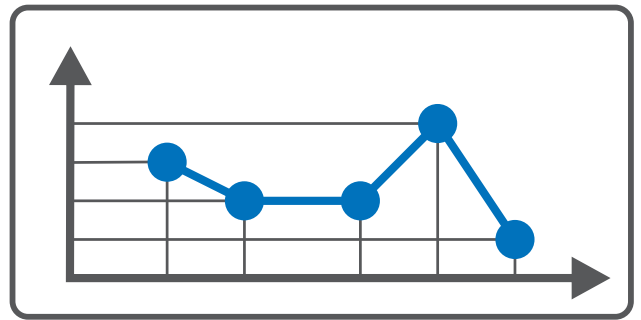
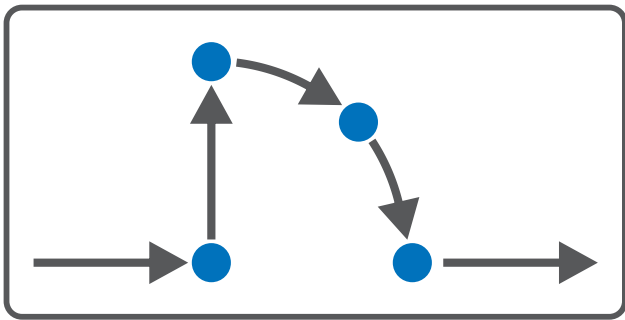


CiA 402



CiA 402 – SC6, SI6 Manual

en-US
10/2026
ID 443080.04



STÖBER

Table of contents

Table of contents	2
1 Foreword	5
2 Information on this documentation.....	6
2.1 Storage and transfer.....	6
2.2 Described product	6
2.3 Directives and standards	6
2.4 Timeliness.....	6
2.5 Original language.....	6
2.6 Limitation of liability.....	7
2.7 Formatting conventions	7
2.7.1 Display of warning messages and information	7
2.7.2 Markup of text elements	8
2.7.3 Mathematics and formulas.....	8
2.8 Trademarks.....	9
3 Safety notes	10
4 What you should know before commissioning	11
4.1 Program interfaces	11
4.1.1 DS6 program interface.....	11
4.1.2 TwinCAT 3 program interface.....	15
4.1.3 CODESYS Development System V3 program interface.....	16
4.2 Meaning of parameters.....	17
4.2.1 Parameter groups.....	17
4.2.2 Parameter types and data types.....	18
4.2.3 Parameter types	19
4.2.4 Parameter structure	19
4.2.5 Parameter visibility	20
4.3 Signal sources	21
4.4 Non-volatile memory.....	21
5 Commissioning.....	22
5.1 DS6: Configuring the drive controller	23
5.1.1 Initiating the project	23
5.1.2 Mapping the mechanical axis model	27
5.1.3 Configuring EtherCAT settings	32
5.1.4 Transmitting and saving a configuration	36
5.1.5 Testing the configuration.....	39

5.2	TwinCAT 3: Setting up an EtherCAT system	41
5.2.1	Creating a standard project	41
5.2.2	EtherCAT MainDevice: Activating Config-Modus	42
5.2.3	Scanning the hardware environment	43
5.2.4	Configuring synchronization using distributed clocks	44
5.2.5	Controller-based axis control	45
5.2.6	Transmitting the configuration	46
5.2.7	Checking the functionality of the axes	47
5.2.8	More information about TwinCAT 3?	48
5.3	CODESYS Development System V3: Setting up the EtherCAT system	55
5.3.1	Creating a standard project	55
5.3.2	Adding an EtherCAT MainDevice	55
5.3.3	Adding a SoftMotion axis	56
5.3.4	Configuring synchronization using distributed clocks	57
5.3.5	Controller-based axis control	58
5.3.6	Transmitting the configuration	59
5.3.7	Checking the functionality of the axes	60
5.3.8	More information about CODESYS Development System V3?	61
6	More about CiA 402?	70
6.1	CiA 402 – Concept	70
6.1.1	Activation	71
6.1.2	Operating modes	71
6.1.3	Data sources	71
6.1.4	User-defined status word	71
6.1.5	Additional functions	71
6.1.6	Control panels	72
6.1.7	Motion core	72
6.2	Axis model	73
6.3	Limit switches	76
6.3.1	Hardware limit switches	77
6.3.2	Software limit switches	79
6.4	Referencing	80
6.4.1	Referencing methods	80
6.4.2	Reference position	106
6.4.3	Reference retaining	106
6.4.4	Reference loss	107
6.5	CiA 402 device control	111
6.5.1	CiA 402 device state machine	111
6.5.2	Commands, states and transitions	112

