E12
- 1.0E-12 ≤ deceleration ≤ 1.0E12

Enable jerk limit

Enable the jerk limit with this check box.

Jerk

Set the desired jerk for ramping up and ramping down in the "Jerk" field.

Motion jobs configured in the command table will be calculated with the selected jerk.

Limit values:

- 1.0E-12 ≤ jerk ≤ 1.0E12

6.4.4.3 Configuration - Limit values

Configure the maximum velocity, the start/stop velocity and the software limit switches of the default axis in the "Limits" configuration window.

If you select a configured axis under "Configuration > Extended parameters > Extended parameters" in the drop-down list "Use axis parameters from", the values of the configured axis is displayed.

If you select the entry "Default axis" under "Configuration > Extended parameters > Extended parameters" in the drop-down list "Use axis parameters from", you can edit the fields described below.

Maximum velocity / Start/stop velocity

Define the maximum permissible velocity and the start/stop velocity of the default axis in these boxes. The start/stop velocity is the minimum permissible velocity of the default axis.

Velocity	Limit value
Start/stop velocity	0.0
	1.0E-12 to 1.0E12
Maximum velocity	0.0
	1.0E-12 to 1.0E12

The value of the maximum velocity must be greater or equal to the value of the start/stop velocity.

Enable SW limit switch

Activate the function of the low and high software limit switch with this check box. Movements in response to software limit switches being reached are not shown in the trend view.

Software high and low limit switch

Enter the position value of the low and high software limit switch in these boxes.

Software limit switch	Limit value
Software low limit switch	-1.0E12 to -1.0E-12
	0.0
	1.0E-12 to 1.0E12
Software high limit switch	-1.0E12 to -1.0E-12
	0.0
	1.0E-12 to 1.0E12

The value of the software high limit switch must be greater than or equal to the value of the software low limit switch.

Download to CPU

7

The data of the Motion Control technology objects are saved in the data blocks. The conditions for downloading of "blocks" therefore apply when loading a new or modified technology object.

CAUTION

Possible malfunctions of the axis when loading without hardware configuration

The hardware configuration is modified when the following modifications are made to the axis or encoder configuration:

- Modification of the pulse generator (PTO)
- Modification of the HW limit switch address
- Modification of the homing switch address
- Modification of the address of the PROFIdrive telegram
- Modification of the address of the analog output
- Modification of address of enable output or ready input

If the modified configuration of the axis or encoder is loaded with the shortcut menu commands "Software" or "Software (all blocks)" without downloading the hardware configuration, the axis may malfunction as a result.

Ensure that the current hardware configuration is downloaded to the CPU under the listed conditions.

Download in CPU S7-1200 RUN mode (from firmware version V2.2)

For CPU S7-1200 from firmware version V2.2, when loading in CPU RUN mode it is checked whether it is possible to load without stopping the CPU.

The following conditions apply when loading data blocks in RUN mode:

	Download to load memory	Download to work memory
Data block modified values	Yes	No
Data block modified structure	Yes (as of firmware V4)	Yes (as of firmware V4) • When downloading with reinitialization • For variables in system reserve for downloading without reinitialization
	No (firmware V2.2...3)	No (firmware V2.2...3)
New data block	Yes	Yes
Data block deleted	Yes	Yes

Also note the following when deleting data blocks and downloading data blocks with reinitialization:

- The axis must be disabled when downloading a positioning axis technology object.
- When downloading a command table technology object, no MC_CommandTable command with this command table must be active (parameter "Busy" = FALSE).
- When downloading an MC_Power instance data block, no MC_Power instruction must be active (parameter "Busy" = FALSE).

From technology version V3.0, Motion Control technology objects (data blocks) can also be downloaded in CPU RUN mode.

Technology objects lower than V3.0 cannot be downloaded in CPU RUN mode.

Select one of the actions described below to download the modified version of a Motion Control technology object (from version V3.0) to the work memory:

- **Technology object positioning axis and command table**
  Change the CPU operating mode from STOP to RUN.
- **Technology object positioning axis**
  Disable the axis and execute a "Restart" using the Motion Control instruction "MC_Reset".
- **Technology object command table**
  Ensure that the command table is not being used. Download the data block of the command table to the work memory using the extended instruction "READ_DBL".

See also

Guidelines on use of motion control (Page 42)

Commissioning

8

8.1 Axis control panel

Use the axis control panel to move the axis in manual mode, to optimize the axis settings, and to test your system.

The axis control panel can only be used if an online connection to the CPU is established. It is recommended to disable any other online communication when the axis control panel and the optimization is in use in order to keep the response times as short as possible.

The axis control panel is divided into the following areas:

- Master control
- Axis
- Command
- Current values
- Axis status

Note

Response times of the axis control panel

The response time during axis control panel operation depends on the communication load of the CPU. Close all other online windows of the TIA Portal to minimize the response time.

You can adjust the timeout in the start dialog.

Master control

In this area, you can take over master control of the technology object, or return it to your user program:

- **"Activate" button**

  With the "Activate" button, you set up an online connection to the CPU and take over master control for the selected technology object. Note the following when taking over master control:

  – To take over master control, the technology object must be disabled in the user program.

  – Until master control is returned, the user program has no influence on the functions of the technology object. Motion Control jobs from the user program to the technology object are rejected with error.

  CAUTION

  Additional axes in automatic mode

  The master control is only applied for the selected technology object. If additional axes are in automatic mode, dangerous situations may arise as a result.

  In this happens, disable all other axes.

- **"Deactivate" button**

  With the "Deactivate" button, you return master control to your user program.

Axis

In this area, enable or disable the technology object for operation with the axis control panel/optimization:

- **"Enable" button**

  With the "Enable" button, you enable the selected technology object.

- **"Disable" button**

  With the "Disable" button, you disable the selected technology object.

Command

Operation in the "command" area is only possible if the axis is enabled. You can select one of the following command inputs:

- **Jog**

  This command is equivalent to Motion Control command "MC_MoveJog" in the user program.

- **Positioning**

  This command is equivalent to the Motion Control commands "MC_MoveAbsolute" and "MC_MoveRelative" in the user program. The axis must be homed for absolute positioning.

- **Homing**

  This command is equivalent to Motion Control command "MC_Home" in the user program.

  – The "Set reference point" button corresponds to Mode = 0 (direct homing absolute)

  – The "Homing" button corresponds to Mode = 3 (active homing)

  For active homing, the homing switch must be configured in the axis configuration.

  The values for approach velocity, homing velocity, and reference position offset are taken from the axis configuration unchanged.

Depending on the selection, the relevant boxes for entry of setpoints and the buttons for starting the command are displayed.

Select the "Enable jerk limitation" check box to activate the jerk limitation. By default, the jerk is applied with 10% of the configured value. This value can be changed as required.

Current values

The following actual values of the axis are displayed in this area:

- Position
- Velocity

Axis status

The current axis status and drive status are shown in the "Axis status" area.

Status message	Description
Enabled	The axis is enabled and ready to be controlled via Motion Control commands.
Homed	The axis is homed and is capable of executing absolute positioning commands of Motion Control instruction "MC_MoveAbsolute".
Ready	The drive is ready for operation.
Axis error	An error has occurred in the positioning axis technology object. The "Error message" box displays detailed information about the cause of the error.
Encoder values valid	The encoder values are valid.
Simulation active	The axis is simulated in the CPU. Setpoints are not output to the drive.
Drive error	The drive has reported an error due to loss of its "Drive ready" signal.
Restart required	A modified configuration of the axis was downloaded to the load memory in CPU RUN mode. To download the modified configuration to the work memory, you need to restart the axis. Use the Motion Control instruction "MC_Reset" to do this.

The "Info message" box displays advanced information about the status of the axis.

The "Error message" box shows the current error.

Click "Acknowledge" to acknowledge all cleared errors.

Note

Initial values for velocity, acceleration/deceleration and jerk

For safety reasons, the "Velocity", "Acceleration/Deceleration" and "Jerk" parameters are initialized with values equivalent to only 10% of the configured values when the axis control panel is activated. The "Jerk" parameter is only used for technology object "Axis" V2.0 and higher.

The values in the configuration view displayed when you select "Extended parameters > Dynamics > General" are used for initialization.

The "Velocity" parameter on the axis control panel is derived from the "Maximum velocity" and the "Acceleration/Deceleration" parameters from "Acceleration" in the configuration.

The "Velocity", "Acceleration/deceleration" and "Jerk" parameters can be changed in the axis control panel. This does not affect the values in the configuration.

See also

Guidelines on use of motion control (Page 42)

8.2 Tuning

The movement of axes with drive connection via PROFIdrive/analog output is position-controlled.

The "Tuning" function supports you in determining the optimal gain (Kv factor) for the control loop (Page 98) of the axis. The axis velocity profile is recorded by means of the Trace function for this purpose for the duration of a configurable positioning movement. Then you can evaluate the recording, and adapt the gain accordingly. It is recommended to disable any other online communication when the axis control panel and the optimization is in use in order to keep the response times as short as possible.

The "Tuning" function for the positioning axis technology object can be found in the project tree under "Technology object > Commissioning".

The "Tuning" dialog is divided into the following areas:

- Master control
- Axis
- Axis status
- Optimize gain setting
- Run measurement
- Trace

Note

No transfer of the parameters

The configured parameter values are discarded after master control is returned. Transfer the values as needed into your configuration.

Start of optimization

The trace is started at the same time the optimization starts. A timeout can be adapted in the start dialog for this.

Master control

In this area, you can take over master control of the technology object, or return it to your user program:

- **"Activate" button**

  With the "Activate" button, you set up an online connection to the CPU and take over master control for the selected technology object. Note the following when taking over master control:

  – To take over master control, the technology object must be disabled in the user program.

  – Until master control is returned, the user program has no influence on the functions of the technology object. Motion Control jobs from the user program to the technology object are rejected with error.

  CAUTION

  Additional axes in automatic mode

  The master control is only applied for the selected technology object. If additional axes are in automatic mode, dangerous situations may arise as a result.

  In this happens, disable all other axes.

- **"Deactivate" button**

  With the "Deactivate" button, you return master control to your user program.

Axis

In this area, enable or disable the technology object for operation with the axis control panel/optimization:

- **"Enable" button**

  With the "Enable" button, you enable the selected technology object.

- **"Disable" button**

  With the "Disable" button, you disable the selected technology object.

Axis status

The current axis status and drive status are shown in the "Axis status" area.

Status message	Description
Enabled	The axis is enabled and ready to be controlled via Motion Control commands.
Homed	The axis is homed and is capable of executing absolute positioning commands of Motion Control instruction "MC_MoveAbsolute".
Ready	The drive is ready for operation.
Axis error	An error has occurred in the positioning axis technology object. The "Error message" box displays detailed information about the cause of the error.
Encoder values valid	The encoder values are valid.
Simulation active	The axis is simulated in the CPU. Setpoints are not output to the drive.
Drive error	The drive has reported an error due to loss of its "Drive ready" signal.
Restart required	A modified configuration of the axis was downloaded to the load memory in CPU RUN mode. To download the modified configuration to the work memory, you need to restart the axis. Use the Motion Control instruction MC_Reset to do this.

The "Info message" box displays advanced information about the status of the axis.

The "Error message" box shows the current error.

Click "Acknowledge" to acknowledge all cleared errors.

Optimize gain setting

You make the settings for optimization of the gain in this area:

- Precontrol

  In this field, configure the current velocity precontrol of the position controller as a percentage.

- Distance

  In this field, configure the load distance for one test step.

- "Customize dynamics" check box

  Select this option to adapt the acceleration and the maximum acceleration for the optimization.

- Velocity

  In this field, you configure the maximum velocity for a test step.

- Acceleration

  In this field, you configure the acceleration for a test step.

- **Measurement duration**

  The measurement duration is recalculated and entered depending on the selected acceleration, velocity and distance.

  You can adapt the value of the measurement duration afterwards.

- **Gain**

  In this field, you configure the actual gain of the position controller (Kv).
  The gain takes effect when it is entered. If the gain of the position controller (Kv) is too large, this can lead to an error on the drive.

Run measurement

Perform the test steps in this area:

- **"Forward" button**

  With the "Forward" button, you start a test step for optimization in the positive direction.

- **"Backward" button**

  With the "Backward" button, you start a test step for optimization in the negative direction.

- **"Stop" button**

  You can use the "Stop" button to end the current movement for optimization and end trace recording.

Trace

With each test step, a Trace recording of the required parameters is automatically started and displayed after completion of the test step. After master control has been returned, the Trace recording is deleted.

You will find a full description of the Trace function in the section on using the trace and logic analyzer function in the TIA Portal help.

Programming 9

9.1 Overview of the Motion Control statements

You control the axis with the user program using Motion Control instructions. The instructions start Motion Control commands that execute the desired functions.

The status of the Motion Control commands and any errors that occur during their execution can be obtained from the output parameters of the Motion Control instructions. The following Motion Control instructions are available:

- MC_Power: Enable, disable axis as of V6 (Page 181)
- MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)
- MC_Home: Home axes, set reference point as of V6 (Page 188)
- MC_Halt: Stop axis as of V6 (Page 193)
- MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)
- MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)
- MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)
- MC_MoveJog: Move axis in jog mode as of V6 (Page 208)
- MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)
- MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)
- MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)
- MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

See also

Creating a user program (Page 155)

Programming notes (Page 158)

Behavior of the Motion Control commands after POWER OFF and restart (Page 160)

Monitoring active commands (Page 161)

Error displays of the Motion Control statements (Page 172)

9.2 Creating a user program

In the section below you learn how to create a user program with the basic configuration for controlling your axis. All available axis functions are controlled using the Motion Control instructions to be inserted.

Requirement

- The technology object has been created and configured without errors.

Before creating and testing the user program, it is advisable to test the function of the axis and the corresponding parts of the system with the axis command table.

Procedure

Proceed as follows to create the user program in accordance with the principles described below:

1. In the project tree, double-click your code block (the code block must be called in the cyclic program).

   The code block is opened in the programming editor and all available instructions are displayed.

2. Open the "Technology" category and the "Motion Control" folder.
3. Use a drag-and-drop operation to move the "MC_Power" instruction to the desired network of the code block.

   The dialog box for defining the instance DB opens.

4. In the next dialog box, select from the following alternatives:

   Single instance

   Click "Single instance" and select whether you want to define the name and number of the instance DB automatically or manually.

   Multi-instance

   Click "Multi-instance" and select whether you want to define the name of the multi-instance automatically or manually.

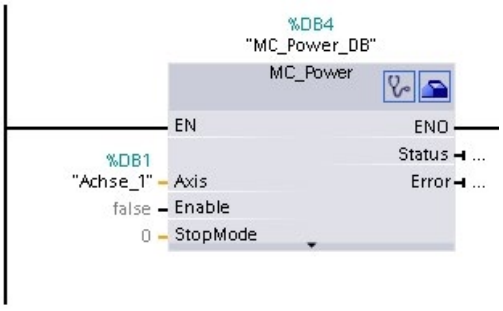
5. Click "OK".

   The Motion Control instruction "MC_Power" is inserted into the network.

   Parameters marked with "<???>" must be initialized; default values are assigned to all other parameters.

   Parameters displayed in black are required for use of the Motion Control instruction.

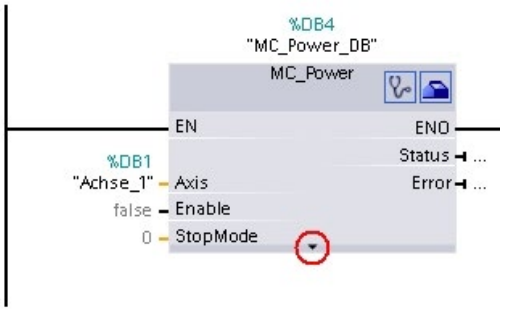
6. Select the technology object in the project tree and drag-and-drop it on <???>.



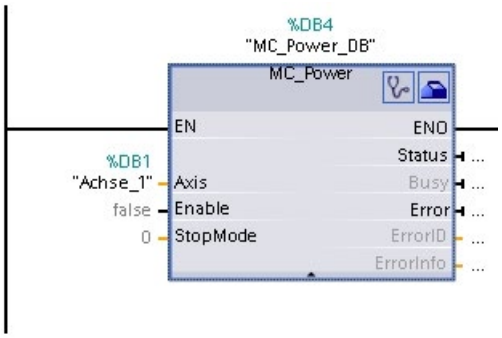
   After the selection of the technology object data block, the following buttons are available:

   Click the stethoscope icon if you want to open the diagnostics dialog for the technology object.

   Click the toolbox icon if you want to open the configuration view of the technology object.



   Click the arrow down icon to view additional parameters of the Motion Control instruction.



The grayed-out parameters now visible can be used optionally.

7. Add your choice of Motion Control instructions in accordance with steps 3 to 6.

Result

You have created the basic configuration for axis control in the user program.

Initialize the input parameters of Motion Control instructions in other parts of the user program to initiate the desired jobs for the "Axis" technology object.

Evaluate the output parameters of the Motion Control instructions and the tags of the data block to track the initiated jobs and the status of the axis.

Refer to the detailed description for details on the parameters of Motion Control instructions.

See also

Overview of the Motion Control statements (Page 154)

Programming notes (Page 158)

Behavior of the Motion Control commands after POWER OFF and restart (Page 160)

Monitoring active commands (Page 161)

Error displays of the Motion Control statements (Page 172)

9.3 Programming notes

When creating your user program, note the following information:

- **Cyclic call of utilized motion control instructions**

  The current status of command execution is available via the output parameters of the motion control instruction. The status is updated with every call of the motion control instruction. Therefore, make sure that the utilized motion control instructions are called cyclically.

- **Transfer of parameter values of a motion control instruction**

  The parameter values pending for the input parameters are transferred with a positive edge at input parameter "Execute" when the block is called.

  The motion control command is started with these parameter values. Parameter values that are subsequently changed for the motion control instruction are not transferred until the next start of the motion control command.

  Exceptions to this are input parameters "StopMode" of motion control instruction "MC_Power" and "Velocity" of motion control instruction "MC_MoveJog". A change in the input parameter is also applied with "Enable" = TRUE or "JogForward" and "JogBackward".

- **Programming under consideration of the status information**

  In a stepwise execution of motion control jobs, make sure to wait for the active command to finish before starting a new command. Use the status messages of the motion control instruction and the "StatusBits" tag of the technology object to check for completion of the active command.

  In the examples below, observe the indicated sequence. Failure to observe the sequence will display an axis or command error.

  – **Axis enable with motion control instruction "MC_Power"**

    You must enable the axis before it can take on motion jobs. Use an AND operation of tag <Axis name>.StatusBits.Enable = TRUE with output parameter Status = TRUE of motion control instruction "MC_Power" to verify that the axis is enabled.

  – **Acknowledge error with motion control instruction "MC_Reset"**

    Prior to starting a motion control command, errors requiring acknowledgement must be acknowledged with "MC_Reset". Eliminate the cause of the error and acknowledge the error with motion control instruction "MC_Reset". Verify that the error has been successfully acknowledged before initiating a new command. For this purpose, use an AND operation of tag <Axis name>.StatusBits.Error = FALSE with output parameter Done = TRUE of motion control instruction "MC_Reset".

  – **Home axis with motion control instruction "MC_Home"**

    Before you can start an MC_MoveAbsolute command, the axis must be homed. Use an AND operation of tag <Axis name>.StatusBits.HomingDone = TRUE with output parameter Done = TRUE of motion control instruction "MC_Home" to verify that the axis has been homed.

- **Override of motion control command processing**

  Motion control jobs for moving an axis can also be executed as overriding jobs.

  If a new motion control command is started for an axis while another motion control command is active, the active command is overridden by the new command before the existing command is completely executed. The overridden command signals this using CommandAborted = TRUE in the motion control instruction. It is possible to override an active MC_MoveRelative command with a MC_MoveAbsolute command.

- **Avoiding multiple use of the same instance**

  All relevant information of a motion control command is stored in its instance.

  Do not start a new command using this instance, if you want to track the status of the current command. Use different instances if you want to track the commands separately. If the same instance is used for multiple motion control commands, the status and error information of the individual commands will overwrite each other.

  In the user program, each axis must be called with a separate call of the Motion Control instruction "MC_Power" with a separate instance data block.

- **Call of motion control instructions in different priority classes (run levels)**

  Motion Control instructions with the same instance may not be called in different priority classes without interlocking. To learn how to call locked motion control instructions, refer to "Tracking commands from higher priority classes (run levels) (Page 224)".

See also

Overview of the Motion Control statements (Page 154)

Creating a user program (Page 155)

Behavior of the Motion Control commands after POWER OFF and restart (Page 160)

Monitoring active commands (Page 161)

Error displays of the Motion Control statements (Page 172)

Tracking jobs from higher priority classes (execution levels) (Page 224)

9.4 Behavior of the Motion Control commands after POWER OFF and restart

A POWER OFF or CPU-STOP aborts all active motion control jobs. All CPU outputs, including pulse and direction outputs, are reset.

After a subsequent POWER ON or CPU restart (CPU RUN), the technology objects and the motion control jobs will be reinitialized.

All actual data of the technology objects as well as all status and error information of the previously active motion control jobs are reset to their initial values.

Before the axis can be reused, it must be enabled again using the Motion Control instruction "MC_Power". If homing is required, the axis must be homed again with Motion Control instruction "MC_Home". When an absolute encoder is used, homing is retained after POWER OFF.

See also

Overview of the Motion Control statements (Page 154)

Creating a user program (Page 155)

Programming notes (Page 158)

Monitoring active commands (Page 161)

Error displays of the Motion Control statements (Page 172)

9.5 Monitoring active commands

9.5.1 Monitoring active commands

There are three typical groups for tracking active Motion Control commands:

- **Motion control instructions with output parameter "Done"**
- **Motion control instruction "MC_MoveVelocity"**
- **Motion control instruction "MC_MoveJog"**

9.5.2 Motion control instructions with "Done" output parameter

Motion control instructions with the output parameter "Done" are started via input parameter "Execute" and have a defined conclusion (for example, with Motion Control instruction "MC_Home": Homing was successful). The command is complete and the axis is at a standstill.

The commands of the following Motion Control instructions have a defined conclusion:

- MC_Reset
- MC_Home
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_CommandTable (technology object as of V2)
- MC_ChangeDynamic (technology object as of V2)
- MC_WriteParam (as of technology object V4)
- MC_ReadParam (as of technology object V4)

The output parameter "Done" indicates the value TRUE, if the command has been successfully completed.

The output parameters "Busy", "CommandAborted", and "Error" signal that the command is still being processed, has been aborted or an error is pending. The Motion Control instruction "MC_Reset" cannot be aborted and thus has no "CommandAborted" output parameter. The Motion Control instruction "MC_ChangeDynamic" is completed immediately and therefore has no "Busy" or "CommandAborted" output parameters.

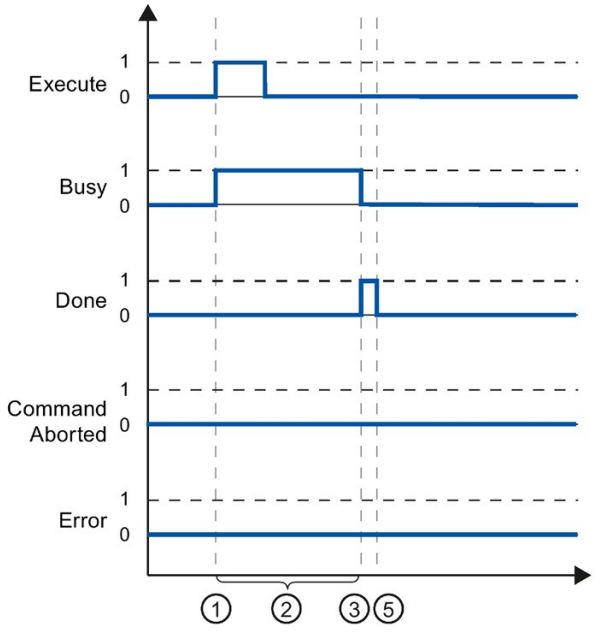
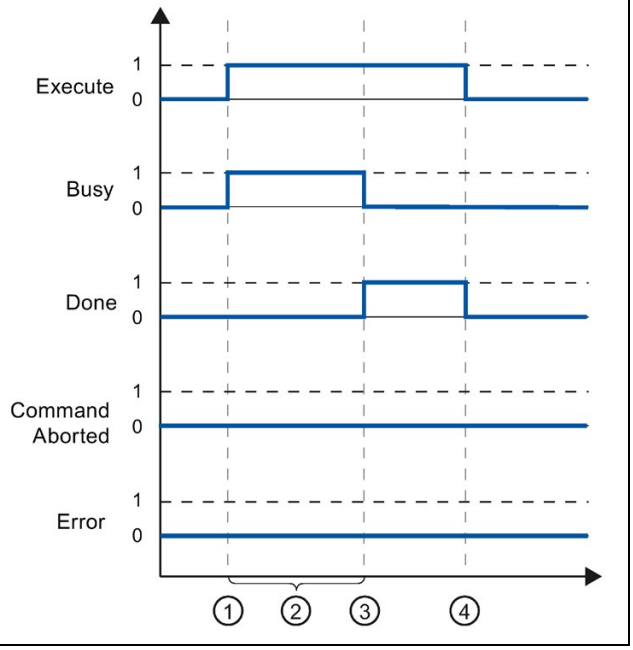
During execution of the Motion Control command, the output parameter "Busy" indicates the value TRUE. If the command has been completed, aborted, or stopped by an error, the output parameter "Busy" changes its value to FALSE. This change occurs regardless of the signal at input parameter "Execute".

Output parameters "Done", "CommandAborted", and "Error" indicate the value TRUE for at least one cycle. These status messages are latched while input parameter "Execute" is set to TRUE.

The behavior of the status bits is presented below for various example situations:

Complete execution of command

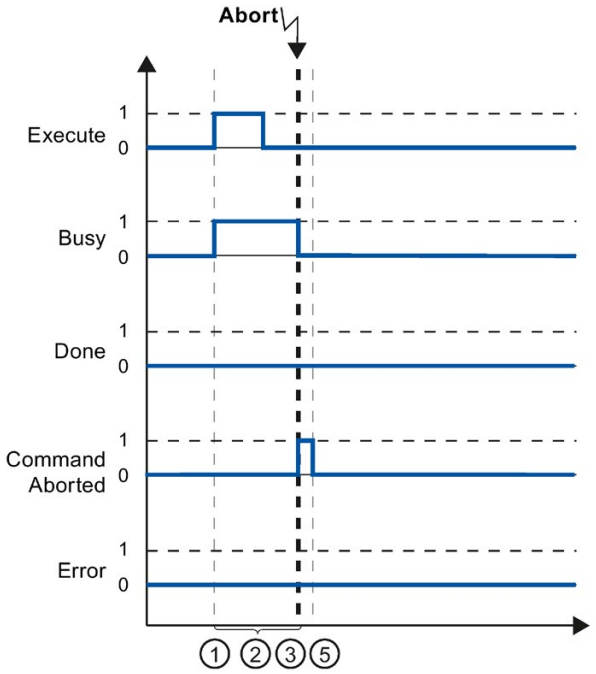
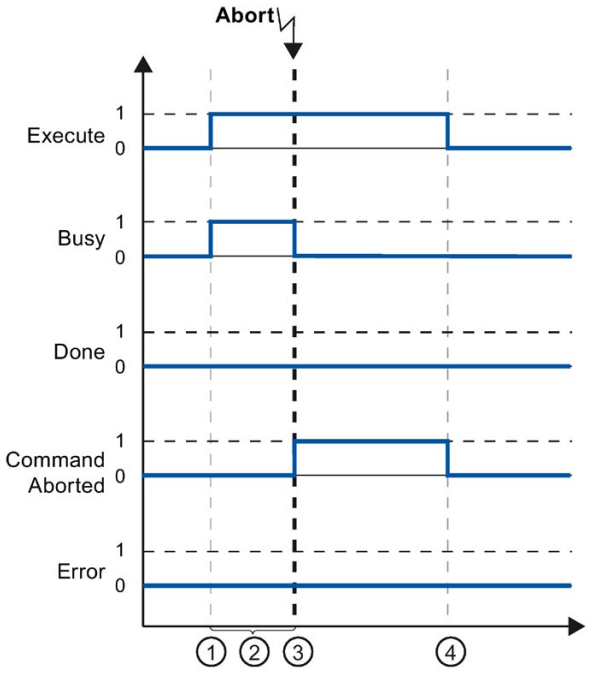
If the Motion Control command has been completely executed by the time of its conclusion, this is indicated by the value TRUE in output parameter "Done". The signal status of the input parameter "Execute" influences the display duration at the output parameter "Done":

"Execute" changes its value to FALSE **during processing** of the command	"Execute" changes its value to FALSE **after completion** of the command
	

①	The command is started with a positive edge at the input parameter "Execute". Depending on the programming, "Execute" can still be reset to the value FALSE during the command, or the value TRUE can be retained until after completion of the command.
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	With conclusion of the command (for example, for Motion Control instruction "MC_Home": Homing was successful), output parameter "Busy" changes to FALSE and "Done" to TRUE.
④	If "Execute" retains the value TRUE until after completion of the command, then "Done" also remains TRUE and changes its value to FALSE together with "Execute".
⑤	If "Execute" has been set to FALSE before the command is complete, "Done" indicates the value TRUE for only one execution cycle.

Abort command

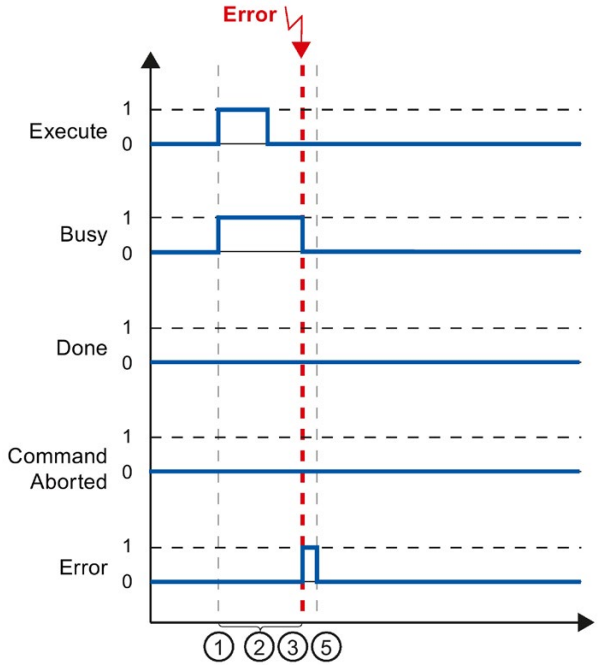
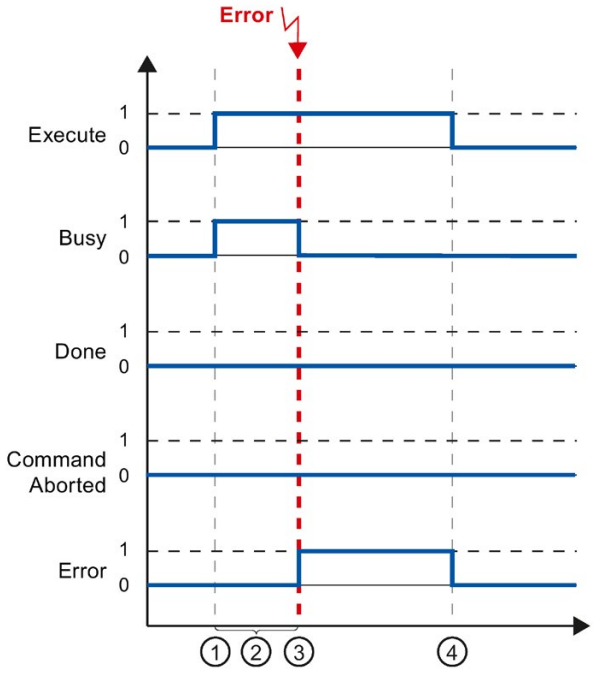
If the Motion Control command is aborted during execution, this is indicated by the value TRUE in output parameter "CommandAborted". The signal status of the input parameter "Execute" influences the display duration at the output parameter "CommandAborted":

"Execute" changes its value to FALSE **before** the command is aborted.	"Execute" changes its value to FALSE **after** the command is aborted.
	

