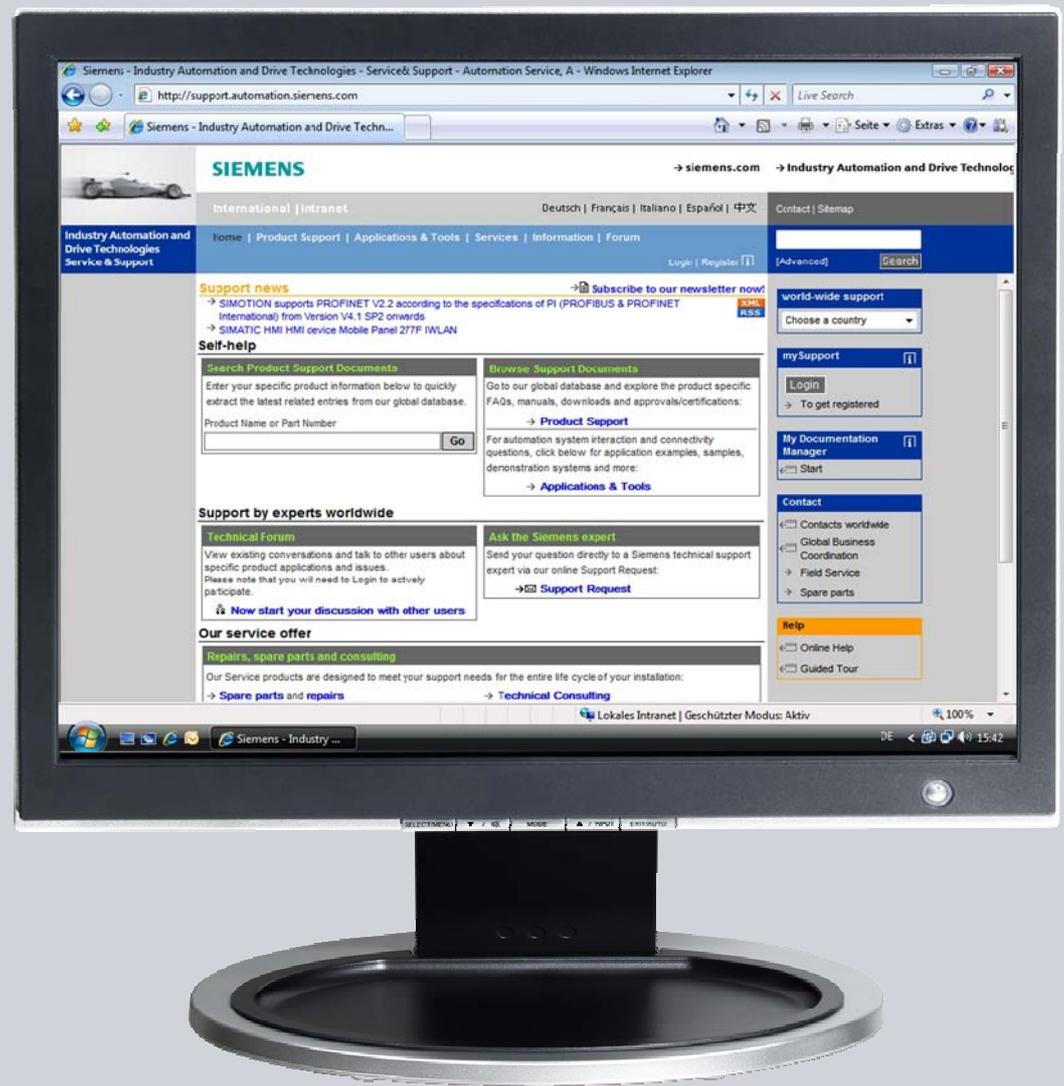


What Motion Control Functions are available in STEP 7 (TIA Portal) V11 and how will you use this functions

STEP 7 (TIA Portal) V11

FAQ • January 2012



Service & Support

Answers for industry.

SIEMENS

This entry is from the Service&Support portal of Siemens AG, Sector Industry, Industry Automation and Drive Technologies. The general terms of use (http://www.siemens.com/terms_of_use) apply.

Clicking the link below directly displays the download page of this document.

<http://support.automation.siemens.com/VW/view/en/41737097>

Question

What Motion Control Functions are available in STEP 7 (TIA Portal) V11 and how will you use this functions?

Answer

The instructions and notes listed in this document provide a detailed answer to this question.

Legal information

Warning notice system

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

⚠ DANGER
indicates that death or severe personal injury will result if proper precautions are not taken.
⚠ WARNING
indicates that death or severe personal injury may result if proper precautions are not taken.
⚠ CAUTION
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.
CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.
NOTICE
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, 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 adhered to. The information in the relevant documentation must be observed.

Trademarks

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

Disclaimer of Liability

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

Table of contents

1	Using Motion Control	7
1.1	Introduction	7
1.1.1	Motion functionality of the CPU S7-1200	7
1.1.2	Hardware components for motion control	8
1.2	Basics for working with motion control.....	11
1.2.1	CPU outputs relevant for motion control	11
1.2.2	Relationship between the travel direction and voltage level at the direction output.....	13
1.2.3	"Axis" technology object and its integration in the overall system	14
1.2.4	Tools for motion control	17
1.2.5	Hardware and software limit switches	19
1.2.6	Homing.....	20
1.3	Guidelines on use of motion control	21
1.4	Add technological object Axis	22
1.5	Motion control tool - Configuration.....	23
1.5.1	Configuring the technological object.....	23
1.5.2	Basic parameters	24
1.5.2.1	Configuration - General.....	24
1.5.3	Extended parameters.....	26
1.5.3.1	Configuration - Drive interface	26
1.5.3.2	Configuration - Mechanics	27
1.5.3.3	Position limits	28
1.5.3.3.1	Requirements for hardware limit switches	28
1.5.3.3.2	Configuration - Position limits	28
1.5.3.3.3	Behavior of axis when position limits is tripped	30
1.5.3.3.4	Changing the position limits configuration in the user program.....	32
1.5.3.4	Dynamics	33
1.5.3.4.1	Configuration - General dynamics	33
1.5.3.4.2	Configuration - Dynamics emergency ramp	34
1.5.3.4.3	Changing the configuration of dynamics in the user program	35
1.5.3.5	Homing.....	36
1.5.3.5.1	Configuration - Homing	36
1.5.3.5.2	Sequence - Active homing	39
1.6	Download to CPU	40
1.7	Motion control tool - Commissioning (control panel).....	41
1.8	Programming.....	44
1.8.1	Overview of the Motion Control statements	44
1.8.2	Creating a user program	45
1.8.3	Programming notes.....	48
1.8.4	Behavior of the Motion Control commands after POWER OFF and restart.....	50
1.8.5	Monitoring active commands	51
1.8.5.1	Monitoring active commands	51
1.8.5.2	Motion control instructions with "Done" output parameter	51
1.8.5.3	Motion control instruction MC_MoveVelocity	55
1.8.5.4	Motion control instruction MC_MoveJog.....	59
1.8.6	Error displays of the Motion Control statements.....	63

1.9	Motion control tool - Diagnostics.....	64
1.9.1	Status and error bits.....	64
1.9.2	Motion status.....	67
1.10	Working with watch tables	68
1.11	Appendix	69
1.11.1	Using multiple axes with the same PTO	69
1.11.2	Using multiple drives with the same PTO	72
1.11.3	Tracking tasks from higher priority classes (execution levels)	73
1.11.4	Special cases for use of software limit switches.....	75
1.11.4.1	Software limit switches in conjunction with a homing operation.....	75
1.11.4.2	Software limit switches in conjunction with dynamic changes.....	79
1.11.5	List of ErrorIDs and ErrorInfos	81
1.11.6	Tags of the technology object.....	87
1.11.6.1	Config. tags	87
1.11.6.2	MotionStatus. tags	97
1.11.6.3	StatusBits. tags	98
1.11.6.4	ErrorBits. tags	100
1.11.6.5	Internal. tabs	102
1.11.6.6	ControlPanel tags	102
1.11.6.7	Update of the technology object tags	102
2	References	103
2.1	MC_Power.....	103
2.1.1	MC_Power: Enable, disable axis	103
2.1.2	MC_Power: Function chart.....	106
2.2	MC_Reset	107
2.2.1	MC_Reset: Acknowledge error	107
2.3	MC_Home	109
2.3.1	MC_Home: Home axes, set home position	109
2.4	MC_Halt	113
2.4.1	MC_Halt: Halt axis	113
2.4.2	MC_Halt: Function chart	115
2.5	MC_MoveAbsolute.....	116
2.5.1	MC_MoveAbsolute: Absolute positioning of axes.....	116
2.5.2	MC_MoveAbsolute: Function chart.....	118
2.6	MC_MoveRelative	119
2.6.1	MC_MoveRelative: Relative positioning of axes.....	119
2.6.2	MC_MoveRelative: Function chart.....	121
2.7	MC_MoveVelocity	122
2.7.1	MC_MoveVelocity: Move axes at preset rotational speed.....	122
2.7.2	MC_MoveVelocity: Function chart	125
2.8	MC_MoveJog	126
2.8.1	MC_MoveJog: Move axes in jogging mode	126
2.8.2	MC_MoveJog: Function chart	128

Using Motion Control

1.1 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, assists you in the control of stepper motors and servo motors with pulse interface:

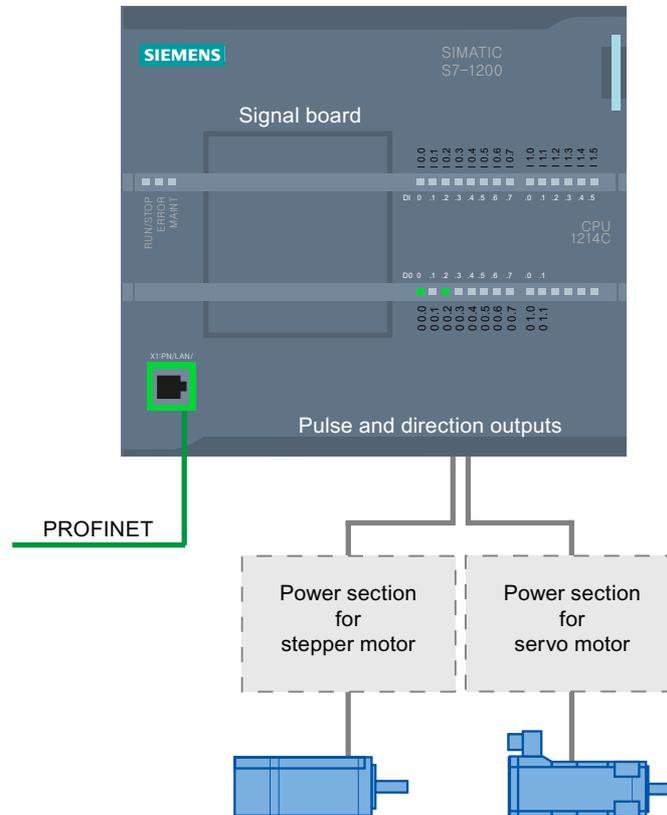
- You configure the "Axis" technology object in the TIA Portal. There, you configure the mechanical drive data, drive interface, dynamic parameters, and other drive properties.
- The CPU S7-1200 controls the pulse and direction outputs of the CPU for controlling the drive.
- In the user program you control the axis with motion control instructions and initiate motion tasks.

See also

Hardware components for motion control (Page 8)

1.1.2 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:

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

The DC/DC/DC variants of the CPU S7-1200 have onboard outputs for direct control of drives. The relay variants of the CPU require the signal board described below for drive control.

Signal board

With the signal board (DI2/DO2 x DC24V), you expand the onboard I/O to include two interrupt-capable digital inputs and two (high-speed) digital outputs. The two digital outputs can be used as pulse and direction outputs for drive control, if required. When a DC/DC/DC variant of the CPU S7-1200 is used together with a signal board, the maximum number of controllable drives is limited to "2".

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.

Maximum number of controllable drives

The maximum number of controllable drives for the different CPU versions is provided in the following table:

CPU		Without signal board	With signal board DI2/DO2 x DC24V
CPU 1211C	DC/DC/DC	2	2
	AC/DC/RLY	-	1
	DC/DC/RLY	-	1
CPU 1212C	DC/DC/DC	2	2
	AC/DC/RLY	-	1
	DC/DC/RLY	-	1
CPU 1214C	DC/DC/DC	2	2
	AC/DC/RLY	-	1
	DC/DC/RLY	-	1

Limit frequencies of pulse outputs

The following limit frequencies apply to the pulse outputs:

Pulse output	Frequency
Onboard	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$
Signal board DI2/DO2 x DC24V	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$

Ordering information

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

Name	MLFB (order no.)
CPU 1211C DC/DC/DC	6ES7211-1AD30-0XB0
CPU 1211C AC/DC/RLY	6ES7211-1BD30-0XB0
CPU 1211C DC/DC/RLY	6ES7211-1HD30-0XB0
CPU 1212C DC/DC/DC	6ES7212-1AD30-0XB0
CPU 1212C AC/DC/RLY	6ES7212-1BD30-0XB0
CPU 1212C DC/DC/RLY	6ES7212-1HD30-0XB0
CPU 1214C DC/DC/DC	6ES7214-1AE30-0XB0
CPU 1214C AC/DC/RLY	6ES7214-1BE30-0XB0
CPU 1214C DC/DC/RLY	6ES7214-1HE30-0XB0
Signal board DI2/DO2 x DC24V	6ES7223-0BD30-0XB0

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 7)

CPU outputs relevant for motion control (Page 11)

1.2 Basics for working with motion control

1.2.1 CPU outputs relevant for motion control

Pulse and direction outputs

The CPU provides one pulse output and one direction output for controlling a stepper motor drive or a servo motor drive with pulse interface. The pulse output provides the drive with the pulses required for motor motion. The direction output controls the travel direction of the drive.

Pulse and direction outputs are permanently assigned to one another. Onboard CPU outputs as well as outputs of signal board DI2/DO2 x DC24V can be used as pulse and direction outputs. You select between onboard CPU outputs and outputs of the signal board during device configuration under Pulse generators (PTO/PWM) on the "Properties" tab.

The following table shows the address assignment of the pulse and direction outputs:

CPU S7-1200:	Without signal board				With signal board DI2/DO2 x DC24V *)			
	Outputs for PTO1		Outputs for PTO2		Outputs for PTO1		Outputs for PTO2	
	Pulse	Direction	Pulse	Direction	Pulse	Direction	Pulse	Direction
CPU 1211C DC/DC/DC	Ax.0	Ax.1	Ax.2	Ax.3	Ax.0	Ax.1	Ax.2	Ax.3
					Ay.0	Ay.1		
CPU 1211C AC/DC/RLY	-	-	-	-	Ay.0	Ay.1	-	-
CPU 1211C DC/DC/RLY	-	-	-	-	Ay.0	Ay.1	-	-
CPU 1212C DC/DC/DC	Ax.0	Ax.1	Ax.2	Ax.3	Ax.0	Ax.1	Ax.2	Ax.3
					Ay.0	Ay.1		
CPU 1212C AC/DC/RLY	-	-	-	-	Ay.0	Ay.1	-	-
CPU 1212C DC/DC/RLY	-	-	-	-	Ay.0	Ay.1	-	-
CPU 1214C DC/DC/DC	Ax.0	Ax.1	Ax.2	Ax.3	Ax.0	Ax.1	Ax.2	Ax.3
					Ay.0	Ay.1		
CPU 1214C AC/DC/RLY	-	-	-	-	Ay.0	Ay.1	-	-
CPU 1214C DC/DC/RLY	-	-	-	-	Ay.0	Ay.1	-	-

x = Initial byte address of onboard CPU outputs (default value = 0)

y = Initial byte address of signal board outputs (default value = 4)

* If a DC/DC/DC variant of the CPU is used together with a signal board DI2/DO2 x DC24V, the signals of PTO1 can be output via the onboard CPU outputs (Ax.0 and Ax.1) or via the outputs of the signal board (Ay.0 and Ay.1).

Drive interface

For motion control, you can optionally parameterize a drive interface for "Drive enabled" and "Drive ready". When drive enabled is used, the digital output for drive enabled and the digital input for "Drive ready" can be selected by the user.

Note

If PTO was selected and assigned to an axis, the firmware assumes control via the associated pulse and direction outputs.

With this takeover of the control function, the connection between the process image and I/O output is also disconnected. While the user has the possibility of writing the process image of pulse and direction outputs via the user program or watch table, this is never 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 merely reflects the value of the process image and does not match the actual status of the I/O output in any respect.

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.

See also

Relationship between the travel direction and voltage level at the direction output (Page 13)

"Axis" technology object and its integration in the overall system (Page 14)

Tools for motion control (Page 17)

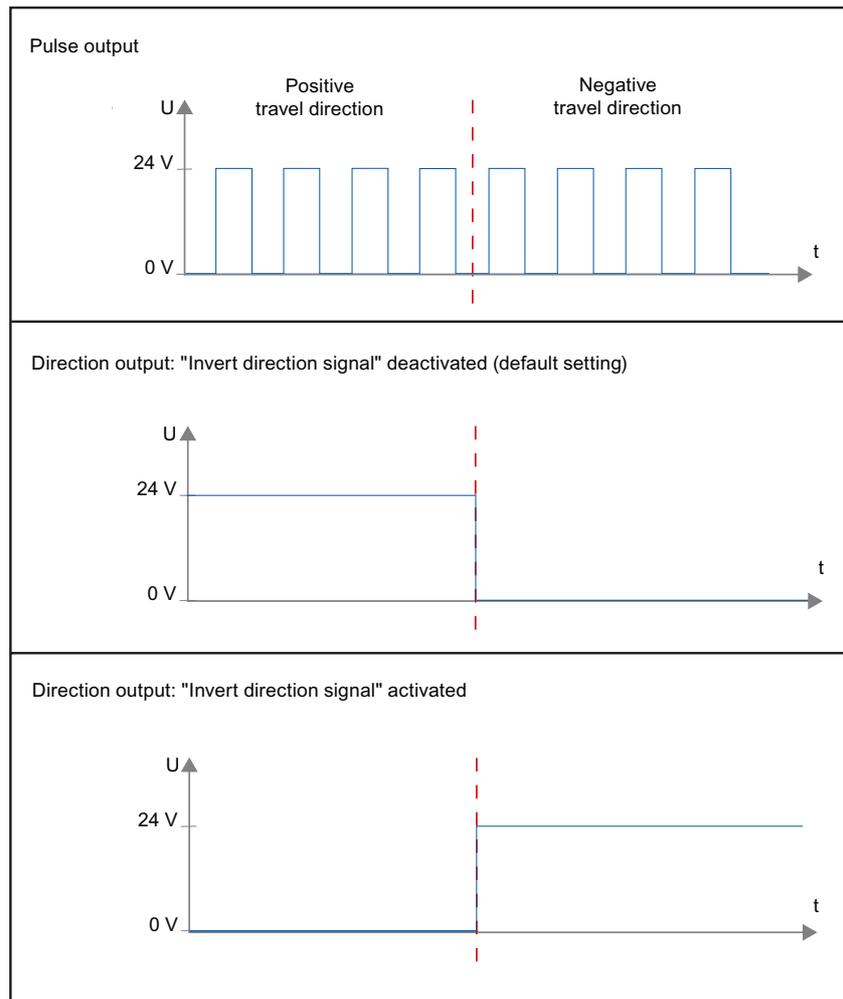
Hardware and software limit switches (Page 19)

Homing (Page 20)

Hardware components for motion control (Page 8)

1.2.2 Relationship between the travel direction and voltage level at the direction output

The direction output of the CPU specifies the travel direction of the drive. You configure the logic of the direction output during axis configuration under "Mechanics". The relationships between configuration, direction output, and travel direction are presented in the following diagram:



If "Invert direction signal" is deactivated in the configuration, a 24 V level is output at the direction output for positive travel direction. If "Invert direction signal" is activated in the configuration, a 0 V level is output at the direction output for positive travel direction.

See also

CPU outputs relevant for motion control (Page 11)

"Axis" technology object and its integration in the overall system (Page 14)

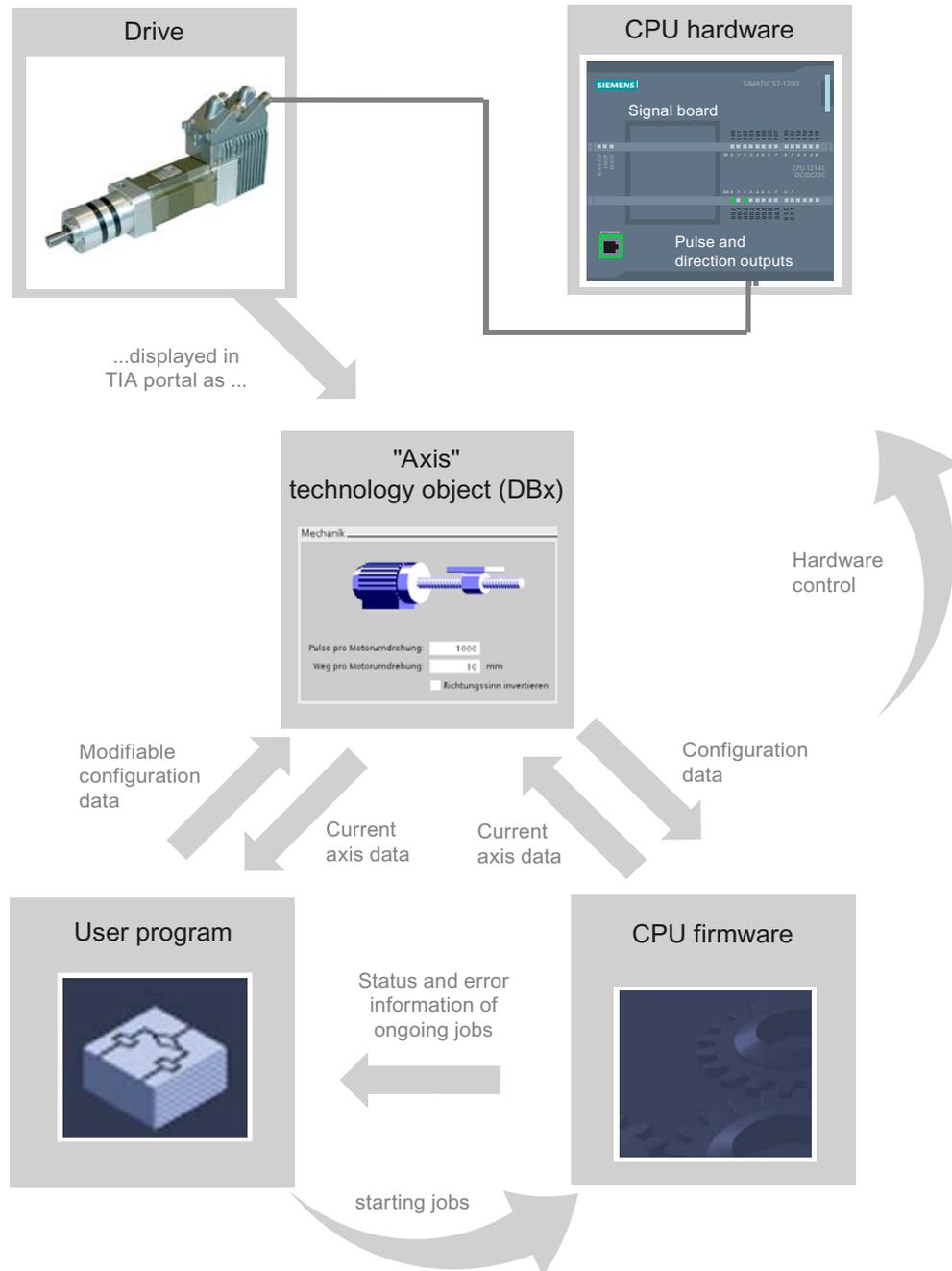
Tools for motion control (Page 17)

Hardware and software limit switches (Page 19)

Homing (Page 20)

1.2.3 "Axis" technology object and its integration in the overall system

The relationships between hardware and software components that come into play when using motion control are shown in the following diagram:



CPU hardware

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

Drive

The drive represents a unit comprising the power section and motor. You can use stepper motors or servo motors with pulse interface.