6.6	Operating modes in detail	118
6.6.1	Interpolated position mode (ip)	118
6.6.2	Cyclic synchronous position mode (csp)	121
6.6.3	Cyclic synchronous velocity mode (csv)	122
6.6.4	Cyclic synchronous torque mode (cst)	123
6.6.5	Profile position mode (pp)	124
6.6.6	Velocity mode (vl)	126
6.6.7	Profile velocity mode (pv)	129
6.6.8	Profile torque mode (pt)	131
6.6.9	Homing mode	132
6.6.10	Jog	134
6.7	Operating modes in accordance with CiA 402 – Commands	137
6.8	Feedforward control	140
6.8.1	Controller-generated external feedforward control	140
6.8.2	Drive-generated internal feedforward control	141
6.8.3	Without feedforward control	141
6.8.4	Setting up feedforward control in DS6	142
6.9	Interpolation	143
6.10	Touch probe – Examples	144
6.11	Cycle times	147
6.12	Lag monitoring	147
7	Appendix	148
7.1	EtherCAT and CiA 402 standard mapping	148
7.1.1	SC6, SI6: CiA 402 receive process data	148
7.1.2	SC6, SI6: CiA 402 transmit process data	150
7.2	Supported communication objects	152
7.2.1	CiA 402 Drives and motion control: 6000 hex – 65FF hex	152
7.2.2	CiA 402 Drives and motion control: 6800 hex – 6DFF hex	156
7.2.3	Touch probe source: Values	160
7.2.4	Manufacturer-specific parameters: 2000 hex – 53FF hex	161
7.2.5	Manufacturer-specific parameters: A000 hex – D3FF hex	163
7.3	Detailed information	165
7.4	Formula symbols	166
7.5	Abbreviations	167
8	Contact	168
8.1	Consultation, service and address	168
8.2	Your opinion is important to us	168
8.3	Close to customers around the world	169
	Glossary	170
	List of figures	171
	List of tables	172

1 Foreword

The DriveControlSuite commissioning software for 6th generation drive controllers offers convenient functions for efficient project configuration and commissioning of drive controllers in multi-axis and single-axis applications.

The CiA 402 application included in the software contains both controller-based and drive-based operating modes along with the associated position, velocity and torque/force control types. For commissioning, emergency operation and maintenance or repair work, the STOBER-specific jog mode is also available.

This documentation describes the general functions of the CiA 402 application and guides you step by step through the setup and project configuration of your drive project in the individual operating modes.

2 Information on this documentation

This documentation supports you in the setup and project configuration of your drive system with the CiA 402 application, which is based on the internationally standardized CANopen CiA 402 device profile for electric drives in accordance with IEC 61800-7.

Commissioning is described using examples of an EtherCAT network with STOBER drive controllers from the 6th generation in combination with a CODESYS SoftMotion controller or, as an alternative, working with a controller from Beckhoff Automation GmbH & Co. KG.

Technical knowledge

To be able to commission one or more drive controllers in combination with a controller using the CiA 402 application, you should have basic knowledge of handling 6th generation STOBER drive controllers and of the DriveControlSuite commissioning software. Operating an EtherCAT network also requires having familiarity with the basics of the EtherCAT network technology.

Technical requirements

Before you begin operating your EtherCAT network, you need to wire the drive controllers and initially check that they are functioning correctly. To do this, follow the instructions in the manual for the relevant drive controller.

2.1 Storage and transfer

As this documentation contains important information for handling the product safely and efficiently, it must be stored in the immediate vicinity of the product until product disposal and be accessible to qualified personnel at all times.