①	The command is started with a positive edge at the input parameter "Execute". Depending on the programming, "Execute" can still be reset to the value FALSE during the command, or the value TRUE can be retained until after completion of the command.
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	During command execution, the command is aborted by another Motion Control command. If the command is aborted, output parameter "Busy" changes to FALSE and "CommandAborted" to TRUE.
④	If "Execute" retains the value TRUE until after the command is aborted, then "CommandAborted" also remains TRUE and changes its value to FALSE together with "Execute".
⑤	If "Execute" has been set to FALSE before the command is aborted, "CommandAborted" indicates the value TRUE for only one execution cycle.

Error during command execution

If an error occurs during execution of the Motion Control command, this is indicated by the value TRUE in the output parameter "Error". The signal status of the input parameter "Execute" influences the display duration at the output parameter "Error":

"Execute" changes its value to FALSE **before** the error occurs	"Execute" changes its value to FALSE **after** the error occurs
	

①	The command is started with a positive edge at the input parameter "Execute". Depending on the programming, "Execute" can still be reset to the value FALSE during the command, or the value TRUE can be retained until after completion of the command.
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	An error occurred during command execution. When the error occurs, the output parameter "Busy" changes to FALSE and "Error" to TRUE.
④	If "Execute" retains the value TRUE until after the error occurs, then "Error" also remains TRUE and only changes its value to FALSE together with "Execute".
⑤	If "Execute" has been set to FALSE before the error occurs, "Error" indicates the value TRUE for only one execution cycle.

9.5.3 Motion control instruction MC_MoveVelocity

A "MC_MoveVelocity" command is started with a positive edge at the "Execute" parameter. The command objective is fulfilled when the assigned velocity is reached and the axis travels at constant velocity. When the assigned velocity is reached and maintained, this is indicated in the "InVelocity" parameter with the value TRUE.

The motion of the axis can, for example, be stopped with an "MC_Halt" command.

The output parameters "Busy", "CommandAborted", and "Error" signal that the command is still being processed, has been aborted or an error is pending.

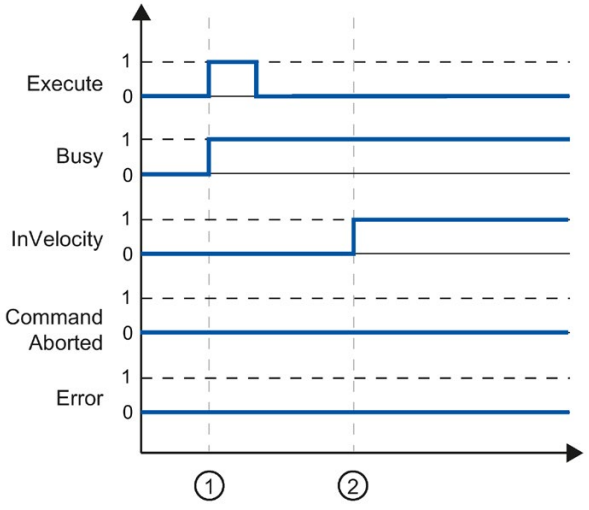
During execution of the Motion Control command, the output parameter "Busy" indicates the value TRUE. If the command has been stopped by another command or by an error, the "Busy" output parameter changes its value to FALSE. This change occurs regardless of the signal at the input parameter "Execute".

The output parameters "CommandAborted" and "Error" show the value TRUE for at least one cycle. These status messages are latched while input parameter "Execute" is set to TRUE.

The behavior of the status bits is presented below for various example situations:

The parameterized velocity is reached

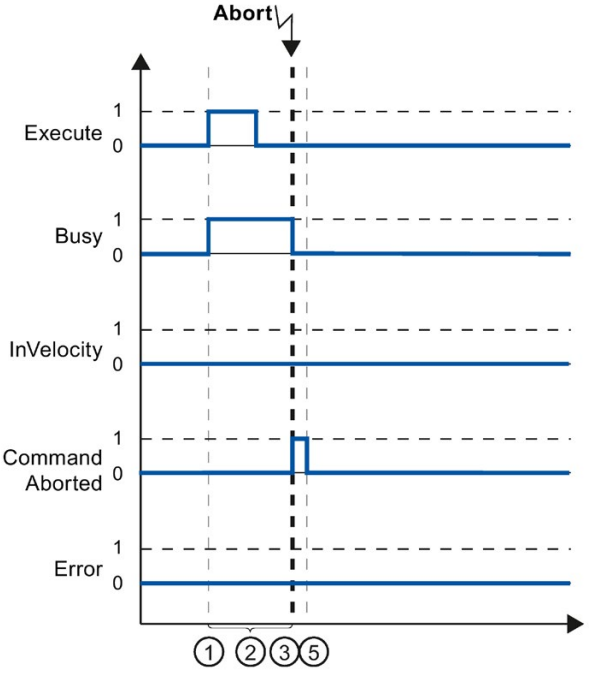
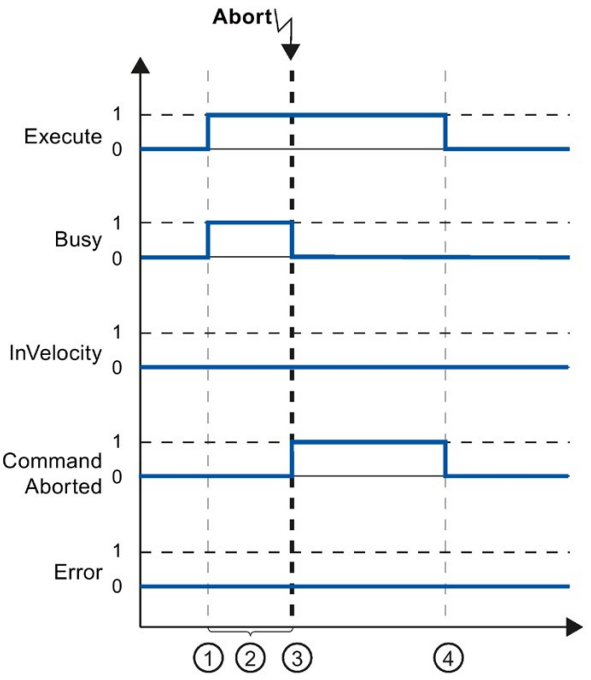
If the Motion Control command has been executed by the time the parameterized velocity is reached, this is indicated by the value TRUE in output parameter "InVelocity". The parameter "Execute" has no effect on the indication duration in the "InVelocity" parameter.



①	The job is started with a positive edge at the "Execute" parameter. Depending on the programming, "Execute" can be reset to the FALSE value before or after the parameterized velocity has been reached. While the job is active, the parameter "Busy" shows the value TRUE.
②	When the assigned velocity is reached, the "InVelocity" parameter changes to TRUE. The "Busy" and "InVelocity" parameters retain the TRUE value until the "MC_MoveVelocity" command is overridden by another Motion Control command or stopped by an error.

The command is aborted prior to reaching the parameterized velocity

If the Motion Control command is aborted before the parameterized velocity is reached, this is indicated by the value TRUE in output parameter "CommandAborted". The signal status of input parameter "Execute" influences the display duration at the output parameter "CommandAborted".

"Execute" changes its value to FALSE **before** the command is aborted.	"Execute" changes its value to FALSE **after** the command is aborted.
	

①	The command is started with a positive edge at the input parameter "Execute". Depending on the programming, "Execute" can still be reset to the value FALSE during the command, or the value TRUE can be retained until after the command is aborted.
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	During command execution, the command is aborted by another Motion Control command. If the command is aborted, output parameter "Busy" changes to FALSE and "CommandAborted" to TRUE.
④	If "Execute" retains the value TRUE until after the command is aborted, then "CommandAborted" also remains TRUE and changes its status to FALSE together with "Execute".
⑤	If "Execute" has been reset to FALSE before the command is aborted, "CommandAborted" indicates the value TRUE for only one execution cycle.

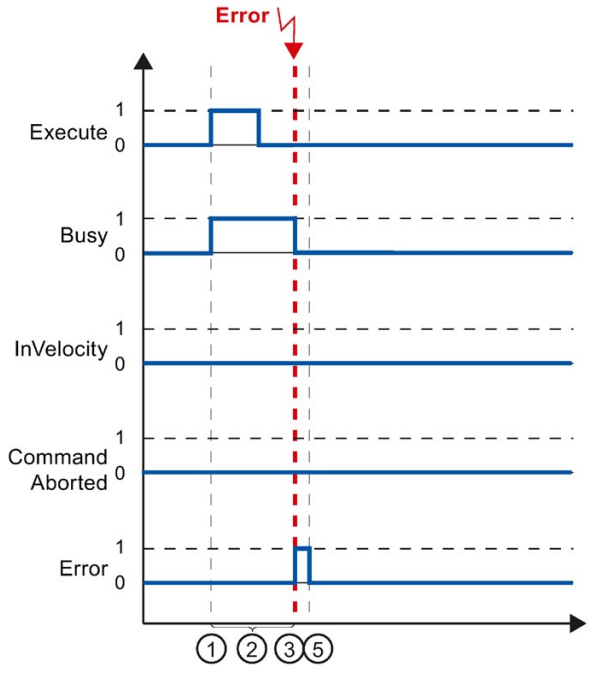
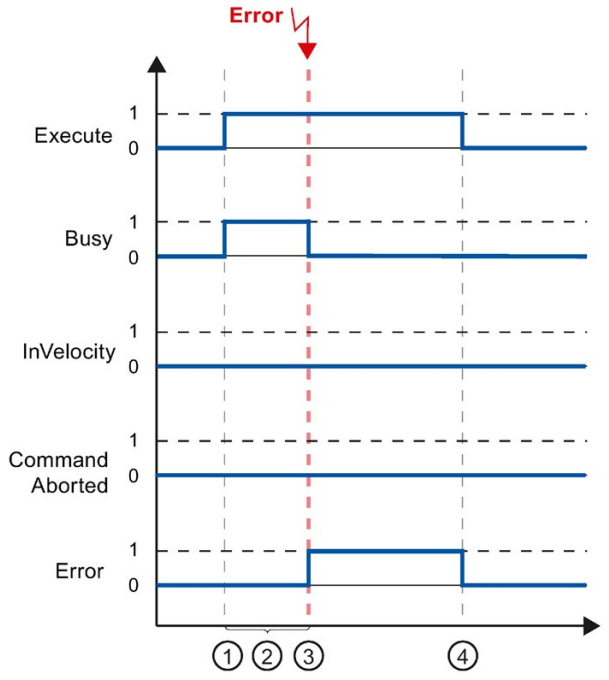
Note

Under the following conditions, an abort is not indicated in output parameter "CommandAborted":

The parameterized velocity has been reached, input parameter "Execute" has the value FALSE, and a new Motion Control command is initiated.

An error has occurred prior to reaching the parameterized velocity

If an error occurs during execution of the Motion Control command before the parameterized velocity has been reached, this is indicated by the value TRUE in the output parameter "Error". The signal status of the input parameter "Execute" influences the display duration at the output parameter "Error":

"Execute" changes its value to FALSE **before** the error occurs	"Execute" changes its value to FALSE **after** the error occurs
	

①	The command is started with a positive edge at the input parameter "Execute". Depending on the programming, "Execute" can still be reset to the value FALSE during the command, or the value TRUE can be retained until after the error has occurred.
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	An error occurred during command execution. When the error occurs, the output parameter "Busy" changes to FALSE and "Error" to TRUE.
④	If "Execute" retains the value TRUE until after the error has occurred, then "Error" also remains TRUE and only changes its status to FALSE together with "Execute".
⑤	If "Execute" has been reset to FALSE before the error occurs, "Error" indicates the value TRUE for only one execution cycle.

Note

Under the following conditions, an error is not indicated in output parameter "Error":

The parameterized velocity has been reached, input parameter "Execute" has the value FALSE, and an axis error occurs (software limit switch is approached, for example).

The error of the axis is only indicated in the "MC_Power" Motion Control instruction.

9.5.4 Motion control instruction MC_MoveJog

The commands of Motion Control instruction "MC_MoveJog" implement a jog operation.

The motion control commands "MC_MoveJog" do not have a defined end. The command objective is fulfilled when the parameterized velocity is reached for the first time and the axis travels at constant velocity. When the parameterized velocity is reached, this is indicated by the value TRUE in output parameter "InVelocity".

The order is complete when input parameter "JogForward" or "JogBackward" has been set to the value FALSE and the axis has come to a standstill.

The output parameters "Busy", "CommandAborted", and "Error" signal that the command is still being processed, has been aborted or an error is pending.

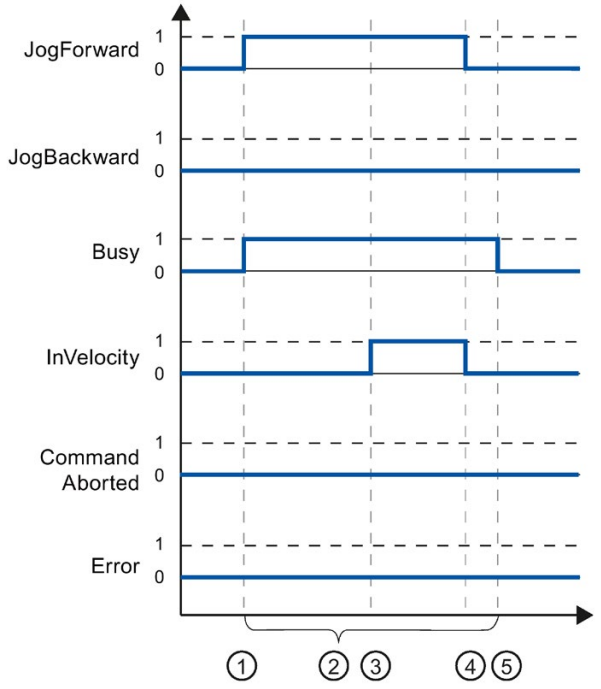
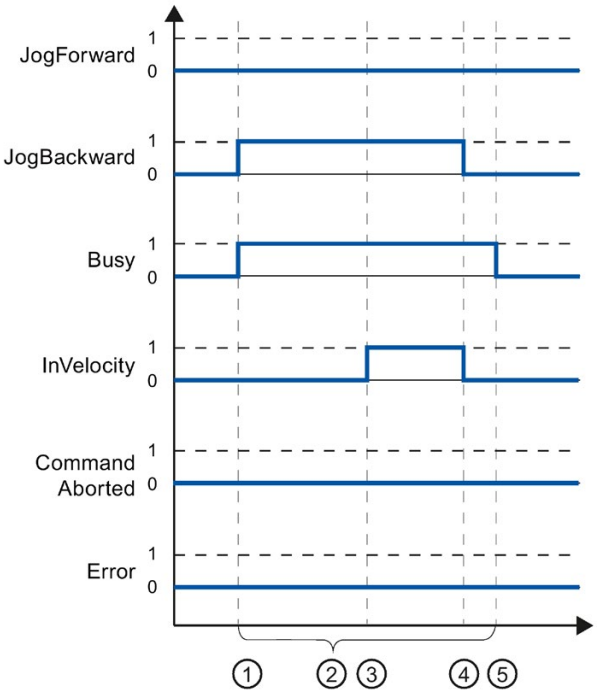
During processing of the motion control command, the output parameter "Busy" indicates the value TRUE. If the command has been completed, aborted, or stopped by an error, the output parameter "Busy" changes its value to FALSE.

The output parameter "InVelocity" indicates the status TRUE, as long as the axis is moving at the parameterized velocity. The output parameters "CommandAborted" and "Error" indicate the status for at least one cycle. These status messages are latched as long as either input parameter "JogForward" or "JogBackward" is set to TRUE.

The behavior of the status bits is presented below for various example situations:

The parameterized velocity is reached and maintained

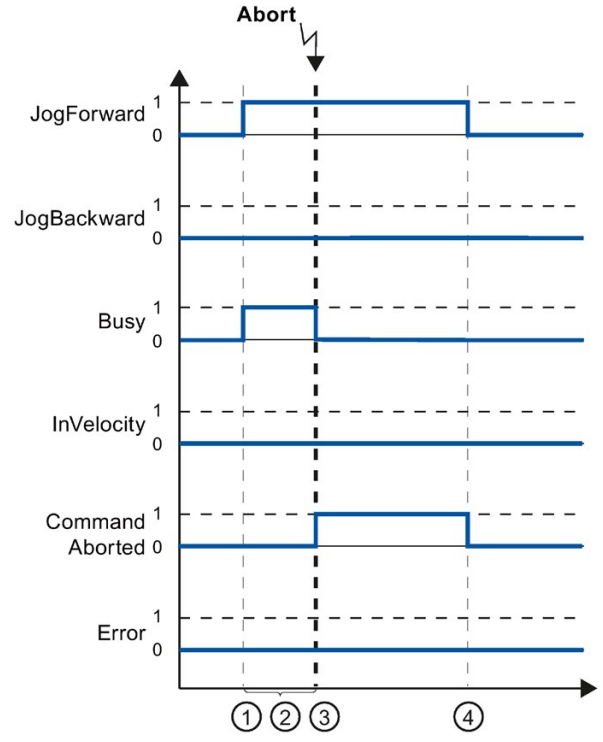
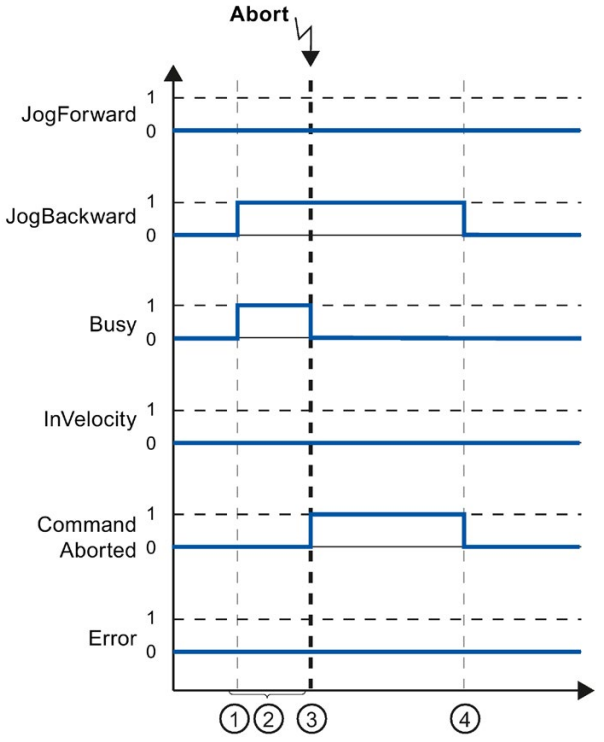
If the motion control command has been executed by the time the parameterized velocity is reached, this is indicated by the value TRUE in output parameter "InVelocity".

Jog mode is controlled by input parameter "JogForward"	Jog mode is controlled by input parameter "JogBackward".
	

①	The command is started with a positive edge at the input parameter "JogForward" or "JogBackward".
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	When the parameterized velocity is reached, the output parameter "InVelocity" changes to TRUE.
④	When the input parameter "JogForward" or "JogBackward" is reset to the value FALSE, the axis motion ends. The axis starts to decelerate. As a result, the axis no longer moves at constant velocity and the output parameter "InVelocity" changes its status to FALSE.
⑤	If the axis has come to a standstill, the motion control command is complete and the output parameter "Busy" changes its value to FALSE.

The command is aborted during execution

If the motion control command is aborted during execution, this is indicated by the value TRUE in output parameter "CommandAborted". The behavior is independent of whether or not the parameterized velocity has been reached.

Jog mode is controlled by input parameter "JogForward".	Jog mode is controlled by input parameter "JogBackward".
	

①	The command is started with a positive edge at the input parameter "JogForward" or "JogBackward".
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	During command execution, the command is aborted by another motion control command. If the command is aborted, output parameter "Busy" changes to FALSE and "CommandAborted" to TRUE.
④	When the input parameter "JogForward" or "JogBackward" is reset to the value FALSE, the output parameter "CommandAborted" changes its value to FALSE.

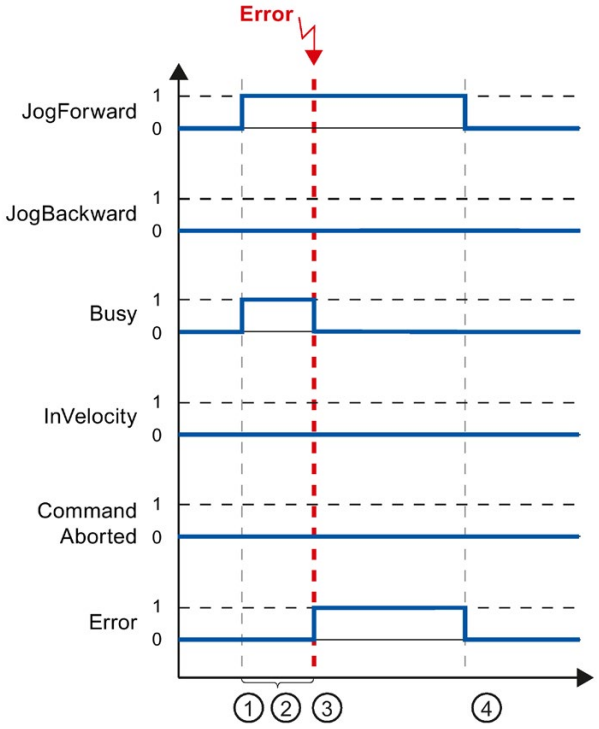
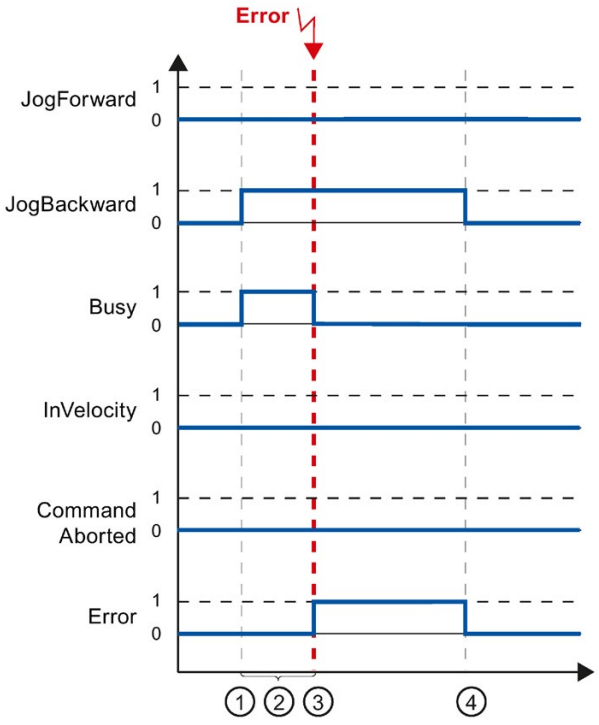
Note

The command abort is indicated in the output parameter "CommandAborted" for only one execution cycle, if all conditions below are met:

The input parameters "JogForward" and "JogBackward" have the value FALSE (but the axis is still decelerating) and a new motion control command is initiated.

An error has occurred during command execution

If an error occurs during execution of the motion control command, this is indicated by the value TRUE in output parameter "Error". The behavior is independent of whether or not the parameterized velocity has been reached.

Jog mode is controlled by input parameter "JogForward".	Jog mode is controlled by input parameter "JogBackward".
	

①	The command is started with a positive edge at the input parameter "JogForward" or "JogBackward".
②	While the command is active, the output parameter "Busy" indicates the value TRUE.
③	An error occurred during command execution. When the error occurs, the output parameter "Busy" changes to FALSE and "Error" to TRUE.
④	When the input parameter "JogForward" or "JogBackward" is reset to the value FALSE, the output parameter "Error" changes its value to FALSE.

Note

An error occurrence is indicated in the output parameter "Error" for only one execution cycle, if all the conditions below are met:

The input parameters "JogForward" and "JogBackward" have the value FALSE (but the axis is still decelerating) and a new error occurs (software limit switch is approached, for example).

9.6 Error displays of the Motion Control statements

The Motion Control instructions indicate any errors in motion control commands and the technology object at the output parameters "Error", "ErrorID" and "ErroInfo" of the Motion Control instructions.

Error display at output parameters "Error", "ErrorID" and "ErrorInfo"

If the output parameter "Error" indicates the value TRUE, the complete command, or portions thereof, could not be executed. The cause of the error is indicated by the value in output parameter "ErrorID". Detailed information about the cause of the error is returned by the value in output parameter ErrorInfo. We distinguish between the following error classes for error indication:

- **Operating error with axis stop (for example, "HW limit switch was approached")**

  Operating errors with axis stop are errors that occur during runtime of the user program. If the axis is in motion, it is stopped with the configured deceleration or emergency stop deceleration, depending on the error. The errors are indicated in the error-triggering Motion Control instruction and in the Motion Control instruction "MC_Power".

- **Operating error without axis stop (for example, "Axis is not homed")**

  Operating errors without axis stop are errors that occur during runtime of the user program. If the axis is in motion, the motion is continued. The errors are only indicated in the Motion Control instruction which triggers the error.

- **Configuration error in Motion Control instruction (for example "Incorrect value in parameter "Velocity")**

  Parameterization errors occur when incorrect information is specified in the input parameters of Motion Control instructions. If the axis is in motion, the motion is continued. The errors are only indicated in the Motion Control instruction which triggers the error.

- **Configuration error on technology object "Axis" (for example, "Value for "Acceleration" is invalid")**

  A configuration error exists if one or more parameters are incorrectly configured in the axis configuration or if editable configuration data have been modified incorrectly during runtime of the program. An axis in motion is stopped with the configured emergency stop deceleration. The error is indicated in the error-triggering Motion Control instruction and in Motion Control instruction "MC_Power".

- **Configuration error on technology object "Command table" (for example "Value for "Velocity" is invalid")**

  There is a configuration error if one or more parameters are incorrectly set in the axis command table or if programmable configuration data have been modified incorrectly during runtime of the program. If the axis is in motion, the motion is continued. The errors are only indicated in the "MC_CommandTable" Motion Control instruction.

- **Internal error**

  When an internal error occurs, the axis is stopped. The errors are indicated in the error-triggering Motion Control instruction and, in some cases, in the Motion Control instruction "MC_Power".

A detailed description of the ErrorIDs and ErrorInfos, as well as their remedies, is available in the Appendix.

See also

Overview of the Motion Control statements (Page 154)

Creating a user program (Page 155)

Programming notes (Page 158)

Behavior of the Motion Control commands after POWER OFF and restart (Page 160)

Monitoring active commands (Page 161)

9.7 Restart of technology objects

Description

After the CPU is switched on, or after technology objects are downloaded into the CPU, the system automatically initializes the technology objects with the start values from the technology data block. If restart-relevant changes are detected during a reload into the CPU, a restart of the technology object is automatically performed.

If restart-relevant data have been changed in RUN mode by the user program, then the technology object must be reinitialized by the user in order for the changes to be used.

If changes in the technology data block should also be retained after the restart of the technology object, then you must write the changes to the start value in load memory using the extended instruction "WRIT_DBL".

Restart necessary

If a restart of the technology object is necessary, this will be indicated under "Technology object > Diagnostics > Status and error bits > Status messages > Restart required", and in the tags of the technology object <Axis name>.StatusBits.RestartRequired.

Restarting a technology object

A restart of the technology object is triggered by the user by means of the "MC_Reset" Motion Control instruction, with parameter "Restart" = TRUE.

A restart resets the "Homed" status of a technology object with incremental actual values (<Axis name>.StatusBits.HomingDone).

9.8 Parameter transfer for function blocks

If you want to reuse a function block with Motion Control instructions for different technology objects, create an input parameter of the data type of the respective technology object in the block interface of the function block. You assign the data type in the block interface with direct input. The parameter is then transferred as a reference to the technology object to the "axis" parameter of the Motion Control instructions. The data types of technology objects correspond to the structure of the associated technology data block.

By specifying the data type, you can address the tags of the technology object in the function block (<parameters of the block interface>.<tag of the technology object>).

If you do not need access to the tags of the technology object, you can use the data type "DB_ANY". The data type "DB_Any" can be used to achieve more variable programming.

The following table shows the data types for reference to the technology objects:

Technology object	Data type for reference to the technology object
Positioning axis	TO_PositioningAxis
Command table	TO_CommandTable

Example 1

The following table shows the definition of the tags used:

Operand	Declaration	Data type	Description
axis	Input	TO_PositioningAxis	Reference to the technology object
on	Input	BOOL	Signal to enable the axis
actPosition	Output	Real	Query of the actual position from the technology data block
instMC_POWER	Static	MC_POWER	Multi-instance of the Motion Control instruction MC_Power

The following SCL program shows how to implement this task:

```
SCL                                                  Explanation
#instMC_POWER(Axis := #axis, Enable := #on);          //Call of the Motion Control instruction MC_Power
                                                     with enable of the axis
#actPosition := #axis.ActualPosition;                //Query of the actual position from the technology
                                                     data block
```

Example 2

The data type "DB_Any" provides another option for transferring the data types of a technology object. The data type "DB_Any" can be assigned in the program during runtime.

The example shows two options for transferring technology-specific data types to a corresponding instruction, e.g. "MC_CommandTable", which has been created as a multi-instance. The first option shows the use of the data type "TO_PositioningAxis". The second option shows the simply transfer of the command table technology object as a function of the "cmdTablToUse" input. Depending on the value at the input, one of the three "cmdTablx" inputs is transferred to the "MC_CommandTable" instruction via "tempCmdTableSel".