"Axis" technology object

The physical drive is mapped as an "Axis" technology object in the TIA Portal. For this purpose, you configure the "Axis" technology object with the following parameters:

- Selection of the PTO to be used and configuration of the drive interface
- Parameters for the mechanics and for the transmission ratio of the drive (or machine or system)
- Parameters for position limits, dynamics, and homing

The configuration of the "Axis" technology object is saved in the data block of the technology object. At the same time, this data block forms the interface between the user program and CPU firmware. During runtime of the user program, the current axis data are stored in the data block of the technology object.

User program

You use motion control instructions in the user program to start tasks in the CPU firmware. The following axis control tasks are available:

- Position axis absolute
- Position axis relative
- Move axis with speed setpoint
- Move axis in jog mode.
- Stop axis
- Home axis; set reference point
- Acknowledge error

Use the input parameters of the motion control instructions and the axis configuration to define the parameters of the task. The output parameters of the instruction provide up-to-date information on the status and on any task errors.

Before you start a task for the axis, you must enable the axis with the motion control instruction "MC_Power".

You can use the technology object tags to read out configuration data and current axis data in the user program. You can change individual, editable tags of the technology object (such as the current acceleration) from the user program.

CPU firmware

The CPU firmware processes the motion control tasks initiated in the user program. When the control panel is used, the motion control tasks are initiated via operator input on the control panel. The CPU firmware has the following functions, according to the axis configuration:

- Calculation of the exact motion profile for motion tasks and emergency stop situations;
- Control of drive enabled, as well as the pulse and direction signal;
- Monitoring of the drive as well as the hardware and software limit switches;
- Up-to-date feedback of status and error information of tasks to motion control instructions in the user program;
- Writing of current axis data to the data block of the technology object.

See also

CPU outputs relevant for motion control (Page 11)

Relationship between the travel direction and voltage level at the direction output (Page 13)

Tools for motion control (Page 17)

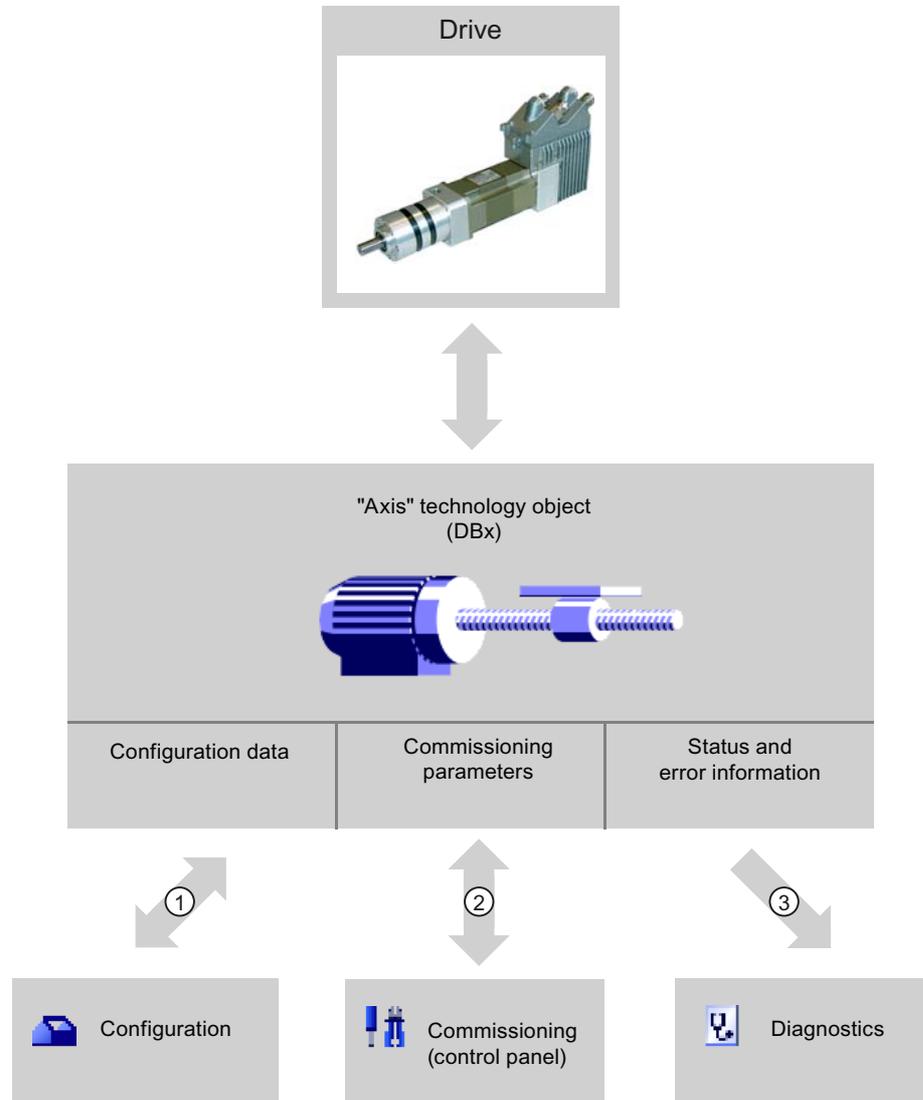
Hardware and software limit switches (Page 19)

Homing (Page 20)

Tags of the technology object (Page 87)

1.2.4 Tools for motion control

The TIA Portal provides the "Configuration", "Commissioning", and "Diagnostics" tools for the "Axis" technology object. The following diagram shows the interaction of the three tools with the technology object and the CPU firmware:



①	Reading and writing of configuration data of the technology object;
②	Drive control via the technology object; Reading of axis status for display on the control panel;
③	Readout of the current status and error information of the technology object.

Configuration

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

- Selection of the PTO to be used and configuration of the drive interface
- Properties of the mechanics and the transmission ratio of the drive (or machine or system)
- Properties for position limits, dynamics, and homing

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 control panel will be displayed.

The following commands are available on the control panel:

- Enable and disable axis
- Move axis in jog mode
- Position axis in absolute and relative terms
- Home axis
- Acknowledge errors

The velocity and the acceleration / deceleration can be specified for the motion commands. The control panel also shows the current axis status.

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 11)

Relationship between the travel direction and voltage level at the direction output (Page 13)

"Axis" technology object and its integration in the overall system (Page 14)

Hardware and software limit switches (Page 19)

Homing (Page 20)

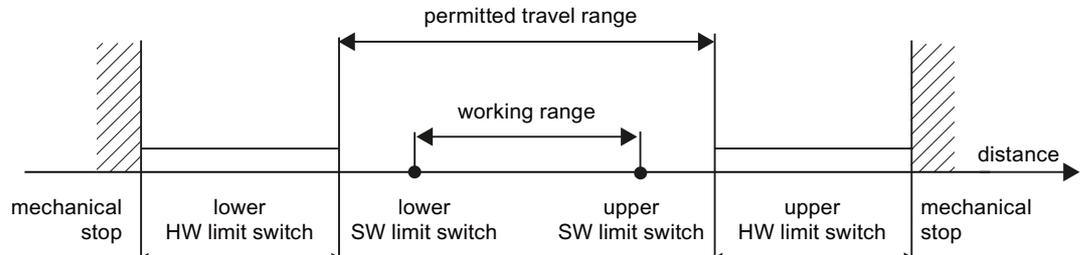
Motion control tool - Configuration (Page 23)

Motion control tool - Commissioning (control panel) (Page 41)

Motion control tool - Diagnostics (Page 64)

1.2.5 Hardware and software limit switches

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



Hardware limit switches are limit switches that limit the maximum "permitted travel 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 travel range. Because the positions of the software limit switches can be set flexibly, the working range of the axis can be restricted 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 11)

Relationship between the travel direction and voltage level at the direction output (Page 13)

"Axis" technology object and its integration in the overall system (Page 14)

Tools for motion control (Page 17)

Homing (Page 20)

Position limits (Page 28)

1.2.6 Homing

Homing refers to the matching of the axis coordinates to the real, physical drive position (if the drive is currently at position x, the axis will be adjusted to be in position x).

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, axis homing is initiated with the motion control instruction "MC_Home".

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 reference point switch is detected, the axis is homed according to the configuration. Active traversing motions are aborted.

- **Passive homing**

During passive homing, the motion control instruction "MC_Home" does not carry out any homing motion. The traversing motion required for this step must be implemented by the user via other motion control instructions. When the reference point 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 reference point 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. If you want to assign the reference point to an exact mechanical position, the axis must be at a standstill at this position at the time of the homing operation.

- **Direct homing relative**

The axis position is set regardless of the reference point 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".

For more detailed information on active and passive homing, refer to Chapter: "Configure -> Extended parameters"

See also

CPU outputs relevant for motion control (Page 11)

Relationship between the travel direction and voltage level at the direction output (Page 13)

"Axis" technology object and its integration in the overall system (Page 14)

Tools for motion control (Page 17)

Hardware and software limit switches (Page 19)

Homing (Page 36)

1.3 Guidelines on use of motion control

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

Requirements

To use the "Axis" technological object, you must create a project with a CPU S7-1200.

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. Add technological object Axis (Page 22)
2. Configuring the technological object (Page 23)
3. Download to CPU (Page 40)
4. Function test of the axis in the commissioning window (Page 41)
5. Programming (Page 44)
6. Diagnostics of the axis control (Page 64)

1.4 Add technological object Axis

Proceed as follows to add an "Axis" technology object in the project tree:

Requirements

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

Procedure

1. Open the CPU folder in the project tree.
2. Open the technology objects folder.
3. Double-click "Add new object".
The "Add new object" dialog box opens.
4. Enter an individual name for the "Axis" technology object in the "Name" input field.
5. Click "Axis".
The required "TO_AXIS_PTO" type is already selected.
6. Select the "Manual" option if you want to change the proposed data block number.
7. Click "More information" if you want to supplement user information for the technology object.
8. Click "OK" to add the technology object. Click "Cancel" to discard your entries.

Result

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

Note

You can select the "Add and open new block" check box at the bottom of the dialog box. This opens the configuration of the technology object after adding has been completed.

See also

Guidelines on use of motion control (Page 21)

1.5 Motion control tool - Configuration

1.5.1 Configuring the technological object

You configure the properties of the "Axis" technology object in the configuration window. To open the configuration window of the technology object, follow these steps:

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

Icons of the configuration window

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

	<p>The configuration contains default values and is complete. The configuration only contains default values. With these default values you can use the technology object without additional changes.</p>
	<p>The configuration contains user-defined values and is complete All values in the input fields of the configuration are valid and at least one default value was modified.</p>
	<p>Incomplete or incorrect configuration At least one input field or a drop-down list box contains no value or an invalid value. The corresponding field, or the drop-down list box, is displayed on a red background. Click the roll-out error message to indicate the cause of error.</p>

See also

Guidelines on use of motion control (Page 21)

Basic parameters (Page 24)

Extended parameters (Page 26)

1.5.2 Basic parameters

1.5.2.1 Configuration - General

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

Axis name

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

Hardware interface

Here, choose the PTO (Pulse Train Output) by which the pulses for control of the stepper motors or servo motors with pulse interface are to be made available. The pulses are output to the power section of the drive by permanently assigned digital outputs.

On CPUs with relay outputs, the pulse signal cannot be output to these outputs because the relays do not support the required switching frequencies. To enable the use of PTO on these CPUs, you must use a signal board with digital outputs.

Note

The PTO requires the internal functionality of a high-speed counter (HSC). This means the corresponding high-speed counter cannot be used elsewhere.

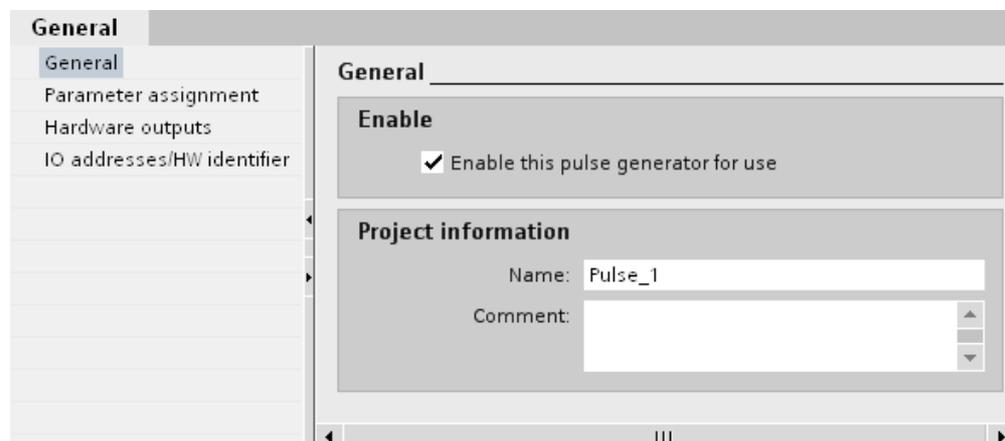
The assignment between PTO and HSC is fixed. When PTO1 is activated, it will be connected to HSC1. If PTO2 is activated, it will be connected to HSC2.

Proceed as follows to configure the desired PTO:

1. Select PTO "Pulse_1" or "Pulse_2" from the "PTO selection of drive control" drop-down list box.
2. Click on "Device configuration".

The device configuration of the pulse generator is displayed.

Enlarge the properties area of the device configuration if the pulse generator configuration is not visible.



3. Select the "Enable this pulse generator for use" check box.
4. Select the "Parameter assignment" entry in the area navigation.
The "Parameter assignment" dialog is displayed.

General

- General
- Parameter assignment**
- Hardware outputs
- IO addresses/HW identifier

Parameter assignment

Pulse options

Pulse generator used as: PTO

Output source: Onboard CPU output

Time base: Milliseconds

Pulse width format: Hundredths

Cycle time: 100 ms

Initial pulse width: 50 Hundredths

5. From the "Pulse generator used as:" drop-down list box select entry "PTO".
6. From the "Output source:" drop-down list box select the entry "Onboard CPU output" or "Signal board output". The "Signal board output" can only be selected for PTO1 when a signal board is installed.
7. Return to the axis configuration.

If the corresponding high-speed counter is not already being used elsewhere, the PTO fields in the "General" axis configuration do not have a red background. If this is not the case, correct the configuration based on the error messages.

User Units

Select the desired unit for the axis dimension system in the drop-down list box. The chosen unit will be used for additional configuring of the "Axis" technology object and for displaying the current axis data.

The values of the input parameters (Position, Distance, Velocity, etc.) of motion control instructions also relate to this unit.

NOTICE

If you change the dimension system at a later time, the values may not be converted correctly in all configuration windows of the technology object. In this case, check the configuration of all axis parameters.

You may have to adapt the values of the input parameters of motion control instructions to the new dimension unit in the user program.

1.5.3 Extended parameters

1.5.3.1 Configuration - Drive interface

Configure the enable signal and the "Drive ready" feedback signal of the drive in the "Drive interface" configuration window.

The enable signal is controlled by motion control instruction "MC_Power" and enables the power to the drive. The signal is provided to the drive via the digital output to be configured.

If the drive is to start executing motions after receipt of the enable signal, it signals "Drive ready" to the CPU. The "Drive ready" signal is signaled back to the CPU via the digital input to be configured.

If the drive does not have any interfaces of this type, you will not have to configure the parameters.

See also

Configuration - Mechanics (Page 27)

Position limits (Page 28)

Dynamics (Page 33)

Homing (Page 36)

1.5.3.2 Configuration - Mechanics

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

Increments per motor revolution

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

Limit values (independent of the selected user unit):

- $0 < \text{increments per motor revolution} \leq 2,147,483,647$

Load distance per motor revolution

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

Limit values (independent of the selected user unit):

- $0.0 < \text{Load distance per motor revolution} \leq 1.0e12$

Invert direction signal

You can adjust the direction output to the direction logic of the drive using the "Invert direction signal" check box.

- **Invert direction signal: deactivated**
 - 0 V level = negative travel direction
 - 24 V level = positive travel direction
- **Invert direction signal: activated**
 - 0 V level = positive travel direction
 - 24V level = negative travel direction

See also

Configuration - Drive interface (Page 26)

Position limits (Page 28)

Dynamics (Page 33)

Homing (Page 36)

1.5.3.3 Position limits

1.5.3.3.1 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 permitted travel range.

See also

Configuration - Position limits (Page 28)

Behavior of axis when position limits is tripped (Page 30)

Changing the position limits configuration in the user program (Page 32)

1.5.3.3.2 Configuration - Position limits

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

Enable hardware position limits

The check box activates the upper and lower hardware limit switches. The hardware limit switches can be used for purposes of direction reversal during a reference point approach. For details, refer to the configuration description for homing.

Low / high HW limit switch input

Select the digital input for the lower or upper hardware limit switch from the drop-down list box. The input must be interrupt-capable. The digital onboard CPU inputs and inputs of an inserted signal board can be selected as inputs for the hardware limit switches.

 CAUTION
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.

Active level

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

- "Low level" selected
0 V (FALSE) at CPU input corresponds to hardware limit switch approached
- "High level" selected
24 V (TRUE) at CPU input corresponds to hardware limit switch approached

Enable software position limits

This check box activates the upper and lower software limit switches.

NOTICE
The enable software position limits setting only affects a homed axis.

Lower / upper SW limit

Specify the position values of the lower and upper software limit switches in these fields.

Limit values (independent of the selected user unit):

- $-1.0e12 \leq \text{lower software limit switch} \leq 1.0e12$
- $-1.0e12 \leq \text{upper software limit switch} \leq 1.0e12$

The value of the upper software limit switch must be greater than or equal to the value of the lower software limit switch.

See also

Requirements for hardware limit switches (Page 28)

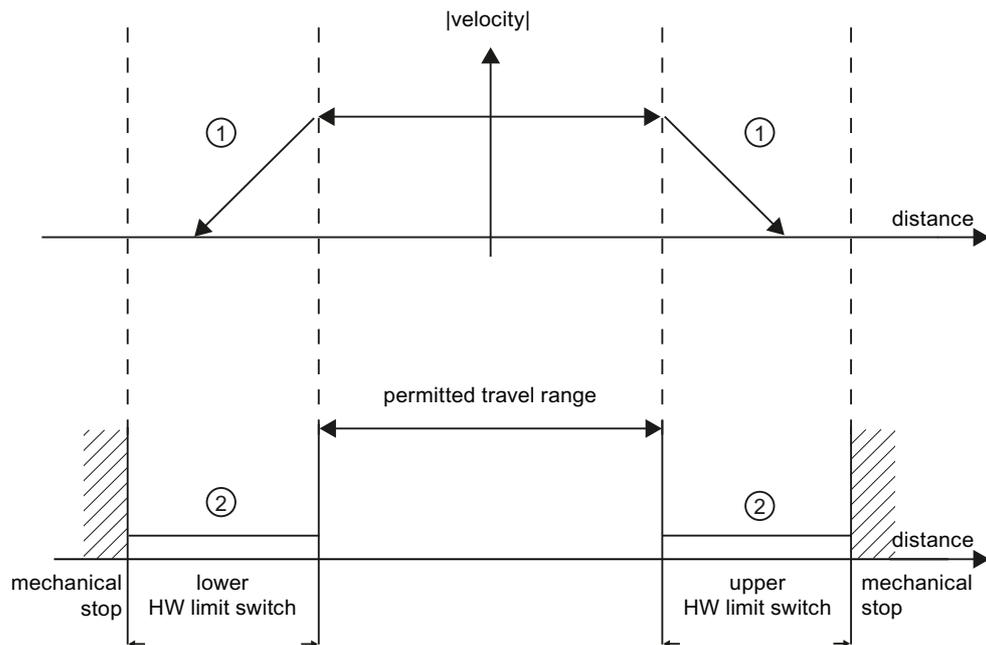
Behavior of axis when position limits is tripped (Page 30)

Changing the position limits configuration in the user program (Page 32)

1.5.3.3.3 Behavior of axis when position limits is tripped

Behavior of axis when hardware limit switches are approached

When the hardware limit switches are approached, the axis brakes to a standstill at the configured emergency deceleration. The specified emergency deceleration must be sufficient to reliably stop the axis before the mechanical stop. The following diagram presents the behavior of the axis after it approaches the hardware limit switches:



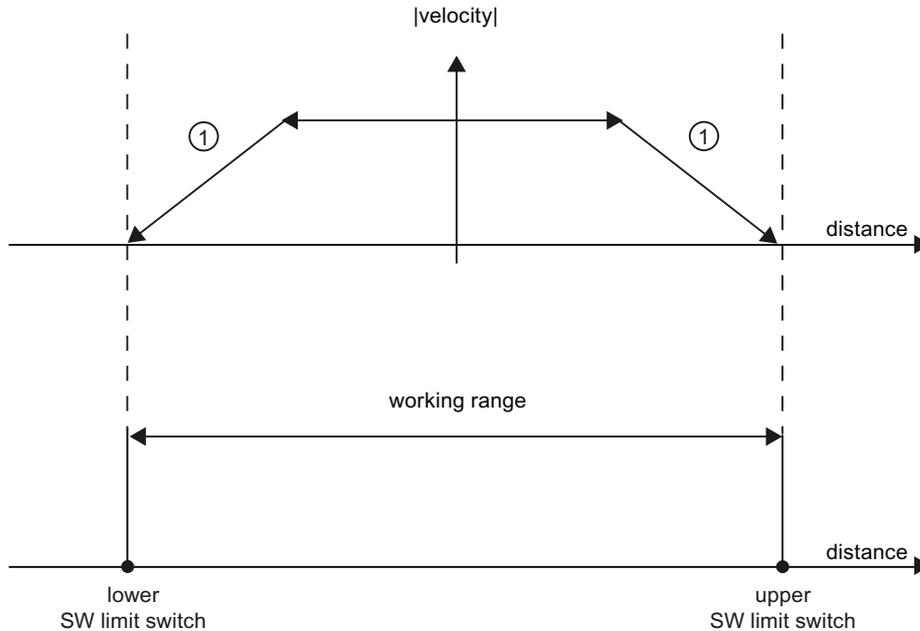
①	The axis brakes to a standstill at the configured emergency deceleration.
②	Range in which the HW limit switches signal the status "approached".

The "HW limit switch approached" error is displayed in the motion control instruction to be initiated, in "MC_Power", and in the technology object tags. 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 reached" error is displayed in the motion control instruction to be initiated, in "MC_Power", and in the technology object tags. Instructions for eliminating errors can be found in the Appendix under "List of ErrorIDs and ErrorInfos".

The circumstances under which the "SW limit switch exceeded" error is displayed can be obtained in the topics "Software limit switches in conjunction with a homing operation (Page 75)" and "Software limit switches in conjunction with dynamic changes (Page 79)".

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 28)

Configuration - Position limits (Page 28)

Changing the position limits configuration in the user program (Page 32)

1.5.3.3.4 Changing the position limits configuration in the user program

The following configuration parameters can be changed in the user program:

Hardware limit switches

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

- `<Axis name>.Config.PositionLimits_HW.Active` for activating and deactivating the hardware limit switches

Refer to the description of technology object tags 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 tags for this purpose:

- `<Axis name>.Config.PositionLimits_SW.Active` for activating and deactivating the software limit switches
- `<Axis name>.Config.PositionLimits_SW.MinPosition` for changing the position of the lower software limit switch
- `<Axis name>.Config.PositionLimits_SW.MaxPosition` for changing the position of the upper software limit switch

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

See also

Requirements for hardware limit switches (Page 28)

Configuration - Position limits (Page 28)

Behavior of axis when position limits is tripped (Page 30)

1.5.3.4 Dynamics

1.5.3.4.1 Configuration - General dynamics

In the "General dynamics" configuration window, configure the maximum velocity, the start/stop velocity, and the acceleration/deceleration of the axis.

Unit for the velocity limits

Select the physical unit for the velocity limits from the drop-down list box. The unit used here is independent of the "User unit" specified in "Configuration - General" and serves only for simple inputs.

Velocity

Define the maximum permitted velocity and the start/stop velocity of the axis in these fields. The start/stop velocity is the minimum permitted velocity of the axis.

Limit values:

The limit values indicated below refer to the user unit **pulses/s**.

- $2 \leq \text{start/stop velocity} \leq 20000/100000$ (signal board/onboard CPU outputs)
- $2 \leq \text{maximum velocity} \leq 20000/100000$ (signal board/onboard CPU outputs)

The value of the maximum velocity must be greater than the value of the start/stop velocity.

The limit values for other user units must be converted to conform to the given mechanics.

Acceleration/deceleration

Set the desired acceleration in the "Ramp-up time" or "Acceleration" fields. The desired deceleration can be set in the "Ramp-down time" or "Deceleration" fields. Motion commands initiated in the user program are executed at the selected acceleration/deceleration.