Also pass on this documentation if the product is transferred or sold to a third party.

2.2 Described product

This documentation is mandatory for SC6 or SI6 series drive controllers in conjunction with version V 6.7-D or later of the DriveControlSuite software (DS6) and V 6.7-D-EC or later of the associated firmware.

2.3 Directives and standards

Refer to the drive controller documentation for the European directives and standards relevant to the drive controller and accessories.

2.4 Timeliness

Check whether you have the latest version of this documentation. The latest document versions for our products are available for download on our website:

<http://www.stoeber.de/en/downloads/>.

2.5 Original language

The original language of this documentation is German; all other language versions are derived from the original language.

2.6 Limitation of liability

This documentation was created taking into account the applicable standards and regulations as well as the current state of technology.

No warranty or liability claims for damage shall result from failure to comply with the documentation or from use that deviates from the intended use of the product. This is especially true for damage caused by individual technical modifications to the product or the project configuration and operation of the product by unqualified personnel.

2.7 Formatting conventions

Orientation guides in the form of signal words, symbols and special text markups are used to emphasize specific information so that you are able identify it in this documentation quickly.

2.7.1 Display of warning messages and information

Warning messages are identified with symbols. They indicate special risks when handling the product and are accompanied by relevant signal words that express the extent of the risk. Furthermore, useful tips and recommendations for efficient, error-free operation are specially highlighted.

ATTENTION!

Attention

This indicates that damage to property may occur

- if the stated precautionary measures are not taken.

⚠ CAUTION!

Caution

This word with a warning triangle indicates that minor personal injury may occur

- if the stated precautionary measures are not taken.

⚠ WARNING!

Warning

This word with a warning triangle means there may be a considerable risk of fatal injury

- if the stated precautionary measures are not taken.

⚠ DANGER!

Danger

This word with a warning triangle indicates that there is a considerable risk of fatal injury

- if the stated precautionary measures are not taken.

Information

Information indicates important information about the product or serves to emphasize a section in the documentation that deserves special attention from the reader.

2.7.2 Markup of text elements

Certain elements of the continuous text are distinguished as follows.

Important information	Words or expressions with a special meaning
Interpolated position mode	Optional: File or product name or other name
<u>Detailed information</u>	Internal cross-reference
http://www.samplelink.com	External cross-reference

Software and other displays

The following formatting is used to identify the various information content of elements referenced by the software interface or a drive controller display, as well as any user entries.

Main menu Settings	Window names, dialog box names, page names or buttons, combined proper nouns, functions referenced by the interface
Select Referencing method A	Predefined entry
Save your <own IP address>	User-defined entry
EVENT 52: COMMUNICATION	Displays (status, messages, warnings, faults)

Keyboard shortcuts and command sequences or paths are represented as follows.

[Ctrl], [Ctrl] + [S]	Key, key combination
Table > Insert table	Navigation to menus/submenus (path specification)

2.7.3 Mathematics and formulas

The following signs are used to represent mathematical relationships and formulas.

–	Subtraction
+	Addition
×	Multiplication
÷	Division
	Absolute value

2.8 Trademarks

The following names used in connection with the device, its optional equipment and its accessories are trademarks or registered trademarks of other companies:

CANopen [®] , CiA [®]	CANopen [®] and CiA [®] are registered trademarks of CAN in AUTOMATION e.V., Germany, the international user and manufacturer association.
CODESYS [®]	CODESYS [®] is a registered trademark of CODESYS GmbH, Germany.
EnDat [®]	EnDat [®] and the EnDat [®] logo are registered trademarks of Dr. Johannes Heidenhain GmbH, Germany.
EtherCAT [®] , Safety over EtherCAT [®]	EtherCAT [®] and TwinCAT [®] are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany.
HIPERFACE [®]	HIPERFACE [®] and the HIPERFACE DSL [®] logo are registered trademarks of SICK AG, Germany.
PLCopen [®]	PLCopen [®] is a registered trademark of the PLCopen Organisation, Netherlands.
TwinCAT [®]	TwinCAT [®] is a registered and licensed trademark of Beckhoff Automation GmbH, Germany.