The following table shows the declaration of the tags used:

Tag	Declaration	Data type	Description
axis	Input	TO_PositioningAxis	Positioning axis
cmdTabl1	Input	DB_ANY	1st command table
cmdTabl2	Input	DB_ANY	2nd command table
cmdTabl3	Input	DB_ANY	3rd command table
cmdTablToUse	Input	Int	Selection, command tables 1 to 3
instMC_CommandTable	Static	MC_CommandTable	Multi-instance of the MC_CommandTable
tempCmdTableSel	Temp	DB_ANY	Current command table

The example below shows the basic procedure:

```
SCL                                          Description
CASE #cmdTablToUse OF
    1:  #tempCmdTableSel := #cmdTabl1;       //Program for scenario 1
    2:  #tempCmdTableSel := #cmdTabl2;       //Program for scenario 2
    3:  #tempCmdTableSel := #cmdTabl3;       //Program for scenario 3
    ELSE                                     //Program for all other values
        #tempCmdTableSel := #cmdTabl1;       //->Default setting 1st command table
END_CASE;
                                             // Call of the "MC_CommandTable" instruction
                                             //with variable transfer of the technology objects
#instMC_CommandTable(Axis:=#axis,            //Assignment of axis
    CommandTable:=#temCmdTableSel);          //Indirect assignment of the command table
```

Additional information

You can find more program examples using the data type "DB_Any" in the following FAQ:

https://support.industry.siemens.com/cs/ww/en/view/109750880
(https://support.industry.siemens.com/cs/ww/en/view/109750880)

Axis - Diagnostics

10

10.1 Status and error bits (technology objects as of V4)

You use the "Status and error bits" diagnostic function to monitor the most important status and error messages for the axis in the TIA Portal. The diagnostic function display is available in online mode in "Manual control" mode and in "Automatic control" when the axis is active. The displayed status and error messages have the following meaning:

Status messages

Status message - Axis	Description
Enabled	The axis is enabled and ready to be controlled via Motion Control commands. (Tag of the technology object: <axis name>.StatusBits.Enable)
Homed	The axis is homed and is capable of executing absolute positioning commands of Motion Control instruction "MC_MoveAbsolute". The axis does not have to be homed for relative positioning. Special situations: • During active homing, the status is FALSE. • If a homed axis undergoes passive homing, the status is set to TRUE during passive homing. (Tag of the technology object: <axis name>.StatusBits.HomingDone)
Axis error	An error has occurred in the "Axis" technology object. Additional information about the error is available in automatic control at the ErrorID and ErrorInfo parameters of the Motion Control instructions. In manual mode, the "Error message" box of the axis control panel displays detailed information about the cause of error. (Tag of the technology object: <axis name>.StatusBits.Error)
Control panel active	The "Manual control" mode was enabled in the axis control panel. The axis control panel has control priority over the "Axis" technology object. The axis cannot be controlled from the user program. (Tag of the technology object: <axis name>.StatusBits.ControlPanelActive)
Restart required	A modified configuration of the axis was downloaded to the load memory in CPU RUN mode. To download the modified configuration to the work memory, you need to restart the axis. Use the Motion Control instruction MC_Reset to do this. (Tag of the technology object: <axis name>.StatusBits.RestartRequired)

Status message - Drive	Description
Ready	The drive is ready for operation. (Tag of the technology object: <axis name>.StatusBits.DriveReady)
Drive error	The drive has reported an error due to loss of its "Drive ready" signal. (Tag of the technology object: <axis name>.ErrorBits.DriveFault)

Status message - Motion	Description
Standstill	The axis is at a standstill. (Tag of the technology object: <axis name>.StatusBits.StandStill)
Acceleration	The axis accelerates. (Tag of the technology object: <axis name>.StatusBits.Accelerating)
Constant velocity	The axis travels at constant velocity. (Tag of the technology object: <axis name>.StatusBits.ConstantVelocity)
Deceleration	The axis decelerates (slows down). (Tag of the technology object: <axis name>.StatusBits.Decelerating)

Status message - Motion type	Description
Positioning	The axis executes a positioning command of the Motion Control instruction "MC_MoveAbsolute", "MC_MoveRelative" or the axis control panel. (Tag of the technology object: <axis name>.StatusBits.PositioningCommand)
Travel with velocity specification	The axis executes a command with velocity specification of the Motion Control instruction "MC_MoveVelocity", "MC_MoveJog" or the axis control panel. (Tag of the technology object: <axis name>.StatusBits.VelocityCommand)
Homing	The axis executes a homing command of the Motion Control instruction "MC_Home" or the axis control panel. (Tag of the technology object: <axis name>.StatusBits.HomingCommand)
Command table active	The axis is controlled by Motion Control instruction "MC_CommandTable". (Tag of the technology object: <axis name>.StatusBits.CommandTableActive)

Limit switch status messages

Limit switch status message	Description
SW low limit switch has been reached	A software limit switch was reached or exceeded. (Tag of the technology object: <axis name>.StatusBits.SWLimitMinActive)
SW high limit switch has been reached	A hardware limit switch was reached or exceeded. (Tag of the technology object: <axis name>.StatusBits.SWLimitMaxActive)
HW low limit switch was reached	The hardware low limit switch was reached or exceeded. (Tag of the technology object: <axis name>.StatusBits.HWLimitMinActive)
HW high limit switch was reached	The hardware high limit switch was reached or exceeded. (Tag of the technology object: <axis name>.StatusBits.HWLimitMaxActive)

Error messages

Error message	Description
SW limit switch has been reached	A software limit switch was reached or exceeded. (Tag of the technology object: <axis name>.ErrorBits.SWLimit)
HW limit switch has been reached	A hardware limit switch was reached or exceeded. (Tag of the technology object: <axis name>.ErrorBits.HWLimit)
Invalid direction of motion	The motion direction of the command does not match the configured motion direction. (Tag of the technology object: <axis name>.ErrorBits.DirectionFault)
PTO already in use	A second axis is using the same PTO (Pulse Train Output) and HSC (High Speed Counter) and is enabled with "MC_Power". (Tag of the technology object: <axis name>.ErrorBits.HWUsed)
Encoder	Error in the encoder system. (Tag of the technology object: <axis name>.ErrorBits.SensorFault)
Data exchange	Error in communication with a connected device. (Tag of the technology object: <axis name>.ErrorBits.CommunicationFault)
Positioning	The axis was not correctly positioned at the end of a positioning motion. (Tag of the technology object: <axis name>.ErrorBits.PositionigFault)
Following error	The maximum permitted following error was exceeded. (Tag of the technology object: <axis name>.ErrorBits.FollowingErrorFault)
Encoder values are invalid	The encoder values are invalid. (Tag of the technology object: <axis name>.StatusSensor.State)
Configuration error	The "Axis" technology object was incorrectly configured or editable configuration data were modified incorrectly during runtime of the user program. (Tag of the technology object: <axis name>.ErrorBits.ConfigFault)
Internal error	An internal error has occurred. (Tag of the technology object: <axis name>.ErrorBits.SystemFault)

The output window below shows the first reported and still unacknowledged error.

See also

Compatibility list of variables V1...3 <-> V4...5 (Page 48)

Motion status (Page 179)

10.2 Motion status

Use the "Motion status" diagnostic function to monitor the motion status of the axis in the TIA Portal. The diagnostic function display is available in online mode in "Manual control" mode and in "Automatic control" when the axis is active. The displayed status information has the following meaning:

Status	Description
Actual position	The "Actual position" box indicates the measured position of the axis. If the axis is not homed, the value indicates the position value relative to the enable position of the axis. (Tag of the technology object: <axis name>.ActualPosition)
Actual velocity	The "Actual velocity" box indicates the measured position of the axis. (Tag of the technology object: <axis name>.ActualVelocity)
Position setpoint	The "Position setpoint" box indicates the measured position setpoint of the axis. If the axis is not homed, the value indicates the position value relative to the enable position of the axis. (Tag of the technology object: <axis name>.Position)
Velocity setpoint	The "Velocity setpoint:" box indicates the calculated velocity setpoint of the axis. (Tag of the technology object: <axis name>.Velocity)
Target position	The "Target position" box indicates the current target position of an active positioning command or of the axis command table. The value of the "Target position" is only valid during execution of a positioning command. (Tag of the technology object: <axis name>.StatusPositioning.TargetPosition)
Remaining travel distance	The "Remaining travel distance" box indicates the travel distance currently remaining for an active positioning command or the axis command table. The "Remaining travel distance" value is only valid during execution of a positioning command. (Tag of the technology object: <axis name>.StatusPositioning.Distance)

See also

Compatibility list of variables V1...3 <-> V4...5 (Page 48)

Tags for position values and velocity values as of V6 (Page 260)

Status and error bits (technology objects as of V4) (Page 176)

10.3 Dynamics settings

Use the "Dynamics settings" diagnostic function to monitor the dynamic limits of the axis in the TIA Portal. The diagnostic function display is available in online mode in "Manual control" mode and in "Automatic control" when the axis is active. The displayed status information has the following meaning:

Dynamic limit	Description
Acceleration	The "Acceleration" box indicates the currently configured acceleration of the axis. (Tag of technology object: <Axis name>.DynamicDefaults.Acceleration)
Deceleration	The "Deceleration" box indicates the currently configured deceleration of the axis. (Tag of technology object: <Axis name>.DynamicDefaults.Deceleration)
Emergency deceleration	The "Emergency deceleration" box indicates the currently configured emergency stop deceleration of the axis. (Tag of technology object: <Axis name>.DynamicDefaults.EmergencyDeceleration)
Jerk (axis technology object as of V2)	The "Velocity" box indicates the current axis step velocity configured. (Tag of technology object: <Axis name>.DynamicDefaults.Jerk)

See also

Compatibility list of variables V1...3 <-> V4...5 (Page 48)

10.4 PROFIdrive frame

The "Technology object > Diagnostics > PROFIdrive telegram" diagnostics function is used in the TIA Portal to monitor the PROFIdrive telegrams returned by the drive and encoder. The display of the Diagnostics function is available in online operation.

"Drive" area

This area displays the following parameters contained in the PROFIdrive telegram from the drive to the controller:

- Status words "SW1" and "SW2"
- The setpoint speed that was output to the drive (NSET)
- The actual speed that was signaled from the drive (NACT)

"Encoder" area

This area displays the following parameters contained in the PROFIdrive telegram from the encoder to the controller:

- Status word "G1_ZSW"
- The actual position value "G1_XIST1" (cyclic actual encoder value)
- The actual position value "G1_XIST2" (absolute value of the encoder)

Instructions 11

11.1 S7-1200 Motion Control as of V6

11.1.1 MC_Power

11.1.1.1 MC_Power: Enable, disable axis as of V6

Description

The Motion Control instruction "MC_Power" enables or disables an axis.

Requirements

- The positioning axis technology object has been configured correctly.
- There is no pending enable-inhibiting error.

Override response

Execution of "MC_Power" cannot be aborted by a Motion Control command.

Disabling the axis (input parameter "Enable" = FALSE) aborts all Motion Control commands for the associated technology object in accordance with the selected "StopMode".

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_Axis	-	Axis technology object	
Enable	INPUT	BOOL	FALSE	TRUE	The axis is enabled.
				FALSE	All current jobs are interrupted in accordance with the "StopMode" configured. The axis is stopped and disabled.
StartMode	INPUT	INT	1	0	Enable positioning axis not position-controlled *)
				1	Enable positioning axis position-controlled *)
				*) This parameter is ignored when a positioning axis with PTO (Pulse Train Output) drive is used. The parameter initially takes effect when the positioning axis is enabled (Enable changes from FALSE to TRUE) and when the axis is enabled after successful acknowledgment of an interrupt that caused the axis to be disabled.	
StopMode	INPUT	INT	0	0	Emergency stop If a request to disable the axis is pending, the axis brakes at the configured emergency deceleration. The axis is disabled after reaching standstill.
				1	Immediate stop If a request to disable the axis is pending, this setpoint zero is output and the axis is disabled. The axis is braked depending on the configuration in the drive, and is brought to a standstill. With drive connection via PTO (Pulse Train Output): When you disable the axis, the pulse output is stopped with a frequency-dependent deceleration: • Output frequency ≥ 100 Hz Deceleration: max. 30 ms • Output frequency < 100 Hz Deceleration: 30 ms up to max. 1.5 s at 2 Hz
				2	Emergency stop with jerk control If a request to disable the axis is pending, the axis brakes at the configured emergency deceleration. If the jerk control is activated, the configured jerk is taken into account. The axis is disabled after reaching standstill.

Parameter	Declaration	Data type	Default value	Description	
Status	OUTPUT	BOOL	FALSE	Status of axis enable	
				FALSE	The axis is disabled. The axis does not execute Motion Control commands and does not accept any new commands (exception: MC_Reset command) For drive connection via PTO (Pulse Train Output): The axis is not homed. Upon disabling, the status does not change to FALSE until the axis reaches a standstill.
				TRUE	The axis is enabled. The axis is ready to execute Motion Control commands. Upon axis enabling, the status does not change to TRUE until the signal "Drive ready" is pending. If the "Drive ready" drive interface was not configured in the axis configuration, the status changes to TRUE immediately.
Busy	OUTPUT	BOOL	FALSE	TRUE	MC_Power" is active.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred in Motion Control instruction "MC_Power" or in the associated technology object. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

Note

If the axis is switched off due to an error, it will be enabled again automatically after the error has been eliminated and acknowledged. This requires that the input parameter "Enable" has retained the value TRUE during this process.

Enabling an axis with configured drive interface

To enable the axis, follow these steps:

1. Check the requirements indicated above.
2. Initialize input parameters "StartMode" and "StopMode" with the desired value. Set the input parameter "Enable" to TRUE.

   The enable output for "Drive enabled" changes to TRUE to enable the power to the drive. The CPU waits for the "Drive ready" signal of the drive.

   When the "Drive ready" signal is available at the configured ready input of the CPU, the axis is enabled. The output parameter "Status" and the variable of the technology object <axis name>.StatusBits.Enable indicate the value TRUE.

Enabling an axis without configured drive interface

To enable the axis, follow these steps:

1. Check the requirements indicated above.
2. Initialize input parameters "StartMode" and "StopMode" with the desired value. Set the input parameter "Enable" to TRUE. The axis is enabled. The output parameter "Status" and the variable of the technology object <axis name>.StatusBits.Enable indicate the value TRUE.

Disabling an axis

To disable an axis, you can follow the steps described below:

1. Bring the axis to a standstill.

   You can identify when the axis is at a standstill in the variable of the technology object <axis name>.StatusBits.StandStill.
2. Set input parameter "Enable" to FALSE after standstill is reached.
3. If output parameters "Busy" and "Status" and variable of technology object <axis name>.StatusBits.Enable indicate the value FALSE, disabling of the axis is complete.

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

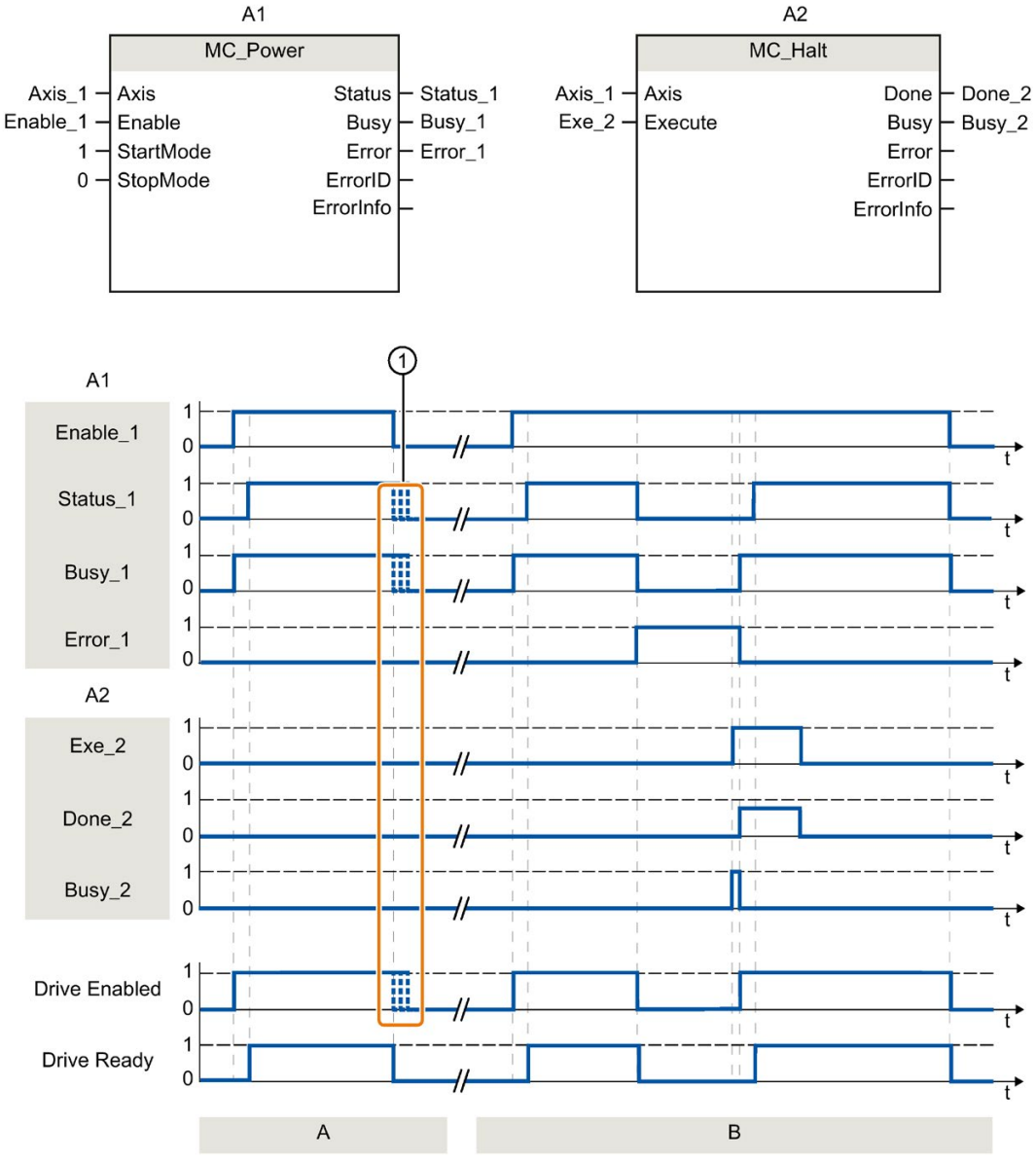
MC_Halt: Stop axis as of V6 (Page 193)

Configuration - Standstill signal (PROFIdrive and analog drive connection only) (Page 97)

Move the axis without position control for servicing (Page 235)

11.1.1.2 MC_Power: Function chart as of V6

Function chart



Section A	An axis is enabled and then disabled again. When the drive has signaled "Drive ready" back to the CPU, the successful enable can be read out via "Status_1".
Section B	Following an axis enable, an error has occurred that caused the axis to be disabled. The error is eliminated and acknowledged with "MC_Reset". The axis is then enabled again.
①	The exact end of the signals depends on the selected drive and the StopMode.

11.1.2 MC_Reset

11.1.2.1 MC_Reset: Acknowledge fault, restart technology object as of V6

Description

Motion Control instruction "MC_Reset" can be used to acknowledge "Operating error with axis stop" and "Configuration error". The errors that require acknowledgment can be found in the "List of ErrorIDs and ErrorInfos" under "Remedy".

The axis configuration can be downloaded to the work memory after a download in RUN mode.

Requirements

- The positioning axis technology object has been configured correctly.
- The cause of a pending configuration error requiring acknowledgment has been eliminated (for example, acceleration in positioning axis technology object has been changed to a valid value).

Override response

The MC_Reset command cannot be aborted by any other Motion Control command.

The new MC_Reset command does not abort any other active Motion Control commands.

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_Axis	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Restart	INPUT	BOOL	FALSE	TRUE	Download the axis configuration from the load memory to the work memory. The command can only be executed when the axis is disabled. Refer to the notes on Download to the CPU (Page 144).
				FALSE	Acknowledges pending errors
Done	OUTPUT	BOOL	FALSE	TRUE	Error has been acknowledged.
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

Acknowledging an error requiring acknowledgment with MC_Reset

To acknowledge an error, follow these steps:

1. Check the requirements indicated above.
2. Start the acknowledgment of the error with a rising edge at input parameter "Execute".
3. If output parameter "Done" indicates the value TRUE and tag of technology object <Axis name>.StatusBits.Error the value FALSE, the error has been acknowledged.

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

Download to CPU (Page 144)

11.1.3 MC_Home

11.1.3.1 MC_Home: Home axes, set reference point as of V6

Description

Motion Control instruction "MC_Home" is used to match the axis coordinates to the real, physical drive position. Homing is required for absolute positioning of the axis. The following types of homing can be executed:

- Active homing (Mode = 3)

  The homing procedure is executed automatically.

- Passive homing (Mode = 2)

  During passive homing, the "MC_Home" Motion Control instruction does not carry out any homing motion. The traversing motion required for this must be implemented by the user via other Motion Control instructions. When the homing switch is detected, the axis is homed.

- Direct homing absolute (Mode = 0)

  The current axis position is set to the value of parameter "Position".

- Direct homing relative (Mode = 1)

  The current axis position is offset by the value of parameter "Position".

- Absolute encoder adjustment relative (Mode = 6)

  The current axis position is offset by the value of parameter "Position".

- Absolute encoder adjustment absolute (Mode = 7)

  The current axis position is set to the value of parameter "Position".

Mode 6 and 7 can only be used with drives with an analog interface and PROFIdrive drive.

Requirements

- The positioning axis technology object has been configured correctly.
- The axis is enabled. (Not valid for PROFIdrive drive/analog drive connection Mode = 0 or 1)
- No MC_CommandTable command may be active upon start with Mode = 0, 1, or 2.

Override response

The override response depends on the selected mode:

Mode = 0, 1, 6, 7

The MC_Home command cannot be aborted by any other Motion Control command.

The MC_Home command does not abort any active Motion Control commands. Position-related motion commands are resumed after homing according to the new homing position (value at input parameter: "Position").

Mode = 2

The MC_Home command can be aborted by the following Motion Control commands:

- MC_Home command Mode = 2, 3

The new MC_Home command aborts the following active Motion Control command:

- MC_Home command Mode = 2

Position-related motion commands are resumed after homing according to the new homing position (value at input parameter: "Position").

Mode = 3

The MC_Home command can be aborted by the following Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The new MC_Home command aborts the following active Motion Control commands:

- MC_Home command Mode = 2, 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_Axis	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Position	INPUT	REAL	0.0	• Mode = 0, 2, and 3 Absolute position of axis after completion of the homing operation • Mode = 1 Correction value for the current axis position Limit values: -1.0E12 ≤ Position ≤ 1.0E12	
Mode	INPUT	INT	0	Homing mode	
				0	Direct homing (absolute) New axis position is the position value of parameter "Position".
				1	Direct homing (relative) New axis position is the current axis position + position value of parameter "Position".
				2	Passive homing Homing according to the axis configuration. Following homing, the value of parameter "Position" is set as the new axis position.
				3	Active homing Homing procedure in accordance with the axis configuration. Following homing, the value of parameter "Position" is set as the new axis position.
				6	Absolute encoder adjustment (relative) The current axis position is offset by the value of parameter "Position". The calculated absolute value offset is stored retentively in the CPU. (<Axis-Name>.StatusSensor.AbsEncoderOffset)
				7	Absolute encoder adjustment (absolute) The current axis position is set to the value of parameter "Position". The calculated absolute value offset is stored retentively in the CPU. (<Axis-Name>.StatusSensor.AbsEncoderOffset)
Done	OUTPUT	BOOL	FALSE	TRUE	Command completed
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	During execution, the command was aborted by another command.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".

Parameter	Declaration	Data type	Default value	Description
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"
ReferenceMark-Position	OUTPUT	REAL	0.0	Display of the position at which the technology object was homed (valid when "Done" = TRUE)

Resetting the "Homed" status

The "Homed" status of a technology object (<Axis name>.StatusBits.HomingDone) is reset under the following conditions:

- **Drive connection via PTO (Pulse Train Output):**
  - Start an "MC_Home" command for active homing

    (After successful completion of the homing operation, the "Homed" status is set again.)
  - Disabling of axis by the "MC_Power" Motion Control instruction
  - Changeover between automatic mode and manual control
  - After POWER OFF → POWER ON of the CPU
  - After restart of CPU (RUN-STOP → STOP-RUN)
- **Technology objects with incremental actual values:**
  - Start an "MC_Home" command for active homing

    (After successful completion of the homing operation, the "Homed" status is set again.)
  - Error in the encoder system, or encoder failure
  - Restart of the technology object
  - After POWER OFF → POWER ON of the CPU
  - Memory reset
  - Modification of the encoder configuration
- **Technology objects with absolute actual values:**
  - Errors in the sensor system/encoder failure
  - Replacement of the CPU
  - Modification of the encoder configuration
  - Restoration of the CPU factory settings
  - Transfer of a different project to the controller

Homing an axis

To home the axis, follow these stops:

1. Check the requirements indicated above.
2. Provide the necessary input parameters with values and start the homing operation with a rising edge at input parameter "Execute".
3. If output parameter "Done" and technology object variable <axis name>.StatusBits.HomingDone indicate the value TRUE, homing is complete. The reference position can be taken from the <axis name>.ReferenceMarkPosition variable.

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

11.1.4 MC_Halt

11.1.4.1 MC_Halt: Stop axis as of V6

Description

The "MC_Halt" Motion Control instruction stops all movements and brings the axis to a standstill with the configured deceleration. The standstill position is not defined.

Requirements

- The positioning axis technology object has been configured correctly.
- The axis is enabled.

Override response

The MC_Halt command can be aborted by the following Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The new MC_Halt command aborts the following active Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_SpeedAxis	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Done	OUTPUT	BOOL	FALSE	TRUE	Zero velocity reached
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed.
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	During execution the command was aborted by another command.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

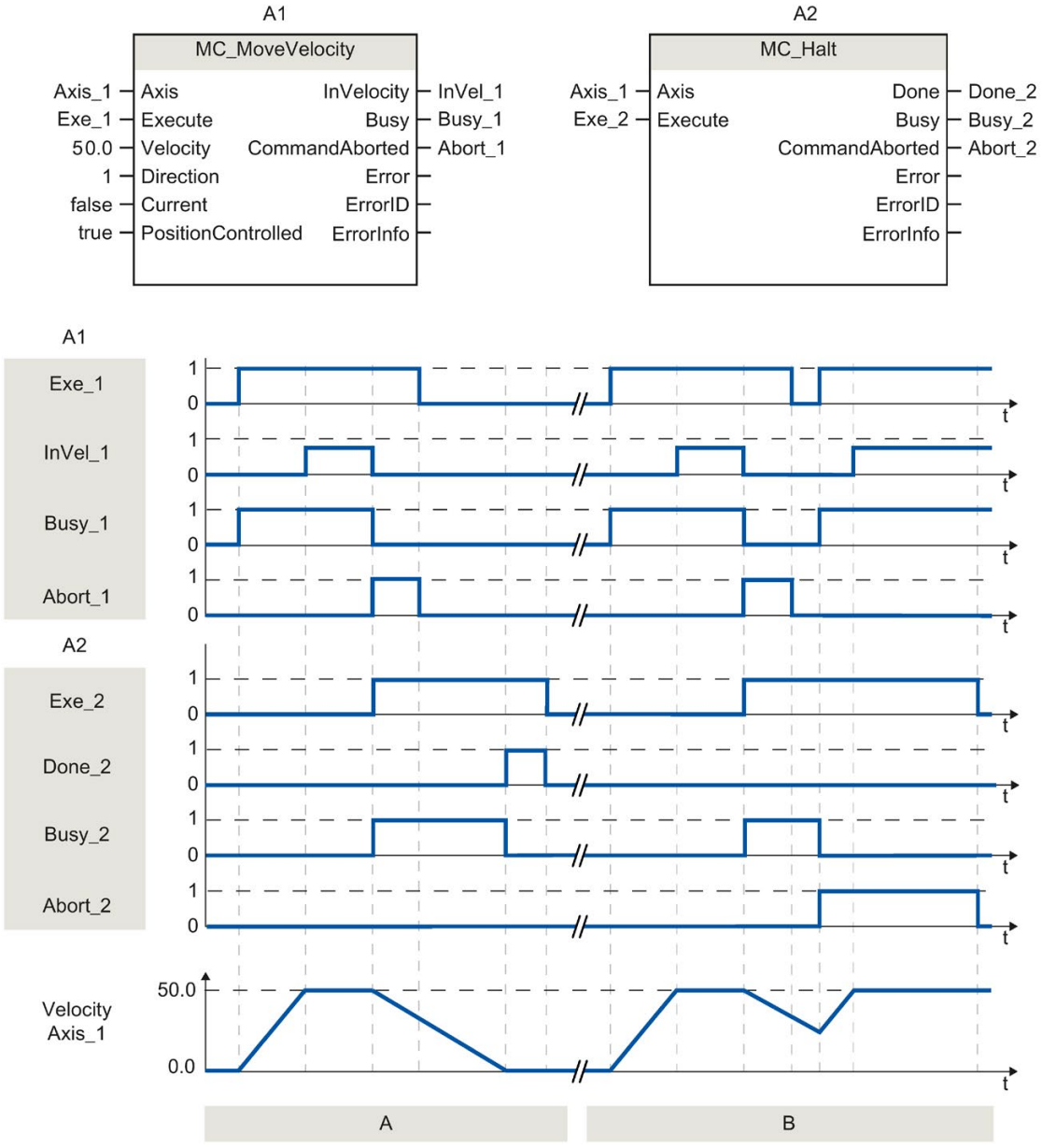
MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

Move the axis without position control for servicing (Page 235)

11.1.4.2 MC_Halt: Function chart as of V6

Function chart



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 5.0

Section A	The axis is braked by an MC_Halt command until it comes to a standstill. The axis standstill is signaled via "Done_2".
Section B	While an MC_Halt command is braking the axis, this command is aborted by another motion command. The abort is signaled via "Abort_2".

11.1.5 MC_MoveAbsolute

11.1.5.1 MC_MoveAbsolute: Absolute positioning of axis as of V6

Description

The "MC_MoveAbsolute" Motion Control instruction starts an axis positioning motion to move it to an absolute position.

Requirements

- The positioning axis technology object has been configured correctly.
- The axis is enabled.
- The axis is homed.