Limit values:

The limit values indicated below refer to the user unit **pulses/s²**.

- $0.28 \leq \text{acceleration} \leq 9.5e9$
- $0.28 \leq \text{deceleration} \leq 9.5e9$

The limit values for other user units must be converted to conform to the given mechanics.

Note

Changes in 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.

See also

Configuration - Dynamics emergency ramp (Page 34)

Changing the configuration of dynamics in the user program (Page 35)

1.5.3.4.2 Configuration - Dynamics emergency ramp

Configure the emergency deceleration of the axis in the "Dynamics emergency ramp" configuration window. When an error occurs and when the axis is disabled with motion control instruction "MC_Power" (input parameter StopMode = 0), the axis is brought to a standstill with this deceleration.

Velocity

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

Deceleration

Set the deceleration limit for emergency stop in the "Emergency ramp time" or "Emergency deceleration" fields.

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 stop.

The emergency deceleration must be selected based on the maximum velocity of the axis.

Limit values:

The limit values indicated below refer to the user unit **pulses/s²**.

- $0.28 \leq \text{emergency deceleration} \leq 9.5e9$

The limit values for other user units must be converted to conform to the given mechanics.

See also

Configuration - General dynamics (Page 33)

Changing the configuration of dynamics in the user program (Page 35)

1.5.3.4.3 Changing the configuration of dynamics in the user program

The following configuration parameters can be changed in the user program:

Acceleration and deceleration

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

- <Axis name>.Config.DynamicDefaults.Acceleration for changing the acceleration
- <Axis name>.Config.DynamicDefaults.Deceleration for changing the deceleration

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

Emergency deceleration

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

- <Axis name>.Config.DynamicDefaults.EmergencyDeceleration for changing the emergency deceleration

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

 WARNING
--

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

See also

Configuration - General dynamics (Page 33)

Configuration - Dynamics emergency ramp (Page 34)

1.5.3.5 Homing

1.5.3.5.1 Configuration - Homing

Configure the parameters for active and passive homing in the "Homing" configuration window. The homing method is set using the "Mode" input parameter of the motion control instruction. Here, Mode = 2 means passive homing and Mode = 3 means active homing.

Input reference point switch (active and passive homing)

Select the digital input for the reference point switch from the drop-down list box. The input must be interrupt-capable. The onboard CPU inputs and inputs of an inserted signal board can be selected as inputs for the reference point 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 reference point switch, this can result in undesired decelerations and thus inaccuracies. Depending on the reduced velocity and extent of the reference point switch, the reference point 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 reference point switch.

Allows auto reverse after reaching the hardware limit switches (active homing only)

Activate the check box to use the hardware limit switch as a reversing cam for the reference point approach. The hardware limit switches must be configured and activated for direction reversal.

If the hardware limit switch is reached during active homing, the axis brakes at the configured deceleration (not with the emergency deceleration) and reverses direction. The reference point 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 reference point approach is aborted with an error and the axis is braked at the emergency deceleration.

NOTICE
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: <ul style="list-style-type: none">• Keep the approach velocity low• Increase the configured acceleration/deceleration• Increase the distance between hardware limit switch and mechanical stop

Approach direction (active and passive homing)

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

Refer to the table under "Reference point switches" for the effect of the approach direction setting on passive homing.

Reference point switch (active and passive homing)

- **Active homing**

Here, select whether the axis is to be referenced on the left or right side of the reference point switch.

Note

Depending on the start position of the axis and the configuration of the homing parameters, the reference point approach sequence can differ from the diagram in the configuration window.

- **Passive homing**

With passive homing, the traversing motions for purposes of homing must be implemented by the user via motion commands. The side of the reference point switch on which homing occurs depends on the following factors:

- "Approach direction" configuration
- "Reference point switch" configuration
- Current travel direction during passive homing

The table below presents details on the effect of factors:

Influencing factors:			Result:
Configuration Approach direction	Configuration Reference point switch	Current travel direction	Homing on Reference point switch
Positive	"Left (negative) side"	Positive direction	left
		Negative direction	right
Positive	"Right (positive) side"	Positive direction	right
		Negative direction	left
Negative	"Left (negative) side"	Positive direction	right
		Negative direction	left
Negative	"Right (positive) side"	Positive direction	left
		Negative direction	right

Approach velocity (active homing only)

In this field, specify the velocity at which the reference point switch is to be searched for during the reference point approach.

Limit values (independent of the selected user unit):

- Start/stop velocity \leq approach velocity \leq maximum velocity

Reduced velocity (active homing only)

In this field, specify the velocity at which the axis approaches the reference point switch for homing.

Limit values (independent of the selected user unit):

- Start/stop velocity \leq reduced velocity \leq maximum velocity

Home position offset (active homing only)

If the desired reference position deviates from the position of the reference point 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 reference point switch:

1. Move the axis at reduced velocity by the value of the home position offset
2. When the position of the home position offset is reached, the axis position is set to the absolute reference position. The absolute reference position is specified via parameter "Position" of motion control instruction "MC_Home".

Limit values (independent of the selected user unit):

- $-1.0e12 \leq$ home position offset $\leq 1.0e12$

See also

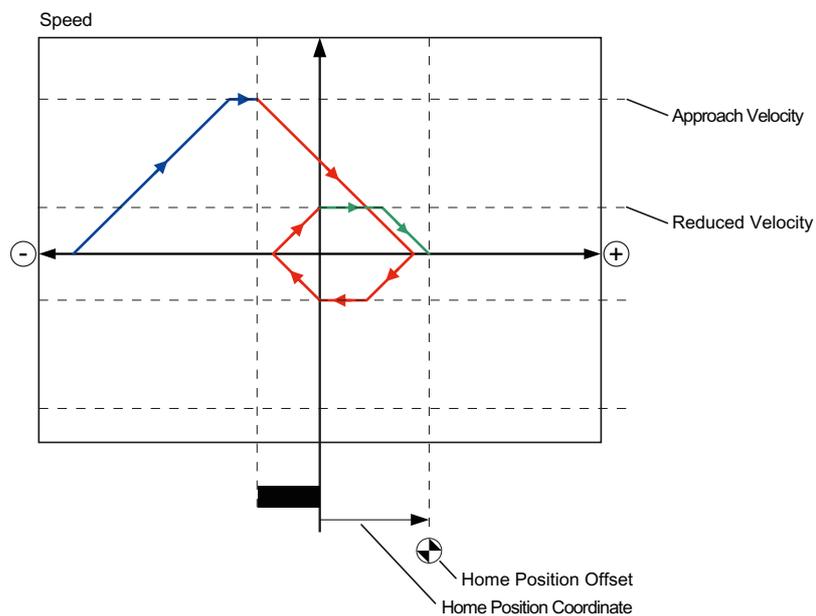
Sequence - Active homing (Page 39)

1.5.3.5.2 Sequence - Active homing

You start active homing with motion control instruction "MC_Home" (input parameter Mode = 3). Input parameter "Position" specifies the absolute reference point coordinates in this case. Alternatively, you can start active homing on the control panel for test purposes.

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

- "Approach direction" = "Positive approach direction"
- "Reference point switch" = "Right (positive) side"
- Value of "home position offset" > 0



Search phase (blue curve segment)

When active homing starts, the axis accelerates to the configured "approach velocity" and searches at this velocity for the reference point switch.

Reference point approach (red curve section)

When the reference point switch is detected, the axis in this example brakes and reverses, to be homed on the configured side of the reference point switch at the configured "reduced velocity".

Travel to reference point position (green curve segment)

After homing at the reference point switch, the axis travels to the "Reference point coordinates" at the "reduced velocity". On reaching the "Reference point coordinates", the axis is stopped at the position value that was specified in input parameter "Position" of motion control instruction "MC_Home".

See also

Configuration - Homing (Page 36)

1.6 Download to CPU

New or modified project data must be downloaded to the CPU.

Downloading the program blocks and the technology object configuration to the CPU S7-1200

Follow the steps below for downloading:

1. Select the "Program blocks" object or the "Technology objects" object in the project tree.
2. Select the "**Online > Download to device**" menu command.

Note

When blocks are downloaded to the CPU S7-1200, the TIA Portal always ensures afterwards that the offline blocks in the project and the online blocks in the CPU are consistent.

Regardless of the marking in the project tree, all new and modified offline blocks and technology objects are downloaded to the CPU.

On each loading operation, all actual values of the data blocks are reset to their initial values.

Downloading of the device configuration to the CPU S7-1200

Follow the steps below for downloading the device configuration:

1. Select the object of the CPU in the project tree
2. Select the "**Download to device > Hardware configuration**" shortcut menu command.

Downloading the device configuration, the program blocks, and the technology object configuration to the CPU S7-1200

Follow the steps below for downloading:

1. Select the object of the CPU in the project tree
2. Select the "**Online > Download to device**" menu command.

See also

Guidelines on use of motion control (Page 21)

1.7 Motion control tool - Commissioning (control panel)

Use the control panel to move the axis in manual mode, to optimize the axis settings, and to test your system.

The control panel can only be used if an online connection to the CPU is set up.

NOTICE

Response times of the axis control panel

The response time during control panel operation is dependent on the communication load of the CPU. Close all other online windows of the TIA Portal to minimize the response time.

"Manual control" button

Click "Manual control" to move the axis in manual control mode. Start by disabling the axis in the user program using motion control instruction "MC_Power". In "Manual control" mode, the control panel takes over control priority for the axis functions. The user program has no influence on the axis functions until manual control is ended.

WARNING

The Manual control is active for one axis only. A second axis could be moved in Automatic control, but this would bring about a dangerous situation.

In this case, set the second axis out of operation.

"Automatic control" button

Click "Automatic control" to end the "Manual control" mode. The control panel passes back the control priority and the axis can be controlled by the user program again. The axis must be re-enabled in the user program and homed, if required.

Complete all active traversing motions before switching to automatic control; otherwise, the axis will be braked with the emergency deceleration.

"Enable" button

Click "Enable" to enable the axis in "Manual control" mode. Once the axis is enabled, the axis control panel functions can be used.

If the axis cannot be enabled because certain conditions are not met, note the error message in the "Last error" field. Information on eliminating errors is available in the Appendix under "List of ErrorIDs and ErrorInfos". After the error has been corrected, enable the axis again.

"Disable" button

Click "Disable" if you want to temporarily disable the axis in "Manual control" mode.

"Command" area

Operation in the "Command" area is only possible if the axis is enabled. You can select one of the following command inputs:

- **Jogging**

This command is equivalent to motion control task "MC_MoveJog" in the user program.

- **Positioning**

This command is equivalent to the motion control tasks "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 task "MC_Home" in the user program.

- The "Set reference point" button corresponds to Mode = 0 (direct homing absolute)
- The "Home" button corresponds to Mode = 3 (active homing)

For active homing, the reference point switch must be configured in the axis configuration.

The values for approach velocity, reduced velocity, and home position offset are taken from the axis configuration unchanged.

Depending on the selection, the relevant fields for entry of setpoints and the buttons for starting the command are displayed.

"Axis status" area

If "Manual control" mode is activated, the current axis status and drive status are shown in the "Axis status" area. The current position and velocity of the axis are displayed at "Actual values".

Click "Acknowledge" to acknowledge all cleared errors.

Last error

In "Manual control" mode, the "Last error" field indicates the most recent error and the corresponding additional error information. Having eliminated the cause of error, click "Acknowledge" to clear the error entry.

Note

Initial values for velocity, acceleration, and deceleration

For safety reasons, when the control panel is activated the "Velocity" and "Acceleration/deceleration" parameters are initialized with values equivalent to only 10% of the configured values.

The values in the configuration view displayed by selecting **Extended parameters > Dynamics > General** are used as the basis for the initialization. The "Velocity" parameter on the control panel is derived from the "Maximum velocity" in the configuration, and the "Acceleration/deceleration" parameter from the "Acceleration".

The "Velocity" and "Acceleration/deceleration" parameters can be changed in the control panel; this does not affect the values in the configuration.

See also

Guidelines on use of motion control (Page 21)

Working with watch tables (Page 68)

1.8 Programming

1.8.1 Overview of the Motion Control statements

You control the axis with the user program using motion control instructions. The instructions start motion control tasks in the firmware that execute the desired functions.

The status of the motion control tasks 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 (Page 103)
- MC_Reset: Acknowledge error (Page 107)
- MC_Home: Home axes, set home position (Page 109)
- MC_Halt: Halt axis (Page 113)
- MC_MoveAbsolute: Absolute positioning of axes (Page 116)
- MC_MoveRelative: Relative positioning of axes (Page 119)
- MC_MoveVelocity: Move axes at preset rotational speed (Page 122)
- MC_MoveJog: Move axes in jogging mode (Page 126)

See also

Creating a user program (Page 45)

Programming notes (Page 48)

Behavior of the Motion Control commands after POWER OFF and restart (Page 50)

Monitoring active commands (Page 51)

Error displays of the Motion Control statements (Page 63)

1.8.2 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.

Requirements

- A technology object was 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 control panel.

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 advanced instructions 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

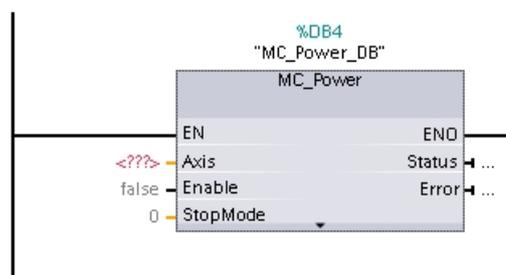
Click "Single instance" and select whether you want to define the name and number of the instance DB automatically or manually.

Multi-instance

Click "Multi-instance" and select whether you want to define the name of the multi-instance automatically or manually.

5. Click "OK".

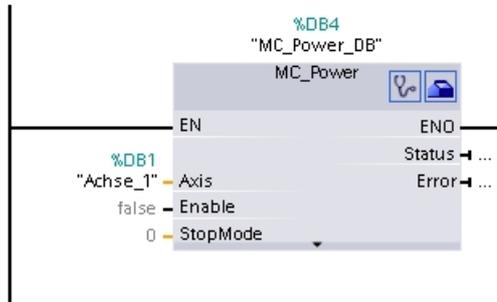
The motion control instruction "MC_Power" is inserted into the network.



Parameters marked with "<???) must be initialized; all other parameters are assigned default values.

Parameters displayed in black are required for use of the motion control instruction.

6. Double-click <??> and select the "Axis" technology object from the operand list.



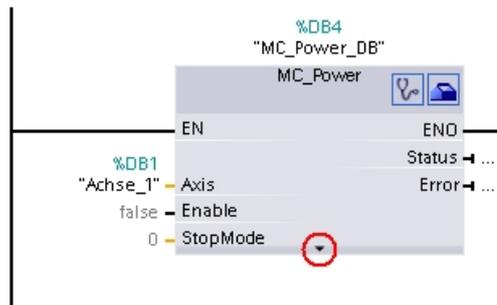
Following selection of the technology object data block, the following buttons are available:



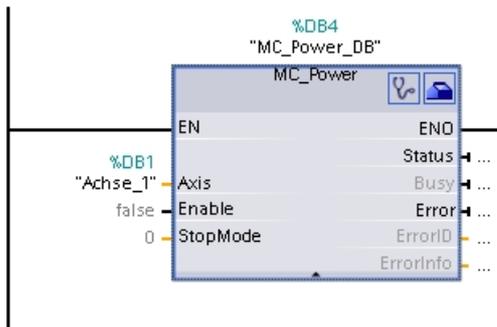
Click the stethoscope icon if you want to open the diagnostics dialog for the technology object.



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. Insert your choice of motion control instructions "MC_Reset", "MC_Home", "MC_Halt", "MC_MoveAbsolute", "MC_MoveRelative", "MC_MoveVelocity", and "MC_MoveJog" 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 tasks 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 tasks 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 44)

Programming notes (Page 48)

Behavior of the Motion Control commands after POWER OFF and restart (Page 50)

Monitoring active commands (Page 51)

Error displays of the Motion Control statements (Page 63)

1.8.3 Programming notes

When creating your user program, note the following information:

- **Cyclic call of utilized motion control instructions**

The current status of task 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 task 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 task.

An exception to this is input parameter "StopMode" of motion control instruction "MC_Power". A change in input parameter "StopMode" is also transferred when "Enable" = TRUE.

- **Programming under consideration of the status information**

In a stepwise execution of motion control tasks, make sure to wait for the active task to finish before starting a new task. Use the status messages of the motion control instruction and the "StatusBits" tag of the technology object to check for completion of the active task.

In the examples below, observe the indicated sequence. Failure to observe the sequence will display an axis or task error.

- **Axis enable with motion control instruction "MC_Power"**

You must enable the axis before it can take on motion tasks. 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 task, 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 task. 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 task, 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 task processing**

Motion control tasks for moving an axis can also be executed as overriding tasks.

If a new motion control task is started for an axis while another motion control task is active, the active task is overridden by the new task before the existing task is completely executed. The overridden task signals this using `CommandAborted = TRUE` in the motion control instruction. It is possible to override an active `MC_MoveRelative` task with a `MC_MoveAbsolute` task.

- **Avoiding multiple use of the same instance data block**

All relevant information of a motion control task is stored in its instance data block.

Do not start a new task using this instance data block, if you want to track the status of the current task. Use different instance data blocks if you want to track the tasks separately. If the same instance data block is used for multiple motion control tasks, the status and error information of the individual tasks will overwrite each other.

See also

Overview of the Motion Control statements (Page 44)

Creating a user program (Page 45)

Behavior of the Motion Control commands after POWER OFF and restart (Page 50)

Monitoring active commands (Page 51)

Error displays of the Motion Control statements (Page 63)

1.8.4 Behavior of the Motion Control commands after POWER OFF and restart

A POWER OFF or CPU-STOP aborts all active motion control tasks. All CPU outputs, including the pulse and direction outputs, are reset.

After a subsequent POWER ON or CPU restart (CPU RUN), the technology objects and the motion control tasks will be reinitialized.

All actual data of the technology objects as well as all status and error information of the previously active motion control tasks 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".

See also

Overview of the Motion Control statements (Page 44)

Creating a user program (Page 45)

Programming notes (Page 48)

Monitoring active commands (Page 51)

Error displays of the Motion Control statements (Page 63)

1.8.5 Monitoring active commands

1.8.5.1 Monitoring active commands

There are three typical groups for tracking active motion control tasks:

- Motion control instructions with output parameter "Done"
- Motion control instruction "MC_MoveVelocity"
- Motion control instruction "MC_MoveJog"

1.8.5.2 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 task is complete and the axis is at a standstill.

The tasks of the following motion control instructions have a defined conclusion:

- MC_Reset
- MC_Home
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative

The output parameter "Done" indicates the value TRUE, if the task has been successfully completed.

The output parameters "Busy", "CommandAborted", and "Error" signal that the task 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.

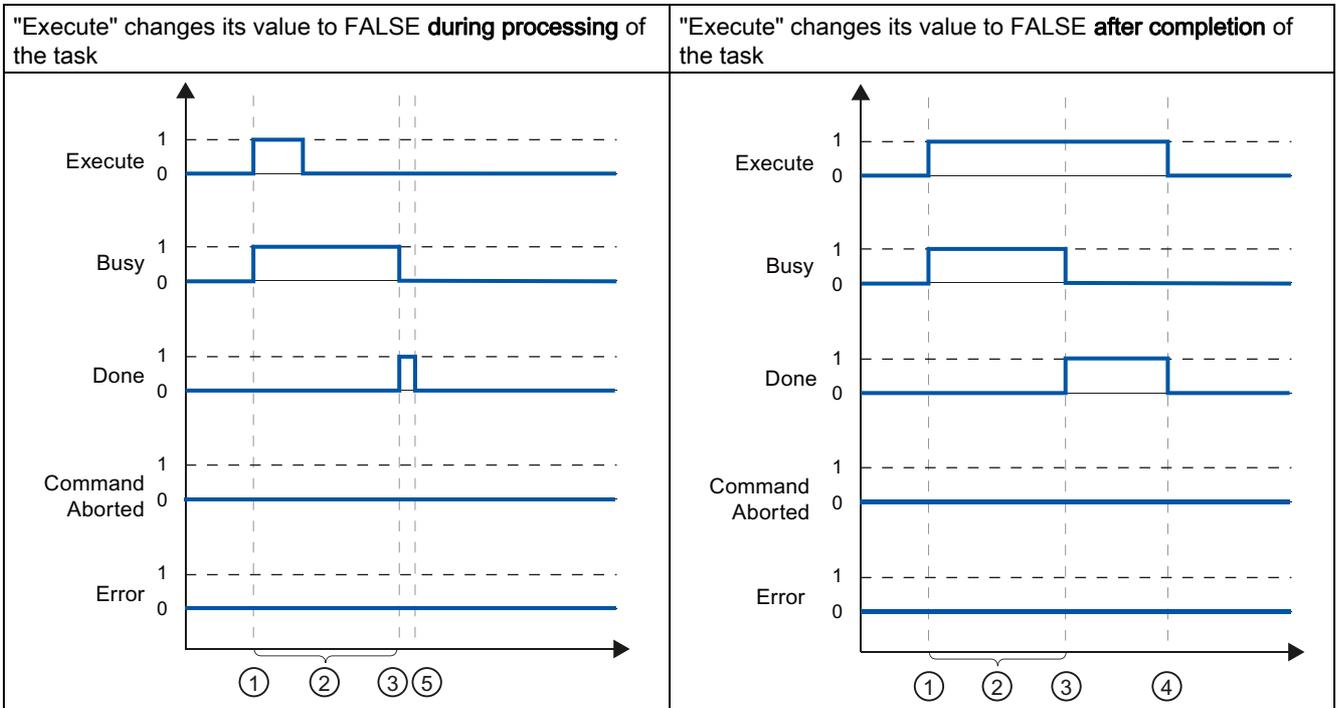
During processing of the motion control task, the output parameter "Busy" indicates the value TRUE. If the task has been completed, aborted, or stopped by an error, the output parameter "Busy" changes its value to FALSE. This change occurs regardless of the signal at input parameter "Execute".

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 task

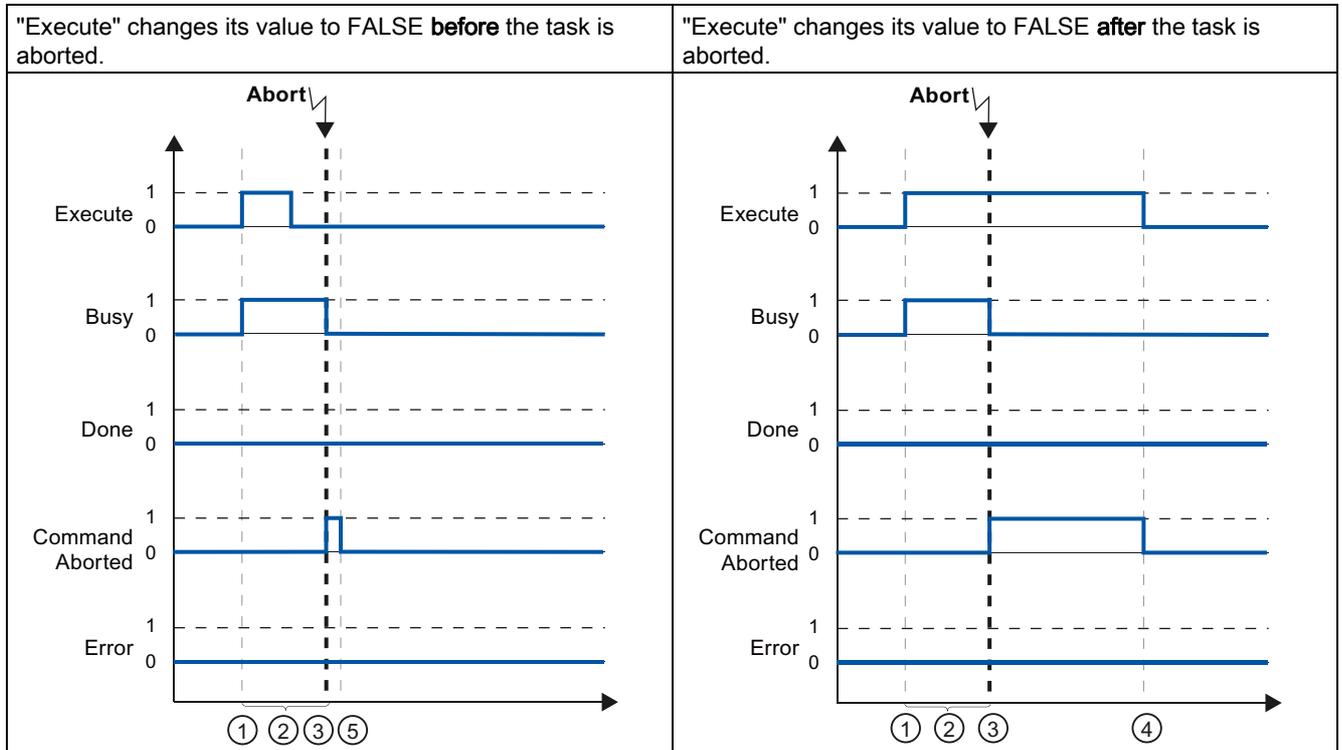
If the motion control task has been completely executed by the time of its conclusion, this is indicated by the value TRUE in output parameter "Done". The signal status of input parameter "Execute" influences the display duration in the output parameter "Done":



①	The task 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 task, or the value TRUE can be retained until after completion of the task.
②	While the task is active, the output parameter "Busy" indicates the value TRUE.
③	With conclusion of the task (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 task, then "Done" also remains TRUE and changes its value to FALSE together with "Execute".
⑤	If "Execute" has been set to FALSE before the task is complete, "Done" indicates the value TRUE for only one execution cycle.