All other trademarks not listed here are the property of their respective owners.

Products that are registered as trademarks are not specially indicated in this documentation. Existing property rights (patents, trademarks, protection of utility models) are to be observed.

3 Safety notes

WARNING!

Risk of fatal injury if safety notes and residual risks are not observed!

Failure to observe the safety notes and residual risks in the drive controller documentation may result in accidents causing serious injury or death.

- Observe the safety notes in the drive controller documentation.
- Consider the residual risks in the risk assessment for the machine or system.

WARNING!

Malfunction of the machine due to incorrect or modified parameterization!

In the event of incorrect or modified parameterization, malfunctions can occur on machines or systems which can lead to serious injuries or death.

- Observe the security notes in the drive controller documentation.
- Protect the parameterization, e.g. from unauthorized access.
- Take appropriate measures for possible malfunctions (e.g. emergency off or emergency stop).

4 What you should know before commissioning

The following chapters provide a quick introduction to the structure of the program interface and accompanying window designations as well as relevant information about parameters and generally saving your project configuration.

4.1 Program interfaces

The following chapters include an overview of the program interfaces for the described software components.

4.1.1 DS6 program interface

Using the graphical interface of the DriveControlSuite commissioning software (DS6), you can project, parameterize and commission your drive project quickly and efficiently. In case of service, you can evaluate diagnostic information such as operating states, fault memories and fault counters of your drive project using DriveControlSuite.

Information

The program interface of DriveControlSuite is available in German, English and French. To change the language of the program interface, select **Settings > Language**.

Information

The DriveControlSuite help in the menu bar can be reached via **Help > Help for DS6** or via the **[F1]** key on your keyboard. When you press **[F1]** in an area of the program, the corresponding help topic opens.