Override response

The MC_MoveAbsolute command can be aborted by the following Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The new MC_MoveAbsolute command aborts the following active Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_Positioning Axis	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Position	INPUT	REAL	0.0	Absolute target position Limit values: -1.0E12 ≤ Position ≤ 1.0E12	
Velocity	INPUT	REAL	10.0	Velocity of axis This velocity is not always reached on account of the configured acceleration and deceleration and the target position to be approached. Limit values: Start/stop velocity ≤ Velocity ≤ maximum velocity	
Direction	INPUT	INT	1	Motion direction of the axis Is only evaluated with "modulo" enabled. "Technology object > Configuration > Extended parameters > Modulo > Enable Modulo" Parameter is ignored with PTO axes.	
				0	The sign for the velocity ("Velocity" parameter) determines the motion direction.
				1	Positive direction (Target position is approached in a positive direction)
				2	Negative direction (Target position is approached in a negative direction)
				3	Shortest distance (Starting from the current position, the technology selects the shortest distance to the target position)
Done	OUTPUT	BOOL	FALSE	TRUE	Absolute target position reached
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed

Parameter	Declaration	Data type	Default value	Description	
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	During execution, the command was aborted by another command.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

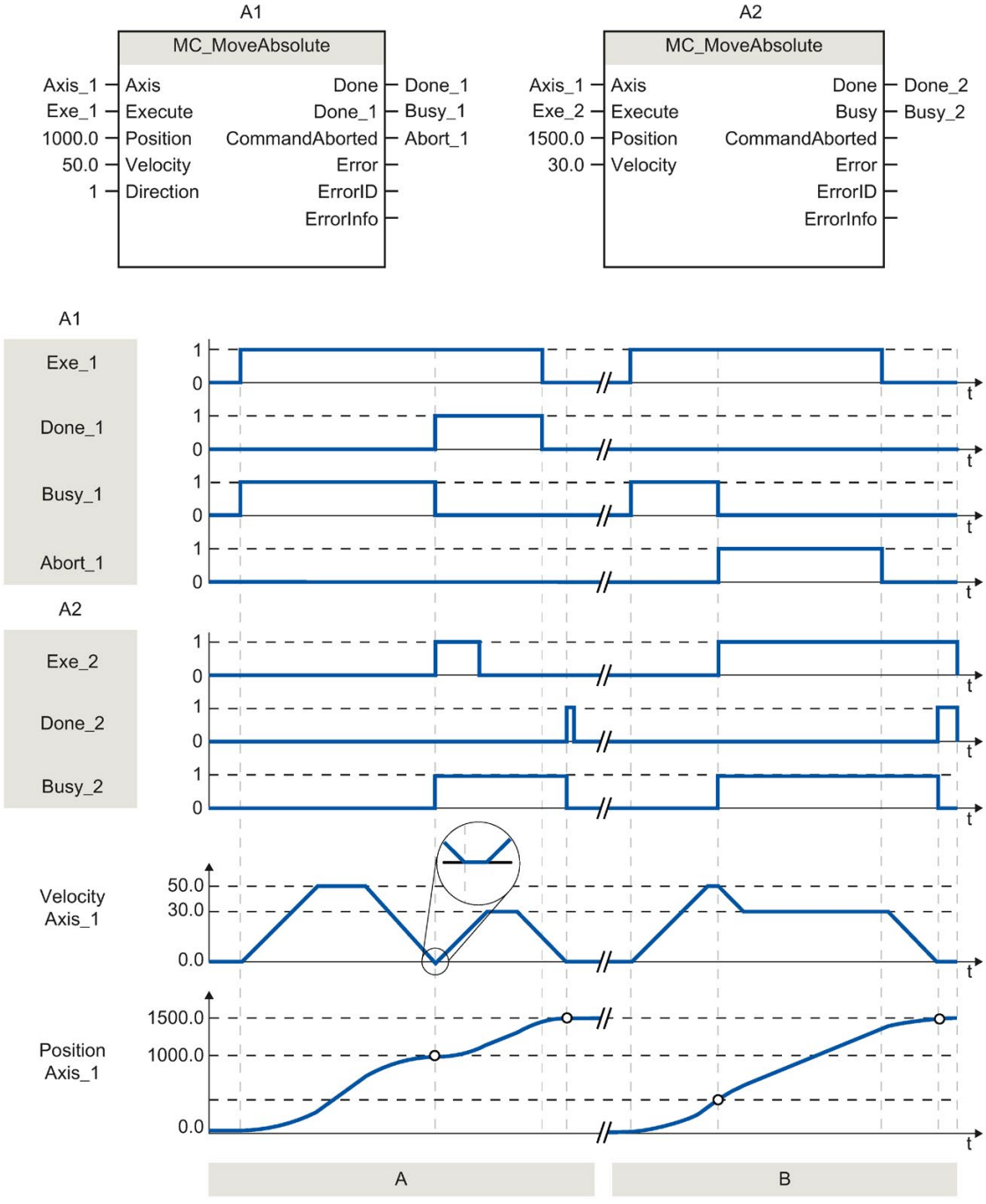
MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

Configuration - Modulo (PROFIdrive/analog drive connection only) (Page 75)

11.1.5.2 MC_MoveAbsolute: Function chart as of V6

Function chart



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 10.0

Section A	An axis is moved to absolute position 1000.0 with an MC_MoveAbsolute command. When the axis reaches the target position, this is signaled via "Done_1". When "Done_1" = TRUE, another MC_MoveAbsolute command, with target position 1500.0, is started. Because of the response times (e.g., cycle time of user program, etc.), the axis comes to a standstill briefly (see zoomed-in detail). When the axis reaches the new target position, this is signaled via "Done_2".
Section B	An active MC_MoveAbsolute command is aborted by another MC_MoveAbsolute command. The abort is signaled via "Abort_1". The axis is then moved at the new velocity to the new target position 1500.0. When the new target position is reached, this is signaled via "Done_2".

11.1.6 MC_MoveRelative

11.1.6.1 MC_MoveRelative: Relative positioning of axis as of V6

Description

The "MC_MoveRelative" Motion Control instruction starts a positioning motion relative to the start position.

Requirements

- The positioning axis technology object has been configured correctly.
- The axis is enabled.

Override response

The MC_MoveRelative command can be aborted by the following Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The new MC_MoveRelative command aborts the following active Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

Parameters

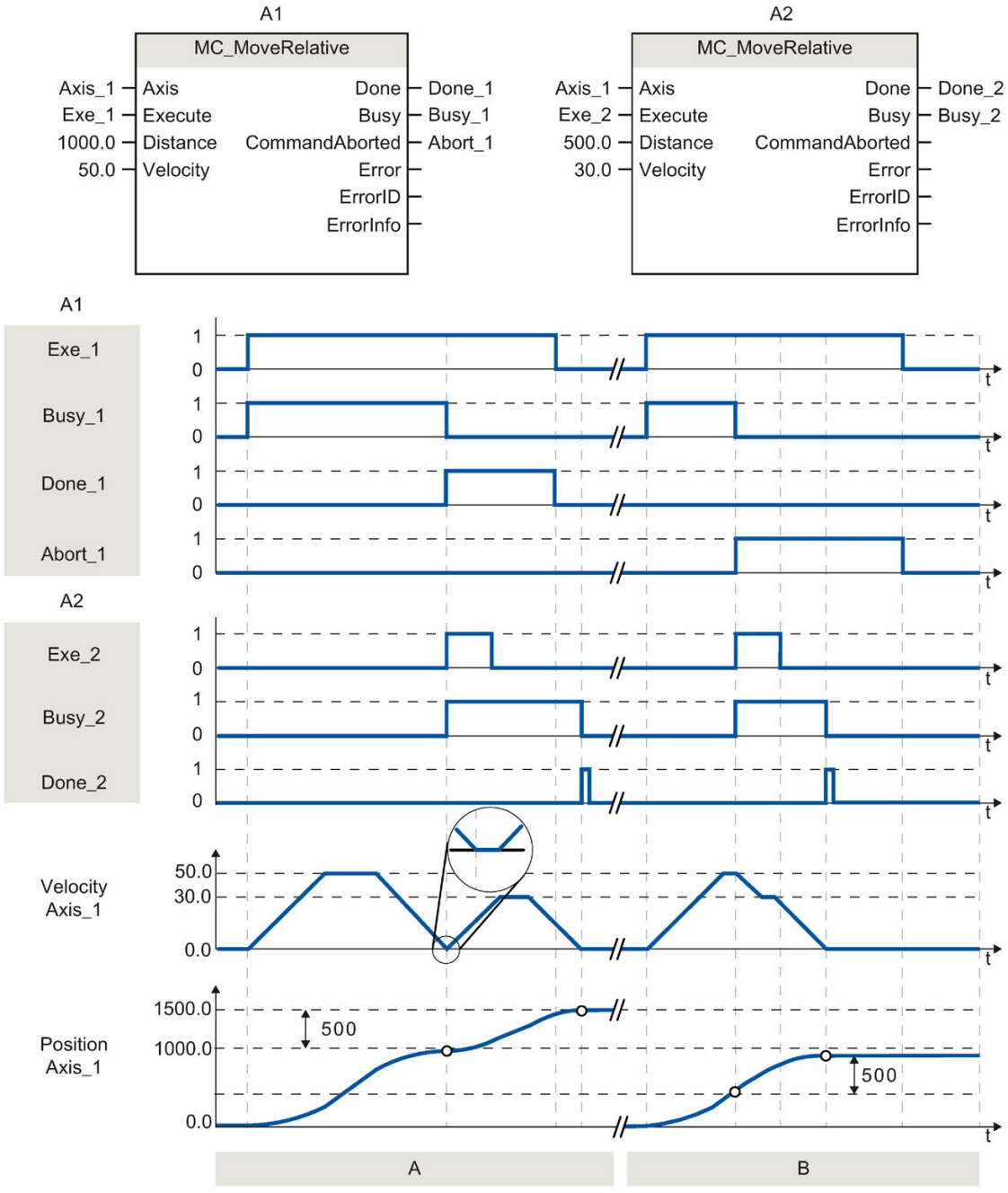
Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_Positioning Axis	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Distance	INPUT	REAL	0.0	Travel distance for the positioning operation Limit values: -1.0E12 ≤ Distance ≤ 1.0E12	
Velocity	INPUT	REAL	10.0	Velocity of axis This velocity is not always reached on account of the configured acceleration and deceleration and the distance to be traveled. Limit values: Start/stop velocity ≤ Velocity ≤ maximum velocity	
Done	OUTPUT	BOOL	FALSE	TRUE	Target position reached
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	During execution, the command was aborted by another command.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)
MC_Power: Enable, disable axis as of V6 (Page 181)
MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)
MC_Home: Home axes, set reference point as of V6 (Page 188)
MC_Halt: Stop axis as of V6 (Page 193)
MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)
MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)
MC_MoveJog: Move axis in jog mode as of V6 (Page 208)
MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)
MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)
MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)
MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

11.1.6.2 MC_MoveRelative: Function chart as of V6

Function chart



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 10.0

Section A	The axis is moved by an MC_MoveRelative command by the distance ("Distance") 1000.0. When the axis reaches the target position, this is signaled via "Done_1". When "Done_1" = TRUE, another MC_MoveRelative command, with travel distance 500.0, is started. Because of the response times (e.g., cycle time of user program, etc.), the axis comes to a standstill briefly (see zoomed-in detail). When the axis reaches the new target position, this is signaled via "Done_2".
Section B	An active MC_MoveRelative command is aborted by another MC_MoveRelative command. The abort is signaled via "Abort_1". The axis is then moved at the new velocity by the new distance ("Distance") 500.0. When the new target position is reached, this is signaled via "Done_2".

11.1.7 MC_MoveVelocity

11.1.7.1 MC_MoveVelocity: Move axis at set velocity as of V6

Description

Motion control instruction "MC_MoveVelocity" moves the axis constantly at the specified velocity.

Requirements

- The positioning axis technology object has been configured correctly.
- The axis is enabled.

Override response

MC_MoveVelocity can be aborted by the following Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The new MC_MoveVelocity command aborts the following active Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_SpeedAxis	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Velocity	INPUT	REAL	10.0	Velocity specification for axis motion Limit values: Start/stop velocity ≤ |Velocity ≤ maximum velocity (Velocity = 0.0 is permitted)	
Direction	INPUT	INT	0	Direction specification	
				0	Direction of rotation corresponds to the sign of the value in parameter "Velocity"
				1	Positive direction of rotation (The sign of the value in parameter "Velocity" is ignored)
				2	Negative direction of rotation (The sign of the value in parameter "Velocity" is ignored)
Current	INPUT	BOOL	FALSE	Maintain current velocity	
				FALSE	"Maintain current velocity" is deactivated. The values of parameters "Velocity" and "Direction" are used.
				TRUE	"Maintain current velocity" is activated. The values in parameters "Velocity" and "Direction" are not taken into account. When the axis resumes motion at the current velocity, the "InVelocity" parameter returns the value TRUE.
PositionControlled	INPUT	BOOL	TRUE	FALSE	Non position-controlled operation
				TRUE	Position-controlled operation
				The parameter applies as long as the "MC_MoveVelocity" command is being executed. After this, the setting of MC_Power applies again. This parameter is ignored when a PTO axis is used.	
InVelocity	OUTPUT	BOOL	FALSE	TRUE	• "Current" = FALSE: The velocity specified in parameter "Velocity" was reached. • "Current" = TRUE: The axis travels at the current velocity at the start time.
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	During execution, the command was aborted by another command.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".

Parameter	Declaration	Data type	Default value	Description
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"

Note

PLCopen Version 2.0

The Motion Control instruction "MC_MoveVelocity" is compatible to PLCopen Version 2.0 as of V4.

The "InVelocity" and "Busy" parameters show their status regardless of the "Execute" parameter until the command is overridden or stopped by an error. For more information, refer to the section " Tracking active commands (Page 161).

Behavior with zero setpoint velocity (Velocity = 0.0)

An MC_MoveVelocity command with "Velocity" = 0.0 (such as an MC_Halt command) aborts active motion commands and stops the axis with the configured deceleration.

When the axis comes to a standstill, output parameter "InVelocity" indicates TRUE for at least one program cycle.

"Busy" indicates the value TRUE during the deceleration process and changes to FALSE together with "InVelocity". If parameter "Execute" = TRUE is set, "InVelocity" and "Busy" are latched.

When the "MC_MoveVelocity" command is started, status bit "SpeedCommand" is set in the technology object. Status bit "ConstantVelocity" is set upon axis standstill. Both bits are adapted to the new situation when a new motion command is started.

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

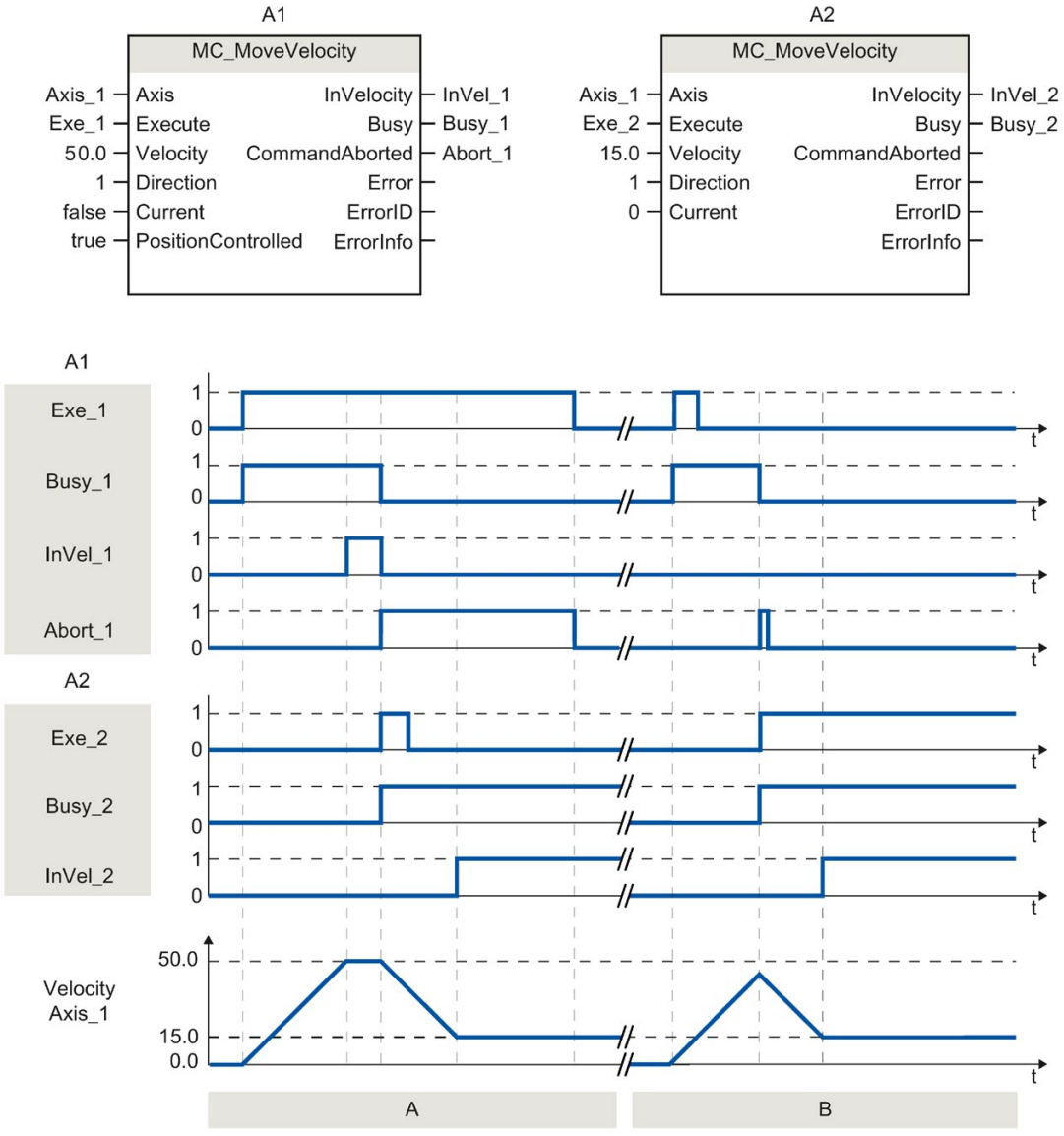
MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

11.1.7.2 MC_MoveVelocity: Function chart as of V6

Function chart



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 10.0

Section A	An active MC_MoveVelocity command signals via "InVel_1" that its target velocity has been reached. It is then aborted by another MC_MoveVelocity command. The abort is signaled via "Abort_1". When the new target velocity 15.0 is reached, this is signaled via "InVel_2". The axis then continues moving at the new constant velocity.
Section B	An active MC_MoveVelocity command is aborted by another MC_MoveVelocity command prior to reaching its target velocity. The abort is signaled via "Abort_1". When the new target velocity 15.0 is reached, this is signaled via "InVel_2". The axis then continues moving at the new constant velocity.

11.1.8 MC_MoveJog

11.1.8.1 MC_MoveJog: Move axis in jog mode as of V6

Description

Motion control instruction "MC_MoveJog" moves the axis constantly at the specified velocity in jog mode. You use this Motion Control instruction, for example, for testing and commissioning purposes.

Requirements

- The positioning axis technology object has been configured correctly.
- The axis is enabled.

Override response

The MC_MoveJog command can be aborted by the following Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The new MC_MoveJog command aborts the following active Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

Parameters

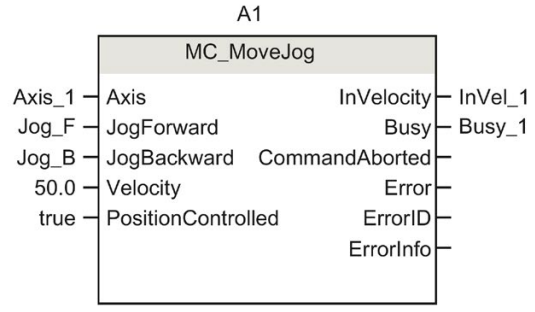
Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_SpeedAxis	-	Axis technology object	
JogForward	INPUT	BOOL	FALSE	As long as the parameter is TRUE, the axis moves in the positive direction at the velocity specified in parameter "Velocity".	
JogBackward	INPUT	BOOL	FALSE	As long as the parameter is TRUE, the axis moves in the negative direction at the velocity specified in parameter "Velocity".	
If both parameters are simultaneously TRUE, the axis stops with the configured deceleration. An error is indicated in parameters "Error", "ErrorID", and "ErrorInfo".					
Velocity	INPUT	REAL	10.0	Preset velocity for jog mode Limit values: Start/stop velocity ≤ velocity ≤ maximum velocity	
PositionControlled	INPUT	BOOL	TRUE	FALSE	Non position-controlled operation
				TRUE	Position-controlled operation
				The parameter applies as long as the "MC_MoveJog" command is being executed. After this, the setting of MC_Power applies again. This parameter is ignored when a PTO axis is used.	
InVelocity	OUTPUT	BOOL	FALSE	TRUE	The velocity specified in parameter "Velocity" was reached.
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	During execution, the command was aborted by another command.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

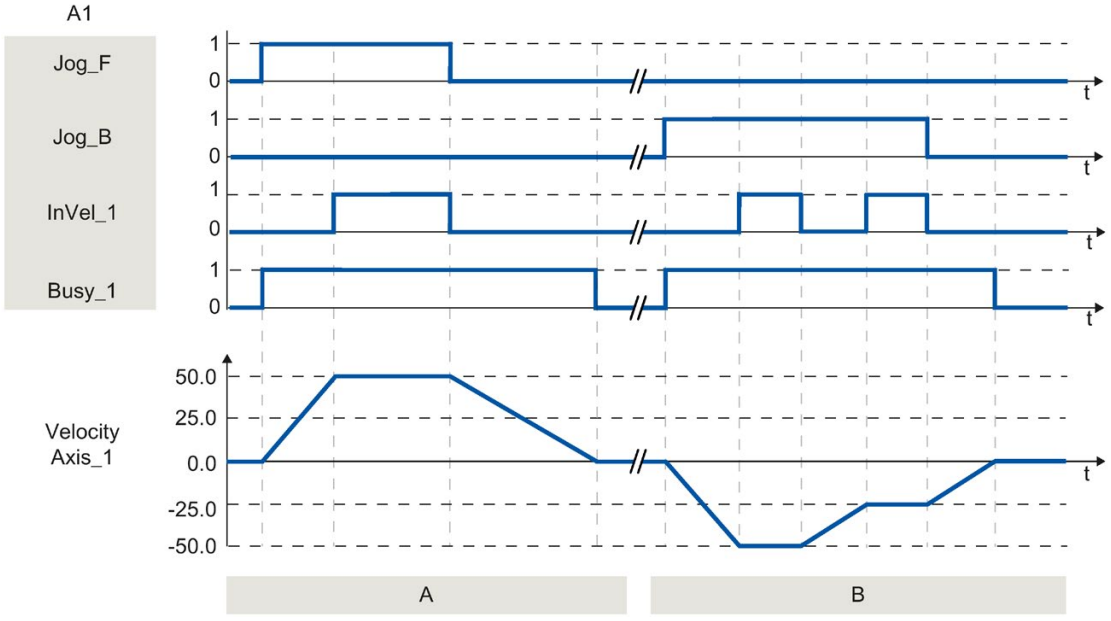
See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)
MC_Power: Enable, disable axis as of V6 (Page 181)
MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)
MC_Home: Home axes, set reference point as of V6 (Page 188)
MC_Halt: Stop axis as of V6 (Page 193)
MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)
MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)
MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)
MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)
MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)
MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)
MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

11.1.8.2 MC_MoveJog: Function chart as of V6

Function chart





The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 5.0

Section A	The axis is moved in the positive direction in jog mode via "Jog_F". When the target velocity 50.0 is reached, this is signaled via "InVel_1". After "Jog_F" is reset, the axis is braked to a standstill.
Section B	The axis is moved in the negative direction in jog mode via "Jog_B". When the target velocity -50.0 is reached, this is signaled via "InVel_1". When "Jog_B" is set, the value at parameter "Velocity" changes to 25.0. "InVel_1" is reset and the axis is braked. When the new target velocity -25.0 is reached, this is signaled via "InVel_1". After "Jog_B" is reset, the axis is braked to a standstill.

11.1.9 MC_CommandTable

11.1.9.1 MC_CommandTable: Run axis commands as motion sequence as of V6

Description

The Motion Control instruction "MC_CommandTable" combines multiple individual axis control commands in one movement sequence. "MC_CommandTable" is available for axes with drive connection via PTO (Pulse Train Output).

Requirements

- The positioning axis technology object has been inserted and correctly configured.
- The drive is connected via PTO (Pulse Train Output).
- The command table technology object has been inserted and correctly configured.
- The axis is enabled.

Override response

The MC_CommandTable command can be aborted by the following Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The new MC_CommandTable command aborts the following active Motion Control commands:

- MC_Home command Mode = 3
- MC_Halt command
- MC_MoveAbsolute command
- MC_MoveRelative command
- MC_MoveVelocity command
- MC_MoveJog command
- MC_CommandTable command

The active Motion Control command is canceled by the start of the first "Positioning Relative", "Positioning Absolute", "Velocity set point" or "Halt" command.

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_SpeedAxis	-	Axis technology object	
Command Table	INPUT	TO_Command Table	-	Command table technology object	
Execute	INPUT	BOOL	FALSE	Command table start with positive edge	
StartStep	INPUT	INT	1	Defines the step at which the execution of the command table should begin Limit values: 1 ≤ StartStep ≤ EndStep	
EndStep	INPUT	INT	32	Defines the step up to which the execution of command table should take place Limit values: StartStep ≤ EndStep ≤ 32	
Done	OUTPUT	BOOL	FALSE	TRUE	Command table has been successfully executed
Busy	OUTPU	BOOL	FALSE	TRUE	The command table is being executed
Command Aborted	OUTPUT	BOOL	FALSE	TRUE	The command table was canceled by another command.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command table. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	
CurrentStep	OUTPUT	INT	0	Step in command table currently being executed	
StepCode	OUTPUT	WORD	16#0000	User-defined numerical value / bit pattern of the step currently being executed	

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

11.1.10 MC_ChangeDynamic

11.1.10.1 MC_ChangeDynamic: Change dynamic settings of axis as of V6

Description

Motion Control instruction "MC_ChangeDynamic" allows you to change the following settings of the axis:

- Change the ramp-up time (acceleration) value
- Change the ramp-down time (deceleration) value
- Change the emergency stop ramp-down time (emergency stop deceleration) value
- Change the smoothing time (jerk) value

For the effectiveness of the change, refer to the description of the tag (Page 259).

Requirements

The positioning axis technology object has been configured correctly.

Override response

A MC_ChangeDynamic command cannot be aborted by any other Motion Control command.

A new MC_ChangeDynamic command does not abort any active Motion Control commands.

Parameters

Parameter	Declaration	Data type	Default value	Description	
Axis	INPUT	TO_SpeedAxis	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Change RampUp	INPUT	BOOL	FALSE	TRUE	Change ramp-up time in line with input parameter "RampUpTime"
RampUp-Time	INPUT	REAL	5.00	Time (in seconds) to accelerate axis from standstill to configured maximum velocity without jerk limit. The change will influence the tag <Axis name>. Config.DynamicDefaults.Acceleration. For the effectiveness of the change, refer to the description of this tag.	
ChangeRampDown	INPUT	BOOL	FALSE	TRUE	Change ramp-down time to correspond to input parameter "RampDownTime"
Ramp-DownTime	INPUT	REAL	5.00	Time (in seconds) to decelerate axis from the configured maximum velocity to standstill without jerk limiter. The change will influence the tag <Axis name>. Config.DynamicDefaults.Deceleration . For the effectiveness of the change, refer to the description of this tag.	
Change Emergency	INPUT	BOOL	FALSE	TRUE	Change emergency stop ramp-down time in line with input parameter "EmergencyRampTime"
Emergency RampTime	INPUT	REAL	2.00	Time (in seconds) to decelerate the axis from configured maximum velocity to standstill without jerk limiter in emergency stop mode. The change will influence the tag <Axis name>. Config.DynamicDefaults.EmergencyDeceleration . For the effectiveness of the change, refer to the description of this tag.	
ChangeJerkTime	INPUT	BOOL	FALSE	TRUE	Change smoothing time according to the input parameter "JerkTime"
JerkTime	INPUT	REAL	0.25	Smoothing time (in seconds) used for the axis acceleration and deceleration ramps The change will influence the tag <Axis name>. Config.DynamicDefaults.Jerk . For the effectiveness of the change, refer to the description of this tag.	
Done	OUTPUT	BOOL	FALSE	TRUE	The changed values have been written to the technology data block. The description of the tags will show when the change becomes effective.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

Note

At the input parameters "RampUpTime", "RampDownTime", "EmergencyRampTime" and "JerkTime", values can be entered which exceed the admissible limits of the resulting parameters: "Acceleration", "Deceleration", "Emergency stop deceleration" and "Jerk".

Ensure that your inputs are within the valid range, taking into consideration the equations and limits in section "Dynamic (Page 81)".

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

Tags of the positioning axis technology object as of V6 (Page 259)

11.1.11 MC_ReadParam

11.1.11.1 MC_ReadParam: Continuously read motion data of a positioning axis as of V6

Description

The Motion Control instruction "MC_ReadParam" enables continuous reading of motion data and status messages of an axis. The current value of the corresponding tags is determined at the start of the command.

The following motion data and status messages can be read:

- As of technology version V4:
  - Setpoint position of the axis
  - Setpoint and actual velocity of the axis
  - Current distance of axis from target position
  - Target position of the axis
- Additional as of technology version V5:
  - Actual position of the axis
  - Actual velocity of the axis
  - Current following error
  - Drive status
  - Encoder status
  - Status bits
  - Error bits

Requirements

The positioning axis technology object has been configured correctly.

Override response

A MC_ReadParam command cannot be aborted by any other Motion Control command.

A new MC_ReadParam command does not abort any active Motion Control commands.

Parameters

Parameter	Declaration	Data type	Default value	Description	
Enable	INPUT	BOOL	FALSE	TRUE	Read the tag specified with the "Parameter" and store the value in the destination address specified with "Value".
				FALSE	Do not update assigned motion data
Parameter	INPUT	VARIANT (REAL)	-	VARIANT pointer to the value to be read. The following tags are permitted: • <Axis name>.Position • <Axis name>.Velocity • <Axis name>.ActualPosition • <Axis name>.ActualVelocity • <Axis name>.StatusPositioning.<Tag name> • <Axis name>.StatusDrive.<Tag name> • <Axis name>.StatusSensor.<Tag name> • <Axis name>.StatusBits.<Tag name> • <Axis name>.ErrorBits.<Tag name> The description of the tags named and the tag structures can be found in the Appendix Tags of the positioning axis technology object as of V6 (Page 259).	
Value	INOUT	VARIANT (REAL)	-	VARIANT pointer to the target tag or destination address to which the read value is to be written.	
Valid	OUTPUT	BOOL	FALSE	TRUE	The read value is valid.
				FALSE	The read value is invalid.
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

MC_WriteParam: Write tag of positioning axis as of V6 (Page 218)

11.1.12 MC_WriteParam

11.1.12.1 MC_WriteParam: Write tag of positioning axis as of V6

Description

Motion Control instruction "MC_WriteParam" enables the writing of tags of the positioning axis technology object in the user program. In contrast to the value assignment of the tags in the user program, "MC_WriteParam" can also change values of read-only tags.

You can learn about the tags, the conditions under which they can be written and the time at which they take effect in the description of the technology object tags (Page 259).

Requirements

- The positioning axis technology object has been configured correctly.
- To write tags that are read-only in the user program, the axis must be disabled.
- Tags whose change requires a restart cannot be written with "MC_WriteParam".

Override response

A MC_WriteParam command cannot be aborted by any other Motion Control command.

A new MC_WriteParam command does not abort any active Motion Control commands.