Abort task

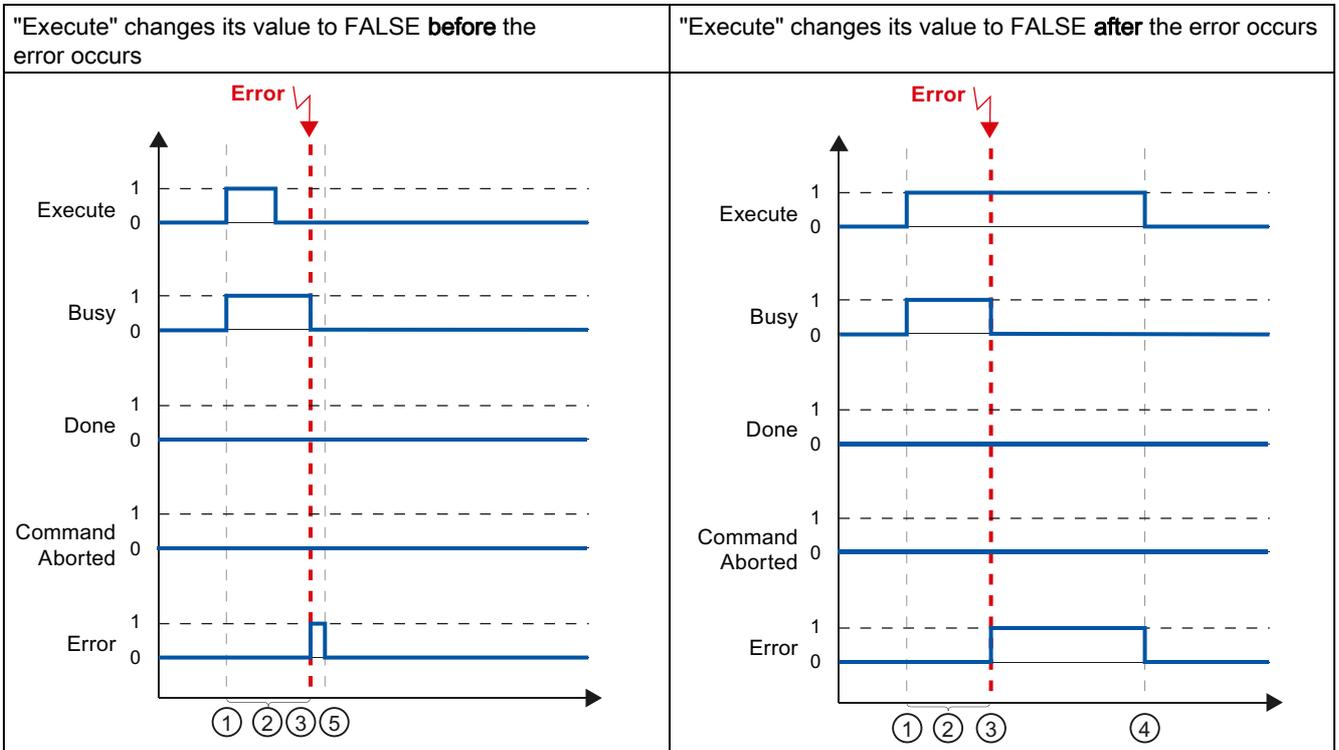
If the motion control task is aborted during execution, this is indicated by the value TRUE in output parameter "CommandAborted". The signal status of the input parameter "Execute" influences the display duration in the output parameter "CommandAborted":



- | | |
|---|---|
| ① | The task 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 task, or the value TRUE can be retained until after completion of the task. |
| ② | While the task is active, the output parameter "Busy" indicates the value TRUE. |
| ③ | During task execution, the task is aborted by another motion control task. If the task is aborted, output parameter "Busy" changes to FALSE and "CommandAborted" to TRUE. |
| ④ | If "Execute" retains the value TRUE until after the task 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 task is aborted, "CommandAborted" indicates the value TRUE for only one execution cycle. |

Error during task execution

If an error occurs during execution of the motion control task, this is indicated by the value TRUE in the output parameter "Error". The signal status of the input parameter "Execute" influences the display duration in the output parameter "Error":



- | | |
|---|---|
| ① | The task 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 task, or the value TRUE can be retained until after completion of the task. |
| ② | While the task is active, the output parameter "Busy" indicates the value TRUE. |
| ③ | An error occurred during task 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. |

1.8.5.3 Motion control instruction MC_MoveVelocity

The tasks of motion control instruction "MC_MoveVelocity" do not have a defined end. The task 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 task is complete when the parameterized velocity has been reached and input parameter "Execute" has been set to the value FALSE. However, the axis motion is not yet complete upon completion of the task. For example, the axis motion can be stopped with motion control task "MC_Halt".

The output parameters "Busy", "CommandAborted", and "Error" signal that the task is still being processed, has been aborted or an error is pending.

During execution of the motion control task, output parameter "Busy" indicates the value TRUE. If the task 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".

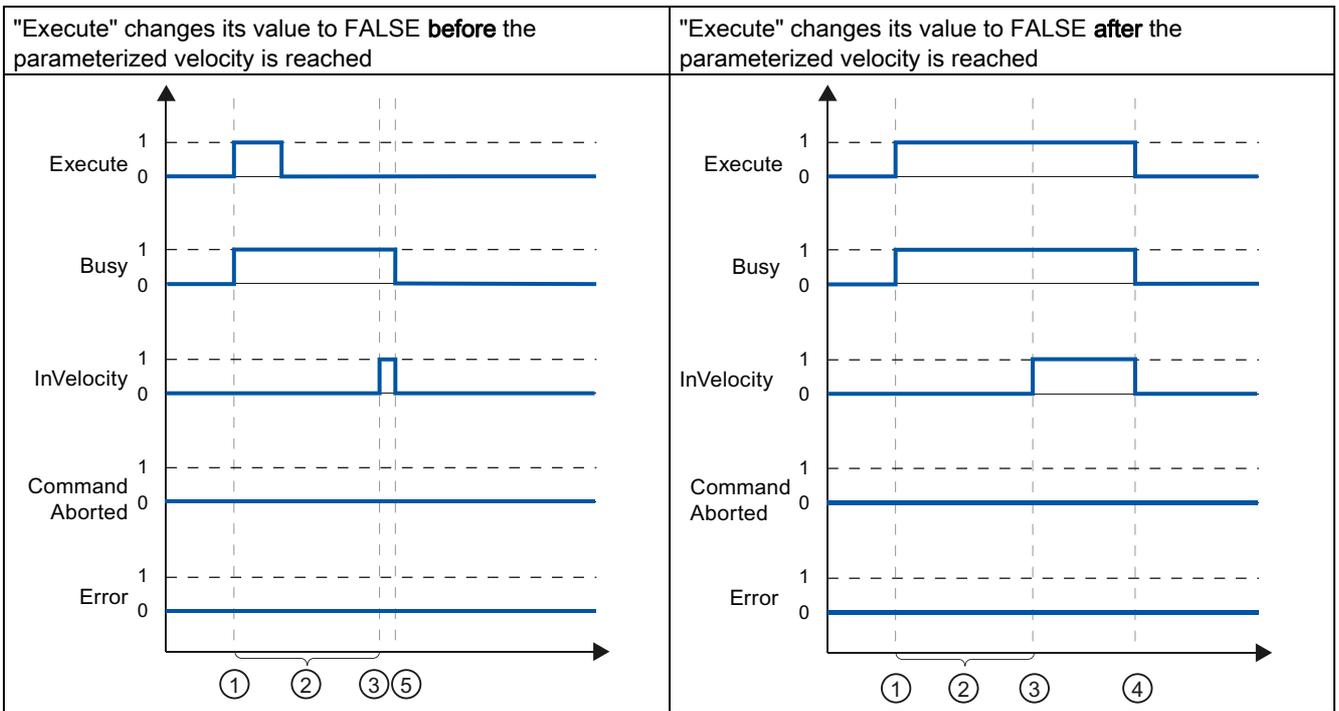
The output parameters "InVelocity", "CommandAborted", and "Error" indicate the value TRUE for at least one cycle, when their conditions are met. These status messages are latched while input parameter "Execute" is set to TRUE.

The behavior of the status bits is presented below for various example situations:

The parameterized velocity is reached

If the motion control task has been executed by the time the parameterized velocity is reached, this is indicated by the value TRUE in output parameter "InVelocity".

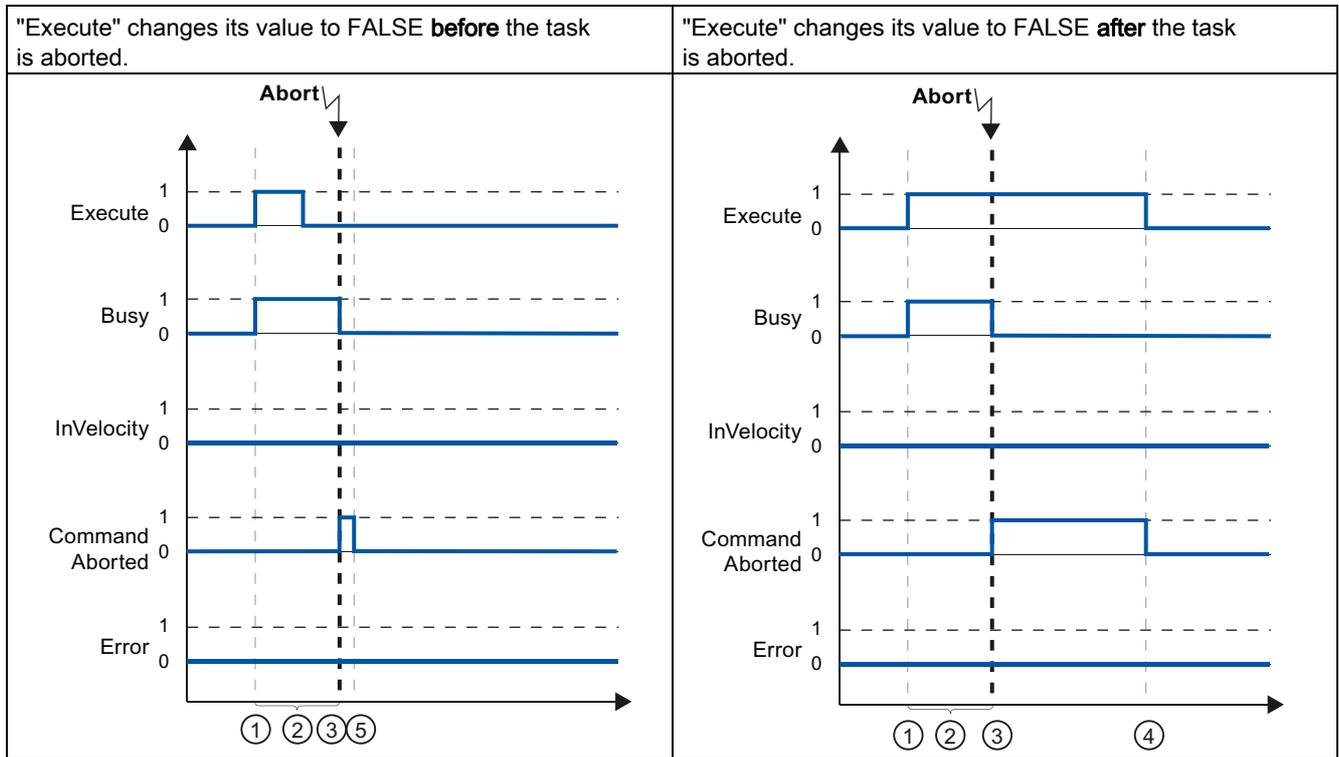
The signal status of the input parameter "Execute" influences the display duration in the output parameter "InVelocity":



①	The task is started with a positive edge at the input parameter "Execute". Depending on the programming, "Execute" can be reset to the value FALSE event before the parameterized velocity is reached, or alternatively only after it has been reached.
②	While the task 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	If "Execute" retains the value TRUE even after the parameterized velocity has been reached, the task remains active. "InVelocity" and "Busy" retain the value TRUE and only change their status to FALSE together with "Execute".
⑤	If "Execute" has been reset to FALSE before the parameterized velocity is reached, the task is complete when the parameterized velocity is reached. "InVelocity" indicates the value TRUE for one execution cycle and changes to FALSE together with "Busy".

The task is aborted prior to reaching the parameterized velocity

If the motion control task 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 in output parameter "CommandAborted".



①	The task 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 task, or the value TRUE can be retained until after the task is aborted.
②	While the task is active, the output parameter "Busy" indicates the value TRUE.
③	During task execution, the task is aborted by another motion control task. If the task is aborted, output parameter "Busy" changes to FALSE and "CommandAborted" to TRUE.
④	If "Execute" retains the value TRUE until after the task 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 task is aborted, "CommandAborted" indicates the value TRUE for only one execution cycle.

Note

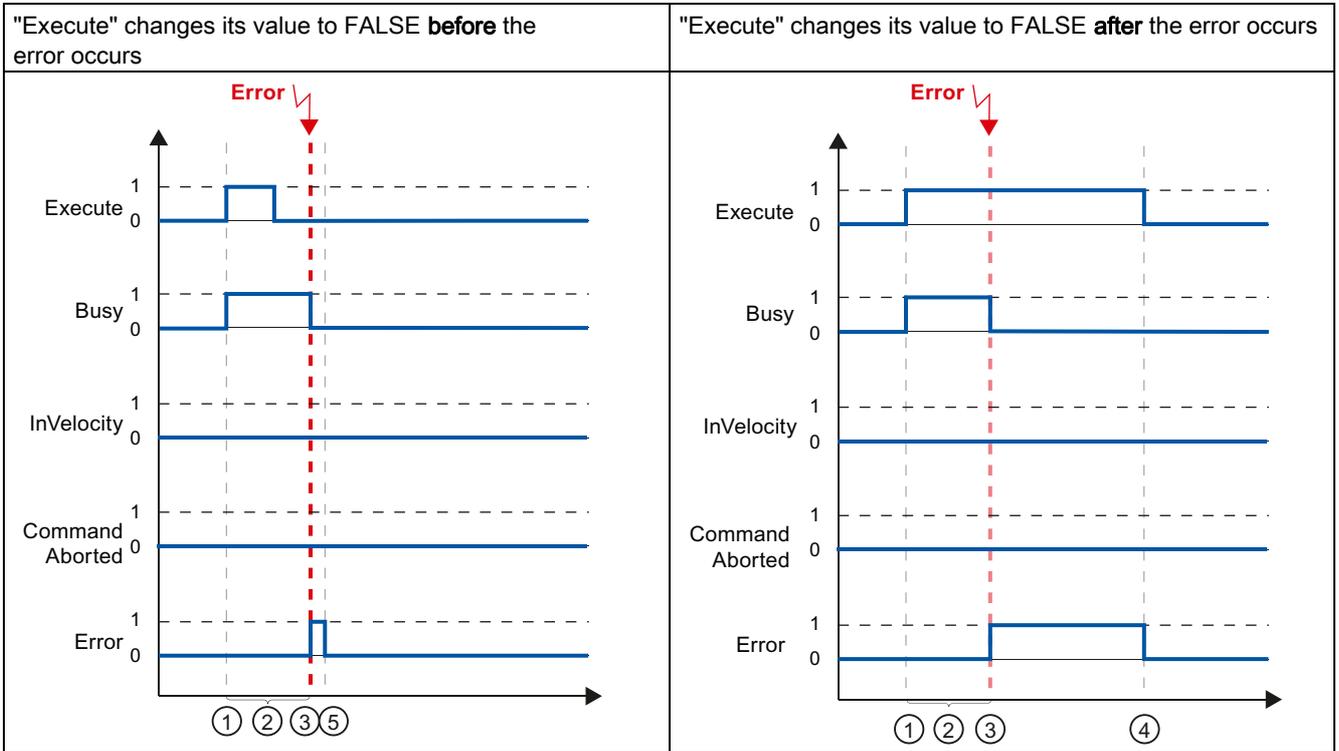
Under the following conditions, an abort is not indicated in output parameter "CommandAborted":

The parameterized velocity has been reached, input parameter "Execute" has the value FALSE, and a new motion control task is initiated.

When the parameterized velocity is reached and input parameter "Execute" has the value FALSE, the task is complete. Therefore, the start of a new task is not indicated as an abort.

An error has occurred prior to reaching the parameterized velocity

If an error occurs during execution of the motion control task 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 in the output parameter "Error":



- | | |
|---|---|
| ① | The task 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 task, or the value TRUE can be retained until after the error has occurred. |
| ② | While the task is active, the output parameter "Busy" indicates the value TRUE. |
| ③ | An error occurred during task 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).

When the parameterized velocity is reached and input parameter "Execute" has the value FALSE, the task is complete. After completion of the task, the axis error is only indicated in the motion control instruction "MC_Power".

1.8.5.4 Motion control instruction MC_MoveJog

The tasks of motion control instruction "MC_MoveJog" implement a jog operation.

The motion control tasks "MC_MoveJog" do not have a defined end. The task 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 task is still being processed, has been aborted or an error is pending.

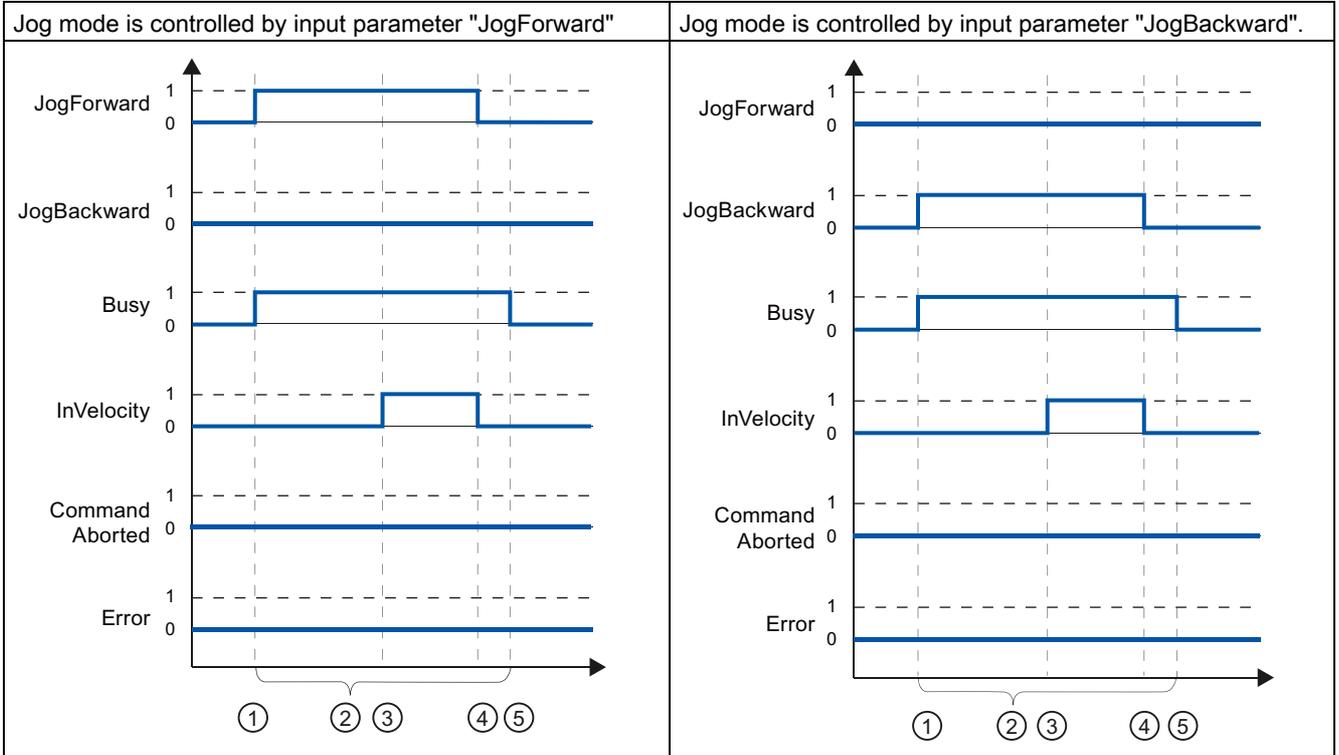
During processing of the motion control task, the output parameter "Busy" indicates the value TRUE. If the task has been completed, aborted, or stopped by an error, the output parameter "Busy" changes its value to FALSE.

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

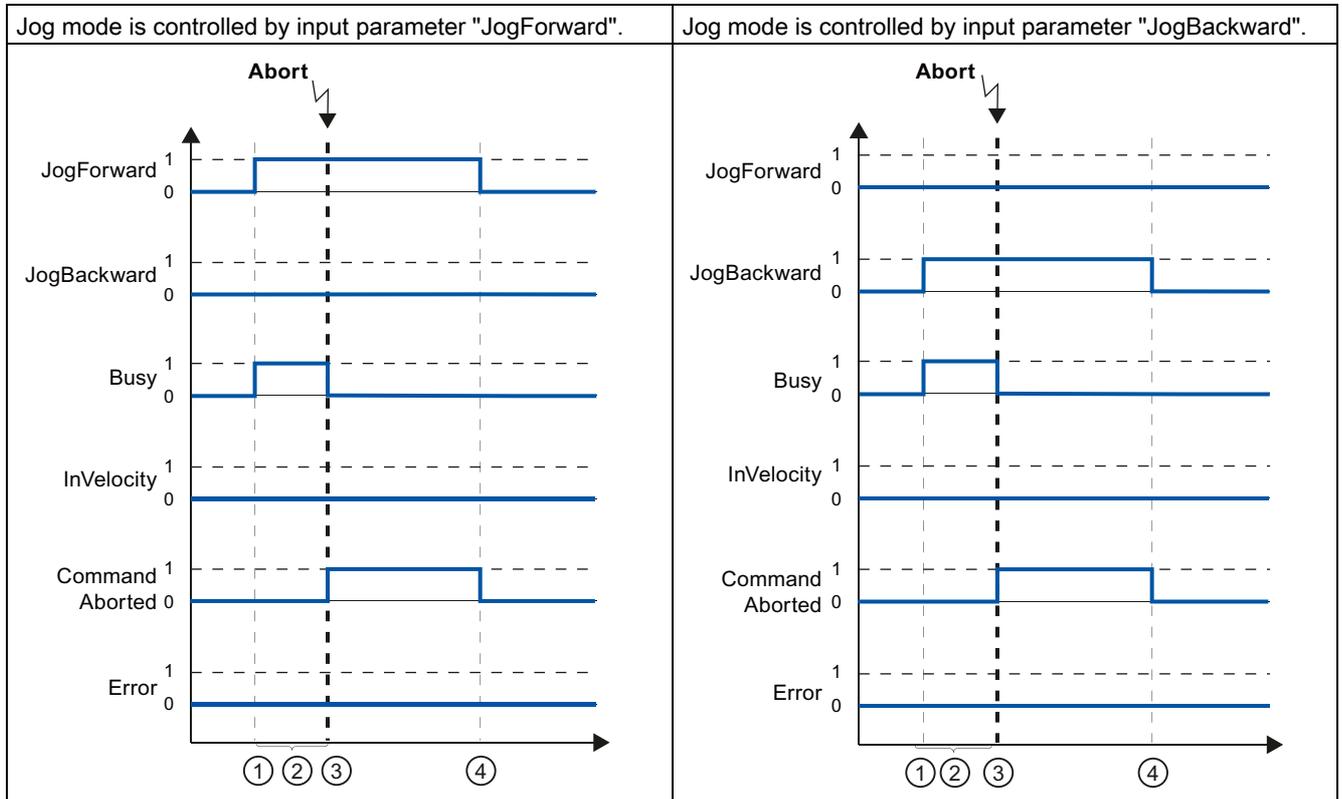
If the motion control task has been executed by the time the parameterized velocity is reached, this is indicated by the value TRUE in output parameter "InVelocity".