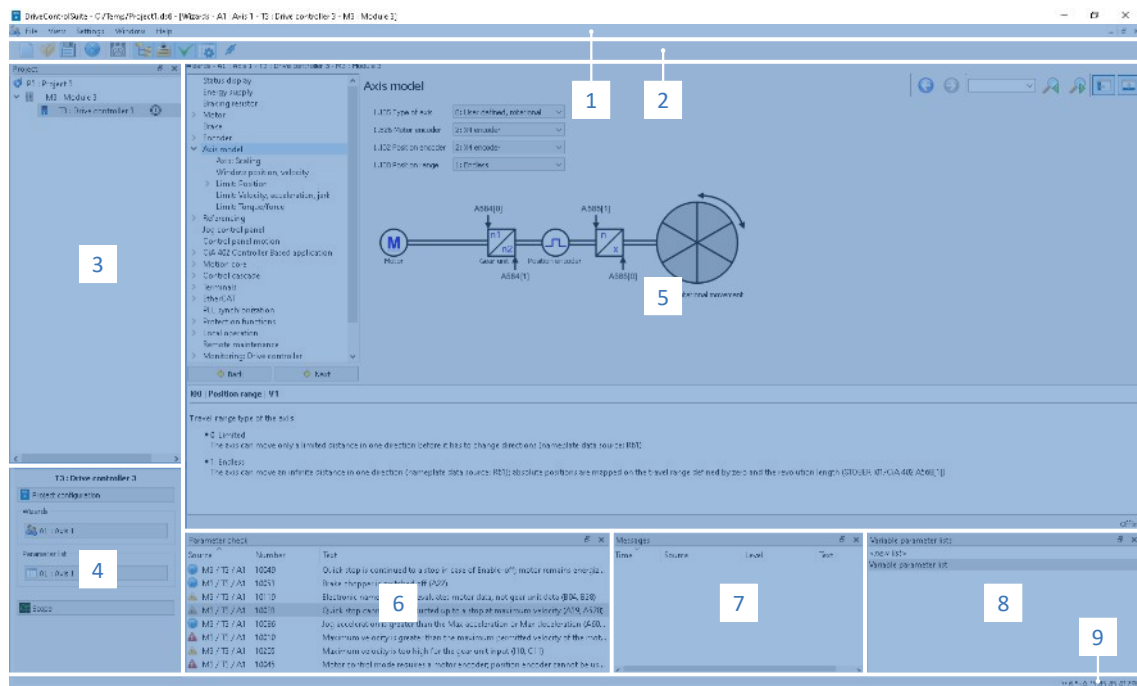


Fig. 1: DS6: Program interface

No.	Area	Description
1	Menu bar	Using the File, View, Settings and Window menus, you can open and save projects, display and hide program windows, select the interface language and access level and change between different windows in the workspace.
2	Toolbar	The toolbar enables quick access to frequently needed functions, like opening and saving projects and hiding and displaying windows in the program interface.
3	Project tree	The project tree forms the structure of your drive project in the form of modules and drive controllers. Select an element using the project tree first in order to edit it using the project menu.
4	Project menu	The project menu offers you various functions for editing the project, module and drive controller. The project menu adapts to the element that you selected in the project tree.
5	Workspace	The different windows which can be used to edit your drive project, such as the configuration dialog, wizards, the parameter list or the scope analysis tool, open in the workspace.
6	Parameter check	The parameter check points out irregularities and inconsistencies that were detected in the plausibility check of calculable parameters.
7	Messages	The entries in the messages log the connection and communication status of the drive controllers, incorrect inputs caught by the system, errors when opening a project or rule violations in the graphical programming.
8	Variable parameter lists	You can use variable parameter lists to compile any parameters in individual parameter lists for a quick overview.
9	Status bar	In the status bar, you can find the specifications of the software version and get additional information about the project file, the devices and the progress of the process during processes such as loading projects.

4.1.1.1 Configuring the view

In DriveControlSuite, you can change the visibility and arrangement of areas and windows, such as to optimize the available space in the workspace when working with smaller screens.

Showing/hiding areas

Use the icons in the toolbar or the items in the **View** menu to show or hide specific areas in DriveControlSuite as needed.

Icon	Item	Description
–	Reset	Resets the view to factory settings.
	Project	Shows/hides the Project window (project tree, project menu).
	Messages	Shows/hides the Messages window.
	Parameter check	Shows/hides the Parameter check window.
	Variable parameter lists	Shows/hides the Variable parameter lists window.

Arrange and group areas

You can undock and rearrange the individual areas via drag and drop. If you drag an undocked window to the edge of DriveControlSuite, you can release it there in a color-highlighted area either next to or on top of another window to redock it.

When you release the window onto another window, the two areas are merged into one window where you can use tabs to switch between the areas.

4.1.1.2 Navigation using sensitive circuit diagrams

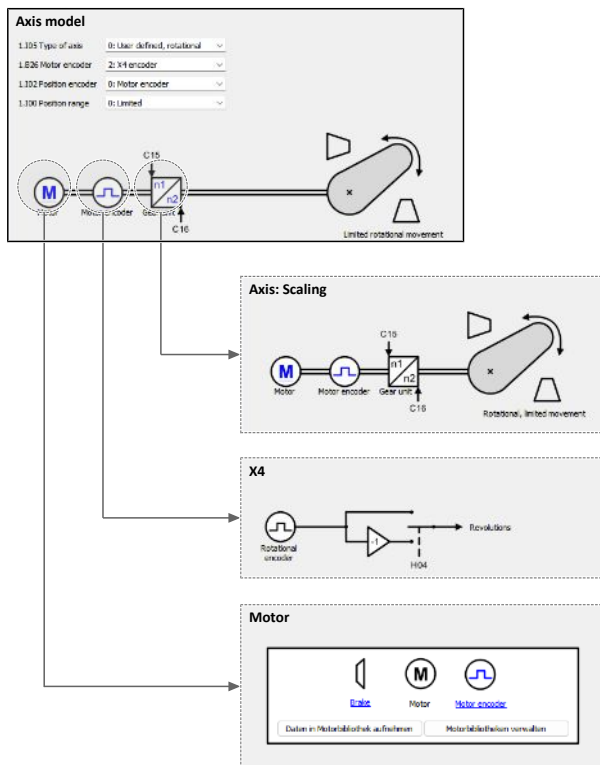


Fig. 2: DriveControlSuite: Navigation using text links and symbols

In order to graphically illustrate the processing sequence of actual and set values, the use of signals or the arrangement of drive components and to make configuring the accompanying parameters easier, they are displayed on the respective wizard pages of the workspace in the form of circuit diagrams.