Parameters

Parameter	Declaration	Data type	Default value	Description	
Parameter	INPUT	VARIANT (BOOL, INT, DINT, UDINT, REAL)	-	VARIANT pointer to the technology object tags (Page 259) positioning axis (destination address) to be written	
Value	INPUT	VARIANT (BOOL, INT, DINT, UDINT, REAL)	-	VARIANT pointer to the value to be written (source address)	
Execute	INPUT	BOOL	FALSE	Start of the command with a positive edge	
Done	OUTPUT	BOOL	FALSE	TRUE	Value was written
Busy	OUTPUT	BOOL	FALSE	TRUE	The command is being executed
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred during execution of the command. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	16#0000	Error ID (Page 236) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	16#0000	Error info ID (Page 236) for parameter "ErrorID"	

See also

List of ErrorIDs and ErrorInfos (technology objects as of V6) (Page 236)

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Reset: Acknowledge fault, restart technology object as of V6 (Page 186)

MC_Home: Home axes, set reference point as of V6 (Page 188)

MC_Halt: Stop axis as of V6 (Page 193)

MC_MoveAbsolute: Absolute positioning of axis as of V6 (Page 196)

MC_MoveRelative: Relative positioning of axis as of V6 (Page 200)

MC_MoveVelocity: Move axis at set velocity as of V6 (Page 204)

MC_MoveJog: Move axis in jog mode as of V6 (Page 208)

MC_CommandTable: Run axis commands as motion sequence as of V6 (Page 211)

MC_ChangeDynamic: Change dynamic settings of axis as of V6 (Page 213)

MC_ReadParam: Continuously read motion data of a positioning axis as of V6 (Page 216)

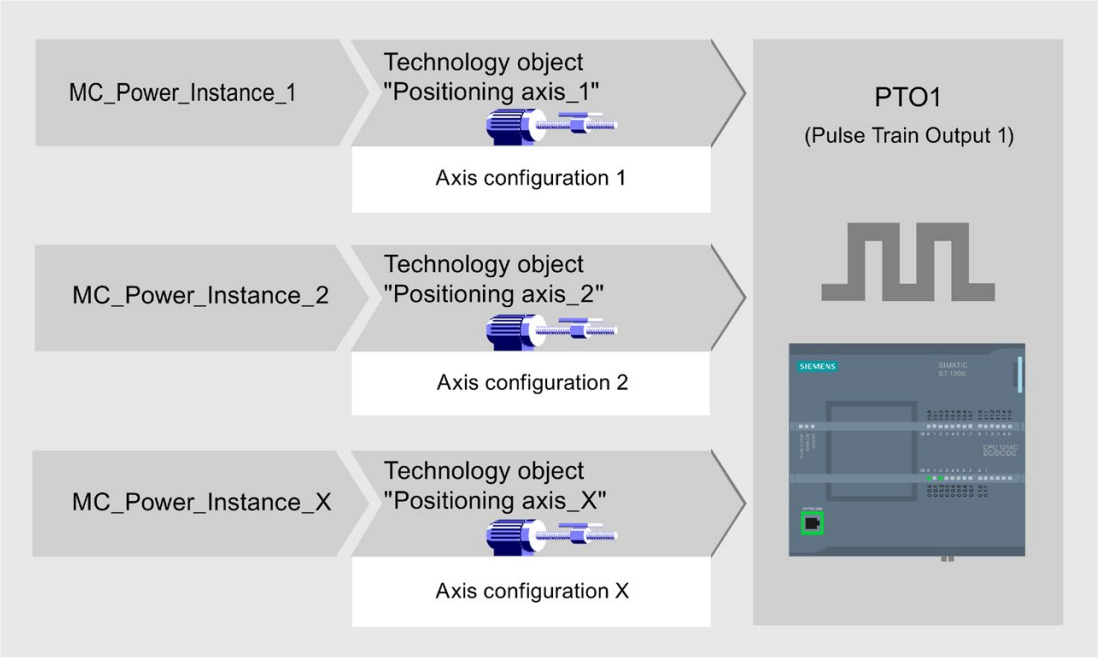
Tags of the positioning axis technology object as of V6 (Page 259)

Appendix

12

12.1 Using multiple axes with the same PTO

Use the Motion Control functionality of the CPU S7-1200 to run multiple positioning axis technology objects with the same PTO (Pulse Train Output) and thus with the same CPU outputs. This is appropriate, for example, if different axis configurations are to be used for different production sequences via one PTO. As described below, it is possible to switch between these axis configurations as often as necessary. The following diagram presents the basic functional relationships:

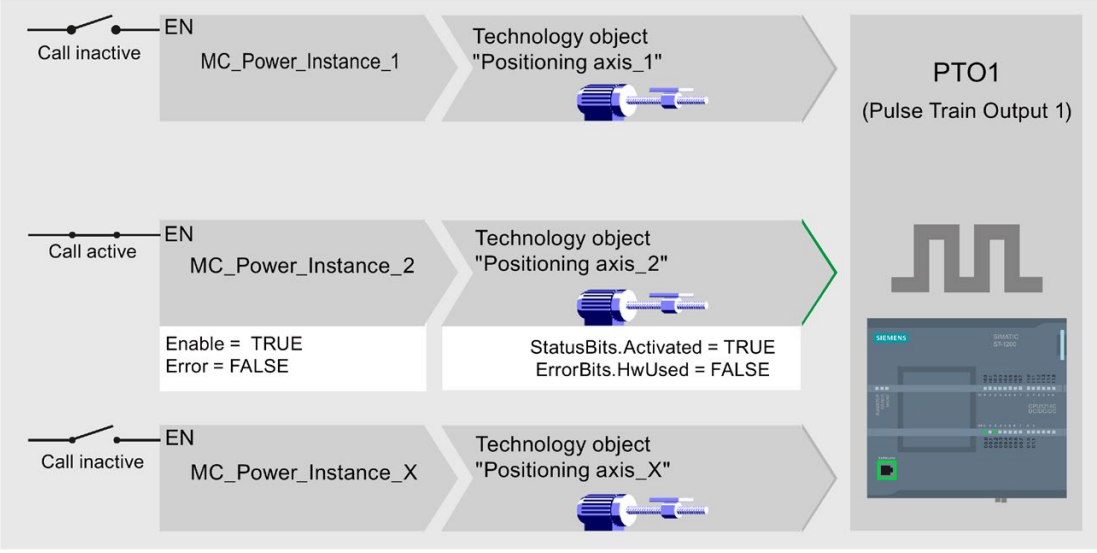


In this example, several positioning axis technology objects, each with its own axis configuration, use the same PTO. Each axis must be called in the user program with a separate call of Motion Control instruction "MC_Power" with a separate instance data block. Only one axis at a time may use the PTO. The axis that is currently using the PTO indicates this with tag <Axis name>.StatusBits.Activated = TRUE.

Switchover of the positioning axis technology object

The program scheme described below shows you how to switch between different technology objects and, thus, between different axis configurations. To use the same PTO with multiple axes without error indications, only the Motion Control instructions of the axis currently being used may be called.

The following diagram presents this principle using Motion Control instruction "MC_Power" as an example:



The tags of the activated axis ("Positioning axis_2" here) show the following typical indicators in the user program:

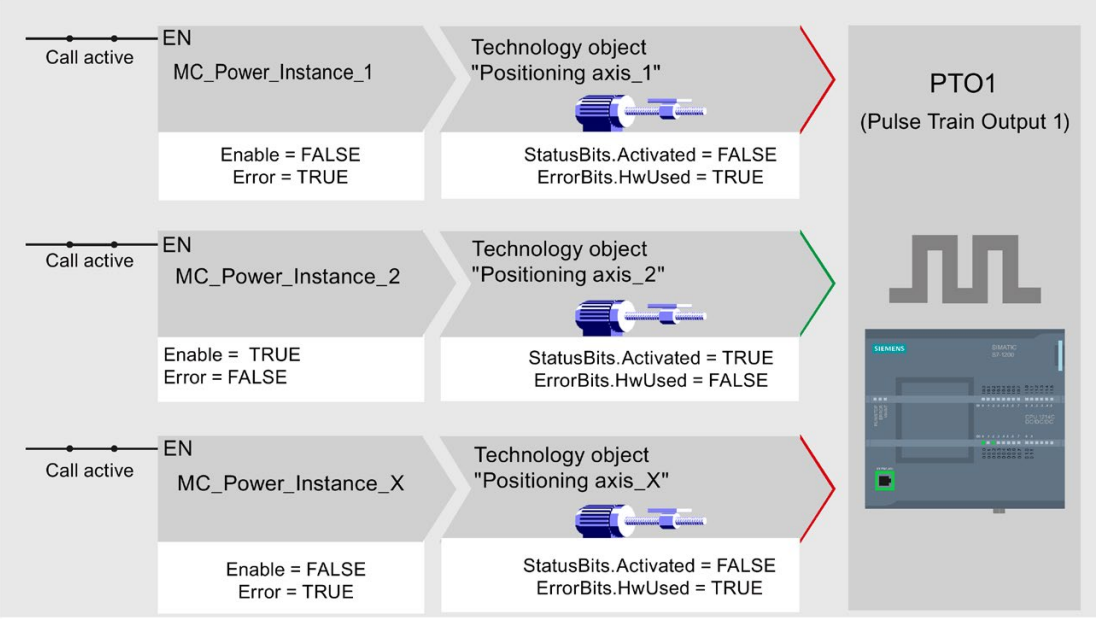
- <Axis name>.StatusBits.Activated = TRUE
- <Axis name>.ErrorBits.HWUsed = FALSE

To switch from one positioning axis technology object to another, follow the steps described below. In the example, a switch is made from "Positioning axis_2" to "Positioning axis_1":

1. End any active traversing motions of activated "Positioning axis_2"
2. Disable "Positioning axis_2" with the associated Motion Control instruction "MC_Power" using input parameter Enable = FALSE
3. To verify that "Positioning axis_2" has been disabled, use an AND operation of output parameter Status = FALSE of Motion Control instruction "MC_Power" and technology object tag <Axis name>.StatusBits.Enable = FALSE.
4. Deactivate the conditional call of the Motion Control instructions for "Positioning axis_2".
5. Activate the conditional call of the Motion Control instruction for "Positioning axis_1". At the first call of the corresponding Motion Control instruction "MC_Power", "Positioning axis_2" is deactivated and "Positioning axis_1" is activated.
6. Enable "Positioning axis_1" with the associated Motion Control instruction "MC_Power" using input parameter Enable = TRUE.

7. To verify that "Positioning axis_1" has been enabled, use an AND operation of output parameter Status = TRUE of Motion Control instruction "MC_Power" and technology object tag <Axis name>.StatusBits.Enable = TRUE.

It is also always possible to cyclically call all Motion Control instructions of all axes working with a single PTO.



When an axis is enabled (here "Positioning axis_2"), this axis becomes active.

In contrast to the conditional call, the Motion Control instructions of the deactivated axes (here "Positioning axis_1" and "Positioning axis_x") indicate errors. The tags of these axes indicate the status <Axis name>.StatusBits.Activated = FALSE and <Axis name>.ErrorBits.HWUsed = TRUE.

Use the conditional call of the Motion Control instructions if you want to implement the user program without error indicators.

See also

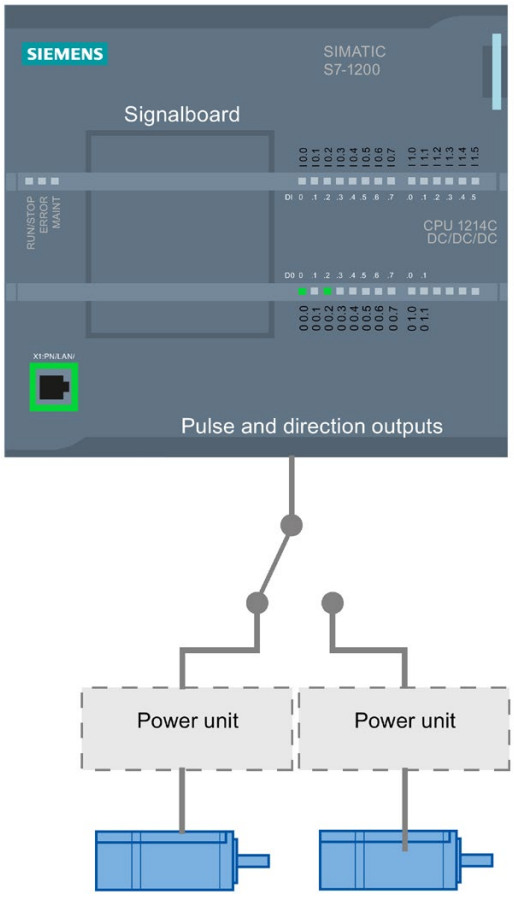
Using multiple drives with the same PTO (Page 223)

Tracking jobs from higher priority classes (execution levels) (Page 224)

Special cases when using software limit switches for drive connection via PTO (Page 227)

12.2 Using multiple drives with the same PTO

If multiple drives are to be used, they can be run with a common PTO (Pulse Train Output) using changeover. The following diagram represents the basic circuit design:



The changeover between drives can be controlled, if required, by the user program via a digital output. If different axis configurations are required for the different drives, a changeover between these configurations is required for the PTO. For additional information on this topic, refer to "Using multiple axes with the same PTO (Page 220)".

See also

Using multiple axes with the same PTO (Page 220)

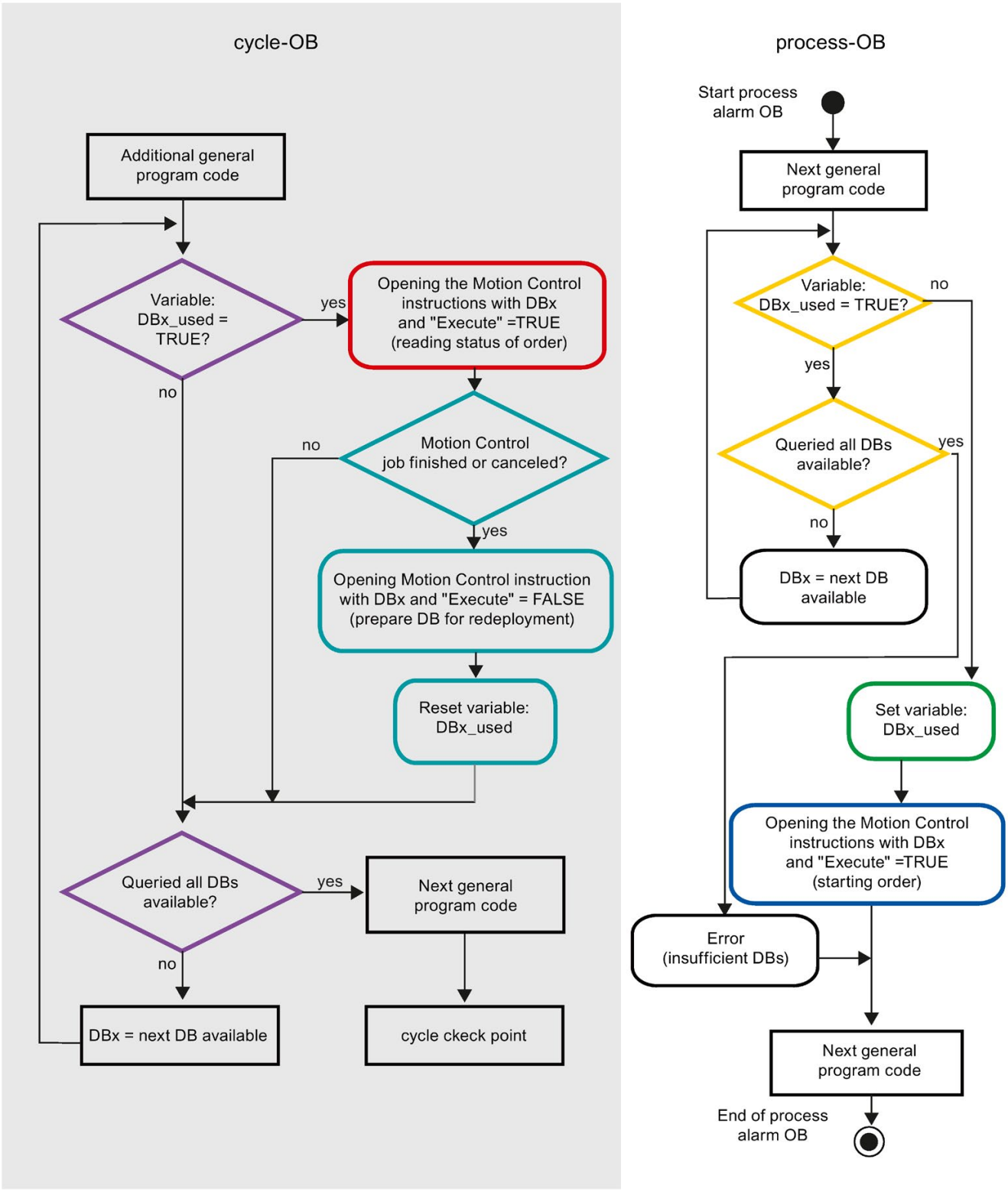
Tracking jobs from higher priority classes (execution levels) (Page 224)

Special cases when using software limit switches for drive connection via PTO (Page 227)

12.3 Tracking jobs from higher priority classes (execution levels)

Depending on the application, it may be necessary to start Motion Control commands (for example, interrupt-controlled) in a higher priority class (execution level).

The Motion Control instructions must be called at short intervals for status monitoring. Motion Control commands cannot be sufficiently closely monitored if the higher priority Motion Control commands are called only once or at too great an interval. Tracking in such cases can be carried out in the cycle OB. An instance data block that is not currently being utilized must be available for each start of a higher priority Motion Control command. Refer to the following flow chart to see how you start Motion Control commands in a higher priority class (for example, hardware interrupt OB) and continue tracking in the program cycle OB:



Depending on the frequency of the Motion Control commands you want to start, you will have to generate a sufficient number of instance data blocks. Users determine which instance data block is currently used in the DBx_used tags.

Start of Motion Control command in the hardware interrupt OB

Binary queries of the DBx_used tags (orange) are used to find an instance data block not currently in use. If such an instance data block is found, the utilized instance data block is marked as "used" (green) and the Motion Control command is started with this instance data block (blue).

Any other program sections of the hardware interrupt OB are then executed, followed by a return to the program cycle OB.

Tracking of started Motion Control commands in the program cycle OB

All instance data blocks available in the cycle OB are checked to determine if they are currently in use by means of the DBx_used tag (violet).

If an instance data block is in use (Motion Control command is being processed), the Motion Control instruction with this instance data block and input parameter Execute = TRUE is called to read out the status messages (red).

If the command is complete or has been aborted, the following actions are taken next (blue green):

- Call of Motion Control instruction with input parameter Execute = FALSE
- Resetting the DBx_used tag

This completes the command tracking, and the instance data block is now available for use again.

See also

Using multiple axes with the same PTO (Page 220)

Using multiple drives with the same PTO (Page 223)

Special cases when using software limit switches for drive connection via PTO (Page 227)

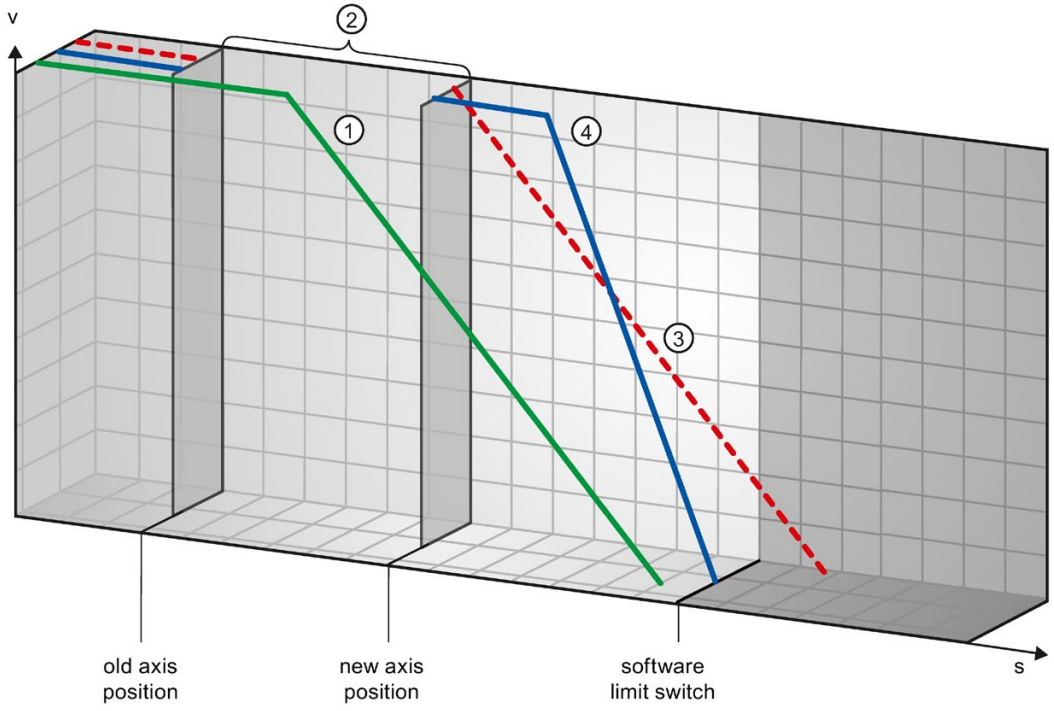
12.4 Special cases when using software limit switches for drive connection via PTO

12.4.1 Software limit switches in conjunction with a homing operation

Due to unfavorably parameterized homing jobs, the braking action of the axis may be influenced at the software limit switch. Take the following examples into consideration when developing your program.

Example 1:

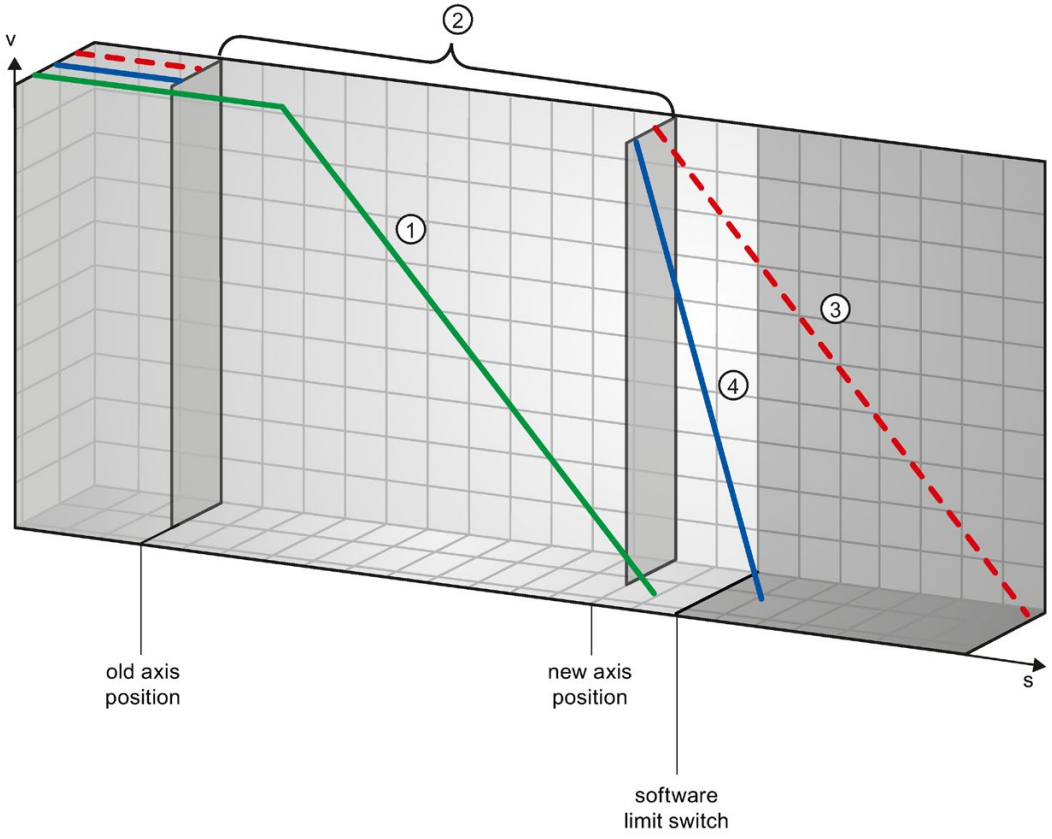
During a travel command, a homing job (for example, Set reference point) offsets the current axis position in the direction of the software limit switch. It is still possible to bring the axis to a standstill before reaching the software limit switch:



①	The green curve shows the motion **without** the homing job. The axis brakes at the configured deceleration and comes to a standstill at a position before the software limit switch.
②	A new axis position is set as a result of the homing job. The area between the old and new axis position is thus "skipped".
③	Based on the new axis position, the axis would theoretically be stopped with the configured deceleration at a position after the software limit switch (red curve).
④	Because braking with the configured deceleration is no longer sufficient, the axis actually follows the blue curve. Following constant motion, the axis brakes at the configured emergency stop deceleration and comes to a standstill at the position of the software limit switch.

Example 2:

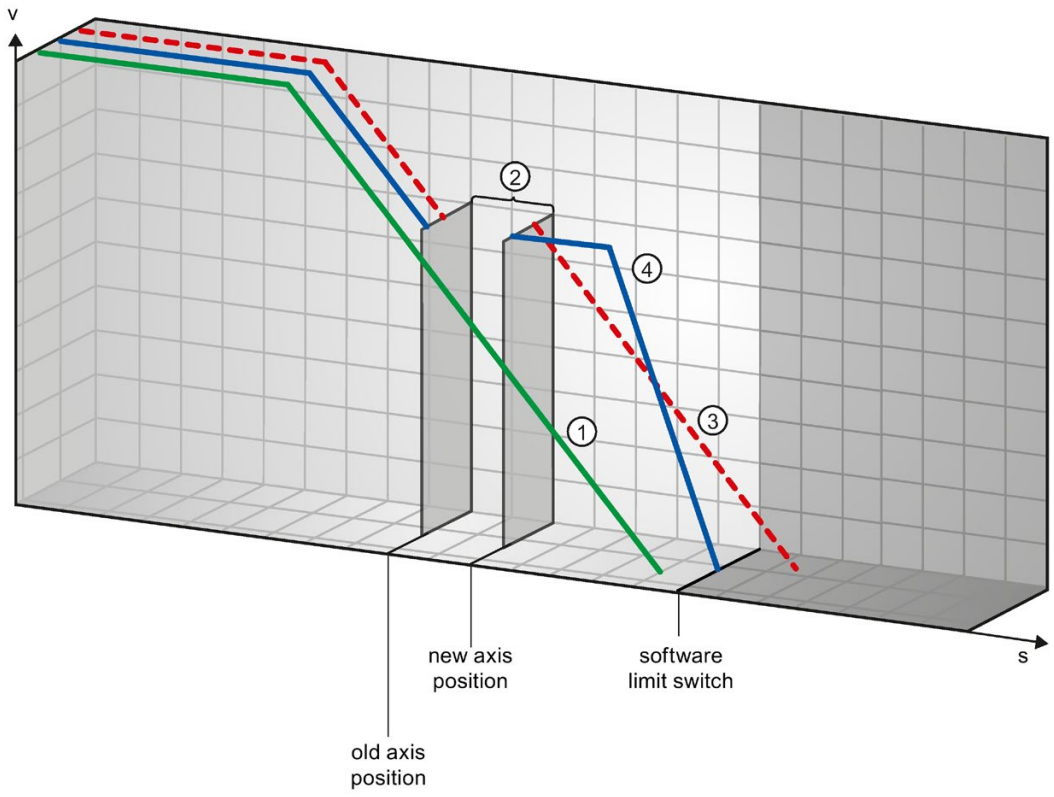
During a travel command, a homing job (for example, Set reference point) offsets the current axis position in the direction of the software limit switch. In contrast to example 1, it is no longer possible to bring the axis to a standstill before reaching the software limit switch. The axis overruns the position of the software limit switch.



①	The green curve shows the motion **without** the homing job. The axis brakes at the configured deceleration and comes to a standstill at a position before the software limit switch.
②	A new axis position is set as a result of the homing job. The area between the old and new axis position is thus "skipped".
③	Based on the new axis position, the axis would theoretically be stopped with the configured deceleration at a position well after the software limit switch (red curve).
④	Because braking with the configured deceleration is no longer sufficient, the axis actually follows the blue curve. The axis brakes with the configured emergency stop deceleration. However, the emergency stop deceleration is not sufficient to stop the axis at the position of the software limit switch. The position of the software limit switch is overrun.

Example 3:

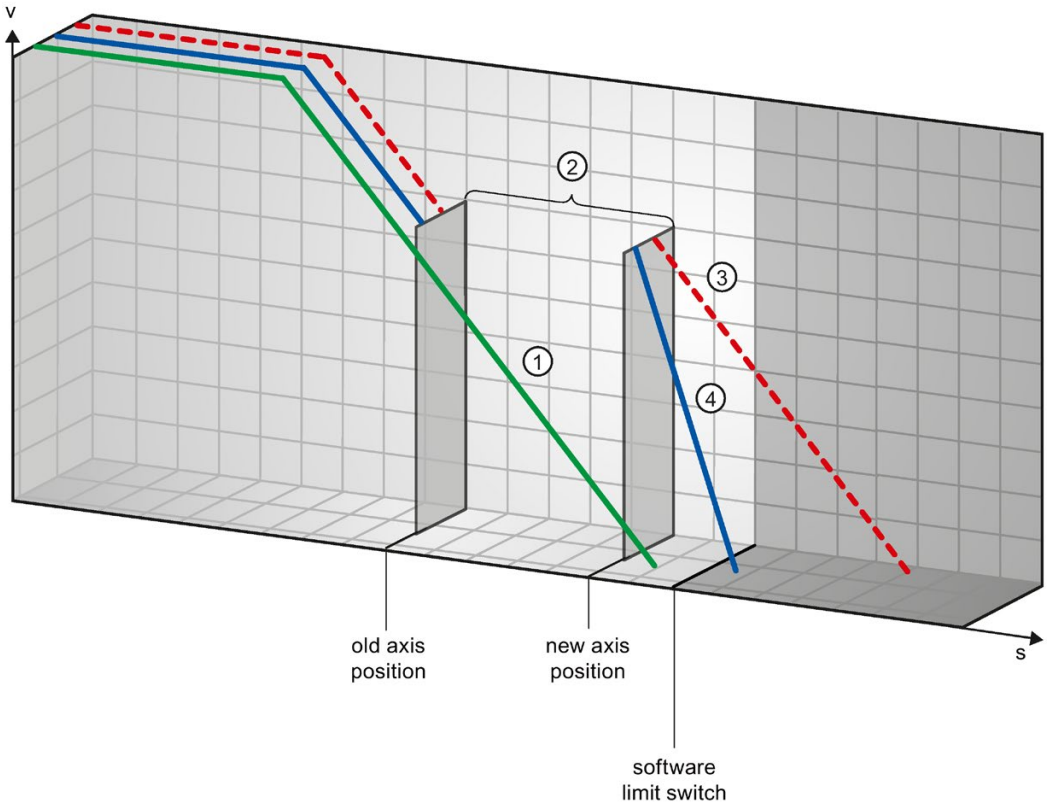
During a braking operation, a homing job (for example, Set reference point) offsets the current axis position in the direction of the software limit switch. It is still possible to bring the axis to a standstill before reaching the software limit switch:



①	The green curve shows the motion **without** the homing job. The axis brakes at the configured deceleration and comes to a standstill at a position before the software limit switch.
②	A new axis position is set as a result of the homing job. The area between the old and new axis position is thus "skipped".
③	Based on the new axis position, the axis would theoretically be stopped with the configured deceleration at a position after the software limit switch (red curve).
④	Because braking with the configured deceleration is no longer sufficient, the axis actually follows the blue curve. Following constant motion, the axis brakes at the configured emergency stop deceleration and comes to a standstill at the position of the software limit switch.

Example 4:

During a braking operation, a homing job (for example, Set reference point) offsets the current axis position in the direction of the software limit switch. In contrast to example 3, it is no longer possible to bring the axis to a standstill before reaching the software limit switch. The axis overruns the position of the software limit switch.



①	The green curve shows the motion **without** the homing job. The axis brakes at the configured deceleration and comes to a standstill at a position before the software limit switch.
②	A new axis position is set as a result of the homing job. The area between the old and new axis position is thus "skipped".
③	Based on the new axis position, the axis would theoretically be stopped with the configured deceleration at a position well after the software limit switch (red curve).
④	Because braking with the configured deceleration is no longer sufficient, the axis actually follows the blue curve. The axis brakes with the configured emergency stop deceleration. However, the emergency stop deceleration is not sufficient to stop the axis at the position of the software limit switch. The position of the software limit switch is overrun.