①	The task is started with a positive edge at the input parameter "JogForward" or "JogBackward".
②	While the task 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 task is complete and the output parameter "Busy" changes its value to FALSE.

The task is aborted during execution

If the motion control task 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.



①	The task is started with a positive edge at the input parameter "JogForward" or "JogBackward".
②	While the task is active, the output parameter "Busy" indicates the value TRUE.
③	During task execution, the task is aborted by another motion control task. If the task 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.

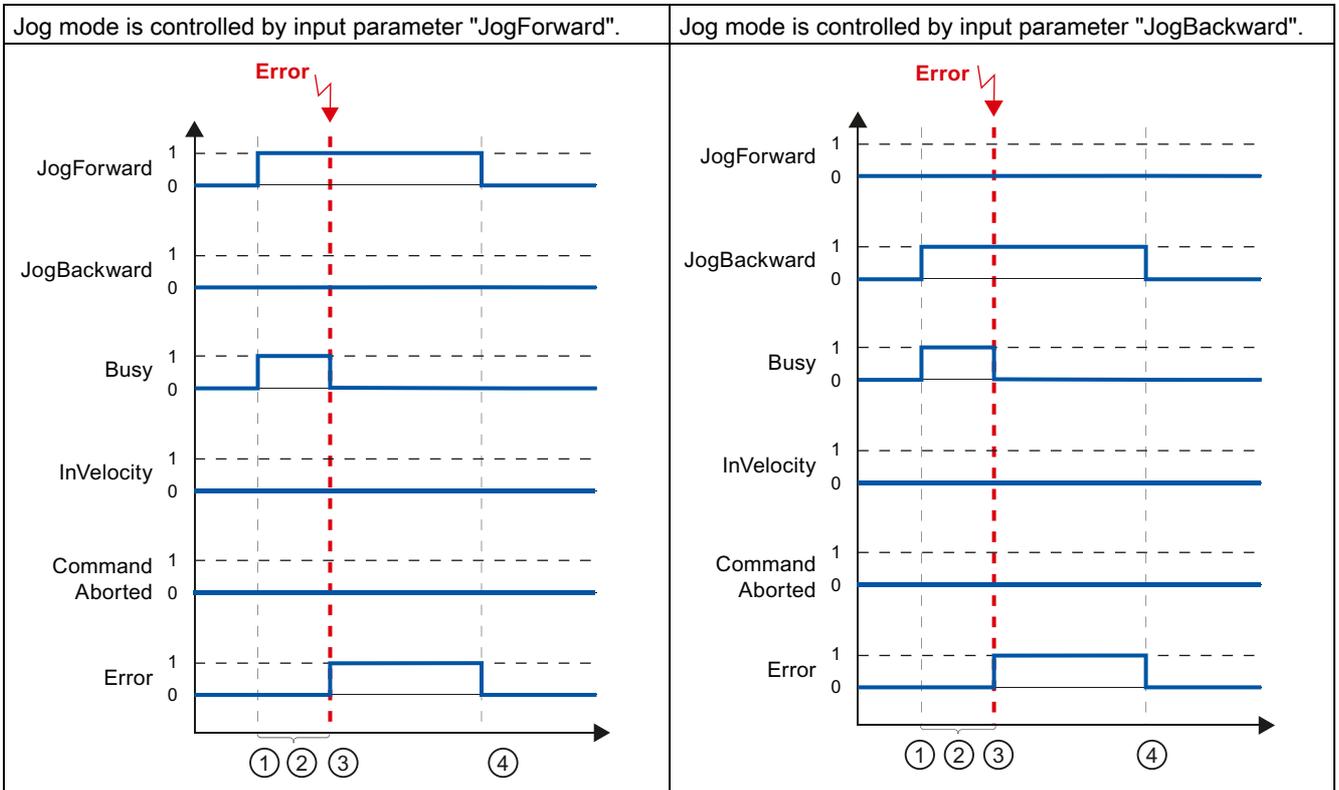
Note

The task 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 task is initiated.

An error has occurred during task execution

If an error occurs during execution of the motion control task, 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.



①	The task is started with a positive edge at the input parameter "JogForward" or "JogBackward".
②	While the task is active, the output parameter "Busy" indicates the value TRUE.
③	An error occurred during task 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).

1.8.6 Error displays of the Motion Control statements

The motion control instructions indicate any motion control task and technology object errors in the output parameters "Error", "ErrorID" and "ErrorInfo" 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 task, 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 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. The errors are indicated only in the error-triggering motion control instruction.
- **Parameterization error of 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. The errors are indicated only in the error-triggering motion control instruction.
- **Configuration error (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. If the axis is in motion, it is stopped according to the "StopMode" in motion control instruction "MC_Power". The error is indicated in the error-triggering motion control instruction and in motion control instruction "MC_Power".
- **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 44)

Creating a user program (Page 45)

Programming notes (Page 48)

Behavior of the Motion Control commands after POWER OFF and restart (Page 50)

Monitoring active commands (Page 51)

1.9 Motion control tool - Diagnostics

1.9.1 Status and error bits

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 status error messages have the following meaning:

Status of the axis

Status	Description
Enabled	The axis is enabled and ready to be controlled via motion control tasks. (Tag of technology object: <Axis name>.StatusBits.Enable)
Homed	The axis is homed and is capable of executing absolute positioning tasks of motion control instruction "MC_MoveAbsolute". The axis does not have to be homed for relative homing. Special situations: <ul style="list-style-type: none"> • 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 technology object: <Axis name>.StatusBits.HomingDone)
Error	An error has occurred in the "Axis" technology object. More information about the error is available in automatic control at the ErrorID and ErrorInfo parameters of the motion control instructions. In manual mode, the "Last error" field of the control panel displays detailed information about the cause of error. (Tag of technology object: <Axis name>.StatusBits.Error)
Control panel active	The "Manual control" mode was enabled in the control panel. The control panel has control priority over the "Axis" technology object. The axis cannot be controlled from the user program. (Tag of technology object: <Axis name>.StatusBits.ControlPanelActive)

Drive status

Status	Description
Drive ready	The drive is ready for operation. (Tag of technology object: <Axis name>.StatusBits.DriveReady)
Error	The drive has reported an error after failure of its ready signal. (Tag of technology object: <Axis name>.ErrorBits.DriveFault)

Status of the axis motion

Status	Description
Standstill	The axis is at a standstill. (Tag of technology object: <Axis name>.StatusBits.StandStill)
Accelerating	The axis accelerates. (Tag of technology object: <Axis name>.StatusBits.Acceleration)
Constant velocity	The axis travels at constant velocity. (Tag of technology object: <Axis name>.StatusBits.ConstantVelocity)
Decelerating	The axis decelerates (slows down). (Tag of technology object: <Axis name>.StatusBits.Deceleration)

Status of the motion mode

Status	Description
Positioning	The axis executes a positioning task of motion control instruction "MC_MoveAbsolute" or "MC_MoveRelative" or of the control panel. (Tag of technology object: <Axis name>.StatusBits.PositioningCommand)
Speed Command	The axis executes a task at set speed of motion control instruction "MC_MoveVelocity" or "MC_MoveJog" or of the control panel. (Tag of technology object: <Axis name>.StatusBits.SpeedCommand)
Homing	The axis executes a homing task of motion control instruction "MC_Home" or the control panel. (Tag of technology object: <Axis name>.StatusBits.Homing)

Error bits

Error	Description
Min software limit reached	The lower software limit switch has been reached. (Tag of technology object: <Axis name>.ErrorBits.SwLimitMinReached)
Min software limit exceeded	The lower software limit switch has been exceeded. (Tag of technology object: <Axis name>.ErrorBits.SwLimitMinExceeded)
Max software limit reached	The upper software limit switch has been reached. (Tag of technology object: <Axis name>.ErrorBits.SwLimitMaxReached)
Max software limit exceeded	The upper software limit switch has been exceeded. (Tag of technology object: <Axis name>.ErrorBits.SwLimitMaxExceeded)
Negative hardware limit	The lower hardware limit switch has been approached. (Tag of technology object: <Axis name>.ErrorBits.HwLimitMin)
Positive hardware limit	The upper hardware limit switch has been approached. (Tag of technology object: <Axis name>.ErrorBits.HwLimitMax)
PTO and HSC already used	A second axis is using the same PTO and HSC and is enabled with "MC_Power". (Tag of technology object: <Axis name>.ErrorBits.HwUsed)
Configuration error	The "Axis" technology object was incorrectly configured or editable configuration data were modified incorrectly during runtime of the user program. (Tag of technology object: <Axis name>.ErrorBits.ConfigFault)
General Error	An internal error has occurred. (Tag of technology object: <Axis name>.ErrorBits.SystemFault)

See also

Motion status (Page 67)
 StatusBits. tags (Page 98)
 ErrorBits. tags (Page 100)

1.9.2 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:

Motion status

Status	Description
Target position	The "Target position" field indicates the current target position of an active positioning task of motion control instruction "MC_MoveAbsolute" or "MC_MoveRelative" or of the control panel. The value of the "Target position" is only valid during execution of a positioning task. (Tag of technology object: <Axis name>.MotionStatus.TargetPosition)
Current position	The "Current position" field indicates the current axis position. If the axis is not homed, the value indicates the position value relative to the enable position of the axis. (Tag of technology object: <Axis name>.MotionStatus.Position)
Current velocity	The "Current velocity" field indicates the actual axis velocity. (Tag of technology object: <Axis name>.MotionStatus.Velocity)

Dynamic limits

Dynamic limit	Description
Velocity	The "Velocity" field indicates the configured maximum velocity of the axis. (Tag of technology object: <Axis name>.Config.DynamicLimits.MaxVelocity)
Acceleration	The "Acceleration" field indicates the currently configured acceleration of the axis. (Tag of technology object: <Axis name>.Config.DynamicDefaults.Acceleration)
Deceleration	The "Deceleration" field indicates the currently configured deceleration of the axis. (Tag of technology object: <Axis name>.Config.DynamicDefaults.Deceleration)

See also

Status and error bits (Page 64)

MotionStatus. tags (Page 97)

1.10 Working with watch tables

Use watch tables if you want to monitor and modify tags of motion control instructions or the "Axis" technology object during commissioning.

To monitor and modify tags, you must specify the complete name of the tag, including object name and all structure names in a watch table.

Example: <Axis name>.Config.DynamicDefaults.Acceleration)

Tip:

You can use a copy & paste operation to avoid entering long tag names.

Procedure

To insert the tag names, follow the steps described below:

1. In the project tree, select the instance data block or the technology object of the axis.
2. **Parameters of the motion control instruction**
 - Right-click and select the **Open** command in the shortcut menu.
Tags of the technology object
 - Right-click and select the **Open in editor** command in the shortcut menu.
3. **Parameters of the motion control instruction**
 - Select the lines of the tags in the Input or Output area
Tags of the technology object
 - In the Static area, open the relevant structures and select the lines of the tags
4. Select the **Edit > Copy** menu command.
5. Double-click to open the watch table.
6. Select the line starting at which the tags are to be inserted
7. Select the **Edit > Paste** menu command.

Insert the tags with their complete names in the watch table.

WARNING

The watch table also gives you write access to tags whose use is blocked for safety reasons in the user program. Modifying these tags can result in damage to the current axis configuration and to undefined responses of the axis. Only modify those tags whose access is marked with "RW" in the tag list of the technology object.

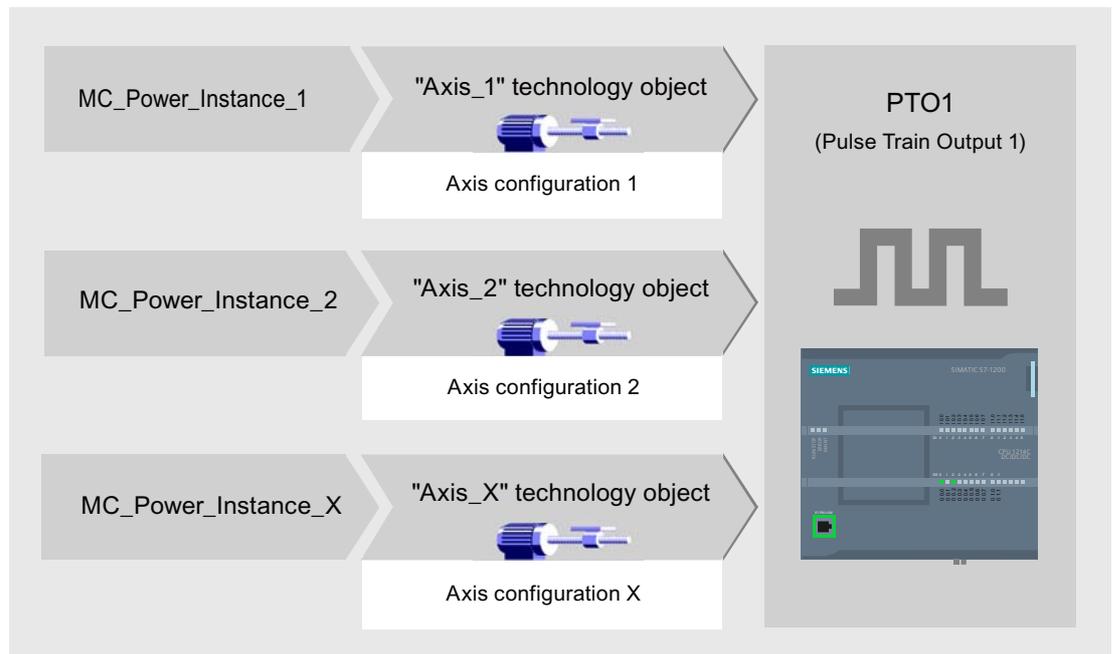
See also

Motion control tool - Commissioning (control panel) (Page 41)

1.11 Appendix

1.11.1 Using multiple axes with the same PTO

Use the motion control functionality of the CPU S7-1200 to run multiple "Axis" technology objects with the same PTO 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:

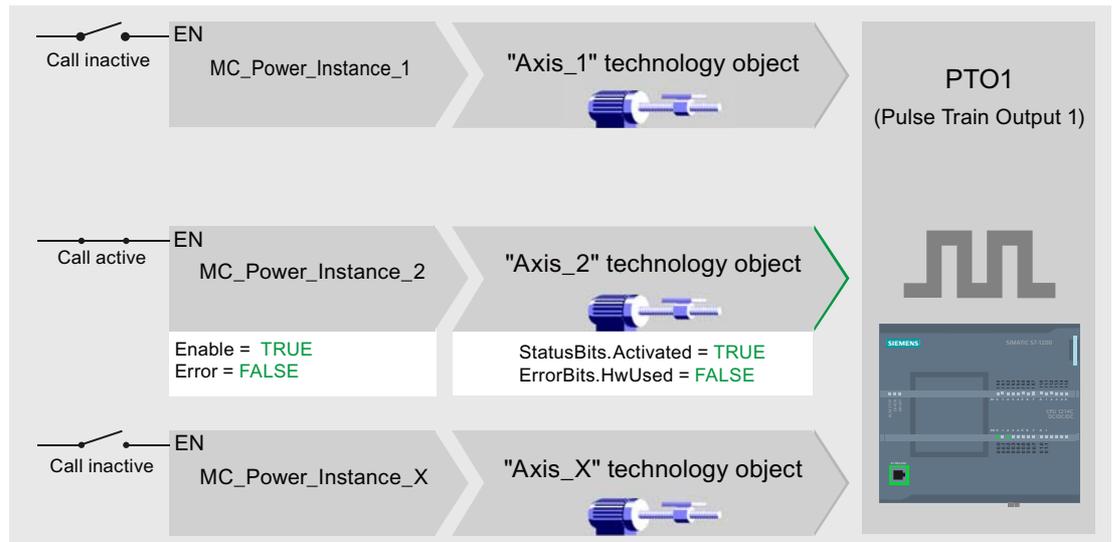


In this example, more than one "Axis" technology object, each with its own axis configuration, uses the same PTO. Each "Axis" must be called in the user program with a separate call of motion control instruction "MC_Power" with a separate instance data block. Only one "Axis" at a time may use the PTO. The axis that is currently using the PTO indicates this with tag <Axis name>.StatusBits.Activated = TRUE.

Switching between "Axis" technology objects

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:



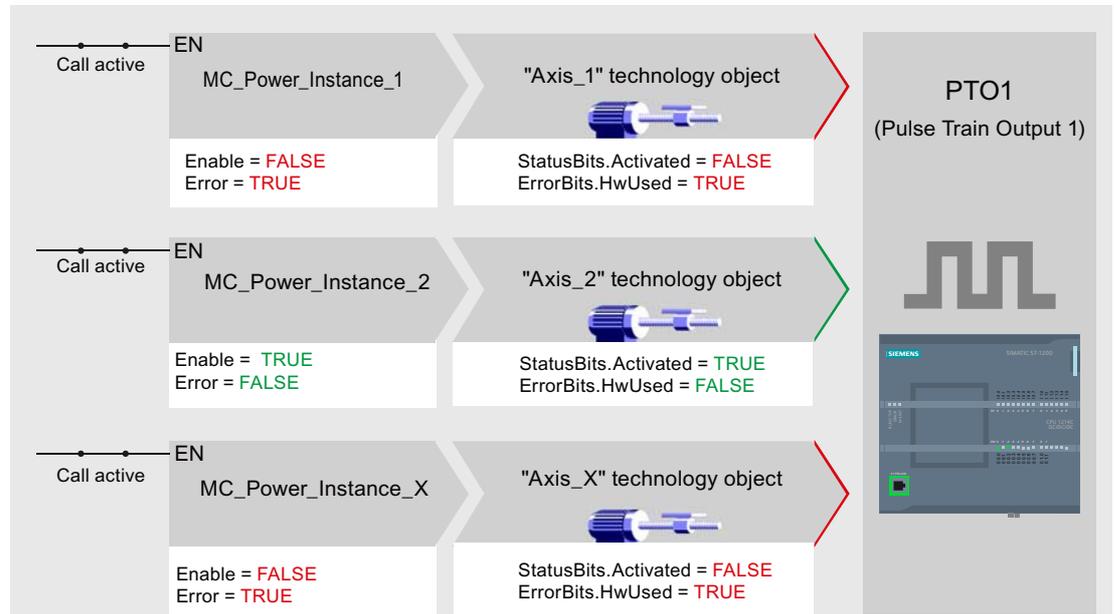
The tags of the activated axis (here "Axis_2") show the following typical indicators in the user program:

- <Axis name>.StatusBits.Activated = TRUE
- <Axis name>.ErrorBits.HwUsed = FALSE

To switch to the "Axis" technology object, follow the steps described below. In the example, a switch is made from "Axis_2" to "Axis_1":

1. End any active traversing motions of activated "Axis_2"
2. Disable "Axis_2" with the associated motion control instruction "MC_Power" using input parameter Enable = FALSE
3. To verify that "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 "Axis_2".
5. Activate the conditional call of the motion control instructions for "Axis_1". On the first call of the corresponding motion control instruction "MC_Power", "Axis_2" becomes deactivated and "Axis_1" becomes activated.
6. Enable "Axis_1" with motion control instruction "MC_Power" using the input parameter Enable = TRUE
7. To verify that "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 "Axis_2"), this axis becomes active.

In contrast to the conditional call, the motion control instructions of the deactivated axes (here "Axis_1" and "Axis_x") will 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 72)

Tracking tasks from higher priority classes (execution levels) (Page 73)

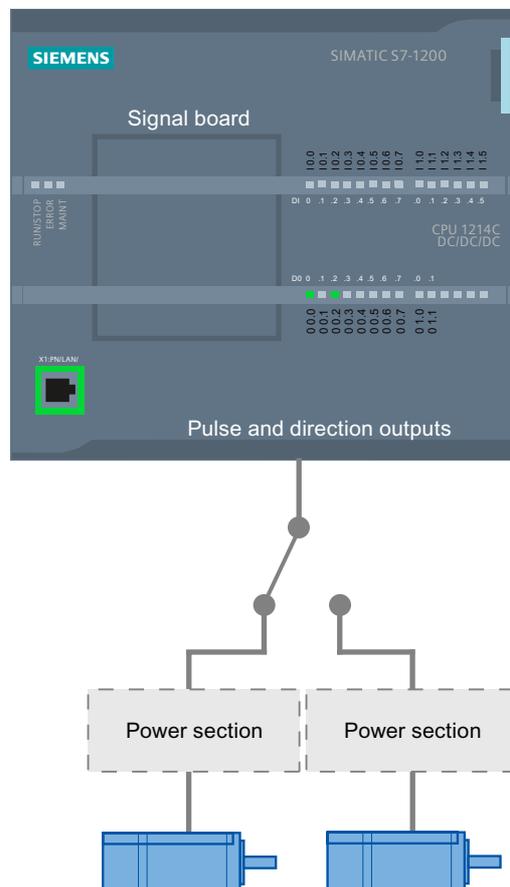
Special cases for use of software limit switches (Page 75)

List of ErrorIDs and ErrorInfos (Page 81)

Tags of the technology object (Page 87)

1.11.2 Using multiple drives with the same PTO

If multiple drives are to be used alternately, they can be run with a common PTO using a switchover approach. The following diagram represents the basic circuit design:



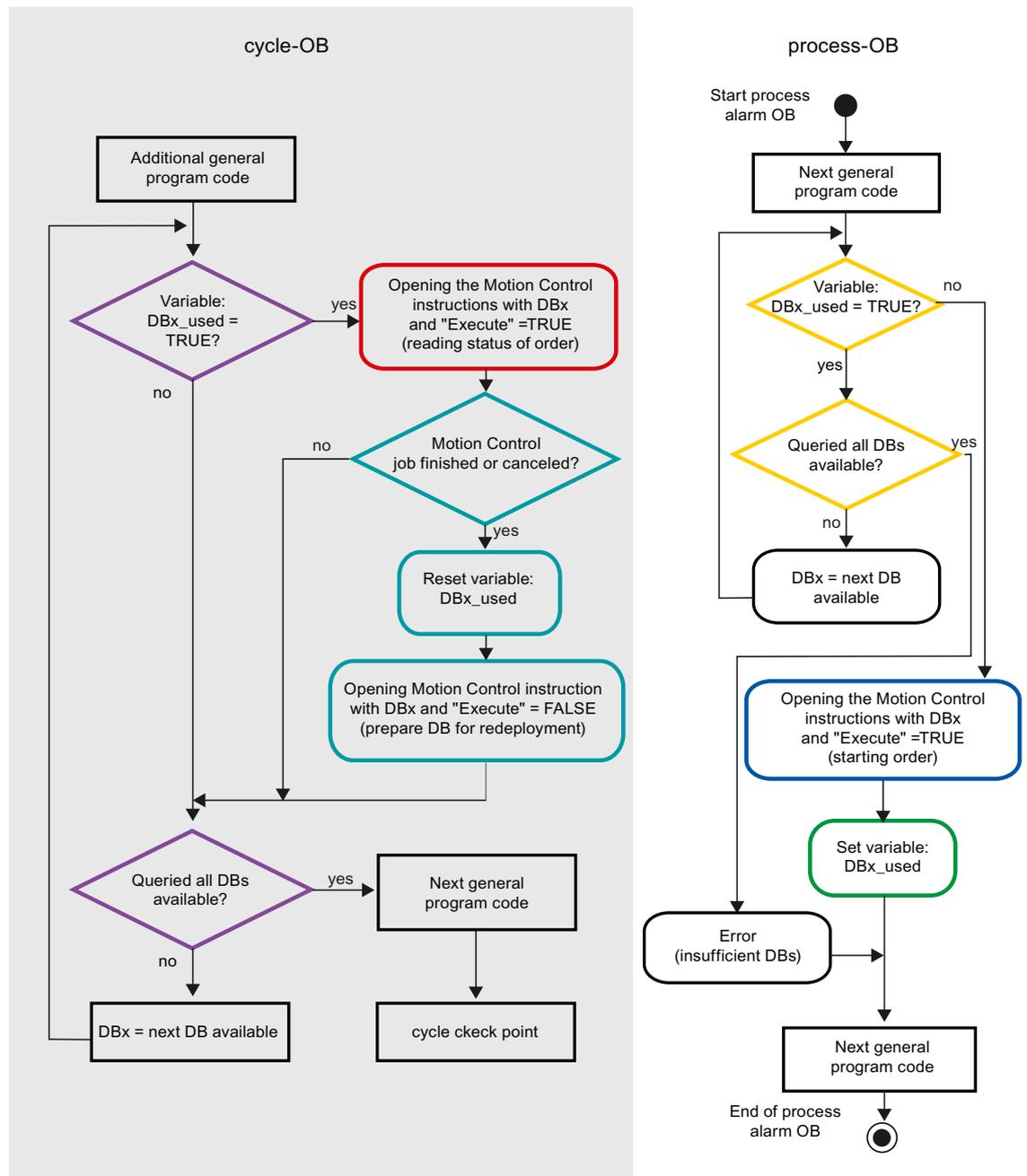
The switchover between drives can be controlled, if required, by the user program via a digital output. If different axis configurations are required for the different drives, a switchover between these configurations is required for the PTO. For additional information on this topic, refer to "Using multiple axes with the same PTO (Page 69)".