Blue text links or clickable icons indicate links within the program. These refer to the corresponding wizard pages and, as a result, allow you to reach additional helpful detail pages with a click.

4.1.2 TwinCAT 3 program interface

In TwinCAT 3, you commission your EtherCAT system via the TwinCAT 3 Engineering development environment. The following graphic shows the interface elements relevant to this documentation.

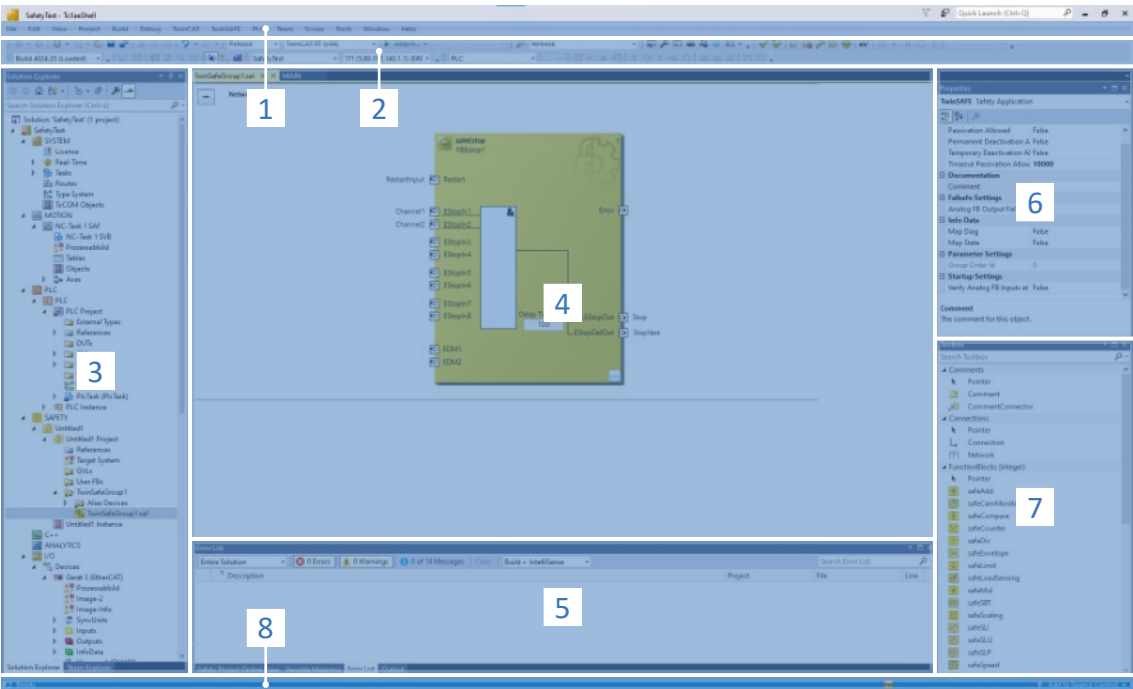


Fig. 3: TwinCAT 3 Engineering: Program interface

No.	Area	Description
1	Menu bar	The menu bar shows the menus set by default. Editor-specific menus appear only when the corresponding editor is open. Using the Tools menu, you can configure the programming interface and add to existing menus or define new ones, for example.
2	Toolbar	The toolbar enables quick access to frequently used functions, such as opening and saving projects.
3	Solution explorer	The solution explorer maps the structure of your project with the project elements it contains. First select an element using the solution explorer to edit it in the editor.
4	Main window (editor)	In the editor, you define and edit objects, e.g. graphical programming elements.
5	Message window	The Messages window informs you of any errors or warnings currently present. You also receive messages about the syntax check, compilation process, etc.
6	Properties window	The properties window shows the properties of the element selected in the solution explorer.
7	Toolbox	The toolbox displays the tools available for the active editor, such as graphical programming elements.
8	Information and status bar	The information and status bar informs you about the state of the system (Config, Run, Stop or Exception mode). In online operation, you can see the current status of the program. If an editor window is active, the current position of the cursor and the set editing mode may also be displayed.