See also

Software limit switches and software limit switch position changes. (Page 231)

Software limit switches in conjunction with dynamic changes (Page 232)

Response of the axis when position limits are triggered (Page 78)

12.4.2 Software limit switches and software limit switch position changes.

An incorrect change in the position of the software limit switch during the runtime of the user program can abruptly reduce the distance between the current axis position and the position of the software limit switch.

The axis response is similar to that described in Software limit switches in conjunction with a homing operation (Page 227).

See also

Software limit switches in conjunction with a homing operation (Page 227)

Software limit switches in conjunction with dynamic changes (Page 232)

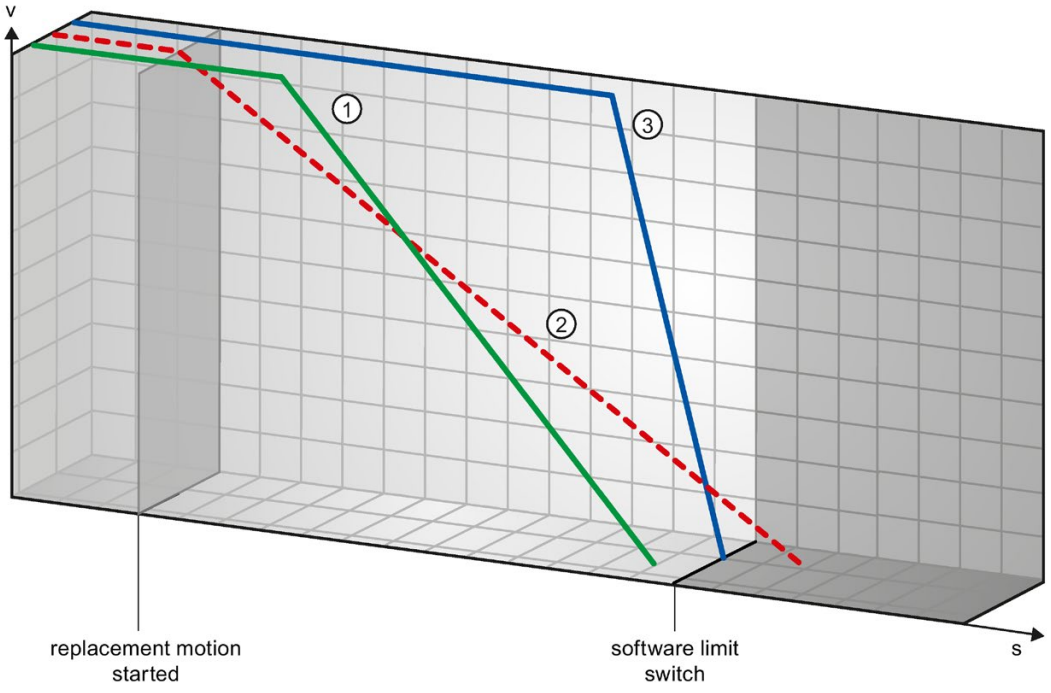
Response of the axis when position limits are triggered (Page 78)

12.4.3 Software limit switches in conjunction with dynamic changes

It is possible to influence the deceleration of the axis in the area of the software limit switches in conjunction with overriding motion commands. This applies when the overriding motion command is started with a lower deceleration (tag <Axis name>.DynamicDefaults.Deceleration). Take the following examples into consideration when developing your program.

Example 1:

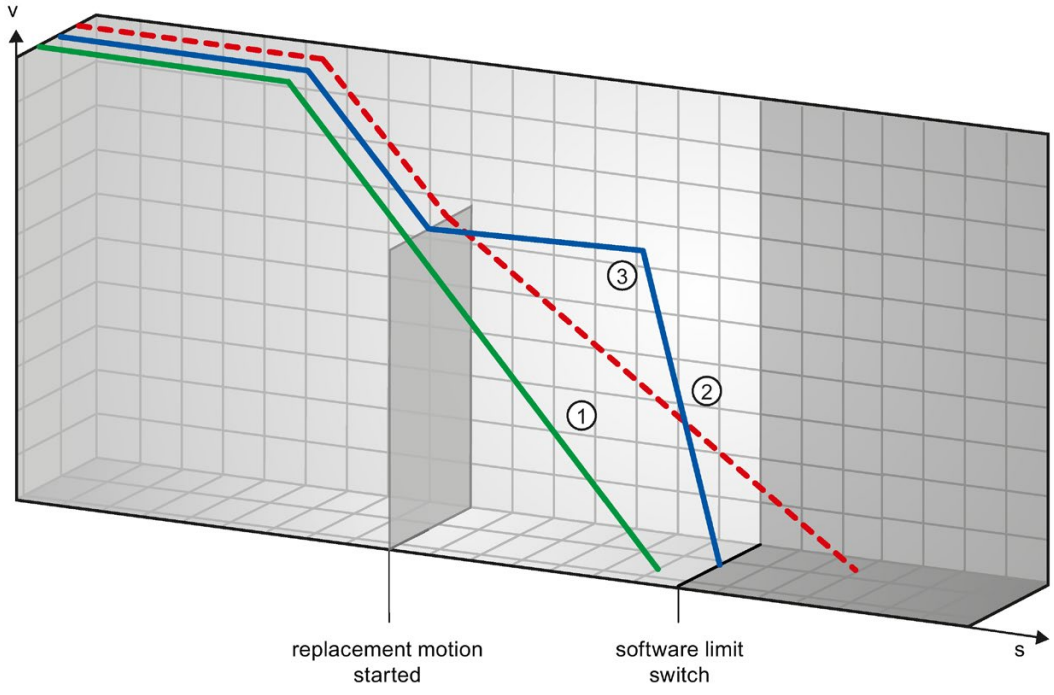
During axis motion, an active motion command is overridden by another motion command with a lower deceleration:



①	The green curve shows the motion of an active command **without** overriding this command. The axis brakes at the configured deceleration and comes to a standstill at a position before the software limit switch.
②	Based on the overriding motion command with lower deceleration, the axis would theoretically be stopped with the configured deceleration at a position after the software limit switch (red curve).
③	Because braking with the configured deceleration of the overriding motion command is no longer sufficient, the axis actually follows the blue curve. Following a constant motion, the axis brakes at the emergency stop deceleration and comes to a standstill at the position of the software limit switch.

Example 2:

During braking of the axis, an active motion command is overridden by another motion command with a lower deceleration:



①	The green curve shows the motion of an active command **without** overriding this command. The axis brakes at the configured deceleration and comes to a standstill at a position before the software limit switch.
②	Based on the overriding motion command with lower deceleration, the axis would theoretically be stopped at a position well after the software limit switch (red curve).
③	Because braking with the configured deceleration of the overriding motion command is no longer sufficient, the axis actually follows the blue curve. Following a constant motion, the axis brakes at the emergency stop deceleration and comes to a standstill at the position of the software limit switch.

See also

Software limit switches in conjunction with a homing operation (Page 227)

Software limit switches and software limit switch position changes. (Page 231)

Response of the axis when position limits are triggered (Page 78)

12.5 Reducing velocity for a short positioning duration

The CPU can reduce the velocity of a positioning command when the planned positioning duration is < 2 ms.

The velocity of command execution will then be reduced for the entire duration. The reduced velocity (pulses per s) is calculated as follows:

- Reduced velocity = Number of pulses to be output * 500Hz

Velocity is **not** reduced if the planned positioning duration is >= 2 ms.

12.6 Dynamic adjustment of start/stop velocity

The configuration of your velocity limits (start/stop velocity and maximum velocity), the dynamic values (acceleration, deceleration and jerk) and the target speed of the traversing command may under certain circumstances result in the start/stop velocity being dynamically adjusted by the CPU.

This is the case, for example, if the required time for the first pulse would be longer than required for the entire acceleration due to a configured low start/stop velocity. The first pulse is in these cases output at a greater velocity than the configured start/stop velocity. The subsequent pulses are also dynamically adjusted to ensure the acceleration process can be completed in the specified time.

If any pulse loss occurs, make sure that hardware (drive) you are using is adapted to this situation or change the dynamic settings of your axis to avoid the dynamic adjustment of the start/stop velocity.

12.7 Move the axis without position control for servicing

If service is required, it may be necessary to move a PROFIdrive drive or a drive with an analog drive interface without position control.

This may be the case for example with invalid or incorrect encoder values.

The following Motion Control instructions influence the status of the position control:

Enable position-controlled drives without position control with MC_Power

Start the axis with the Motion Control instruction MC_Power StartMode = 0 in non-position-controlled operation.

The non-position-controlled mode is in effect until another Motion Control instruction changes the status of the position control.

MC_MoveVelocity

MC_MoveVelocity with PositionControlled = FALSE forces non-position-controlled operation.

MC_MoveVelocity with PositionControlled = TRUE forces position-controlled operation.

The selected position-controlled operation is retained even after the termination of MC_MoveVelocity.

MC_MoveJog

MC_MoveJog with PositionControlled = FALSE forces non-position-controlled operation.

MC_MoveJog with PositionControlled = TRUE forces position-controlled operation.

The selected position-controlled operation is retained even after the termination of MC_MoveJog.

MC_Home, MC_MoveRelative, MC_MoveAbsolute

The Motion Control instructions MC_Home, MC_MoveRelative and MC_MoveAbsolute force position-controlled operation.

The position control remains active even when the command has ended.

MC_Halt

The Motion Control instruction MC_Halt is executed in position-controlled and non-position-controlled operation.

The status of the position control is not changed.

See also

MC_Power: Enable, disable axis as of V6 (Page 181)

MC_Halt: Stop axis as of V6 (Page 193)

12.8 List of ErrorIDs and ErrorInfos (technology objects as of V6)

The following table lists all ErrorIDs and ErrorInfos that can be indicated in Motion Control instructions. In addition to the cause of the error, remedies for eliminating the error are also listed:

Depending on the error reaction, the axis is stopped in the case of operating errors with stop of axis. The following error reactions are possible:

- **Remove enable**

  The setpoint zero is output and the enable is removed. The axis is braked depending on the configuration in the drive, and is brought to a standstill.

- **Stop with emergency stop ramp**

  Active motion commands are aborted. The axis is braked with the emergency stop deceleration configured under "Technology object > Extended parameters > Dynamics > Emergency stop ramp" without any jerk limit and brought to a standstill.

Operating error with axis stop

ErrorID	ErrorInfo	Description	Remedy	Error reaction
16#8000		**Drive error, loss of "Drive ready"**		-
	16#0001	-	Acknowledge error with instruction "MC_Reset"; provide drive signal; restart command, if necessary	
16#8001		**SW low limit switch has been tripped**		-
	16#000E	The position of the SW low limit switch was reached with the currently configured deceleration	Acknowledge the error with instruction "MC_Reset"; use a motion command to move the axis in the positive direction out of the range of the SW limit switch	
	16#000F	The position of the SW low limit switch was reached with the emergency stop deceleration		
	16#0010	The position of the SW low limit switch was exceeded with the emergency stop deceleration		Remove enable
16#8002		**SW high limit switch has been tripped**		-
	16#000E	The position of the SW high limit switch was reached with the currently configured deceleration	Acknowledge the error with instruction "MC_Reset"; use a motion command to move the axis in the negative direction out of the range of the SW limit switch	
	16#000F	The position of the SW high limit switch was reached with the emergency stop deceleration		
	16#0010	The position of the SW high limit switch was exceeded with the emergency stop deceleration		Remove enable

ErrorID	ErrorInfo	Description	Remedy	Error reaction
16#8003		**HW low limit switch was reached**		For drive connection via PTO (Pulse Train Output): Stop with emergency stop ramp For drive connection via PROFIdrive/ analog output: Remove enable
	16#000E	The HW low limit switch was reached. The axis was stopped with the emergency stop deceleration. (During an active homing procedure, the homing switch was not found)	Acknowledge the error for an enabled axis with instruction "MC_Reset"; use a motion command to move the axis in the positive direction out of the range of the HW limit switch.	
16#8004		**HW high limit switch was reached**		For drive connection via PTO (Pulse Train Output): Stop with emergency stop ramp For drive connection via PROFIdrive/ analog output: Remove enable
	16#000E	The HW high limit switch was reached. The axis was stopped with the emergency stop deceleration. (During an active homing procedure, the homing switch was not found)	Acknowledge the error for an enabled axis with instruction "MC_Reset"; use a motion command to move the axis in the negative direction out of the range of the HW limit switch.	
16#8005		**PTO/HSC are already being used by another axis**		-
	16#0001	-	**The axis was configured incorrectly:** Correct the configuration of the PTO (Pulse Train Output) / HSC (High Speed Counter) and download it to the controller	
			More than one axis is to run with one PTO: Another axis is using the PTO / HSC. If the current axis is to assume the control, the other axis must be disabled with "MC_Power" Enable = FALSE. (See also Using multiple axes with the same PTO (Page 220))	
16#8006		**A communication error in the control panel has occurred**		Remove enable
	16#0012	A timeout has occurred	Check the cable connection and press the "Manual control" button again	
16#8007		**The axis cannot be enabled**		-
	16#0025	Restarting	Wait until the axis restart is complete.	
	16#0026	Executing loading process in RUN mode	Wait until the loading process is complete.	

ErrorID	ErrorInfo	Description	Remedy	Error reaction
16#8008		**Invalid direction of motion**		-
	16#002E	The selected motion direction is not allowed.	• Adjust the motion direction and restart the command. • Adjust the allowed direction of rotation in the technology object configuration under "Extended parameters > Mechanics". Restart the command.	
	16#002F	A reversing motion is not possible with the selected motion direction.		
16#8009		**Reference switch/encoder zero mark not found**		Stop with emergency stop ramp
	16#0033	Error in the configuration, hardware or installation of the encoder or at the homing switch.	• Connect a suitable device. • Check the device (I/Os). • Compare the configuration of HW Config and the technology object.	
16#800A		**Alarm message from encoder**		Remove enable
	16#0001	-	Check the device with regard to function, connections and I/Os.	
	16#0034	Hardware error at encoder		
	16#0035	Encoder dirty		
	16#0036	Error during reading of encoder absolute value	Compare the encoder type in the drive or encoder parameter P979 with the configuration data of the technology object.	
	16#0037	Zero mark monitoring of the encoder	Encoder reports error in zero mark monitoring (fault code 0x0002 in Gx_XIST2, see PROFIdrive profile). Check the plant for electromagnetic compatibility (EMC).	
	16#0038	Encoder is in "Parking" state	• Search for the cause of the error in the connected drive or encoder. • Check whether the error message was possibly triggered by a commissioning action at the drive or encoder.	
	16#0040	PROFIdrive: Encoder on bus failed (station failure).	Check the device with regard to function, connections and I/Os.	
	16#0041	PROFIdrive: Signs of life of encoder faulty.		
16#800B		**Range violation of the position**		Remove enable
	16#0039	Range violation in positive direction	Home the axis to a valid actual value range.	
	16#003A	Range violation in negative direction		
	16#003B	The change of the actual position in a position control clock cycle is greater than the modulo length.	Adjust the modulo length of the employed encoder.	

ErrorID	ErrorInfo	Description	Remedy	Error reaction
16#800C		**Alarm message from drive**		Remove enable
	16#0001	-	Check the device with regard to function, connections and I/Os. In the "Tuning" dialog box, select a smaller gain (Kv).	
	16#003C	PROFIdrive: Drive signal "Control requested" failed.		
	16#003D	PROFIdrive/analog drive connection: Drive has shut down.		
	16#003E	PROFIdrive: Drive on bus failed (station failure)		
	16#003F	PROFIdrive: Signs of life of drive faulty.	• Check the device with regard to function, connections and I/Os. • Compare the clock parameters of HW Config (PROFIBUS line, slave OM for drive or encoder) and the execution system. Tmapc and servo must be configured with the same clock cycle time.	
16#800D		**The permitted following error was exceeded**		Remove enable
	16#0001	-	• Check the configuration of the control loop. • Check the direction signal of the encoder. • Check the configuration of following error monitoring.	
16#800E		**Error at the hardware limit switch**		Remove enable
	16#0042	Illegal free travel direction with active hardware limit switch	The programmed direction of movement is disabled due to the active hardware limit switch. Retract the axis in the opposite direction.	
	16#0043	Hardware limit switch polarity is reversed, axis cannot be freed	Check the mechanical configuration of the hardware limit switch.	
	16#0044	Both hardware limit switches are active, axis cannot be freed		
16#800F		**Error in target range**		Remove enable
	16#0045	Target range not reached	Target range was not reached within the positioning tolerance time. • Check the configuration of the position monitoring. • Check the configuration of the control loop.	
	16#0046	Exit target range again	The target range was exited within the minimum dwell time. • Check the configuration of the position monitoring. • Check the configuration of the control loop.	

ErrorID	ErrorInfo	Description	Remedy	Error reaction
16#8010		**Position of the SW low limit switch is greater than that of the SW high limit switch when the axis is not a modulo axis**		Remove enable
	16#0001	-	Change the position of the software limit switches.	
16#8011		**Approach velocity to the homing switch / zero mark equals zero.**		Remove enable
	16#000A	Value is less than or equal to 0.	Approach velocity > Select zero	
16#8012		**Homing velocity for setting the home position equals zero.**		Remove enable
	16#000A	Value is less than or equal to 0.	Homing velocity > Select zero	
16#8013		**The axis cannot occupy the PTO because it is being used by "CTRL_PTO".**		Remove enable
	16#0001	-	Select another PTO in the configuration.	

Operating error without axis stop

ErrorID	ErrorInfo	Description	Remedy
16#8200		**Axis is not enabled**	
	16#0001	-	Enable the axis; restart the command
16#8201		**Axis has already been enabled by another "MC_Power" instance**	
	16#0001	-	Enable the axis with only one "MC_Power" instance
16#8202		**The maximum number of simultaneous Motion Control commands has been exceeded (max. 200 commands for drive connection via PTO (Pulse Train Output), max. 100 commands for drive connection via PROFIdrive/analog output)**	
	16#0001	-	Reduce the number of simultaneously active commands; restart the command A command is active if parameter "Busy" = TRUE in the Motion Control instruction.
16#8203		**Axis is currently operated in "Manual control" (axis control panel)**	
	16#0001	-	Exit "Manual control"; restart the command
16#8204		**Axis is not homed**	
	16#0001	-	Home the axis with instruction "MC_Home"; restart the command
16#8205		**The axis is currently controlled by the user program (the error is only displayed in the axis control panel)**	
	16#0013	The axis is enabled in the user program	Disable axis with instruction "MC_Power" and select "Manual control" again in the axis control panel
16#8206		**Technology object not activated yet**	
	16#0001	-	Enable the axis with instruction "MC_Power" Enable = TRUE or enable the axis in the axis control panel.
16#8207		**Command rejected**	
	16#0001	-	
	16#0016	Active homing is running; another homing method cannot be started.	Wait for active homing to finish or abort the active homing with a motion command, for example, "MC_Halt".
	16#0018	The axis cannot be moved with a command table while it is being actively or passively homed.	Wait until direct or passive homing is complete.

ErrorID	ErrorInfo	Description	Remedy
	16#0019	The axis cannot be actively or passively homed while a command table is being processed.	Wait for command table to finish or abort the command table with a motion command, for example, "MC_Halt".
	16#0052	The specified position exceeds the numerical limit.	Enter a valid position value at the Motion Control instruction.
	16#0053	The axis is ramping up.	Wait until the axis is ready for operation.
	16#0054	Actual value is invalid	To execute a "MC_Home" command, the actual values must be valid. Check the status of the actual values. The variable of the technology object <axis name>."StatusSensor.State" must show the value 2 (valid).
	16#0058	Command is already used in another execution level.	Call axis only via an "MC_Power" instance.
	16#006B	Call only prohibited in position-controlled mode	Enable the axis with position control with the "MC_Power" with StartMode = 1.
16#8208		**Difference between maximum and start/stop velocity is invalid**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#000A	Value is less than or equal to 0.	
16#8209		**Invalid acceleration for technology object "Axis"**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#000A	Value is less than or equal to 0.	
16#820A		**It is not possible to restart the axis**	
	16#0013	The axis is enabled in the user program	Disable the axis with the "MC_Power" instruction; restart again
	16#0027	The axis is currently being operated in "Manual control" (axis control panel)	Exit "Manual control"; restart again
	16#002C	The axis is not disabled.	Disable the axis; restart the command
	16#0047	The technology object is not ready for restart.	Download the project again.
	16#0048	Condition for restart of the technology object is not satisfied.	Disable the technology object.
16#820B		**It is not possible to execute the command table**	
	16#0026	Executing loading process in RUN mode	Wait until the loading process is complete.
16#820C		**No configuration available**	
	16#0001	-	Internal error Contact the hotline.
	16#0014	The selected hardware is used by another application	

Block parameter error

ErrorID	ErrorInfo	Description	Remedy
16#8400		**Invalid value at parameter "Position" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#0005	Value is outside the number range (greater than 1.0E12)	
	16#0006	Value is outside the number range (less than -1.0E12)	
	16#0030	Value has an incorrect number format or is outside the valid number range	
16#8401		**Invalid value at parameter "Distance" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#0005	Value is outside the number range (greater than 1.0E12)	
	16#0006	Value is outside the number range (less than -1.0E12)	
16#8402		**Invalid value at parameter "Velocity" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#0008	Value is greater than the configured maximum velocity	
	16#0009	Value is less than the configured start/stop velocity	
	16#0024	Value is less than 0	
	16#0030	Value has an incorrect number format or is outside the valid number range	
16#8403		**Invalid value at parameter "Direction" of the Motion Control instruction**	
	16#0011	The selection value is invalid	Correct the selection value; restart the command
16#8404		**Invalid value at parameter "Mode" of the Motion Control instruction**	
	16#0011	The selection value is invalid	Correct the selection value; restart the command
	16#0015	Active/passive homing is not configured	Correct the configuration and download it to the controller; enable the axis and restart the command
	16#0017	The direction reversal is activated at the hardware limit switch, despite the fact that the hardware limit switches are disabled	• Activate the HW limit switch using the variable <Axis name>.PositionLimits_HW.Active = TRUE, restart the command • Correct the configuration and download it to the controller; enable the axis and restart the command
	16#0055	Invalid mode at incremental encoder	Start a homing process for an incremental encoder using parameter "Mode" = 0, 1, 2, 3.
	16#0056	Invalid mode at absolute encoder	Passive and active homing ("Mode" = 2, 3) are not possible for an absolute value encoder. Start a homing process for an absolute encoder using parameter "Mode" = 0, 1.

ErrorID	ErrorInfo	Description	Remedy
16#8405		**Invalid value at parameter "StopMode" of the Motion Control instruction**	
	16#0011	The selection value is invalid	Correct the selection value; enable the axis again
16#8406		**Simultaneous forward and backward jogging is not allowed**	
	16#0001	-	Take steps to ensure that parameters "JogForward" and "JogBackward" do not have signal status TRUE simultaneously; restart the command.
16#8407		**Switching to another axis with instruction "MC_Power" is only permitted after disabling the active axis.**	
	16#0001	-	Disable the active axis; it is then possible to switch to the other axis and enable it.
16#8408		**Invalid value at parameter "Axis" of the Motion Control instruction**	
	16#001A	The specified value does not match the required technology object version	Correct the value; restart the command
	16#001B	The specified value does not match the required technology object type	
	16#001C	The specified value is not a Motion Control technology data block	
16#8409		**Invalid value at parameter "CommandTable" of the Motion Control instruction**	
	16#001A	The specified value does not match the required technology object version	Correct the value; restart the command
	16#001B	The specified value does not match the required technology object type	
	16#001C	The specified value is not a Motion Control technology data block	
16#840A		**Invalid value at parameter "StartStep" of the Motion Control instruction**	
	16#000A	Value is less than or equal to 0.	Correct the value; restart the command
	16#001D	The start step is greater than the end step	
	16#001E	Value is greater than 32	
16#840B		**Invalid value at parameter "EndStep" of the Motion Control instruction**	
	16#000A	Value is less than or equal to 0.	Correct the value; restart the command
	16#001E	Value is greater than 32	
16#840C		**Invalid value at parameter "RampUpTime" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#000A	Value is less than or equal to 0.	
16#840D		**Invalid value at parameter "RampDownTime" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#000A	Value is less than or equal to 0.	
16#840E		**Invalid value at parameter "EmergencyRampTime" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#000A	Value is less than or equal to 0.	

ErrorID	ErrorInfo	Description	Remedy
16#840F		**Invalid value at parameter "JerkTime" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#000A	Value is less than or equal to 0.	
16#8410		**Invalid value at parameter "Parameter" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
	16#000B	Address is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#0028	Data type of VARIANT pointer "Parameter" and "Value" do not match.	Use a suitable data type; restart command
	16#0029	VARIANT pointer "Parameter" does not point to a data block of the technology object.	Correct the VARIANT pointer; restart the command
	16#002A	The value at the VARIANT pointer "Parameter" cannot be read.	Correct the VARIANT pointer; restart the command
	16#002B	The value at the VARIANT pointer "Parameter" cannot be written.	Correct the VARIANT pointer or value; restart the command
	16#002C	The axis is not disabled.	Disable the axis; restart the command
16#8411		**Invalid value at parameter "Value" of the Motion Control instruction**	
	16#0002	The value has an invalid number format	Correct the value; restart the command
16#8412		**Value at "Start Mode" parameter of the Motion Control instruction invalid**	
	16#0011	The selection value is invalid	Correct the selection value; enable the axis again

Configuration error of the axis

ErrorID	ErrorInfo	Description	Remedy
16#8600		**Parameter assignment of pulse generator (PTO is invalid**	
	16#000B	The address is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#0014	The selected hardware is used by another application	
16#8601		**Parameter assignment of the high-speed counter (HSC) is invalid**	
	16#000B	The address is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#0014	The selected hardware is used by another application	
16#8602		**Invalid parameter assignment of "Enable output"**	
	16#000B	The address is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8603		**Invalid parameter assignment of "Ready input"**	
	16#000B	The address is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8604		**Invalid "Pulses per motor revolution" value**	
	16#000A	Value is less than or equal to zero	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"

ErrorID	ErrorInfo	Description	Remedy
16#8605		**Invalid "Distance per revolution" value**	
	16#0002	The value has an invalid number format	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#0005	Value is outside the number range (greater than 1.0E12)	
	16#000A	Value is less than or equal to zero	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8606		**Invalid "Start/stop velocity" value**	
	16#0002	The value has an invalid number format	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#0003	Value is higher than the hardware high limit	
	16#0004	Value is lower than the hardware low limit	
	16#0007	The start/stop velocity is greater than the maximum velocity	
16#8607		**Invalid "maximum velocity" value**	
	16#0002	The value has an invalid number format	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#0003	Value is higher than the hardware high limit	
	16#0004	Value is lower than the hardware low limit	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8608		**Invalid "Acceleration" value**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0003	Value is higher than the hardware high limit	
	16#0004	Value is lower than the hardware low limit	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary

ErrorID	ErrorInfo	Description	Remedy
16#8609		**Invalid "Deceleration" value**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0003	Value is higher than the hardware high limit	
	16#0004	Value is lower than the hardware low limit	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#860A		**Invalid "Emergency stop deceleration" value**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0003	Value is higher than the hardware high limit	
	16#0004	Value is lower than the hardware low limit	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#860B		**Value for position of the SW low limit switch is invalid**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0005	Value is outside the number range (greater than 1.0E12)	
	16#0006	Value is outside the number range (less than -1.0E12)	
	16#0030	The position value of the software low limit switch is greater than that of the software high limit switch	
16#860C		**Value for position of the SW high limit switch is invalid**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0005	Value is outside the number range (greater than 1.0E12)	
	16#0006	Value is outside the number range (less than -1.0E12)	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary

ErrorID	ErrorInfo	Description	Remedy
16#860D		**Invalid address of the HW low limit switch**	
	16#000B	Invalid address	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#000C	The address of the falling edge is invalid	
	16#000D	The address of the rising edge is invalid	
16#860E		**Invalid address of the HW high limit switch**	
	16#000B	Invalid address	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#000C	The address of the falling edge is invalid	
	16#000D	The address of the rising edge is invalid	
16#860F		**Invalid "home position offset" value**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0005	Value is outside the number range (greater than 1.0E12)	
	16#0006	Value is outside the number range (less than -1.0E12)	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8610		**Invalid "approach velocity" value**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0008	The velocity is greater than the maximum velocity	
	16#0009	The velocity is less than the start/stop velocity	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8611		**Invalid "Homing velocity" value**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0008	The velocity is greater than the maximum velocity	
	16#0009	The velocity is less than the start/stop velocity	

ErrorID	ErrorInfo	Description	Remedy
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8612		**Invalid address of the homing switch**	
	16#000B	Invalid address	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#000C	The address of the falling edge is invalid	
	16#000D	The address of the rising edge is invalid	
16#8613		**During active homing, direction reversal at the hardware limit switch is activated although the hardware limit switches are not configured**	
	16#0001	-	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8614		**Invalid "Jerk" value**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#001F	Value is greater than the maximum jerk	
	16#0020	Value is less than the minimum jerk	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8615		**Value for "Unit of measurement" is invalid**	
	16#0011	The selection value is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8616		**Address of homing switch is invalid (passive homing as of V4)**	
	16#0011	The selection value is invalid	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary

ErrorID	ErrorInfo	Description	Remedy
16#8617		**Value of variable <axis name>.Sensor.Sensor[1].ActiveHoming.Mode is invalid**	
	16#0011	The selection value is invalid (Valid value: 2 = Homing via digital input)	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8618		**Value of variable <axis name>.Sensor.Sensor[1].PassiveHoming.Mode is invalid**	
	16#0011	The selection value is invalid (Valid value: 2 = Homing via digital input)	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8619		**Value of variable <axis name>.Actor.Type is invalid**	
	16#0011	The selection value is invalid (Valid value: 2 = Connection via pulse interface)	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#861A		**Value for "Permitted direction of rotation" is invalid**	
	16#0011	The selection value is invalid	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
	16#002D	"Both directions" not allowed when direction output is deactivated	
16#861B		**Faulty load gear factors**	
	16#0031	Valid is invalid.	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#861C		**Illegal combination of data for homing with incremental encoder**	
	16#0031	Valid is invalid.	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#861D		**The set encoder mounting type is invalid. Invalid value in <Axis name>.Sensor.Sensor[1].MountingMode**	
	16#0011	The selection value is invalid	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary

ErrorID	ErrorInfo	Description	Remedy
16#861E		**The configuration of the measuring wheel circumference of the encoder is invalid. Invalid value in <Axis name>.Sensor.Sensor[1].Parameter.DistancePerRevolution**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#861F		**The configuration for the resolution of the linear encoder is faulty. Invalid value in <Axis name>.Sensor.Sensor[1].Parameter.Resolution**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8620		**The set fine resolution for Gn_XIST1 is invalid. Invalid value in <Axis name>.Sensor.Sensor[1].Parameter.FineResolutionXist1**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8621		**The set fine resolution for Gn_XIST1 in <Axis name>.Sensor.Sensor[1].Parameter.FineResolutionXist1 is not consistent with the setting in PROFIdrive parameter P979**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8622		**Invalid value for the configuration date <Axis name>.Actor.Interface.AddressIn or <Axis name>.Actor.Interface.AddressOut**	
	16#0011	The selection value is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8623		**The value set in the variable <axis name>.Sensor.Sensor[1].Type is invalid.**	
	16#0011	The selection value is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8624		**The set encoder system is invalid. Invalid value in <Axis name>.Sensor.Sensor[1].System**	
	16#0011	The selection value is invalid	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary

ErrorID	ErrorInfo	Description	Remedy
16#8625		**Parameter of position monitoring is faulty. Invalid value in <Axis name>.PositioningMonitoring.MinDwellTime**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8626		**Parameter of position monitoring is faulty. Invalid value in <Axis name>.PositioningMonitoring.Window**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8627		**The configuration of the PROFIdrive interface of the actual value is faulty. Invalid value in <Axis name>.Sensor.Sensor[1].Interface.AddressIn or <Axis name>.Sensor.Sensor[1].Interface.AddressOut**	
	16#0011	The selection value is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8628		**Faulty controller factors**	
	16#0030	Value has an incorrect number format or is outside the valid number range	The value for the gain or the precontrol of the control loop is faulty. • Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary (<Axis name>.PositionControl.Kv, <Axis name>.PositionControl.Kpc)
16#8629		**Limit for standstill signal is faulty. Invalid value in <Axis name>.StandStillSignal.VelocityThreshold**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#862A		**Parameter of position monitoring is faulty. Invalid value in <Axis name>.PositioningMonitoring.ToleranceTime**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary

ErrorID	ErrorInfo	Description	Remedy
16#862B		**Inconsistent PROFIBUS parameterization; the sum of Ti and To is greater than one DP cycle**	
	16#0030	Value has an incorrect number format or is outside the valid number range	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#862C		**Parameter of standstill monitoring is faulty. Invalid value in <Axis name>.StandStillSignal.MinDwellTime**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#862D		**Parameter of following error monitoring is faulty. Invalid value in <Axis name>.FollowingError.MinValue**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#862E		**Invalid value for the configuration date <Axis name>.Modulo.Length**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#862F		**Invalid value for the configuration date <Axis name>.Modulo.StartValue**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8630		**Invalid value for the configuration date <Axis name>.Actor.DriveParameter.ReferenceSpeed**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8631		**The set fine resolution for Gn_XIST2 is invalid. Invalid value in <Axis name>.Sensor.Sensor[1].Parameter.FineResolutionXist2**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary

ErrorID	ErrorInfo	Description	Remedy
16#8632		**The number of determinable encoder revolutions is invalid. Invalid value in <Axis name>.Sensor.Sensor[1].Parameter.DeterminableRevolutions**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8634		**Parameter of the following error monitoring is faulty. Invalid value in <Axis name>.FollowingError.MaxValue**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8635		**Parameter of the following error monitoring is faulty. Invalid value in <Axis name>.FollowingError.MinVelocity**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8636		**Controller factor is incorrect. Invalid value of the precontrol factor <Axis name>.PositionControl.Kpc**	
	16#0030	Value has an incorrect number format or is outside the valid number range	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online; acknowledge error with instruction "MC_Reset" and restart the command, if necessary
16#8637		**Invalid value for the configuration date <Axis name>.Sensor.Sensor[1].Interface.Type**	
	16#0011	The selection value is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8638		**Invalid value for the configuration date <Axis name>.Sensor.Sensor[1].Interface.HSC**	
	16#0011	The selection value is invalid	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
16#8639		**Error at the drive**	
	16#0049	Configuration error at device	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#004A	The technology needs a smaller servo clock.	Internal system error. Check the project for consistency and reload it into the controller.
	16#004B	Device driver not initialized during ramp-up.	To enable a technology object, the actuator driver must be initialized. Execute the command again later.