See also

- Using multiple axes with the same PTO (Page 69)
- Tracking tasks from higher priority classes (execution levels) (Page 73)
- Special cases for use of software limit switches (Page 75)
- List of ErrorIDs and ErrorInfos (Page 81)
- Tags of the technology object (Page 87)

1.11.3 Tracking tasks from higher priority classes (execution levels)

Depending on the application, it may be necessary to start motion control tasks (for example, interrupt-controlled) in a higher priority class (execution level). Because the motion control task cannot be tracked, or only inadequately tracked, in the higher priority class, the option exists to perform tracking in the program cycle OB. For every start of the motion control task, an instance data block that is not currently being utilized must be available. Refer to the following flow chart to see how you start motion control tasks 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 tasks you want to start, you will have to generate a sufficient number of instance data blocks. Users determine which instance data block is used at the moment in the "DBx_used" tags.

Start of motion control task in the hardware interrupt OB

Binary queries of the "DBx_used" tags (orange) are used to find an instance data block that is currently not in use. If such an instance data block is found, the motion control task is started with this instance data block and the utilized instance data block is marked as "used" (green).

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 tasks in the program cycle OB

All instance data blocks available in the program 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 task 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 task is complete or has been aborted, the following actions are taken next (blue green):

- Reset of "DBx_used" tag
- Call of motion control instruction with input parameter Execute = FALSE

This completes the task tracking, and the instance data block is now available for use again.

See also

Using multiple axes with the same PTO (Page 69)

Using multiple drives with the same PTO (Page 72)

Special cases for use of software limit switches (Page 75)

List of ErrorIDs and ErrorInfos (Page 81)

Tags of the technology object (Page 87)

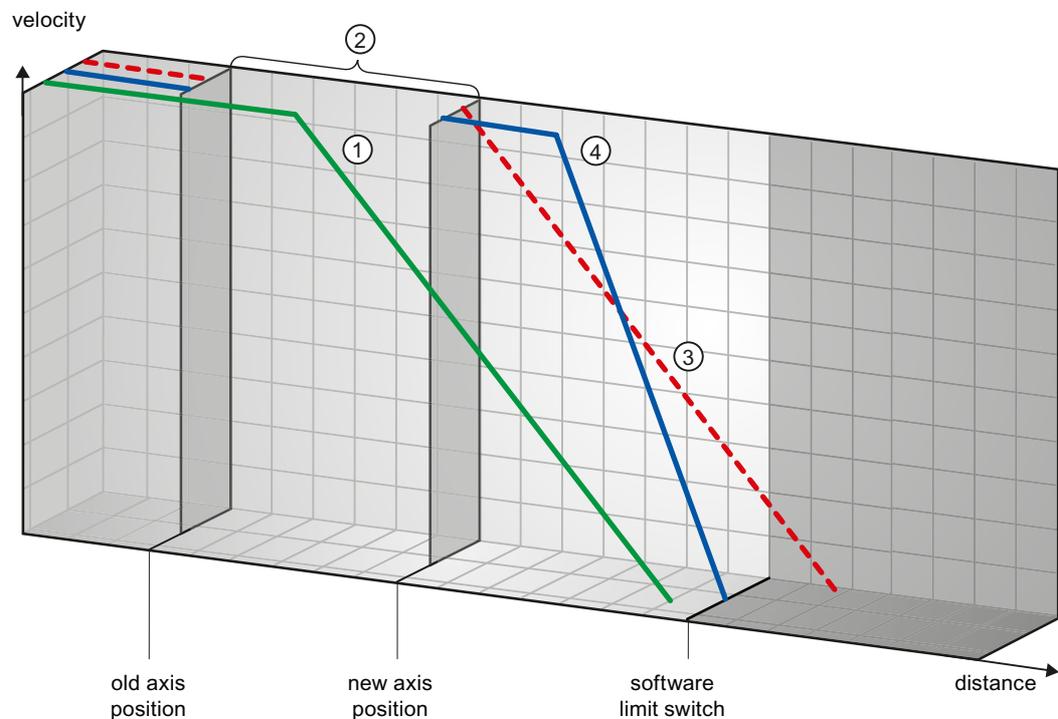
1.11.4 Special cases for use of software limit switches

1.11.4.1 Software limit switches in conjunction with a homing operation

Due to unfavorably parameterized homing tasks, 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:

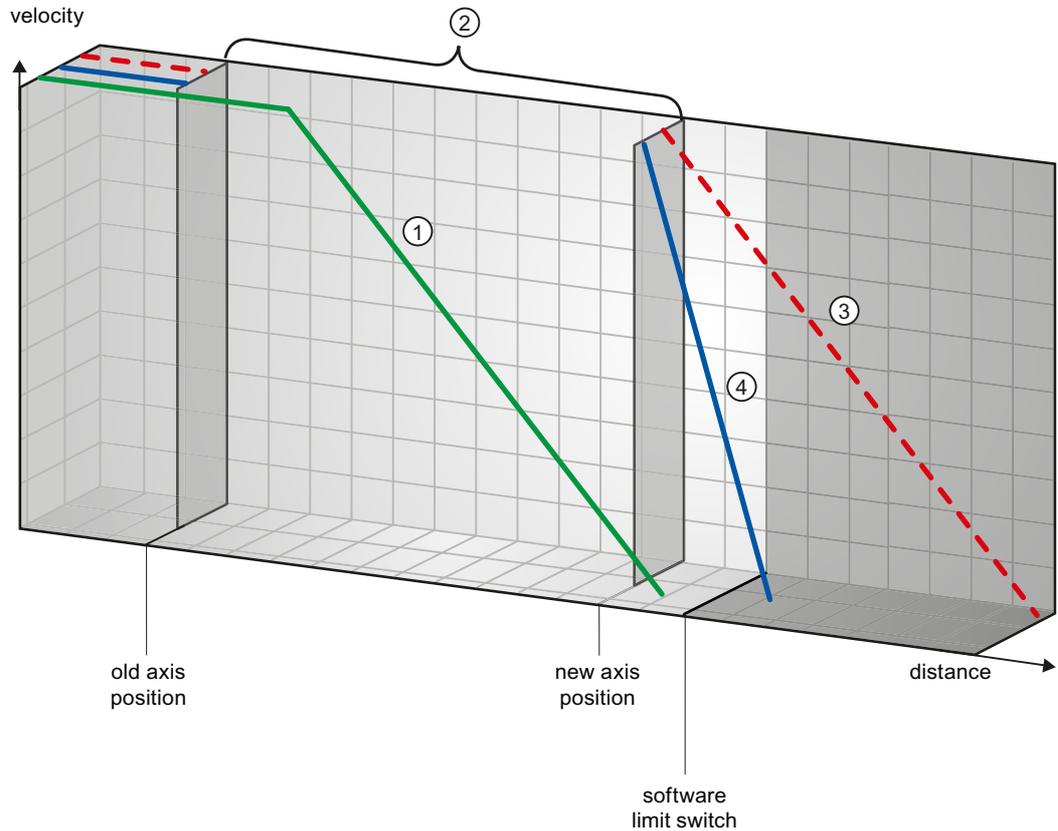
During a travel command, a homing task (for example, Set reference point) offsets the current axis position in the direction of the software limit switch. It is still possible to bring the axis to a standstill before reaching the software limit switch:



①	The green curve shows the motion without the homing task. 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 task. 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 a constant motion, the axis brakes at the emergency deceleration and comes to a standstill at the position of the software limit switch.

Example 2:

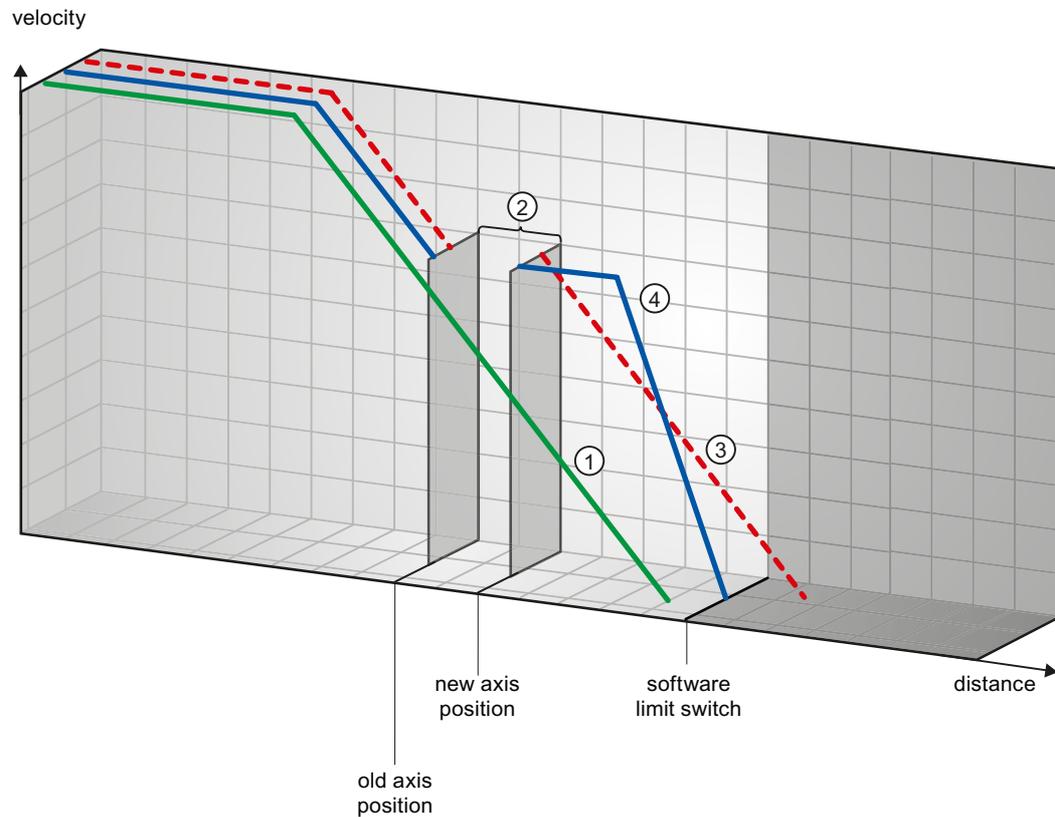
During a travel command, a homing task (for example, Set reference point) offsets the current axis position in the direction of the software limit switch. In contrast to example 1, it is no longer possible to bring the axis to a standstill before reaching the software limit switch. The axis overruns the position of the software limit switch.



①	The green curve shows the motion without the homing task. 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 task. 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 at the emergency deceleration. However, the emergency 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:

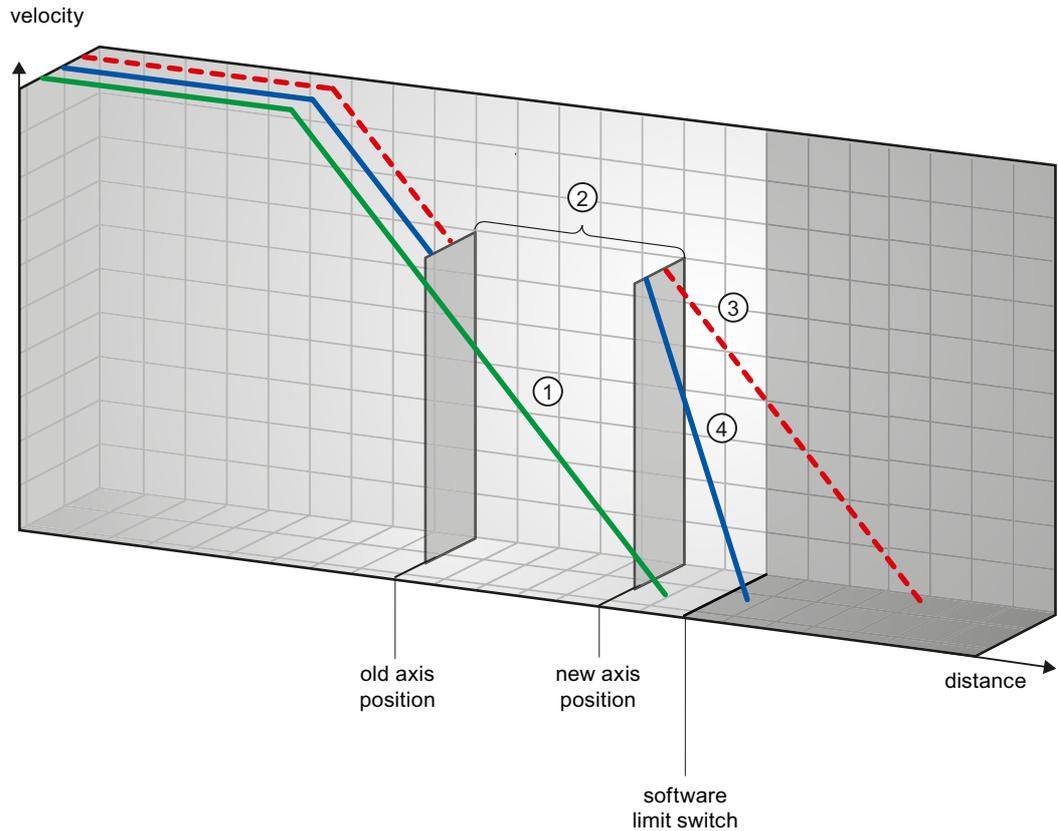
During a braking operation, a homing task (for example, Set reference point) offsets the current axis position in the direction of the software limit switch. It is still possible to bring the axis to a standstill before reaching the software limit switch:



①	The green curve shows the motion without the homing task. 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 task. 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 a constant motion, the axis brakes at the emergency deceleration and comes to a standstill at the position of the software limit switch.

Example 4:

During a braking operation, a homing task (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 task. 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 task. 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 at the emergency deceleration. However, the emergency 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 in conjunction with dynamic changes (Page 79)

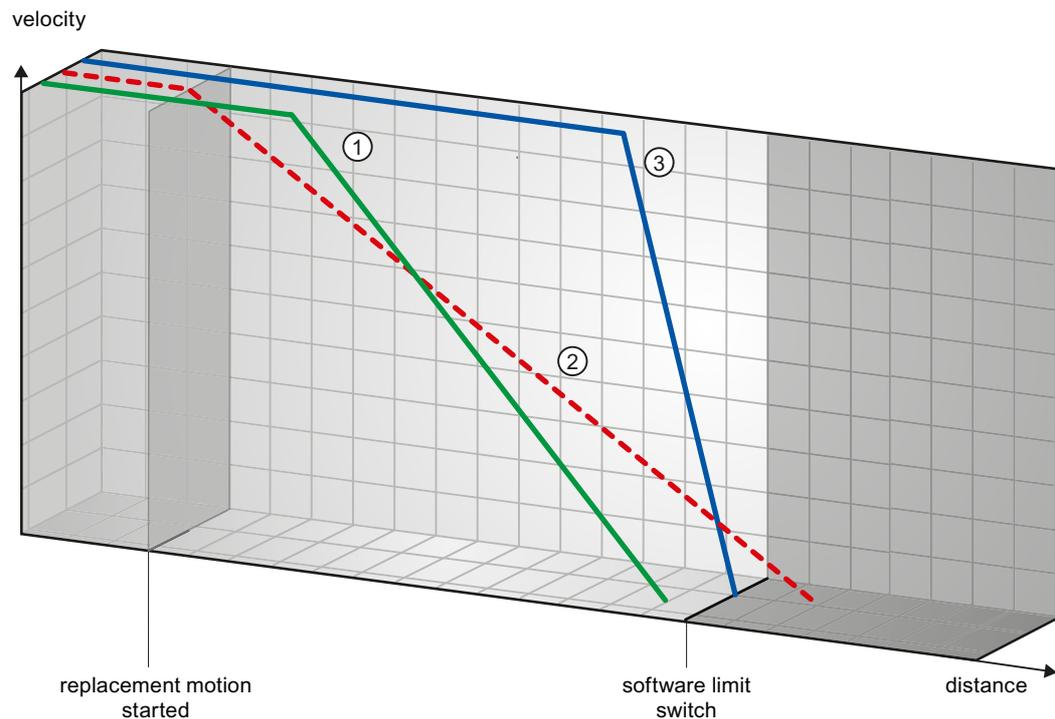
Behavior of axis when position limits is tripped (Page 30)

1.11.4.2 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 tasks. This applies when the overriding motion task with a lower deceleration (tag "<Axis name>.Config.DynamicDefaults.Deceleration") is started. Take the following examples into consideration when developing your program.

Example 1:

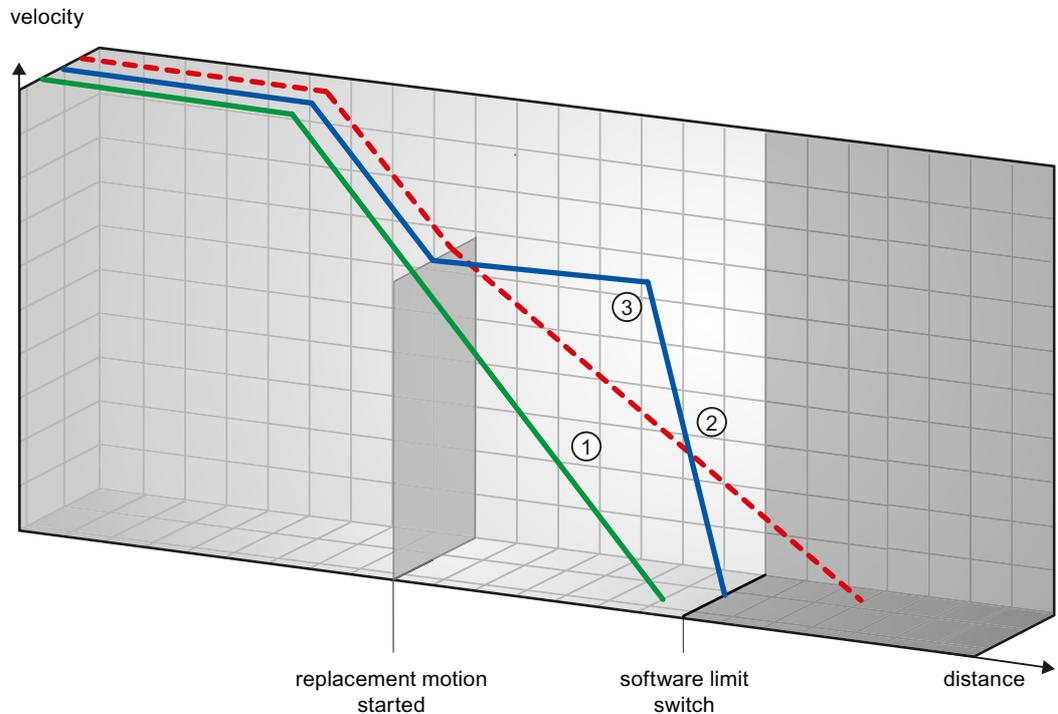
During axis motion, an active motion task is overridden by another motion task with a lower deceleration:



①	The green curve shows the motion of an active task without this task being overridden. 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 task 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 task is no longer sufficient, the axis actually follows the blue curve. Following a constant motion, the axis brakes at the emergency deceleration and comes to a standstill at the position of the software limit switch.

Example 2:

During braking of the axis, an active motion task is overridden by another motion task with a lower deceleration:



①	The green curve shows the motion of an active task without this task being overridden. 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 task 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 task is no longer sufficient, the axis actually follows the blue curve. Following a constant motion, the axis brakes at the emergency 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 75)

Behavior of axis when position limits is tripped (Page 30)

1.11.5 List of ErrorIDs and ErrorInfos

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:

Operating error with stop of the axis

ErrorID	ErrorInfo	Description	Remedy
16#8000		Drive error, "Drive ready" failure	
	16#0001	-	Acknowledge error with instruction "MC_Reset"; provide drive signal; possibly restart task
16#8001		Lower software limit switch was reached/exceeded	
	16#000E	The position of the lower software limit switch was reached with the currently configured deceleration	Acknowledge the error with instruction "MC_Reset"; use a motion task to move the axis in the positive direction out of the range of the software limit switch
	16#000F	The position of the lower software limit switch was reached with the emergency deceleration	
	16#0010	The position of the low software limit switch was exceeded with the emergency deceleration	
16#8002		Upper software limit switch was reached/exceeded	
	16#000E	The position of the upper software limit switch was reached with the currently configured deceleration	Acknowledge the error with instruction "MC_Reset"; use a motion task to move the axis in the negative direction out of the range of the software limit switch
	16#000F	The position of the upper software limit switch was reached with the emergency deceleration	
	16#0010	The position of the upper software limit switch was exceeded with the emergency deceleration	
16#8003		Lower hardware limit switch was approached	
	16#000E	The lower hardware limit switch was approached. The axis was stopped with the emergency deceleration. (During an active reference point approach, the reference point switch was not found)	Acknowledge the error for an enabled axis with instruction "MC_Reset"; use a motion task to move the axis in the positive direction out of the range of the hardware limit switch.
16#8004		Upper hardware limit switch was approached	
	16#000E	The upper hardware limit switch was approached. The axis was stopped with the emergency deceleration. (During an active reference point approach, the reference point switch was not found)	Acknowledge the error for an enabled axis with instruction "MC_Reset"; use a motion task to move the axis in the negative direction out of the range of the hardware 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/HSC 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 69))

Operating error without stop of the axis

ErrorID	ErrorInfo	Description	Remedy
16#8200		Axis is not enabled	
	16#0001	-	Enable the axis; restart the task
16#8201		Axis has already been enabled by another "MC_Power" instruction	
	16#0001	-	Enable the axis with only one "MC_Power" instruction
16#8202		The maximum number of simultaneously active motion control tasks was exceeded (maximum of 200 tasks for all motion control technology objects)	
	16#0001	-	Reduce the number of simultaneously active tasks; restart the task A task is active if parameter "Busy" = TRUE in the motion control instruction.
16#8203		Axis is currently operated in "Manual control" (control panel)	
	16#0001	-	Exit "Manual control"; restart the task
16#8204		Axis is not homed	
	16#0001	-	Home the axis with instruction "MC_Home"; restart the task
16#8205		The axis is currently controlled by the user program (the error is only displayed in the control panel)	
	16#0001	-	Disable axis with instruction "MC_Power" and select "Manual control" again on the control panel
16#8206		Technology object not activated yet	
	16#0001	-	Activate the axis with instruction "MC_Power" Enable = TRUE, or enable the axis on the control panel.
16#8207		Task rejected	
	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 task, for example, "MC_Halt". The other homing type can then be started.