4.1.3 CODESYS Development System V3 program interface

You can commission your EtherCAT system via the graphical user interface of the CODESYS Development System V3 programming software for industrial control and automation technology. The following graphic shows the interface elements relevant to this documentation.

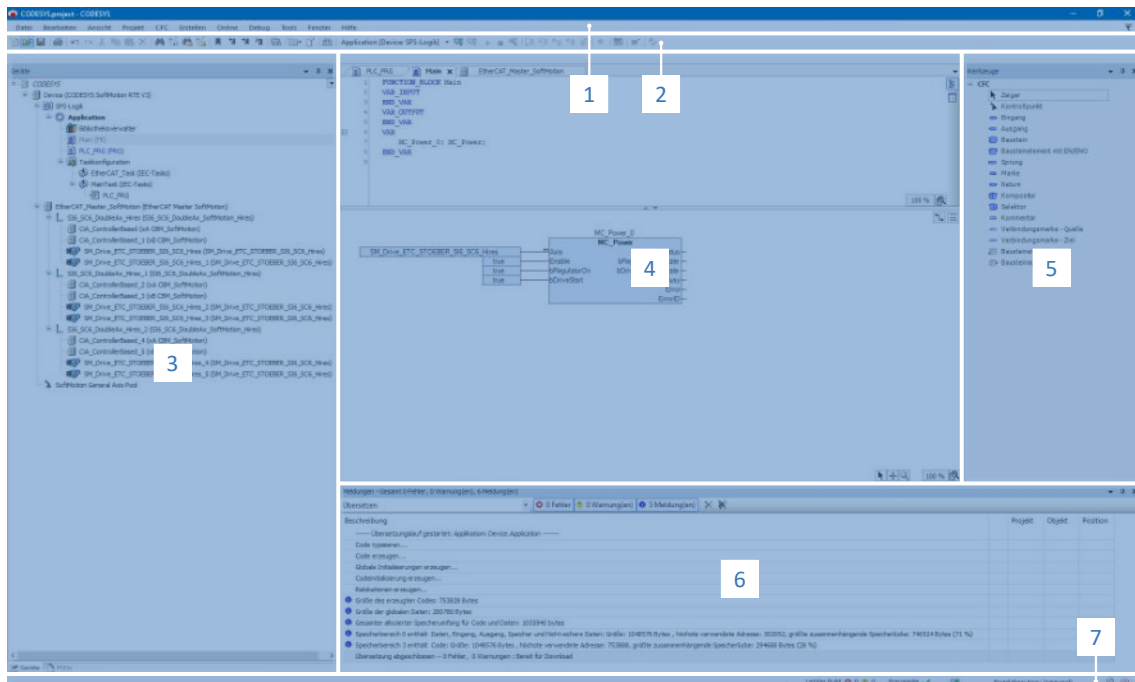


Fig. 4: CODESYS Development System V3: Program interface

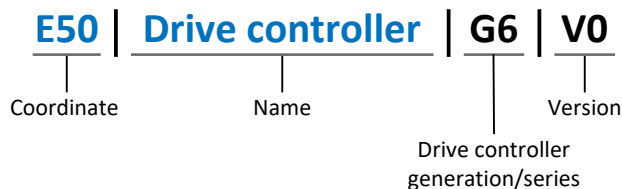
No.	Area	Description
1	Menu bar	The menu bar shows the menus set by default. Editor-specific menus only appear if the corresponding editor is active. Using the Tools menu, you can configure the programming interface and add to existing menus or define new ones, for example.
2	Toolbar	The toolbar enables quick access to frequently used functions, such as opening and saving projects.
3	Device tree	The device tree (Devices tab) shows the structure of your project with the device objects it contains. First select a device from the device tree to edit it in the editor.
4	Main window (editor)	In the editor, you define and edit objects, e.g. device settings and function blocks.
5	Tools	The Tools window displays the tools available for the active editor, e.g. graphical programming elements or elements for visualization.
6	Messages	The Messages window informs you of any errors or warnings currently present. You also receive messages about the syntax check, compilation process, etc.
7	Information and status bar	The information and status bar informs you about the state of the system and the current number of active errors and warnings. If an editor window is active, the current position of the cursor and the set editing mode may also be displayed.