ErrorID	ErrorInfo	Description	Remedy
16#863A		**Communication to the drive is faulty**	
	16#004C	Configuration error at device	• Connect a suitable device. • Check the device (I/Os). • Compare the configuration of HW Config and the technology object.
	16#004D	The device driver needs a smaller servo clock.	• Connect a suitable device. • Check the device (I/Os). • Compare the configuration of HW Config and the technology object.
	16#004E	Error in internal communication with the device	Check the project for consistency and reload it into the controller.
16#863B		**Error at encoder**	
	16#0049	Configuration error at device	Download error-free configuration to the controller; enable the axis again with instruction "MC_Power"
	16#004A	The technology needs a smaller servo clock.	Internal system error. Check the project for consistency and reload it into the controller.
	16#004B	Device driver not initialized during ramp-up.	To enable a technology object, the actuator driver must be initialized. Execute the command again later.
16#863C		**Communication with encoder is faulty**	
	16#004C	Configuration error at device	• Connect a suitable device. • Check the device (I/Os). • Compare the configuration of HW Config and the technology object.
	16#004D	The device driver needs a smaller servo clock.	• Connect a suitable device. • Check the device (I/Os). • Compare the configuration of HW Config and the technology object.
	16#004E	Error in internal communication with the device	Check the project for consistency and reload it into the controller.
16#863D		**Communication to the device (drive or encoder) is faulty**	
	16#004F	The requested logical address is invalid.	• Connect a suitable device. • Check the device (I/Os). • Check the topological configuration in HW Config. • Compare the configuration of HW Config and the technology object.
	16#0050	The requested logical output address is invalid.	
	16#0051	The requested logical output address is invalid.	
16#863E		**Value of variable "ControlPanel.Input.TimeOut" is invalid (axis control panel)**	
	16#0030	Value has an incorrect number format or is outside the valid number range	Correct the value in the variables of the technology object <Axis name>.ControlPanel.Input.TimeOut. The value is specified in milliseconds (ms).

ErrorID	ErrorInfo	Description	Remedy
16#863F		**Invalid value for the configuration date <Axis name>.Actor.DriveParameter.MaxSpeed**	
	16#0030	Value has an incorrect number format or is outside the valid number range	Correct the reference value in the drive and in the configuration of the technology object to Actuator.MaxSpeed/2. With analog drive connection, correct the reference value in the drive and in the configuration of the technology object to Actuator.MaxSpeed/1.17.
16#8640		**Error with automatic transfer of drive parameters in the device**	
	16#0030	Value has an incorrect number format or is outside the valid number range	Correct the value.
	16#0059	The device is not assigned to any SINAMICS drive unit or does not support the services required for the adaptation.	Disable automatic transfer of the parameters in the device. Complete the required parameters in the axis configuration and reload the axis configuration in the device.
	16#005A	Automatic transfer of parameters canceled due to insufficient resources.	
	16#005B	The parameters can only be automatically transferred when the device is connected directly to an I/O area.	
	16#005C	Maximum speed/velocity (p1082): Either the parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	Check the causes. Disable automatic transfer of the parameters in the device if you cannot eliminate the causes. Complete the required parameters in the axis configuration and reload the axis configuration in the device.
	16#005D	Maximum torque/force (p1520): Either the parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	
	16#005E	Maximum torque/force (p1521): Either the parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	
	16#005F	Fine resolution torque/force limiting (p1544): Either the parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	
	16#0060	Nominal speed/velocity (p2000): Either the parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	

ErrorID	ErrorInfo	Description	Remedy
	16#0061	Nominal torque/force (p2003): Either the parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	
16#8641		**Error with automatic transfer of encoder parameters in the device**	
	16#0030	Value has an incorrect number format or is outside the valid number range	Correct the value.
	16#0059	The device is not assigned to any SINAMICS drive unit or does not support the services required for the adaptation.	Disable automatic transfer of the parameters in the device. Complete the required parameters in the axis configuration and reload the axis configuration in the device.
	16#005A	Automatic transfer of parameters canceled due to insufficient resources.	
	16#005B	The parameters can only be automatically transferred when the device is connected directly to an I/O area.	
	16#0062	Encoder system (r0979[1/11].0): Either a parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	Check the causes. Disable automatic transfer of the parameters in the device if you cannot eliminate the causes. Complete the required parameters in the axis configuration and reload the axis configuration in the device.
	16#0063	Encoder resolution (r0979[2/12]): Either a parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	
	16#0064	Encoder fine resolution Gx_XIST1 (r0979[3/13]): Either a parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	
	16#0065	Encoder fine resolution Gx_XIST2 (r0979[4/14]): Either a parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	
	16#0066	Number of determinable encoder revolutions (r0979[5/15]): Either a parameter does not exist or its value cannot be read or is outside the permitted limits. Reading of parameters has been aborted due to an error signaled by the hardware.	

ErrorID	ErrorInfo	Description	Remedy
16#8642		**Configuration is internally adapted**	
	16#0067	1: Illegal value for Actor.MaxSpeed (Actor.MaxSpeed greater than 2*Actor.ReferenceSpeed) Remedy: set e.g. P2000 = P1082 in drive.	Adjust values.
16#8643		**Inconsistency between the configuration of the axis and the drive configuration**	
	16#0068	The configured telegram type is not compatible with the telegram type on the device (P922 or P2079).	Synchronize the telegram type in the configuration of the axis and in the configuration of the drive.
	16#0069	A function module with linear motor is set in the drive.	Set rotation motor in drive.
16#8644		**Inconsistency between the configuration of the axis and the encoder configuration**	
	16#0068	The configured telegram type is not compatible with the telegram type on the device (P922 or P2079).	Synchronize the telegram type in the configuration of the axis and in the configuration of the encoder.
	16#006A	The encoder on the drive is not an absolute encoder (P979).	Synchronize the encoder type in the configuration of the axis and in the encoder configuration.
16#8645		**The maximum velocity cannot be reached with the configured drive and axis parameters.**	
	16#0001	-	Adapt drive or axis configuration and load configuration in the device.
16#8646		**Illegal value in Sensor.Interface.Number**	
	16#0030	Value has an incorrect number format or is outside the valid number range	Correct the value and load configuration in the device.
16#8647		**Simulation is not supported for PTO axes**	
	16#0001	-	Disable simulation

Configuration error of the command table

ErrorID	ErrorInfo	Description	Remedy
16#8700		**Value for "Command type" in the command table is invalid**	
	16#0001	-	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online and restart the command, if necessary
16#8701		**Value for "Position / travel path" in the command table is invalid**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online and restart the command, if necessary
	16#0005	Value is outside the number range (greater than 1.0E12)	
	16#0006	Value is outside the number range (less than -1.0E12)	
16#8702		**Value for "Velocity" in the command table is invalid**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online and restart the command, if necessary
	16#0008	Value is greater than the configured maximum velocity	
	16#0009	Value is less than the configured start/stop velocity	
16#8703		**Value for "Duration" in the command table is invalid**	
	16#0002	The value has an invalid number format	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online and restart the command, if necessary
	16#0021	Value is greater than 64800 s	
	16#0022	Value is less than 0.001 s	
16#8704		**Value for "Next step" in the command table is invalid**	
	16#0011	The selection value is invalid	• Download error-free configuration to the controller; enable the axis again with instruction "MC_Power" • Correct the incorrect value online and restart the command, if necessary
	16#0023	The command transition is not permitted for this command	

Internal errors

ErrorID	ErrorInfo	Description	Remedy
16#8FFF		**Internal error**	
	16#F0**	-	POWER OFF and POWER ON the CPU If this does not work, contact Customer Support. Have the following information ready: • ErrorID • ErrorInfo • Diagnostic buffer entries

12.9 Tags of the positioning axis technology object as of V6

12.9.1 Legend

Tag	Name of the tag	
Data type	Data type of the tag	
Values	Start value (Value range of the tag - minimum value to maximum value) If no specific value is shown, the value range limits of the relevant data type apply or the information under "Description".	
Access	Access to the tag in the user program	
	OPR	The tag can be read by the Openness application.
	OPRW	The tag can be read and written by the Openness application.
	R	The tag can be read in the user program and in the HMI.
	RCCP	The tag can be read in the user program and in the HMI and is updated at each cycle control point.
	RP	The variable can be read with the Motion Control instruction "MC_ReadParam". The current value of the corresponding variables is determined at the start of the command.
	RW	The tag can be read and written in the user program and in the HMI. The variable can be written with Motion Control instruction "MC_WriteParam".
	WD BL	For drive connection via PROFIdrive/analog output: The tag can be written to the start value in the load memory with the extended instruction "WRIT_DBL".
	WP	Independent of the drive connection: If the axis is disabled (MC_Power.Status = FALSE), the tag can be written with the Motion Control instruction "MC_WriteParam".
	WP_PD	For drive connection via PROFIdrive/analog output: If the axis is disabled (MC_Power.Status = FALSE), the tag can be written with the Motion Control instruction "MC_WriteParam".
	WP_PTO	For drive connection via PTO: If the axis is disabled (MC_Power.Status = FALSE), the tag can be written with the Motion Control instruction "MC_WriteParam".
	-	The variable cannot be used in the user program.
W	Effectiveness of changes in the technology data block	
	1	For drive connection via PTO: When axis is activated, disabled, or enabled
	2	For drive connection via PTO: When axis is enabled
	5	For drive connection via PTO: The next time an MC_MoveAbsolute, MC_MoveRelative, MC_MoveVelocity, MC_MoveJog, MC_Halt, MC_CommandTable, or active MC_Home command is started (Mode = 3)
	6	For drive connection via PTO: When a MC_MoveJog command is stopped
	7	For drive connection via PTO: When a passive homing command is started
	8	For drive connection via PTO: When an active homing command is started
	9	With the restart of the technology object
	10	For drive connection via PROFIdrive/analog output: With the next call of the MC-Servo [OB91]
Description	Description of the tag	

Access to the tags is with "<TO>.<tag name>". The placeholder <TO> represents the name of the technology object.

12.9.2 Tags for position values and velocity values as of V6

The tag structure contains the setpoint and actual values of the position and the velocity of the axis.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
Position	REAL	0.0 (-9.0E15 to 9.0E15)	RCCP, RP, OPR	-	Setpoint position of the axis (indicated in the configured unit of measurement) If the axis is not homed, the tag indicates the position value relative to the enable position of the axis. Name in Openness: Position
Velocity	REAL	0.0	RCCP, RP, OPR	-	Velocity setpoint of the axis (indicated in the configured unit of measurement) Name in Openness: Velocity
ActualPosition	REAL	0.0 (-9.0E15 to 9.0E15)	RCCP, RP, OPR	-	Actual position of the axis (indicated in the configured unit of measurement) If the axis is not homed, the tag indicates the position value relative to the enable position of the axis. Name in Openness: ActualPosition
ActualVelocity	REAL	0.0 (-9.0E15 to 9.0E15)	RCCP, RP, OPR	-	Actual velocity of the axis (indicated in the configured unit of measurement) Name in Openness: ActualVelocity

See also

Motion status (Page 179)

12.9.3 Simulation tags as of V6

The tag structure <axis name>.Simulation.Mode contains the simulation mode.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description	
Simulation.						
Mode	UDINT	0 (0 to 1)	R WDBL OPRW	- 2, 9	Simulation mode Name in Openness: Simulation.Mode	
					0	No simulation, normal operation
					1	Simulation mode In simulation mode, you can simulate axes without a real drive in the CPU.

12.9.4 Actuator tags as of V6

The tag structure <axis name>.Actor.<tag name> contains the drive parameters.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
Actor.	STRUCT				TO_Struct_Actor
Type	DINT	2 (0 to 2)	R OPRW	-	Name in Openness: Actor.Type
					Positioning axis technology object as of V5:
					0 The drive is connected via an analog output. All movements of the axis are position-controlled.
					1 The drive is connected via PROFIdrive telegrams. All movements of the axis are position-controlled.
					2 The drive is connected via a pulse interface.
					Positioning axis technology object V4:
					2 The drive is connected via a pulse interface.
InverseDirection	BOOL	FALSE	R WP_PTO OPRW	- 2	Name in Openness: Actor.InverseDirection
					FALSE The direction is not inverted.
					TRUE The direction is inverted.
DirectionMode	INT	0 (0 to 2)	R WP_PTO OPRW	- 2	Permitted direction of rotation Name in Openness: Actor.DirectionMode
					0 Both directions
					1 Positive direction
					2 Negative direction
DataAdaption	DINT	0 (0 to 1)	R OPRW	-	Name in Openness: Actor.DataAdaption
					0 Automatic transfer of drive parameters in the device is disabled.
					1 Automatic transfer of drive parameters in the device is enabled.
Interface.	STRUCT				TO_Struct_ActorInterface
AddressIn.	VREF	-	-	-	Input address for the PROFIdrive telegram (internal parameter)
AREA	BYTE	-	OPR	-	Name in Openness: Actor.Interface.AddressIn.AREA
DB_NUMBER	UINT	-	OPR	-	Name in Openness: Actor.Interface.AddressIn.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Name in Openness: Actor.Interface.AddressIn.OFFSET

Tag	Data type	Values	Access	W	Description
RID	DWORD	-	OPR	-	Name in Openness: Actor.Interface.AddressIn.RID
ProfiDriveIn*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.ProfiDriveIn Valid input address, which is the part of a telegram Valid tag name
AddressOut.	VREF	-	-	-	Output address for the PROFIdrive telegram (internal parameter)
AREA	BYTE	-	OPR	-	Name in Openness: Actor.Interface.AddressOut.AREA
DB_NUMBER	UINT	-	OPR	-	Name in Openness: Actor.Interface.AddressOut.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Name in Openness: Actor.Interface.AddressOut.OFFSET
RID	DWORD	-	OPR	-	Name in Openness: Actor.Interface.AddressOut.RID
ProfiDriveOut*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.ProfiDriveOut Valid output address, which is the part of a telegram Valid tag name
DataBlock*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.DataBlock Valid data block address:
Analog*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.Analog Valid analog output, valid data block address, valid tag name
DataConnection*	INT	0 (0 to 1)	OPRW	-	Name in Openness: _Actor.Interface.DataConnection 0 Drive 1 Data block
EnableDriveOutput	VREF	-	-	-	Enable output (internal parameter)
AREA	BYTE	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyOutput.AREA
DB_NUMBER	UINT	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyOutput.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyOutput.OFFSET
RID	DWORD	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyOutput.RID
EnableDriveOutput*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.EnableDriveOutput Valid input, valid output, valid memory address, valid tag name
DriveReadyInput	VREF	-	-	-	Ready input (internal parameter)

Tag	Data type	Values	Access	W	Description
AREA	BYTE	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyInput.AREA
DB_NUMBER	UINT	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyInput.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyInput.OFFSET
RID	DWORD	-	OPR	-	Name in Openness: Actor.Interface.DriveReadyInput.RID
DriveReady Input*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.DriveReadyInput Valid input, valid output, valid memory address, valid tag name
PTO	DWORD	0	OPR	-	Pulse output (internal parameter)
PTO*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.PTO Name of pulse generators as in the hardware configuration
PTO_OutputA*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.PTO_OutputA Valid input address, valid tag name Only onboard CPU or Dignalboard addresses are accepted.
PTO_OutputB Enable*	BOOL		OPRW	-	Name in Openness: _Actor.Interface.PTO_OutputBEnable Only possible with PTO_SignalType = 2
					FALSE: Output B is disabled.
					TRUE: Output B is enabled.
PTO_OutputB*	STRING	-	OPRW	-	Name in Openness: _Actor.Interface.PTO_OutputB Valid input address, valid tag name Only onboard CPU or Dignalboard addresses are accepted.
PTO_Signal Type*	INT	(2 to 5)	OPRW	-	Name in Openness: _Actor.Interface.PTO_SignalType
					2: Pulse A and direction B
					3: Clock up A and clock down B
					4: A/B phase-shifted
					5: A/B phase-shifted - quadruple
DriveParameter.	STRUCT				TO_Struct_ActorDriveParameter

Tag			Data type	Values	Access	W	Description
		Reference Speed	REAL	3000.0	R OPRW	-	Reference value (100 %) for the speed setpoint (NSET) of the drive The setpoint speed is transmitted in the PROFIdrive telegram as a standardized value from -200% to 200% of the "ReferenceSpeed". For setpoint specification via an analog output, the analog output can be operated in the range from -117 % to 117 %, provided the drive permits this. Name in Openness: Actor.DriveParameter.ReferenceSpeed
		MaxSpeed	REAL	3000.0	R OPRW	-	Maximum value for the speed setpoint of the drive (NSET) (PROFIdrive: MaxSpeed ≤ 2 × ReferenceSpeed Analog setpoint: MaxSpeed ≤ 1.17 × ReferenceSpeed) Name in Openness: Actor.DriveParameter.MaxSpeed
		PulsesPerDrive Revolution	DINT	1000 (1 to 2147483648)	R WP_PTO OPRW	- 2	Pulses per motor revolution Name in Openness: Actor.DriveParameter.PulsesPerDriveRevolution

*) Available in Openness

12.9.5 Sensor[1] tags as of V6

The tag structure <axis name>.Sensor[1].<tag name> contains the encoder parameters.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
Sensor[1].	STRUCT				ARRAY[1..1] TO_Struct_Sensor
Type	DINT	0 (0 to 1)	R, OPRW	-	Encoder type (internal parameter) Name in Openness: Sensor[1].Type
				0	Incremental
				1	Absolute
InverseDirection	BOOL	FALSE	R, OPRW	-	Inversion of the actual value Name in Openness: Sensor[1].InverseDirection
				FALSE	Actual value is not inverted
				TRUE	Actual value is inverted
System	DINT	1 (0 to 1)	R, OPRW	-	Encoder system Name in Openness: Sensor[1].System
				0	Linear encoder
				1	Rotary encoder
MountingMode	DINT	0 (0 to 2)	R, OPRW	-	Mounting type of encoder Name in Openness: Sensor[1].MountingMode
				0	On the motor shaft
				2	External measuring system
DataAdaption	DINT	0 (0 to 1)	R, OPRW	-	Name in Openness: Sensor[1].DataAdaption
				0	Automatic transfer of encoder parameters in the device is disabled.
				1	Automatic transfer of encoder parameters in the device is enabled.
Interface.	STRUCT				TO_Struct_SensorInterface
Type	DINT	4 (0 to 4)	OPR	-	Encoder connection (internal parameter) Name in Openness: Sensor[1].Interface.Type
				0	PROFIdrive encoder on PROFINET
				1	Encoder on technology module (TM)
				2	Encoder on the drive
				4	Encoder on high-speed counter
AddressIn.	VREF	-	-	-	Input address for the PROFIdrive telegram (internal parameter)

Tag	Data type	Values	Access	W	Description
AREA	BYTE	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressIn.AREA
DB_NUMBER	UINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressIn.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressIn.OFFSET
RID	DWORD	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressIn.RID
ProfiDriveIn*	STRING	-	OPRW	-	Name in Openness: _Sensor[1].Interface.ProfiDriveIn Valid input address, which is the part of a telegram Valid tag name
AddressOut.	VREF	-	-	-	Output address for the PROFIdrive telegram (internal parameter)
AREA	BYTE	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressOut.AREA
DB_NUMBER	UINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressOut.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressOut.OFFSET
RID	DWORD	-	OPR	-	Internal parameter Name in Openness: Sensor[1].Interface.AddressOut.RID
ProfiDriveOut*	STRING	-	OPRW	-	Name in Openness: _Sensor[1].Interface.ProfiDriveOut Valid output address, which is the part of a telegram Valid tag name
DataBlock*	STRING	-	OPRW	-	Name in Openness: _Sensor[1].Interface.DataBlock Valid data block address:
DataConnec-tion*	UDINT	(0 to 1)	OPRW	-	Name in Openness: _Sensor[1].Interface.DataConnection
					0 Encoder
					1 Data block
EncoderCon-nection*	INT	(4 to 7)	OPRW	-	Name in Openness: _Sensor[1].Interface.EncoderConnection
					4 Encoder on high-speed counter (HSC)
					7 Encoder on PROFIBUS/PROFINET

Tag	Data type	Values	Access	W	Description
Number	UDINT	1	OPRW	-	Encoder number Name in Openness: Sensor[1].Interface.Number
HSC	DWORD	0	OPR	-	High-speed counter to which the encoder transfers the actual value (internal parameter)
HSC*	STRING	-	OPRW	-	Name in Openness: _Sensor[1].Interface.HSC Name of fast counters from the hardware configuration
HSC_Operating Mode*	INT	(1 to 3)	OPRW	-	Name in Openness: _Sensor[1].Interface.HSC_OperatingMode
					1 Two-phase
					2 A/B counter
					3 A/B counter quadruple
HSC_InputA*	STRING	-	OPRW	-	Name in Openness: _Sensor[1].Interface.HSC_InputA Valid input address, valid tag name
HSC.InputB*	STRING	-	OPRW	-	Name in Openness: _Sensor[1].Interface.HSC.InputB Valid input address, valid tag name
Parameter.	STRUCT				TO_Struct_SensorParameter
Resolution	REAL	0.001 (-1.0E12 to 1.0E12)	R, OPRW	-	Resolution of a linear encoder (offset between two encoder pulses) Name in Openness: Sensor[1].Parameter.Resolution
StepsPer Revolution	UDINT	2048 (1 to 8388608)	R, OPRW	-	Increments per rotary encoder revolution Name in Openness: Sensor[1].Parameter.StepsPerRevolution
FineResolution Xist1	UDINT	11 (0 to 31)	R, OPRW	-	Number of bits for fine resolution Gn_XIST1 (cyclic actual encoder value) Name in Openness: Sensor[1].Parameter.FineResolutionXist1
FineResolution Xist2	UDINT	9 (0 to 31)	R, OPRW	-	Number of bits for fine resolution Gn_XIST2 (absolute value of the encoder) Name in Openness: Sensor[1].Parameter.FineResolutionXist2
Determinable Revolutions	UDINT	1 (0 to 8388608)	R, OPRW	-	Number of differentiable encoder revolutions for a multi-turn absolute encoder Name in Openness: Sensor[1].Parameter.DeterminableRevolutions
					0 Incremental encoder
					1 Single return absolute encoder
DistancePer-Revolution	REAL	100.0 (0.0 to 1.0E12)	R, OPRW	-	Load distance per revolution of an externally mounted encoder Name in Openness: Sensor[1].Parameter.DistancePerRevolution
ActiveHoming.	STRUCT				TO_Struct_SensorActiveHoming

Tag	Data type	Values	Access	W	Description
Mode	DINT	2 (0 to 2)	R, WP_PTO, OPRW	- 2	Active homing mode Name in Openness: Sensor[1].ActiveHoming.Mode
					Positioning axis technology object as of V5:
					0 — Zero mark via PROFIdrive telegram (not PTO)
					1 — Zero mark via PROFIdrive telegram and proximity switch (not PTO)
					2 — Homing via digital input
					Positioning axis technology object V4:
					2 — Homing via digital input
SideInput	BOOL	FALSE	RW, WP, OPRW	1, 8, 10	End of the homing switch to which the axis is homed during active homing Name in Openness: Sensor[1].ActiveHoming.SideInput
					FALSE — Bottom side
					TRUE — Top side
DigitalInput Address.	VREF	-	-	-	Symbolic input address of the homing switch (internal parameter)
AREA	BYTE	-	OPR	-	Internal parameter Name in Openness: Sensor[1].ActiveHoming.DigitalInputAddress.AREA
DB_NUMBER	UINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].ActiveHoming.DigitalInputAddress.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].ActiveHoming.DigitalInputAddress.OFFSET
RID	DWORD	-	OPR	-	Internal parameter Name in Openness: Sensor[1].ActiveHoming.DigitalInputAddress.RID
DigitalInput*	STRING	-	OPRW	-	Name in Openness: _Sensor[1].ActiveHoming.DigitalInput Valid input address, valid tag name
HomePosition-Offset	REAL	0.0 (-1.0E12 to 1.0E12)	RW, WP, OPRW	1, 8, 10	Home position offset (active homing) (indicated in the configured unit of measurement) Name in Openness: Sensor[1].ActiveHoming.HomePositionOffset
SwitchLevel	BOOL	TRUE	RW, WP, OPRW	1, 8, 10	Selection of signal level that is present at the CPU input when the homing switch is reached Name in Openness: Sensor[1].ActiveHoming.SwitchLevel
					FALSE — Low level (Low active)
					TRUE — High level (high-enabled)

Tag	Data type	Values	Access	W	Description
PassiveHoming.	STRUCT				TO_Struct_SensorPassiveHoming
DigitalInput Address.	VREF	-	-	-	Symbolic input address of the homing switch (internal parameter)
AREA	BYTE	-	OPR	-	Internal parameter Name in Openness: Sensor[1].PassiveHoming.DigitalInputAddress.AREA
DB_NUMBER	UINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].PassiveHoming.DigitalInputAddress.DB_NUMBER
OFFSET	UDINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].PassiveHoming.DigitalInputAddress.OFFSET
RID	DWORD	-	OPR	-	Internal parameter Name in Openness: Sensor[1].PassiveHoming.DigitalInputAddress.RID
Mode	DINT	2 (0 to 2)	R WP_PTO OPRW	- 2	Passive homing mode Name in Openness: Sensor[1].PassiveHoming.Mode
					Positioning axis technology object as of V5:
					0 — Zero mark via PROFIdrive telegram (not PTO)
					1 — Zero mark via PROFIdrive telegram and proximity switch (not PTO)
					2 — Homing via digital input
					Positioning axis technology object V4:
					2 — Homing via digital input
SideInput	BOOL	FALSE	RW, WP, OPRW	1, 7, 10	End of the homing switch to which the axis is homed during passive homing Name in Openness: Sensor[1].PassiveHoming.SideInput
					FALSE — Bottom side
					TRUE — Top side
DigitalInput Address.	VREF	-	-	-	Symbolic input address of the homing switch (internal parameter)
AREA	BYTE	-	OPR	-	Internal parameter Name in Openness: Sensor[1].PassiveHoming.DigitalInputAddress.AREA
DB_NUMBER	UINT	-	OPR	-	Internal parameter Name in Openness: Sensor[1].PassiveHoming.DigitalInputAddress.DB_NUMBER

Tag	Data type	Values	Access	W	Description	
OFFSET	UDINT	-	OPR	-	Internal parameter Name in Openness: Sen- sor[1].PassiveHoming.DigitalInputAddress.OFFSET	
RID	DWORD	-	OPR	-	Internal parameter Name in Openness: Sensor[1].PassiveHoming.DigitalInputAddress.RID	
DigitalInput*	STRING		OPRW	-	Name in Openness: _Sensor[1].PassiveHoming.DigitalInput Valid input address, valid tag name	
SwitchLevel	BOOL	TRUE	RW, WP, OPRW	1, 7, 10	Selection of level that is present at the CPU input when the homing switch is reached Name in Openness: Sensor[1].PassiveHoming.SwitchLevel	
					FALSE	Low level (Low active)
					TRUE	High level (high-enabled)

*) Available in Openness

12.9.6 Units tag as of V6

The tag structure <axis name>.Units.LengthUnit contains the configured units of measurement of the parameters.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description	
Units.	STRUCT				TO_Struct_Units	
LengthUnit	INT	1013 (-32768 to 32767)	R WP_PTO OPRW	- 2	Configured unit of measurement of the parameter Name in Openness: Units.LengthUnit	
					-1	Pulses
					1005	°
					1010	m
					1013	mm
					1018	ft
					1019	in

12.9.7 Mechanics tag as of V6

The tag structure <axis name>.Mechanics.LeadScrew contains the distance traveled per motor revolution.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
Mechanics.	STRUCT				TO_Struct_Mechanics
LeadScrew	REAL	10.0 (-1.0E12 to 1.0E12)	R, WP_PTO, OPRW	-	Distance per revolution (indicated in the configured unit of measurement) Name in Openness: Mechanics.LeadScrew