Block parameter error

ErrorID	ErrorInfo	Description	Remedy
16#8400		Incorrect value in parameter "Position" of the motion control instruction	
	16#0002	Number format of value is invalid	Correct the "position" value; restart the task
	16#0005	The value is outside the number range (greater than $1e^{12}$)	
	16#0006	The value is outside the number range (less than $-1e^{12}$)	
16#8401		Incorrect value in parameter "Distance" of the motion control instruction	
	16#0002	Number format of value is invalid	Correct the "Distance" value; restart the task
	16#0005	The value is outside the number range (greater than $1e^{12}$)	
	16#0006	The value is outside the number range (less than $-1e^{12}$)	
16#8402		Incorrect value in parameter "Velocity" of the motion control instruction	
	16#0002	Number format of value is invalid	Correct the "Velocity" value; restart the task
	16#0008	Velocity is greater than the maximum velocity	
	16#0009	Velocity is less than the start/stop velocity	
16#8403		Incorrect value in parameter "Direction" of the motion control instruction	
	16#0011	Invalid selection value	Correct the selection value; restart the task
16#8404		Incorrect value in parameter "Mode" of motion control instruction "MC_Home"	
	16#0011	Invalid selection value	Correct the selection value; restart the task
	16#0015	Active/passive homing is not configured	Correct the configuration and download it to the controller; enable the axis and restart the task
	16#0017	Axis reversal is activated at the HW limit switch, despite the fact that the hardware limit switches are disabled	<ul style="list-style-type: none"> • Activate the hardware limit switch using tag "Axis".Config.PositionLimits_HW.Active = TRUE, restart the task • Correct the configuration and download it to the controller; enable the axis and restart the task
16#8405		Incorrect value in parameter "StopMode" of motion control instruction "MC_Power"	
	16#0011	Invalid selection value	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 task.
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.

Configuration error

ErrorID	ErrorInfo	Description	Remedy
16#8600		Parameterization of pulse generator (PTO) is invalid	
	16#000B	Address is invalid	Correct the configuration and download it to the controller
16#8601		Parameterization of the high-speed counter (HSC) is invalid	
	16#000B	Address is invalid	Correct the configuration and download it to the controller
16#8602		Invalid configuration of the "Drive enabled" output	
	16#000D	Address is invalid	Correct the configuration and download it to the controller
16#8603		Invalid configuration of the "Drive ready" input	
	16#000D	Address is invalid	Correct the configuration and download it to the controller
16#8604		Invalid "Increments per motor revolution" value	
	16#000A	The value is less than or equal to zero	Correct the configuration and download it to the controller
16#8605		Invalid "Load distance per motor revolution" value	
	16#0002	Number format of value is invalid	Correct the configuration and download it to the controller
	16#000A	The value is less than or equal to zero	
16#8606		Invalid "Start / stop velocity" value	
	16#0002	Number format of value is invalid	Correct the configuration and download it to the controller
	16#0003	Value exceeds the hardware limit	
	16#0004	Value is less than the hardware limit	
	16#0007	The start/stop velocity is greater than the maximum velocity	
16#8607		Invalid "maximum velocity" value	
	16#0002	Number format of value is invalid	Correct the configuration and download it to the controller
	16#0003	Value exceeds the hardware limit	
	16#0004	Value is less than the hardware limit	
16#8608		Invalid "Acceleration" value	
	16#0002	Number format of value is invalid	<ul style="list-style-type: none"> 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 task, if necessary
	16#0003	Value exceeds the hardware limit	
	16#0004	Value is less than the hardware limit	
16#8609		Invalid "Deceleration" value	
	16#0002	Number format of value is invalid	<ul style="list-style-type: none"> 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 task, if necessary
	16#0003	Value exceeds the hardware limit	
	16#0004	Value is less than the hardware limit	
16#860A		Invalid "Emergency deceleration" value	
	16#0002	Number format of value is invalid	<ul style="list-style-type: none"> 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 task, if necessary
	16#0003	Value exceeds the hardware limit	
	16#0004	Value is less than the hardware limit	

ErrorID	ErrorInfo	Description	Remedy
16#860B Invalid position value for the lower SW limit switch			
	16#0002	Number format of value is invalid	<ul style="list-style-type: none"> 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 task, if necessary
	16#0005	The value is outside the number range (greater than $1e^{12}$)	
	16#0006	The value is outside the number range (less than $-1e^{12}$)	
	16#0007	The position value of the lower SW limit switch is greater than that of the upper SW limit switch	
16#860C Invalid position value for the upper SW limit switch			
	16#0002	Number format of value is invalid	<ul style="list-style-type: none"> 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 task, if necessary
	16#0005	The value is outside the number range (greater than $1e^{12}$)	
	16#0006	The value is outside the number range (less than $-1e^{12}$)	
16#860D Invalid address of the lower HW limit switch			
	16#000C	Address of falling edge is invalid	Correct the configuration and download it to the controller
	16#000D	Address of rising edge is invalid	
16#860E Invalid address of the upper HW limit switch			
	16#000C	Address of falling edge is invalid	Correct the configuration and download it to the controller
	16#000D	Address of rising edge is invalid	
16#860F Invalid "home position offset" value			
	16#0002	Number format of value is invalid	Correct the configuration and download it to the controller
	16#0005	The value is outside the number range (greater than $1e^{12}$)	
	16#0006	The value is outside the number range (less than $-1e^{12}$)	
16#8610 Invalid "approach velocity" value			
	16#0002	Number format of value is invalid	Correct the configuration and download it to the controller
	16#0008	Velocity is greater than the maximum velocity	
	16#0009	Velocity is less than the start/stop velocity	
16#8611 Invalid "reduced velocity" value			
	16#0002	Number format of value is invalid	Correct the configuration and download it to the controller
	16#0008	Velocity is greater than the maximum velocity	
	16#0009	Velocity is less than the start/stop velocity	
16#8612 Invalid address of the reference point switch			
	16#000C	Address of falling edge is invalid	Correct the configuration and download it to the controller
	16#000D	Address of 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	-	Correct the configuration and download it to the controller

Internal errors

ErrorID	ErrorInfo	Description	Remedy
16#8FFF		Internal error	
	16#F0**	-	<p>POWER OFF and POWER ON the CPU If this does not work, contact Customer Support. Have the following information ready:</p> <ul style="list-style-type: none"> • ErrorID • ErrorInfo • Diagnostic buffer entries

See also

- Using multiple axes with the same PTO (Page 69)
- Using multiple drives with the same PTO (Page 72)
- Tracking tasks from higher priority classes (execution levels) (Page 73)
- Special cases for use of software limit switches (Page 75)
- Tags of the technology object (Page 87)

1.11.6 Tags of the technology object

1.11.6.1 Config. tags

Config.General. tags

Table 1- 1 Legend

Data type	Data type of the tag	
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.	
Access	Access to the tag in the user program:	
	RW	The tag can be read and written in the user program.
	R	The tag can be read in the user program.
	-	The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.	
HMI	The tag can be used in an HMI system.	

<Axis name>.Config.General.PTO)				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWORD	DW#16#00000000	-	-	-

<Axis name>.Config.General.HSC)				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWORD	DW#16#00000000	-	-	-

Config.DriveInterface. tags

Table 1- 2 **Legend**

Data type	Data type of the tag
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.Config.DriveInterface.EnableOutput...				
Tags cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
-	-	-	-	-

<Axis name>.Config.DriveInterface.ReadyInput...				
Tags cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
-	-	-	-	-

Config.Mechanics. tags

Table 1- 3 Legend

Data type	Data type of the tag
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.Config.Mechanics.PulsesPerDriveRevolution)				
Increments per motor revolution				
Data type	Initial value	Access	Effective	HMI
DInt	L#1000	R	-	X

<Axis name>.Config.Mechanics.LeadScrew				
Load distance per motor revolution (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	1.0E+001	R	-	X

<Axis name>.Config.Mechanics.InverseDirection				
Invert direction signal				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

Config.DynamicLimits. tags

Table 1- 4 Legend

Data type	Data type of the tag
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.Config.DynamicLimits.MinVelocity				
Start/stop velocity of axis (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	1.0E+001	R	-	X

<Axis name>.Config.DynamicLimits.MaxVelocity				
Maximum velocity of axis (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	2.5E+002	R	-	X

Config.DynamicDefaults. tags

Table 1- 5 Legend

Data type	Data type of the tag
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.Config.DynamicDefaults.Acceleration				
Acceleration of axis (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	4.8E+001	RW	Change takes effect on the next start of a motion control task	X

<Axis name>.Config.DynamicDefaults.Deceleration				
Deceleration of axis (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	4.8E+001	RW	Change takes effect on the next start of a motion control task	X

<Axis name>.Config.DynamicDefaults.EmergencyDeceleration				
Emergency deceleration of axis (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	1.2E+002	RW	<ul style="list-style-type: none"> Change takes effect on the next start of a motion control task Change takes effect after disabling and enabling the axis 	X

Config.PositionLimits_SW. tags

Table 1- 6 Legend

Data type	Data type of the tag
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.Config.PositionLimits_SW.Active				
true = The software limit switches are activated				
Data type	Initial value	Access	Effective	HMI
Bool	false	RW	A change takes effect after standstill of the axis on the next start of a motion control task. The axis standstill can be checked with tag <Axis name>. StatusBits. Standstill.	X

<Axis name>.Config.PositionLimits_SW.MinPosition				
Position of negative software limit switch (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	-1.0E+004	RW	A change takes effect after standstill of the axis on the next start of a motion control task. The axis standstill can be checked with tag <Axis name>. StatusBits. Standstill.	X

<Axis name>.Config.PositionLimits_SW.MaxPosition				
Position of positive software limit switch (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	1.0E+004	RW	A change takes effect after standstill of the axis on the next start of a motion control task. The axis standstill can be checked with tag <Axis name>. StatusBits. Standstill.	X

Config.PositionLimits_HW. tags

Table 1- 7 Legend

Data type	Data type of the tag
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.Config.PositionLimits_HW.Active				
true = The hardware limit switches are active.				
Data type	Initial value	Access	Effective	HMI
Bool	false	RW	*)	X
*) A change in the tag takes effect based on the following conditions:				
<ul style="list-style-type: none"> On the next start of a motion control task after a standstill of the axis. The axis standstill can be checked with tag <Axis name>. StatusBits.Standstil. After enable of the axis (the axis must have been disabled beforehand and be at a standstill). The axis enable/disable can be checked with tag <Axis name>.StatusBits.Enable and with output parameter "Status" of motion control instruction "MC_Power". 				

<Axis name>.Config.PositionLimits_HW.MinSwitchedLevel				
true = 24 V at CPU input corresponds to lower hardware limit switch approached				
false = 0 V at CPU input corresponds to lower hardware limit switch approached				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.Config.PositionLimits_HW.MinFallingEvent				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWord	DW#16#00000000	-	-	-

<Axis name>.Config.PositionLimits_HW.MinRisingEvent				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWord	DW#16#00000000	-	-	-

<Axis name>.Config.PositionLimits_HW.MaxSwitchedLevel				
true = 24 V at CPU input corresponds to upper hardware limit switch approached				
false = 0 V at CPU input corresponds to upper hardware limit switch approached				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.Config.PositionLimits_HW.MaxFallingEvent				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWord	DW#16#00000000	-	-	-

<Axis name>.Config.PositionLimits_HW.MaxRisingEvent				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWord	DW#16#00000000	-	-	-

Config.Homing. tags

Table 1- 8 Legend

Data type	Data type of the tag
Initial value	Initial value of tag The initial value can be overwritten by the axis configuration.
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.Config.Homing.AutoReversal				
true = Direction reversal at hardware limit switch activated false = Direction reversal at hardware limit switch deactivated				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	

<Axis name>.Config.Homing.Direction				
true = positive approach direction for the search of the reference point switch and positive homing direction false = negative approach direction for the search of the reference point switch and positive homing direction				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	

<Axis name>.Config.Homing.RisingEdge				
true = Homing on right (positive) side of the reference point switch (active homing) false = Homing on left (negative) side of the reference point switch (active homing) For information on the effect of the tag on passive homing, refer to the description in "Configuration - Homing".				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	

<Axis name>.Config.Homing.Offset				
Home position offset (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	0.0	R	-	

<Axis name>.Config.Homing.FastVelocity				
Approach velocity (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	2.0E+002	R	-	

<Axis name>.Config.Homing.SlowVelocity				
Reduced velocity (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	4.0E+001	R	-	

<Axis name>.Config.Homing.FallingEvent				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWord	DW#16#00000000	-	-	

<Axis name>.Config.Homing.RisingEvent				
Tag cannot be evaluated in the user program.				
Data type	Initial value	Access	Effective	HMI
DWord	DW#16#00000000	-	-	

1.11.6.2 MotionStatus. tags

Table 1- 9 Legend

Data type	Data type of the tag
Initial value	Initial value of tag
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.MotionStatus.Position				
Current position of the axis (specified in the configured dimension unit) If the axis is not homed, the tag indicates the position value relative to the enable position of the axis.				
Data type	Initial value	Access	Effective	HMI
Real	0.0	R	-	X

<Axis name>..MotionStatus.Velocity				
Current velocity of the axis (specified in the configured dimension unit)				
Data type	Initial value	Access	Effective	HMI
Real	0.0	R	-	X

<Axis name>.MotionStatus.Distance				
Current distance to the target position of the axis (specified in the configured dimension unit) The value of the tag is only valid during execution of a positioning task with "MC_MoveAbsolute" or "MC_MoveRelative" or of the control panel.				
Data type	Initial value	Access	Effective	HMI
Real	0.0	R	-	X

<Axis name>.MotionStatus.TargetPosition				
Target position of axis (specified in the configured dimension unit) The value of the tag is only valid during execution of a positioning task with "MC_MoveAbsolute" or "MC_MoveRelative" or of the control panel.				
Data type	Initial value	Access	Effective	HMI
Real	0.0	R	-	X

See also

Motion status (Page 67)

1.11.6.3 StatusBits. tags

Table 1- 10 Legend

Data type	Data type of the tag
Initial value	Initial value of tag
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
	- The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.StatusBits.Activated				
true = The axis is activated. It is connected to the assigned PTO. The data of the technology data block will be updated cyclically.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.Enable				
true = The axis is enabled and ready to take on motion control tasks.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.HomingDone				
true = The axis is homed and is capable of executing positioning tasks of motion control instruction "MC_MoveAbsolute". The axis does not have to be homed for relative homing. During active homing with instruction "MC_Home", the status is "false".				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.Done				
true = A motion control task is not active on the axis.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.Error				
true = An error occurred in the axis technology object. Detailed information about the error is available in automatic control in the ErrorID" and ErrorInfo" parameters of the motion control instructions. In manual mode, the "Last error" field of the control panel displays detailed information about the cause of error.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.StandStill				
true = The axis must be at a standstill.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.PositioningCommand				
true = The axis executes a positioning task of motion control instruction "MC_MoveAbsolute" or "MC_MoveRelative" or on the control panel.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.SpeedCommand				
true = The axis executes a motion task with velocity specification of motion control instruction "MC_MoveVelocity", "MC_MoveJog" or of the control panel.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.Homing				
true = The axis executes a homing task of motion control instruction "MC_Home" or of the control panel.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.ConstantVelocity				
true = The axis travels at constant velocity.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.Acceleration				
true = The axis accelerates.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.Deceleration				
true = The axis decelerates (slows down).				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.ControlPanelActive				
true = The "Manual control" mode was enabled on the control panel. The control panel takes control priority over the axis technology object. The axis cannot be controlled from the user program.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.StatusBits.DriveReady				
true = The drive is ready.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

See also

Status and error bits (Page 64)

1.11.6.4 ErrorBits. tags

Table 1- 11 Legend

Data type	Data type of the tag
Initial value	Initial value of tag
Access	Access to the tag in the user program:
	RW The tag can be read and written in the user program.
	R The tag can be read in the user program.
-	The tag cannot be used in the user program.
Effective	Specifies when a change in the tag takes effect.
HMI	The tag can be used in an HMI system.

<Axis name>.ErrorBits.SystemFault				
true = Internal system error.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.ConfigFault				
true = Incorrect configuration of axis.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.DriveFault				
true = The drive has indicated an error after failure of its ready signal.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.SwLimitMinReached				
true = The lower software limit switch has been reached.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.SwLimitMinExceeded				
true = The lower software limit switch has been exceeded.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.SwLimitMaxReached				
true = The upper software limit switch has been reached.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.SwLimitMaxExceeded				
true = The upper software limit switch has been exceeded.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.HwLimitMin				
true = The lower hardware limit switch has been approached.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.HwLimitMax				
true = The upper hardware limit switch has been approached.				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

<Axis name>.ErrorBits.HwUsed				
true = A second axis is using the same PTO and is enabled with "MC_Power".				
Data type	Initial value	Access	Effective	HMI
Bool	false	R	-	X

See also

Status and error bits (Page 64)

1.11.6.5 Internal. tabs

The "Internal" tags contain no user-relevant data; these tags cannot be accessed in the user program.

1.11.6.6 ControlPanel tags

The "ControlPanel" tags contain no user-relevant data; these tags cannot be accessed in the user program.

1.11.6.7 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.

The change in values of editable configuration tags does not take effect immediately. For information on the conditions under which a change takes effect, refer to the detailed description of the relevant tag.

References

2.1 MC_Power

2.1.1 MC_Power: Enable, disable axis

Description

The "MC_Power" motion control instruction enables or disables an axis.

Requirements

- The 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 task.

Disabling the axis (input parameter "Enable" = FALSE) aborts all motion control tasks for the associated technology object.

Parameter

Parameter	Declaration	Data type	Initial value	Description	
Axis	INPUT	TO_Axis_1	-	Axis technology object	
Enable	INPUT	BOOL	FALSE	TRUE	Motion Control attempts to enable the axis.
				FALSE	All active tasks are aborted according to the parameterized "StopMode" and the axis is stopped.
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 axis is disabled without deceleration. Pulse output is stopped immediately.
Status	OUTPUT	BOOL	FALSE	Status of axis enable	
				FALSE	The axis is disabled. The axis does not execute motion control tasks and does not accept any new tasks (exception: MC_Reset task). 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 tasks. 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 has 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	0000	Error ID (Page 81) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	0000	Error info ID (Page 81) for parameter "ErrorID"	

NOTICE

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 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 parameter "StopMode" with the desired value. Set 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 becomes enabled. Output parameter "Status" and technology object tag <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 parameter "StopMode" with the desired value. Set input parameter "Enable" to TRUE. The axis is enabled. Output parameter "Status" and technology object tag <Axis name>.StatusBits.Enable indicate the value TRUE.

Disabling an axis

To disable an axis, you can follow the steps described below:

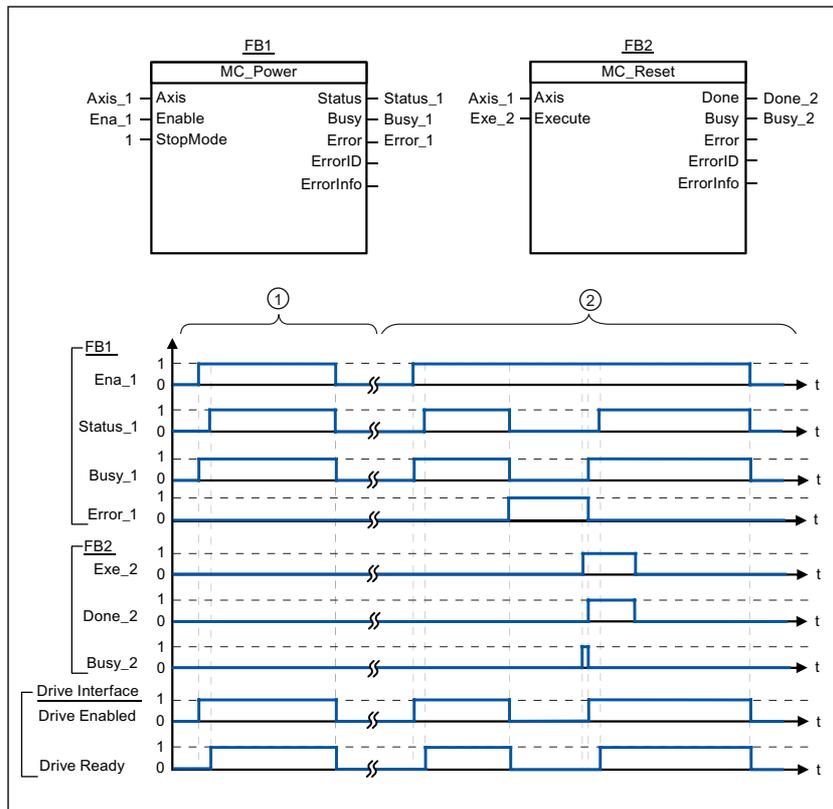
1. Bring the axis to a standstill.
You can identify when the axis is at a standstill in technology object tag <Axis name>.StatusBits.StandStill.
2. Set input parameter "Enable" to FALSE after standstill is reached.
3. If output parameters "Busy" and "Status" and technology object tag <Axis name>.StatusBits.Enable indicate the value FALSE, disabling of the axis is complete.

See also

List of ErrorIDs and ErrorInfos (Page 81)
MC_Power: Function chart (Page 106)
MC_Reset: Acknowledge error (Page 107)
MC_Home: Home axes, set home position (Page 109)
MC_Halt: Halt axis (Page 113)
MC_MoveAbsolute: Absolute positioning of axes (Page 116)
MC_MoveRelative: Relative positioning of axes (Page 119)
MC_MoveVelocity: Move axes at preset rotational speed (Page 122)
MC_MoveJog: Move axes in jogging mode (Page 126)
Overview of the Motion Control statements (Page 44)

2.1.2 MC_Power: Function chart

Description



- | | |
|---|--|
| ① | An axis is enabled and then disabled again. Once the drive has signaled "Drive ready" back to the CPU, the successful enable can be read out via "Status_1". |
| ② | 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. |

See also

MC_Power: Enable, disable axis (Page 103)

2.2 MC_Reset

2.2.1 MC_Reset: Acknowledge error

Description

Motion control instruction "MC_Reset" can be used to acknowledge "Operating error with axis stop" and "Configuration error". The errors that require acknowledgement can be found in the "List of ErrorIDs and ErrorInfos" under "Remedy".

Requirements

- The cause of a pending configuration error requiring acknowledgement has been eliminated (for example, acceleration in "Axis" technology object has been changed to a valid value).

Override response

The MC_Reset task cannot be aborted by any other motion control task.

The new MC_Reset task does not abort any other active motion control tasks.

Parameters

Parameter	Declaration	Data type	Initial value	Description
Axis	INPUT	TO_Axis_1	-	Axis technology object
Execute	INPUT	BOOL	FALSE	Start of the task with a positive edge
Done	OUTPUT	BOOL	FALSE	TRUE Error has been acknowledged.
Busy	OUTPUT	BOOL	FALSE	TRUE The task is being executed.
Error	OUTPUT	BOOL	FALSE	TRUE An error has occurred during execution of the task. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	0000	Error ID (Page 81) for parameter "Error"
ErrorInfo	OUTPUT	WORD	0000	Error info ID (Page 81) for parameter "ErrorID"

Acknowledging an error requiring acknowledgement with MC_Reset

To acknowledge an error, follow these steps:

1. Check the requirements indicated above.
2. Start the acknowledgement of the error with a rising edge at input parameter "Execute".
3. If output parameter "Done" indicates the value TRUE and technology object tag <Axis name>.StatusBits.Error the value FALSE, the error has been acknowledged.

See also

List of ErrorIDs and ErrorInfos (Page 81)

MC_Power: Enable, disable axis (Page 103)

MC_Home: Home axes, set home position (Page 109)

MC_Halt: Halt axis (Page 113)

MC_MoveAbsolute: Absolute positioning of axes (Page 116)

MC_MoveRelative: Relative positioning of axes (Page 119)

MC_MoveVelocity: Move axes at preset rotational speed (Page 122)

MC_MoveJog: Move axes in jogging mode (Page 126)

Overview of the Motion Control statements (Page 44)

2.3 MC_Home

2.3.1 MC_Home: Home axes, set home position

Description

Use the motion control instruction "MC_Home" 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 motion control instruction "MC_Home" does not carry out any homing motion. The traversing motion required for this step must be implemented by the user via other motion control instructions. When the reference point 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".

Requirements

- The axis is enabled.

Override response

The override response is dependent on the selected mode:

Mode = 0, 1

The MC_Home task cannot be aborted by any other motion control task.

The new MC_Home task does not abort any active motion control tasks. Position-related motion tasks are resumed after homing according to the new homing position (value at input parameter: "Position").

Mode = 2

The MC_Home task can be aborted by the following motion control tasks:

- MC_Home task Mode = 2, 3

The new MC_Home task aborts the following active motion control task.

- MC_Home task Mode = 2

Position-related motion tasks are resumed after homing according to the new homing position (value at input parameter: "Position").

Mode = 3

The MC_Home task can be aborted by the following motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

The new MC_Home task aborts the following active motion control tasks:

- MC_Home task Mode = 2, 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

Parameters

Parameter	Declaration	Data type	Initial value	Description	
Axis	INPUT	TO_Axis_PTO	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the task with a positive edge	
Position	INPUT	REAL	0.0	<ul style="list-style-type: none"> 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: $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$	
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 Reference point approach in accordance with the axis configuration. Following homing, the value of parameter "Position" is set as the new axis position.
Done	OUTPUT	BOOL	FALSE	TRUE	Task completed
Busy	OUTPUT	BOOL	FALSE	TRUE	The task is being executed.
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	During execution the task was aborted by another task.
Error	OUTPUT	BOOL	FALSE	TRUE	An error has occurred during execution of the task. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	0000		Error ID (Page 81) for parameter "Error"
ErrorInfo	OUTPUT	WORD	0000		Error info ID (Page 81) for parameter "ErrorID"

Note

Axis homing is lost under the following conditions:

- Disabling of axis by motion control instruction "MC_Power"
- Switchover between automatic control and manual control
- Upon start of active homing. After successful completion of the homing operation, axis homing is again available.
- After POWER OFF -> POWER ON of the CPU
- After CPU restart (RUN-STOP -> STOP-RUN)

Homing an axis

To home the axis, follow these steps:

1. Check the requirements indicated above.
2. Initialize 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 tag <Axis name>.StatusBits.HomingDone indicate the value TRUE, homing is complete.

See also

List of ErrorIDs and ErrorInfos (Page 81)

MC_Power: Enable, disable axis (Page 103)

MC_Reset: Acknowledge error (Page 107)

MC_Halt: Halt axis (Page 113)

MC_MoveAbsolute: Absolute positioning of axes (Page 116)

MC_MoveRelative: Relative positioning of axes (Page 119)

MC_MoveVelocity: Move axes at preset rotational speed (Page 122)

MC_MoveJog: Move axes in jogging mode (Page 126)

Overview of the Motion Control statements (Page 44)

2.4 MC_Halt

2.4.1 MC_Halt: Halt axis

Description

The "MC_Halt" motion control instruction stops all motions and brings the axis to a standstill. The standstill position is not defined.

Requirements

- The axis is enabled.

Override response

The MC_Halt task can be aborted by the following motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

The new MC_Halt task aborts the following active motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

Parameter

Parameter	Declaration	Data type	Initial value	Description
Axis	INPUT	TO_Axis_1	-	Axis technology object
Execute	INPUT	BOOL	FALSE	Start of the task with a positive edge
Done	OUTPUT	BOOL	FALSE	TRUE Zero velocity reached
Busy	OUTPUT	BOOL	FALSE	TRUE The task is being executed.
CommandAborted	OUTPUT	BOOL	FALSE	TRUE During execution the task was aborted by another task.
Error	OUTPUT	BOOL	FALSE	TRUE An error has occurred during execution of the task. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	0000	Error ID (Page 81) for parameter "Error"
ErrorInfo	OUTPUT	WORD	0000	Error info ID (Page 81) for parameter "ErrorID"

See also

List of ErrorIDs and ErrorInfos (Page 81)

MC_Halt: Function chart (Page 115)

MC_Power: Enable, disable axis (Page 103)

MC_Reset: Acknowledge error (Page 107)

MC_Home: Home axes, set home position (Page 109)

MC_MoveAbsolute: Absolute positioning of axes (Page 116)

MC_MoveRelative: Relative positioning of axes (Page 119)

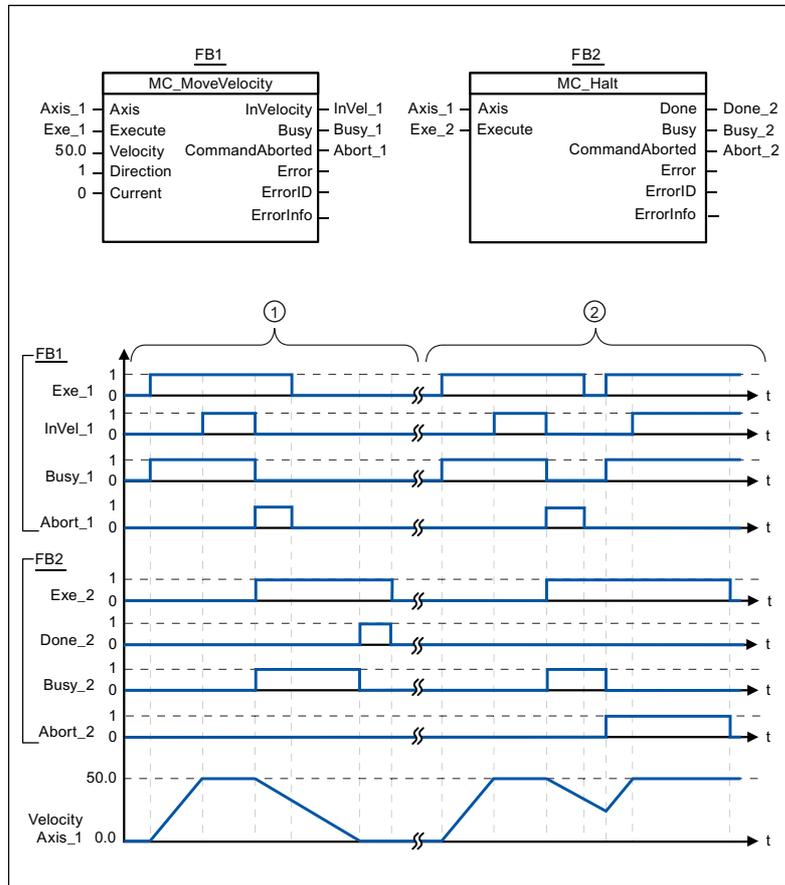
MC_MoveVelocity: Move axes at preset rotational speed (Page 122)

MC_MoveJog: Move axes in jogging mode (Page 126)

Overview of the Motion Control statements (Page 44)

2.4.2 MC_Halt: Function chart

Description



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 5.0

①	The axis is braked by an MC_Halt task until it comes to a standstill. The axis standstill is signaled via "Done_2".
②	While an MC_Halt task is braking the axis, this task is aborted by another motion task. The abort is signaled via "Abort_2".

See also

MC_Halt: Halt axis (Page 113)

2.5 MC_MoveAbsolute

2.5.1 MC_MoveAbsolute: Absolute positioning of axes

Description

The "MC_MoveAbsolute" motion control instruction starts a positioning motion of the axis to an absolute position.

Requirements

- The axis is enabled.
- The axis is homed.

Override response

The MC_MoveAbsolute task can be aborted by the following motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

The new MC_MoveAbsolute task aborts the following active motion control tasks:

- MC_Home task Mode = 3
- MC_Halt-Auftrag
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

Parameter

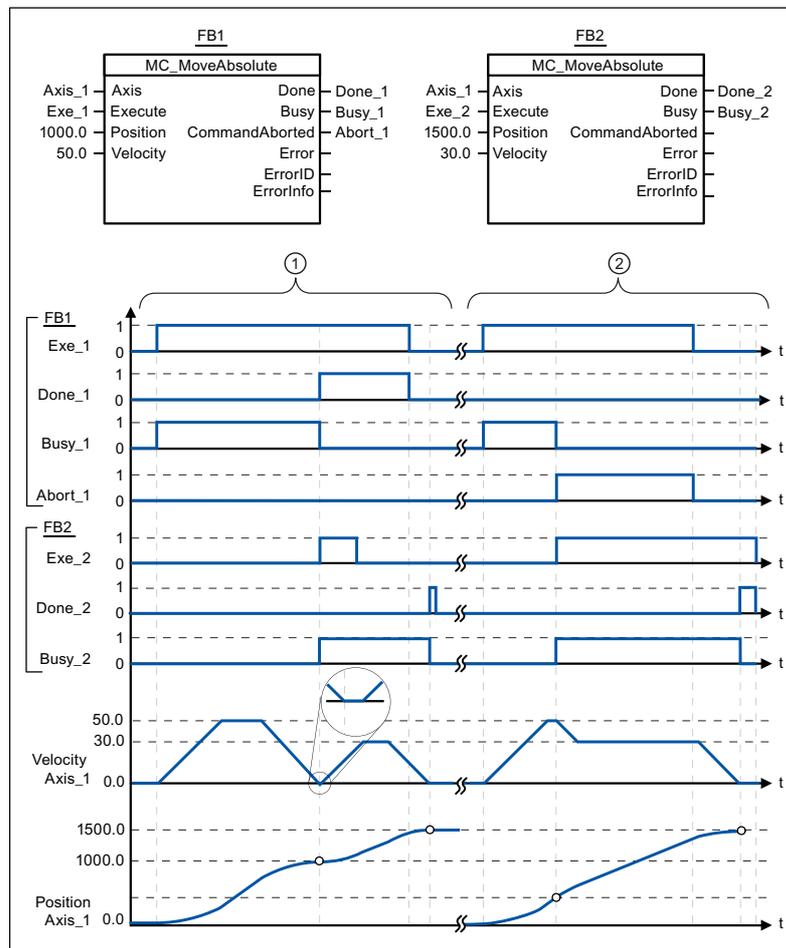
Parameter	Declaration	Data type	Initial value	Description
Axis	INPUT	TO_Axis_1	-	Axis technology object
Execute	INPUT	BOOL	FALSE	Start of the task with a positive edge
Position	INPUT	REAL	0.0	Absolute target position Limit values: $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Velocity	INPUT	REAL	10.0	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 \leq Velocity \leq maximum velocity
Done	OUTPUT	BOOL	FALSE	TRUE Absolute target position reached
Busy	OUTPUT	BOOL	FALSE	TRUE The task is being executed.
CommandAborted	OUTPUT	BOOL	FALSE	TRUE During execution the task was aborted by another task.
Error	OUTPUT	BOOL	FALSE	TRUE An error has occurred during execution of the task. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	0000	Error ID (Page 81) for parameter "Error"
ErrorInfo	OUTPUT	WORD	0000	Error info ID (Page 81) for parameter "ErrorID"

See also

- List of ErrorIDs and ErrorInfos (Page 81)
- MC_MoveAbsolute: Function chart (Page 118)
- MC_Power: Enable, disable axis (Page 103)
- MC_Reset: Acknowledge error (Page 107)
- MC_Home: Home axes, set home position (Page 109)
- MC_Halt: Halt axis (Page 113)
- MC_MoveRelative: Relative positioning of axes (Page 119)
- MC_MoveVelocity: Move axes at preset rotational speed (Page 122)
- MC_MoveJog: Move axes in jogging mode (Page 126)
- Overview of the Motion Control statements (Page 44)

2.5.2 MC_MoveAbsolute: Function chart

Description



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 10.0

①	An axis is moved to absolute position 1000.0 with a MC_MoveAbsolute task. When the axis reaches the target position, this is signaled via "Done_1". When "Done_1" = TRUE, another MC_MoveAbsolute task, 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".
②	An active MC_MoveAbsolute task is aborted by another MC_MoveAbsolute task. 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".

See also

MC_MoveAbsolute: Absolute positioning of axes (Page 116)

2.6 MC_MoveRelative

2.6.1 MC_MoveRelative: Relative positioning of axes

Description

The "MC_MoveRelative" motion control instruction starts a positioning motion relative to the start position.

Requirements

- The axis is enabled.

Override response

The MC_MoveRelative task can be aborted by the following motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

The new MC_MoveRelative task aborts the following active motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

Parameter

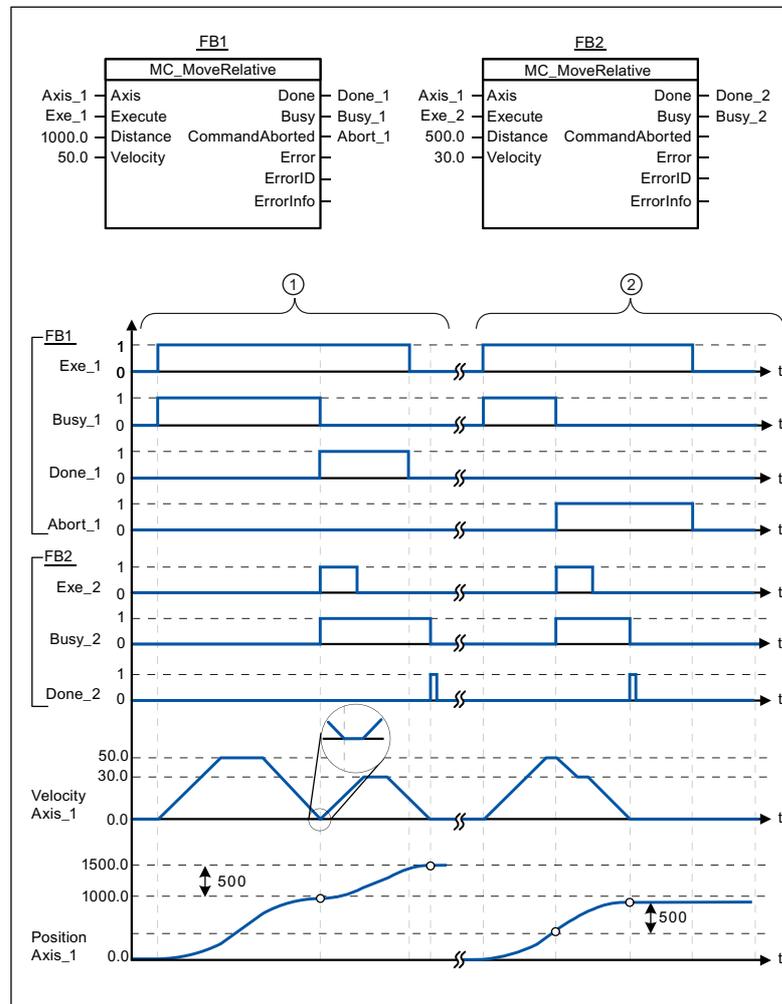
Parameter	Declaration	Data type	Initial value	Description
Axis	INPUT	TO_Axis_1	-	Axis technology object
Execute	INPUT	BOOL	FALSE	Start of the task with a positive edge
Distance	INPUT	REAL	0.0	Travel distance for the positioning operation Limit values: $-1.0e^{12} \leq \text{Distance} \leq 1.0e^{12}$
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 \leq Velocity \leq maximum velocity
Done	OUTPUT	BOOL	FALSE	TRUE Target position reached
Busy	OUTPUT	BOOL	FALSE	TRUE The task is being executed.
CommandAborted	OUTPUT	BOOL	FALSE	TRUE During execution the task was aborted by another task.
Error	OUTPUT	BOOL	FALSE	TRUE An error has occurred during execution of the task. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	0000	Error ID (Page 81) for parameter "Error"
ErrorInfo	OUTPUT	WORD	0000	Error info ID (Page 81) for parameter "ErrorID"

See also

- List of ErrorIDs and ErrorInfos (Page 81)
- MC_MoveRelative: Function chart (Page 121)
- MC_Power: Enable, disable axis (Page 103)
- MC_Reset: Acknowledge error (Page 107)
- MC_Home: Home axes, set home position (Page 109)
- MC_Halt: Halt axis (Page 113)
- MC_MoveAbsolute: Absolute positioning of axes (Page 116)
- MC_MoveVelocity: Move axes at preset rotational speed (Page 122)
- MC_MoveJog: Move axes in jogging mode (Page 126)
- Overview of the Motion Control statements (Page 44)

2.6.2 MC_MoveRelative: Function chart

Description



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 10.0

①	The axis is moved by an MC_MoveRelative task 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 task, 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".
②	An active MC_MoveRelative task is aborted by another MC_MoveRelative task. 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".

See also

MC_MoveRelative: Relative positioning of axes (Page 119)

2.7 MC_MoveVelocity

2.7.1 MC_MoveVelocity: Move axes at preset rotational speed

Description

Motion control instruction "MC_MoveVelocity" moves the axis constantly at the specified velocity.

Requirements

- The axis is enabled.

Override response

MC_MoveVelocity can be aborted by the following motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

The new MC_MoveVelocity task aborts the following active motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

Parameter

Parameter	Declaration	Data type	Initial value	Description	
Axis	INPUT	TO_Axis_1	-	Axis technology object	
Execute	INPUT	BOOL	FALSE	Start of the task with a positive edge	
Velocity	INPUT	REAL	10.0	Velocity specification for axis motion Limit values: Start/stop velocity \leq Velocity \leq 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.
InVelocity	OUTPUT	BOOL	FALSE	TRUE <ul style="list-style-type: none"> "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 task is being executed.	
CommandAborted	OUTPUT	BOOL	FALSE	TRUE During execution the task was aborted by another task.	
Error	OUTPUT	BOOL	FALSE	TRUE An error has occurred during execution of the task. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".	
ErrorID	OUTPUT	WORD	0000	Error ID (Page 81) for parameter "Error"	
ErrorInfo	OUTPUT	WORD	0000	Error info ID (Page 81) for parameter "ErrorID"	

Behavior with zero set velocity (Velocity = 0.0)

An MC_MoveVelocity task with "Velocity" = 0.0 (such as an MC_Halt task) aborts active motion tasks 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 operation and changes to FALSE together with "InVelocity". If parameter "Execute" = TRUE is set, "InVelocity" and "Busy" are latched.

When the MC_MoveVelocity task 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 task is started.

See also

List of ErrorIDs and ErrorInfos (Page 81)

MC_MoveVelocity: Function chart (Page 125)

MC_Power: Enable, disable axis (Page 103)

MC_Reset: Acknowledge error (Page 107)

MC_Home: Home axes, set home position (Page 109)

MC_Halt: Halt axis (Page 113)

MC_MoveAbsolute: Absolute positioning of axes (Page 116)

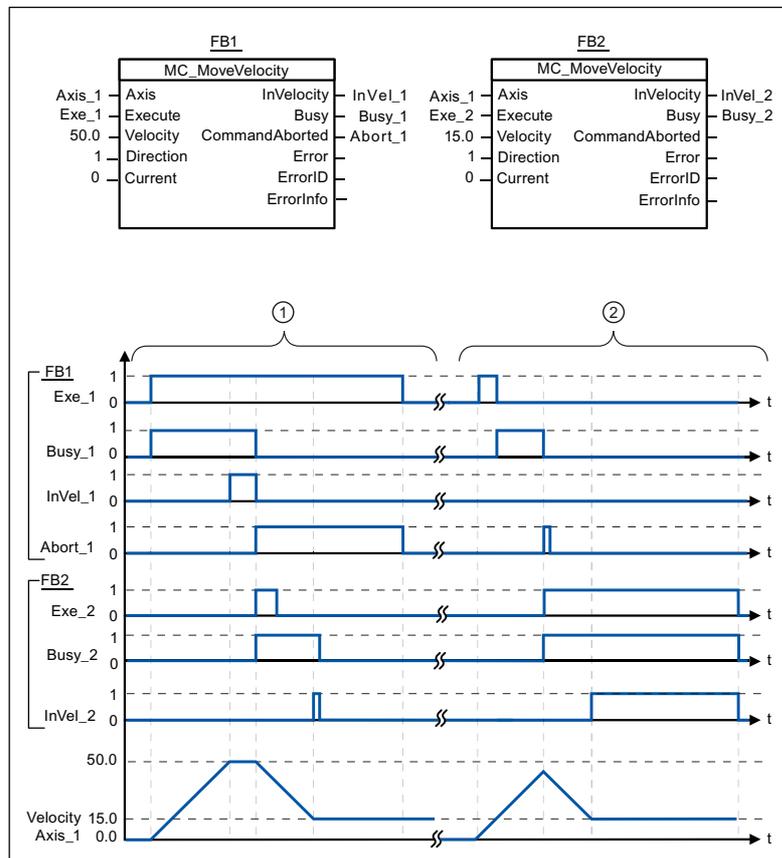
MC_MoveRelative: Relative positioning of axes (Page 119)

MC_MoveJog: Move axes in jogging mode (Page 126)

Overview of the Motion Control statements (Page 44)

2.7.2 MC_MoveVelocity: Function chart

Description



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 10.0

①	An active MC_MoveVelocity task signals via "InVel_1" that its target velocity has been reached. It is then aborted by another MC_MoveVelocity task. 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.
②	An active MC_MoveVelocity task is aborted by another MC_MoveVelocity task 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.

See also

MC_MoveVelocity: Move axes at preset rotational speed (Page 122)

2.8 MC_MoveJog

2.8.1 MC_MoveJog: Move axes in jogging mode

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 axis is enabled.

Override response

The MC_MoveJog task can be aborted by the following motion control tasks:

- MC_Home task Mode = 3
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

The new MC_MoveJog task aborts the following active motion control tasks:

- MC_Home task (Mode = 3)
- MC_Halt task
- MC_MoveAbsolute task
- MC_MoveRelative task
- MC_MoveVelocity task
- MC_MoveJog task

Parameter

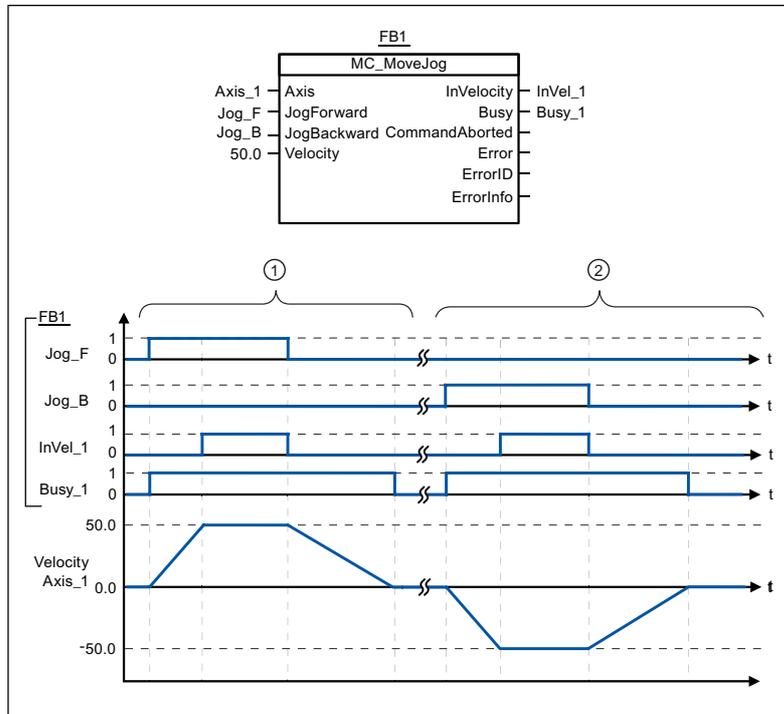
Parameter	Declaration	Data type	Initial value	Description
Axis	INPUT	TO_Axis_1	-	Axis technology object
JogForward	INPUT	BOOL	FALSE	As long as the parameter is TRUE, the axis moves in the positive direction at the velocity specified in parameter "Velocity". (The sign of the value in parameter "Velocity" is ignored)
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". (The sign of the value in parameter "Velocity" is ignored)
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 \leq Velocity \leq maximum velocity
InVelocity	OUTPUT	BOOL	FALSE	TRUE The velocity specified in parameter "Velocity" was reached.
Busy	OUTPUT	BOOL	FALSE	TRUE The task is being executed.
CommandAborted	OUTPUT	BOOL	FALSE	TRUE During execution the task was aborted by another task.
Error	OUTPUT	BOOL	FALSE	TRUE An error has occurred during execution of the task. The cause of the error can be found in parameters "ErrorID" and "ErrorInfo".
ErrorID	OUTPUT	WORD	0000	Error ID (Page 81) for parameter "Error"
ErrorInfo	OUTPUT	WORD	0000	Error info ID (Page 81) for parameter "ErrorID"

See also

- List of ErrorIDs and ErrorInfos (Page 81)
- MC_MoveJog: Function chart (Page 128)
- MC_Power: Enable, disable axis (Page 103)
- MC_Reset: Acknowledge error (Page 107)
- MC_Home: Home axes, set home position (Page 109)
- MC_Halt: Halt axis (Page 113)
- MC_MoveAbsolute: Absolute positioning of axes (Page 116)
- MC_MoveRelative: Relative positioning of axes (Page 119)
- MC_MoveVelocity: Move axes at preset rotational speed (Page 122)
- Overview of the Motion Control statements (Page 44)

2.8.2 MC_MoveJog: Function chart

Description



The following values were configured in the configuration window **Dynamics > General**:

- Acceleration: 10.0
- Deceleration: 5.0

①	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 "InVelo_1". The axis brakes to a standstill again after Jog_F is reset.
②	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 "InVelo_1". The axis brakes to a standstill again after Jog_B is reset.

See also

MC_MoveJog: Move axes in jogging mode (Page 126)