12.9.8 Modulo tags as of V6

The tag structure <axis name>.Modulo.<tag name> contains the modulo settings.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description	
Modulo.	STRUCT				TO_Struct_Modulo	
Enable	BOOL	FALSE	R, OPRW	-	Name in Openness: Modulo.Enable	
					FALSE	Modulo conversion disabled
					TRUE	Modulo conversion enabled
					When modulo conversion is enabled, a check is made for modulo length > 0.0	
Length	REAL	360.0 (0.001 to 1.0E12)	R, OPRW	-	Modulo length Name in Openness: Modulo.Length	
StartValue	REAL	0.0 (-1.0E12 to 1.0E12)	R, OPRW	-	Modulo start value Name in Openness: Modulo.StartValue	

12.9.9 DynamicLimits tags as of V6

The tag structure <axis name>.DynamicLimits.<tag name> contains the configuration of the dynamic limits.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
DynamicLimits.	STRUCT				TO_Struct_DynamicLimits
MaxVelocity	REAL	250.0	R WP_PTO OPRW	- 2	Maximum velocity of the axis (indicated in the configured unit of measurement) Name in Openness: DynamicLimits.MaxVelocity
MinVelocity	REAL	10.0	R WP_PTO OPRW	- 2	Start/stop velocity of the axis (indicated in the configured unit of measurement) Name in Openness: DynamicLimits.MinVelocity

12.9.10 DynamicDefaults tags as of V6

The tag structure <axis name>.DynamicDefaults.<tag name> contains the configuration of the dynamic defaults.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
DynamicDefaults.	STRUCT				TO_Struct_DynamicDefaults
Acceleration	REAL	48.0 (0.0 to 1.0E12)	RW, WP, OPRW	5, 6, 10	Default setting of the acceleration of the axis (indicated in the configured unit of measurement) Name in Openness: DynamicDefaults.Acceleration
Deceleration	REAL	48.0 (0.0 to 1.0E12)	RW, WP, OPRW	5, 6, 10	Default decelaration of the axis (indicated in the configured unit of measurement) Name in Openness: DynamicDefaults.Deceleration
Jerk	REAL	192.0 (0.0 to 1.0E12)	RW, WP, OPRW	5, 10	Setting the jerk default during acceleration and deceleration ramp of the axis (indicated in the configured unit of measurement) The jerk is activated if the configured jerk is greater than 0.00004 mm/s². Name in Openness: DynamicDefaults.Jerk
EmergencyDeceleration	REAL	120.0 (0.0 to 1.0E12)	RW, WP, OPRW	1, 5, 6, 10	Emergency stop deceleration of the axis (indicated in the configured unit of measurement) Name in Openness: DynamicDefaults.EmergencyDeceleration

12.9.11 PositionLimits_SW variables as of V6

The tag structure <axis name>.PositionLimits_SW.<tag name> contains the configuration for position monitoring with software limit switches. Software limit switches are used to limit the operating range of a positioning axis.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
PositionLimits_SW.	STRUCT				TO_Struct_PositionLimitsSW
Active	BOOL	FALSE	RW, WP, OPRW	1, 5, 6, 10	Name in Openness: PositionLimits_SW.Active
					FALSE: The software limit switches are deactivated.
					TRUE: The software limit switches are activated.
MinPosition	REAL	-10000.0 (-1.0E12 to 1.0E12)	RW, WP, OPRW	1, 5, 6, 10	Position of the software low limit switches (specification in the configured measurement unit) Name in Openness: PositionLimits_SW.MinPosition
MaxPosition	REAL	10000.0 (-1.0E12 to 1.0E12)	RW, WP, OPRW	1, 5, 6, 10	Position of the software high limit switch (indicated in the configured unit of measurement) Name in Openness: PositionLimits_SW.MaxPosition

12.9.12 PositionLimits_HW variables as of V6

The tag structure <axis name>.PositionLimits_HW.<tag name> contains the configuration for position monitoring with hardware limit switches. Hardware limit switches are used to limit the traversing range of a positioning axis.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
PositionLimits_HW.	STRUCT				TO_Struct_PositionLimitsHW
Active	BOOL	FALSE	RW, WP, OPRW	1, 5, 6, 10	Name in Openness: PositionLimits_HW.Active
					FALSE: The hardware limit switches are deactivated.
					TRUE: The hardware limit switches are activated.
MinSwitchLevel	BOOL	FALSE	RW WP_PTO	- 2	Selection of signal level that is present at the CPU input when the hardware low limit switch is reached Name in Openness: PositionLimits_HW.MinSwitchLevel
					FALSE: Low level (Low active)
					TRUE: High level (high-enabled)
MinSwitchAddress.	VREF	-	-	-	Symbolic input address of the hardware low limit switch (internal parameter)
AREA	BYTE	-	OPR	-	Name in Openness: PositionLimits_HW.MinSwitchAddress.AREA
DB_NUMBER	USHORT	-	OPR	-	Name in Openness: PositionLimits_HW.MinSwitchAddress.DB_NUMBER
OFFSET	UINT	-	OPR	-	Name in Openness: PositionLimits_HW.MinSwitchAddress.OFFSET
RID	UINT	-	OPR	-	Name in Openness: PositionLimits_HW.MinSwitchAddress.RID
MinSwitch*	STRING	-	OPRW	-	Name in Openness: _PositionLimits_HW.MinSwitch Valid input address, valid tag name
MaxSwitchLevel	BOOL	FALSE	RW WP_PTO	- 2	Selection of signal level that is present at the CPU input when the hardware high limit switch is reached Name in Openness: PositionLimits_HW.MaxSwitchLevel
					FALSE: Low level (Low active)
					TRUE: High level (high-enabled)
MaxSwitch Address.	VREF	-	-	-	Input address of the hardware high limit switch (internal parameter)
AREA	BYTE	-	OPR	-	Name in Openness: PositionLimits_HW.MaxSwitchAddress.AREA
DB_NUMBER	USHORT	-	OPR	-	Name in Openness: PositionLimits_HW.MaxSwitchAddress.DB_NUMBER

Tag	Data type	Values	Access	W	Description
OFFSET	UINT	-	OPR	-	Name in Openness: PositionLimits_HW.MaxSwitchAddress.OFFSET
RID	UINT	-	OPR	-	Name in Openness: PositionLimits_HW.MaxSwitchAddress.RID
MaxSwitch*	STRING	-	OPRW	-	Name in Openness: _PositionLimits_HW.MaxSwitch Valid input address, valid tag name

*) Available in Openness

12.9.13 Homing tags as of V6

The tag structure <axis name>.Homing.<tag name> contains the configuration for homing the axis.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description	
Homing.	STRUCT				TO_Struct_Homing	
AutoReversal	BOOL	FALSE	RW, WP, OPRW	1, 8, 10	Name in Openness: Homing.AutoReversal	
					FALSE	Auto reversal at the hardware limit switch is deactivated.
					TRUE	Auto reversal at the hardware limit switch is activated.
ApproachDirection	BOOL	TRUE	RW, WP, OPRW	1, 8, 10	Name in Openness: Homing.ApproachDirection	
					FALSE	Negative approach direction for finding the homing switch and negative homing direction
					TRUE	Positive approach direction to search for reference point switch and positive homing direction
ApproachVelocity	REAL	200.0 (0.0 to 1.0E12)	RW, WP, OPRW	1, 8, 10	Approach velocity of the axis during active homing (indicated in the configured unit of measurement) Name in Openness: Homing.ApproachVelocity	
Referencing Velocity	REAL	40.0 (0.0 to 1.0E12)	RW, WP, OPRW	1, 8, 10	Homing velocity of the axis during active homing (indicated in the configured unit of measurement) Name in Openness: Homing.ReferencingVelocity	

12.9.14 PositionControl tag as of V6

The tag structure <axis name>.PositionControl.<tag name> contains the position control settings.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
PositionControl.	STRUCT				TO_Struct_PositionControl
Kv	REAL	10.0 (0.0 to 2147480.0)	R WP_PD OPRW	- 10	Proportional gain of the closed loop position control ("Kv" > 0.0) Name in Openness: PositionControl.Kv
Kpc	REAL	100.0 (0.0 to 150.0)	R WP_PD OPRW	- 10	Velocity precontrol of the position control as a percentage Name in Openness: PositionControl.Kpc

12.9.15 FollowingError tags as of V6

The tag structure <axis name>.FollowingError.<tag name> contains the configuration of the dynamic following error monitoring.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
FollowingError.	STRUCT				TO_Struct_FollowingError
EnableMonitoring	BOOL	TRUE	R OPRW	-	Name in Openness: FollowingError.EnableMonitoring
					FALSE: Following error monitoring deactivated
					TRUE: Following error monitoring enabled
MinValue	REAL	10.0 (0.0 to 1.0E12)	R WP_PD OPRW	- 10	Permissible following error at velocities below the value of "MinVelocity". Name in Openness: FollowingError.MinValue
MaxValue	REAL	100.0 (0.0 to 1.0E12)	R WP_PD OPRW	- 10	Maximum permissible following error, which may be reached at the maximum velocity. Name in Openness: FollowingError.MaxValue
MinVelocity	REAL	10.0 (0.0 to 1.0E12)	R WP_PD OPRW	- 10	"MinValue" is permissible below this velocity and is held constant. Name in Openness: FollowingError.MinVelocity

12.9.16 PositionMonitoring tags as of V6

The tag structure <axis name>.PositionMonitoring.<tag name> contains the configuration for position monitoring at the end of a positioning motion.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
PositionMonitoring.	STRUCT				TO_Struct_PositionMonitoring
ToleranceTime	REAL	1.0 (0.0 to 1.0E12)	R WP_PD OPRW	- 10	Tolerance time Maximum permitted duration from reaching velocity setpoint zero, until entrance into the positioning window. Name in Openness: PositionMonitoring.ToleranceTime
MinDwellTime	REAL	0.1 (0.0 to 1.0E12)	R WP_PD OPRW	- 10	Minimum dwell time in positioning window Name in Openness: PositionMonitoring.MinDwellTime
Window	REAL	1.0 (0.001 to 1.0E12)	R WP_PD OPRW	- 10	Positioning window Name in Openness: PositionMonitoring.Window

12.9.17 StandstillSignal tags as of V6

The tag structure <axis name>.StandstillSignal.<tag name> contains the configuration for the standstill signal.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
StandstillSignal.	STRUCT				TO_Struct_StandstillSignal
VelocityThreshold	REAL	5.0 (0.0 to 1.0E12)	R WP_PD OPRW	- 10	Velocity threshold If velocity is below this threshold, then the minimum dwell time begins. Name in Openness: StandStillSignal.VelocityThreshold
MinDwellTime	REAL	0.01 (0.0 to 1.0E12)	R WP_PD OPRW	- 10	Minimum dwell time Name in Openness: StandStillSignal.MinDwellTime

12.9.18 StatusPositioning tags as of V6

The tag structure <axis name>.StatusPositioning.<tag name> indicates the status of a positioning motion.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
StatusPositioning.	STRUCT				TO_Struct_StatusPositioning
Distance	REAL	0.0 (-9.0E15 to 9.0E15)	RCCP, RP, OPR	-	Current distance of axis from target position (indicated in the configured unit of measurement) The value of the tag is only valid during execution of a positioning command with "MC_MoveAbsolute", "MC_MoveRelative", or the axis control panel. Name in Openness: StatusPositioning.Distance
TargetPosition	REAL	0.0 (-9.0E15 to 9.0E15)	RCCP, RP, OPR	-	Target position of the axis (indicated in the configured unit of measurement) The value of the tag is only valid during execution of a positioning command with "MC_MoveAbsolute", "MC_MoveRelative", or the axis control panel. Name in Openness: StatusPositioning.TargetPosition
FollowingError	REAL	0.0 (-9.0E15 to 9.0E15)	RCCP, RP, OPR	-	Current following error of the axis (indicated in the configured unit of measurement) FollowingError = 0.0 for drive connection via PTO (Pulse Train Output). Name in Openness: StatusPositioning.FollowingError

12.9.19 StatusDrive tags as of V6

The tag structure <axis name>.StatusDrive.<tag name> indicates the status of the drive.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description	
StatusDrive.	STRUCT				TO_Struct_StatusDrive	
InOperation	BOOL	FALSE	RCCP, RP, OPR	-	Operational status of the drive Name in Openness: StatusDrive.InOperation	
					FALSE	Drive not ready. Setpoints will not be executed.
					TRUE	Drive ready. Setpoints can be executed.
CommunicationOK	BOOL	FALSE	RCCP, RP, OPR	-	Cyclic BUS communication between controller and drive Name in Openness: StatusDrive.CommunicationOK	
					FALSE	Communication not established
					TRUE	Communication established
AdaptionState	DINT	0 (0 to 4)	R, OPR	10	Transfer status of the drive Name in Openness: StatusDrive.AdaptionState	
					0	Data not transferred
					1	Data in transfer
					2	Data transferred
					3	Transfer not possible or not selected
					4	Error during data transfer

12.9.20 StatusSensor tags as of V6

The tag structure <axis name>.StatusSensor[1].<tag name> indicates the status of the measuring system.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description	
StatusSensor[1].	STRUCT				TO_Struct_StatusSensor	
State	DINT	0 (0 to 2)	RCCP, RP, OPR	-	Status of the encoder value Name in Openness: StatusSensor.State	
					0	Invalid
					1	Waiting for valid status
					2	Valid
CommunicationOK	BOOL	FALSE	RCCP, RP, OPR	-	Cyclic BUS communication between controller and encoder Name in Openness: StatusSensor.CommunicationOK	
					FALSE	Communication not established
					TRUE	Communication established
AbsEncoderOffset	REAL	0.0 (-9.0E15 to 9.0E15)	RCCP, RP, OPR	-	Home point offset to the value of an absolute value encoder. The value will be retentively stored in the CPU. Name in Openness: StatusSensor.AbsEncoderOffset	
AdaptionState	DINT	0 (0 to 1)	R, OPR	10	Transfer status of the encoder Name in Openness: StatusSensor.AdaptionState	
					0	Data not transferred
					1	Data in transfer

12.9.21 StatusBits tags as of V6

The tag structure <axis name>.StatusBits.<tag name> contains the status information of the technology object.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description	
StatusBits.	STRUCT				TO_Struct_StatusBits	
Activated	BOOL	FALSE	RCCP, RP, OPR	-	Activation of the axis Name in Openness: StatusBits.Activated	
					FALSE	The axis is not activated.
					TRUE	The axis is activated. It axis is connected to the assigned PTO (Pulse Train Output). The data of the technology data block will be updated cyclically.
Enable	BOOL	FALSE	RCCP, RP, OPR	-	Enable status of the axis Name in Openness: StatusBits.Enable	
					FALSE	The axis is not enabled.
					TRUE	The axis is enabled and ready to accept Motion Control commands.
AxisSimulation	BOOL	FALSE	RCCP, RP, OPR	-	Name in Openness: StatusBits.AxisSimulation	
					FALSE	The simulation is disabled.
					TRUE	The simulation is enabled.
NonPosition Controlled	BOOL	FALSE	RCCP, RP, OPR	-	Name in Openness: StatusBits.NonPositionControlled	
					FALSE	The axis is in position-controlled mode.
					TRUE	The axis is not in position-controlled mode.
HomingDone	BOOL	FALSE	RCCP, RP, OPR	-	Homing status of the axis Name in Openness: StatusBits.HomingDone	
					FALSE	The axis is not homed.
					TRUE	The axis is homed and is capable of executing absolute positioning commands.
					The axis does not have to be homed for relative positioning. During active homing, the status is FALSE. The status remains TRUE during passive homing if the axis was already homed beforehand.	
Done	BOOL	FALSE	RCCP, RP, OPR	-	Command execution on the axis Name in Openness: StatusBits.Done	

Tag	Data type	Values	Access	W	Description	
					FALSE	A Motion Control command is active on the axis.
					TRUE	A Motion Control command is not active on the axis.
Error	BOOL	FALSE	RCCP, RP, OPR	-	Error status on the axis Name in Openness: StatusBits.Error	
					FALSE	No error is active on the axis.
					TRUE	An error has occurred on the axis.
					Additional information about the error is available in automatic mode at the "ErrorID" and "ErrorInfo" parameters of the Motion Control instructions. In manual mode, the "Error message" box of the axis control panel displays detailed information about the cause of error.	
Standstill	BOOL	FALSE	RCCP, RP, OPR	-	Standstill of the axis Name in Openness: StatusBits.Standstill	
					FALSE	The axis is in motion.
					TRUE	The axis is at a standstill.
Positioning Command	BOOL	FALSE	RCCP, RP, OPR	-	Execution of a positioning command Name in Openness: StatusBits.PositioningCommand	
					FALSE	A positioning command is not active on the axis.
					TRUE	The axis executes a positioning command of the "MC_MoveRelative" or "MC_MoveAbsolute" Motion Control instructions.
VelocityCommand	BOOL	FALSE	RCCP, RP, OPR	-	Execution of a command with velocity specification Name in Openness: StatusBits.VelocityCommand	
					FALSE	A command with velocity specification is not active on the axis.
					TRUE	The axis is executing a motion command with velocity specification of the "MC_MoveVelocity" or MC_MoveJog Motion Control instruction.
HomingCommand	BOOL	FALSE	RCCP, RP, OPR	-	Execution of a homing command Name in Openness: StatusBits.HomingCommand	
					FALSE	A homing command is not active on the axis.
					TRUE	The axis is executing a homing command of the "MC_Home" Motion Control instruction.

Tag	Data type	Values	Access	W	Description	
CommandTable Active	BOOL	FALSE	RCCP, RP, OPR	-	Execution of a command table Name in Openness: StatusBits.CommandTableActive	
					FALSE	A command table is not active on the axis.
					TRUE	The axis is controlled by Motion Control instruction "MC_CommandTable".
ConstantVelocity	BOOL	FALSE	RCCP, RP, OPR	-	Constant velocity Name in Openness: StatusBits.ConstantVelocity	
					FALSE	The axis is accelerating, decelerating, or at a standstill.
					TRUE	The setpoint velocity has been reached. The axis is moving at constant velocity.
Accelerating	BOOL	FALSE	RCCP, RP, OPR	-	Acceleration process Name in Openness: StatusBits.Accelerating	
					FALSE	The axis is decelerating, moving at constant velocity, or is at a standstill.
					TRUE	Axis is being accelerated.
Decelerating	BOOL	FALSE	RCCP, RP, OPR	-	Deceleration process Name in Openness: StatusBits.Decelerating	
					FALSE	The axis is accelerating, moving at constant velocity, or is at a standstill.
					TRUE	The axis is being decelerated.
ControlPanelActive	BOOL	FALSE	RCCP, RP, OPR	-	Activation status of the axis control panel Name in Openness: StatusBits.ControlPanelActive	
					FALSE	"Automatic mode" is activated. The user program has control priority over the axis.
					TRUE	The "Manual control" mode was enabled in the axis control panel. The axis control panel has control priority over the axis. The axis cannot be controlled from the user program.
DriveReady	BOOL	FALSE	RCCP, RP, OPR	-	Operational status of the drive Name in Openness: StatusBits.DriveReady	
					FALSE	The drive is not ready. Setpoints will not be executed.
					TRUE	The drive is ready. Setpoints can be executed.
RestartRequired	BOOL	FALSE	RCCP, RP, OPR	-	Restart of axis required Name in Openness: StatusBits.RestartRequired	
					FALSE	A restart of the axis is not required.
					TRUE	Values were modified in the load memory.

Tag	Data type	Values	Access	W	Description	
					To download the values to the work memory while the CPU is in RUN mode, the axis must be restarted. Use the MC_Reset Motion Control instruction to do this.	
SWLimitMinActive	BOOL	FALSE	RCCP, RP, OPR	-	Status of the software low limit switch Name in Openness: StatusBits.SWLimitMinActive	
					FALSE	The axis is kept within its configured work area.
					TRUE	The software low limit switch was reached or exceeded.
SWLimitMaxActive	BOOL	FALSE	RCCP, RP, OPR	-	Status of the software high limit switch Name in Openness: StatusBits.SWLimitMaxActive	
					FALSE	The axis is kept within its configured work area.
					TRUE	The software high limit switch was reached or exceeded.
HWLimitMinActive	BOOL	FALSE	RCCP, RP, OPR	-	Status of the hardware low limit switch Name in Openness: StatusBits.HWLimitMinActive	
					FALSE	The axis is kept within its configured permitted traversing range.
					TRUE	The hardware low limit switch was reached or exceeded.
HWLimitMaxActive	BOOL	FALSE	RCCP, RP, OPR	-	Status of the hardware high limit switch Name in Openness: StatusBits.HWLimitMaxActive	
					FALSE	The axis is kept within its configured permitted traversing range.
					TRUE	The hardware high limit switch was reached or exceeded.

12.9.22 ErrorBits tags as of V6

The tag structure <axis name>.ErrorBits.<tag name> indicates error at the technology object.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
ErrorBits.	STRUCT				TO_Struct_ErrorBits
SystemFault	BOOL	FALSE	RCCP, RP, OPR	-	Internal system error Name in Openness: ErrorBits.SystemFault
ConfigFault	BOOL	FALSE	RCCP, RP, OPR	-	Incorrect configuration of the axis Name in Openness: ErrorBits.ConfigFault
DriveFault	BOOL	FALSE	RCCP, RP, OPR	-	Error in the drive. Loss of the "Drive ready" signal. Name in Openness: ErrorBits.DriveFault
SWLimit	BOOL	FALSE	RCCP, RP, OPR	-	Software limit switch reached or exceeded Name in Openness: ErrorBits.SWLimit
HWLimit	BOOL	FALSE	RCCP, RP, OPR	-	Hardware limit switch reached or exceeded Name in Openness: ErrorBits.HWLimit
DirectionFault	BOOL	FALSE	RCCP, RP, OPR	-	Impermissible motion direction Name in Openness: ErrorBits.DirectionFault
HWUsed	BOOL	FALSE	RCCP, RP, OPR	-	Another axis is using the same PTO (Pulse Train Output) and is enabled with "MC_Power". Name in Openness: ErrorBits.HWUsed
SensorFault	BOOL	FALSE	RCCP, RP, OPR	-	Error in the encoder system Name in Openness: ErrorBits.SensorFault
Communication-Fault	BOOL	FALSE	RCCP, RP, OPR	-	Communication error Error in communication with a connected device. Name in Openness: ErrorBits.CommunicationFault
FollowingError	BOOL	FALSE	RCCP, RP, OPR	-	Maximum permitted following error exceeded Name in Openness: ErrorBits.FollowingError
PositioningFault	BOOL	FALSE	RCCP, RP, OPR	-	Positioning error The axis was not correctly positioned at the end of a positioning motion. Name in Openness: ErrorBits.PositioningFault

Tag	Data type	Values	Access	W	Description
AdaptionError	BOOL	FALSE	RCCP, RP, OPR	-	Error during data transfer Name in Openness: ErrorBits.AdaptionError

12.9.23 ControlPanel tags as of V6

The "ControlPanel" tags do not contain any user-relevant data. These tags cannot be accessed in the user program.

Tags

The following tags "ControlPanel" are readable in Openness.

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
ControlPanel.	STRUCT				TO_Struct_ControlPanel
Input	STRUCT				TO_Struct_Input
TimeOut	DINT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.TimeOut
EsLifeSign	DINT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.EsLifeSign
Command[1].	STRUCT				ARRAY[1..1] TO_Struct_Command
ReqCounter	DINT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.Command[1].ReqCounter
Type	DINT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.Command[1].Type
Position	REAL	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.Command[1].Position
Velocity	REAL	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.Command[1].Velocity
Acceleration	REAL	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.Command[1].Acceleration
Jerk	REAL	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.Command[1].Jerk

Tag	Data type	Values	Access	W	Description
Param	INT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Input.Command[1].Param
Output.	STRUCT		-		TO_Struct_Output
RTLifeSign	INT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Output.RTLifeSign
Command[1].	STRUCT		-		ARRAY[1..1] TO_Struct_Command
AckCounter	INT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Output.Command[1].AckCounter
ErrorID	USHORT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Output.Command[1].ErrorID
ErrorInfo	USHORT	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Output.Command[1].ErrorInfo
Done	BOOL	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Output.Command[1].Done
Aborted	BOOL	-	OPR	-	(Internal parameter) Name in Openness: ControlPanel.Output.Command[1].Aborted

12.9.24 Internal tags as of V6

The "Internal" tags do not contain any user-relevant data. These tags cannot be accessed in the user program.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
Internal[1..4].	STRUCT				ARRAY [1..4] TO_Struct_Internal
Id	INT	0 (-32768 to 32767)	OPRW	-	(Internal parameter) Name in Openness: Internal[1..4].Id
Value	REAL	0 (-9.0E15 to 9.0E15)	OPRW	-	(Internal parameter) Name in Openness: Internal[1..4].Value

12.9.25 Update of the technology object tags

The status and error information of the axis indicated in the technology object tags is updated at each cycle control point.

Changes to the values of configuration tags do not take effect immediately. For information on the conditions under which a change takes effect, refer to the detailed description of the relevant tag.

12.10 Tags of the command table V6 technology object

The tag structure <command table>.Command[n].<tag name> contains the configured command parameters.

Tags

Legend (Page 259)

Tag	Data type	Values	Access	W	Description
Command[n].	STRUCT				ARRAY[1..32] TO_Struct_Command
Type	INT	0 (0 to 151)	RW, OPRW	-	Command type Name in Openness: Command[n].Type
					0 "Empty" command
					2 "Stop" command
					5 "Relative positioning" command
					6 "Absolute positioning" command
					7 Command "Velocity setpoint"
					151 Command "Wait"
Position	REAL	0.0	RW, OPRW	-	Target position/traversing distance of the command Name in Openness: Command[n].Position
Velocity	REAL	0.0	RW, OPRW	-	Command velocity Name in Openness: Command[n].Velocity
Duration	REAL	0.0	RW, OPRW	-	Command duration Name in Openness: Command[n].Duration
NextStep	INT	0 (0 to 1)	RW, OPRW	-	Mode for the transition to the next command Name in Openness: Command[n].NextStep
					0 "Complete command"
					1 "Blend motion"
StepCode	WORD	0	RW, OPRW	-	Command step code Name in Openness: Command[n].StepCode

Tag	Data type	Values	Access	W	Description	
WarningEnabled*	BOOL	FALSE	OPRW	-	Name in Openness: _WarningEnabled	
					FALSE	Warning is disabled.
					TRUE	Warning is enabled
UseAxisParametersFrom*	INT/STRING	-	OPRW	-	Name in Openness: _UseAxisParametersFrom Axis number, "default axis", name of axis.	

*) Available in Openness

Index

A

Add empty line, 131
Add separator line, 131
Axis and command table technology object
    List of ErrorIDs and ErrorInfo, 236

B

Basic knowledge
    Required, 3

C

Closed loop control, 98
Command table technology object
    Add new object, 124
    Basic parameters, 127
    Command table configuration, 127
    Configuration window icons, 125
    Configuring activate warnings, 127
    Configuring duration, 130
    Configuring position / travel path, 129
    Configuring the command type, 128
    Configuring the next step, 130
    Configuring the step code, 130
    Configuring use axis parameters of, 127
    Configuring velocity, 129
    Extended parameters, 141
    General configuration, 127
    Tag command[1...32], 291
    Tools, 123
    Usage, 123

D

DB_ANY, 174
Direction output and travel direction
    relation, 17
Drive connection S7-1200 Motion Control, 22

E

Encoder connection S7-1200 Motion Control, 22

F

Following error monitoring, 97

H

Hardware and software limit switches
    Function, 38
Hardware configuration for Motion Control S7-1200, 10
Homing
    Homing modes, 40

I

Insert empty line, 131
Insert separator line, 131
Interpolator OB, 33, 35

J

Jerk limit
    Function, 39

M

Manual
    Purpose, 3
    Scope, 3
MC_ChangeDynamic
    Instruction, 213
    Parameter, 214
MC_CommandTable
    Instruction, 211
    Parameter, 212
MC_Halt
    Function chart, 195
    Instruction, 193
    Parameters, 194
MC_Home
    Instruction, 188
    Parameter, 190
MC_MoveAbsolute
    Function chart, 199
    Instruction, 196
    Parameter, 197

MC_MoveJog
Function chart, 210
Instruction, 208
Parameter, 209
MC_MoveRelative
Function chart, 203
Instruction, 200
Parameter, 201
MC_MoveVelocity
Function chart, 207
Instruction, 204
MC_Power
Function chart, 185
Instruction, 181
Parameters, 182
MC_ReadParam
Instruction, 216
Parameter, 217
MC_Reset, 186
MC_WriteParam
Instruction, 218
Parameter, 219
MC-Interpolator OB, 33, 35
MC-Servo OB, 33, 35
Motion Control CPU S7-1200
Guidelines, 42
Motion Control S7-1200
Drive and encoder connections, 22
PROFIdrive, 22
Telegram, 22
MoveVelocity
Parameter, 205

O

OB servo, 34
Optimization, 150
Optimize position controller, 150

P

Position control, 98
Position monitoring, 96
Positioning axis
Tags, 259
Positioning axis technology object
Active homing, 92
Actor tags, 262
Add new object, 58
Axis name configuration, 61
Basic parameters, 61
Changing the configuration parameters for dynamics in the user program, 86
Changing the configuration parameters for homing in the user program, 94
Commissioning overview, 57
Configuration overview, 57
Configuration window icons, 59
Configuring acceleration, 82
Configuring active homing, 88
Configuring approach/homing direction, 89
Configuring deceleration, 82
Configuring distance per motor revolution, 73
Configuring end of homing switch, 89, 91
Configuring end of homing switch, 89, 91
Configuring home position offset, 90
Configuring homing switch input, 88, 91
Configuring invert direction signal, 74
Configuring jerk limiter, 82
Configuring maximum velocity / start/stop velocity, 81
Configuring passive homing, 90
Configuring pulses per motor revolution, 73
Configuring ramp-down time, 82
Configuring ramp-up time, 82
Configuring reference point position, 90, 91
Configuring smoothing time, 83
Configuring the homing speed, 89
Configuring the velocity limiting unit configuration, 81
Diagnostics overview, 57
Drive signal configuration, 65
DynamicDefaults tag, 274
DynamicLimits tag, 273
Emergency stop deceleration configuration, 84
ErrorBits tag, 288
FollowingError tag, 279
General dynamics configuration, 81
Hardware and software components, 53
Hardware interface configuration, 63
Homing tag, 277
Mechanics configuration, 73, 74
Mechanics tag, 272
Modulo tag, 272
Passive homing, 93
Position tag, 260
PositionControl tag, 278
PositionLimitsHW tag, 276
PositionLimitsSW tag, 275
PositionMonitoring tag, 280
PTO and HSC configuration, 63
Response when jerk limiter is activated, 85
Sensor tag, 266

Simulation tag, 261
StandstillSignal tag, 280
StatusBits tag, 284
StatusDrive tag, 282
StatusPositioning tag, 281
StatusSensor tag, 283
Tag ActualPosition, 260
Tag ActualVelocity, 260
Tools, 56
Unit of measurement configuration, 62
Units tag, 271
Updating the tags, 291
Velocity tag, 260
PROFIdrive, 22
Pulse interface
Principle, 16
Purpose
Manual, 3

R

Reinitialization of technology objects, 173
Required basic knowledge, 3
Restart of technology objects, 173

S

Scope
Manual, 3
Servo motor, 10
Servo OB, 33, 35
Standstill signal, 97
Stepper motor, 10

T

Tags of the positioning axis technology object, 259
Technology object
Data types, 174
Technology object command table: Shortcut menu commands,
Telegram S7-1200 Motion Control, 22
TO_PositioningAxis, 58