4.2 Meaning of parameters

You can use parameters to adapt the function of the drive controller to your individual application. In addition, parameters visualize the current actual values (actual velocity, actual torque, etc.) and trigger actions such as Save values, Test phase, etc.

Interpretation of parameter identification

Parameter identification consists of the following elements, where short forms are also possible, i.e. only specifying a coordinate or the combination of coordinate and name.



4.2.1 Parameter groups

Parameters are assigned to individual groups by topic. The drive controllers differentiate between the following parameter groups.

Group	Topic
A	Drive controllers, communication, cycle times
B	Motor
C	Machine, velocity, torque/force, comparators
D	Set value
E	Display
F	Terminals, analog and digital inputs and outputs, brake
G	Technology – Part 1 (application-dependent)
H	Encoder
I	Motion (all motion settings)
J	Motion blocks
K	Control panel
L	Technology – Part 2 (application-dependent)
M	Profiles (application-dependent)
N	Additional functions (application-dependent; e.g. extended cam control unit)
P	Customer-specific parameters (programming)
Q	Customer-specific parameters, instance-dependent (programming)
R	Production data for the drive controller, motor, brakes, motor adapter, gearbox and geared motor
S	Safety (safety technology)
T	Scope
U	Protection functions
Z	Fault counter

Tab. 1: Parameter groups

4.2.2 Parameter types and data types

In addition to topic-based sorting in individual groups, all parameters belong to a certain data type and parameter type. The data type of a parameter is displayed in the parameter list, properties table. The connections between parameter types, data types and their value range can be found in the following table.

Data type	Parameter type	Length	Value range (decimal)
INT8	Integer or selection	1 byte (signed)	-128 – 127
INT16	Integer	2 bytes (1 word, signed)	-32768 – 32767
INT32	Integer or position	4 bytes (1 double word, signed)	-2 147 483 648 – 2 147 483 647
BOOL	Binary number	1 bit (internal: LSB in 1 byte)	0, 1
BYTE	Binary number	1 byte (unsigned)	0 – 255
WORD	Binary number	2 bytes (1 word, unsigned)	0 – 65535
DWORD	Binary number or parameter address	4 bytes (1 double word, unsigned)	0 – 4 294 967 295
REAL32 (single type according to IEE754)	Floating-point number	4 bytes (1 double word, signed)	$-3.40282 \times 10^{38} - 3.40282 \times 10^{38}$
STR8	Text	8 characters	—
STR16	Text	16 characters	—
STR80	Text	80 characters	—

Tab. 2: Parameters: Data types, parameter types, possible values

Parameter types: Use

- Integer, floating-point number
For general computing processes
Example: Set and actual values
- Selection
Numeric value to which a direct meaning is assigned
Example: Sources for signals or set values
- Binary number
Bit-oriented parameter information that is collected in binary
Example: Control and status words
- Position
Integer combined with associated units and decimal places
Example: Actual and set values of positions
- Velocity, acceleration, deceleration, jerk
Floating-point number combined with associated units
Example: Actual and set values for velocity, acceleration, deceleration, jerk
- Parameter address
Referencing of a parameter
Example: In F40 AO1 source, for example, E08 n-motor filtered can be parameterized
- Text
Outputs or messages

4.2.3 Parameter types

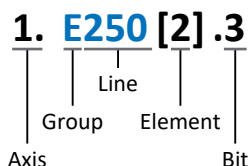
The following types of parameters are differentiated.

Parameter type	Description	Example
Simple parameters	Consist of one group and one line with a defined value.	A21 Brake resistor R: Value = 100 ohms
Array parameters	Consist of a group, a line and multiple sequential (listed) elements, which have the same properties but different values.	A10 Access level - A10[0] access level: value = Access level via operating unit - A10[2] Access level: value = Access level via CANopen, EtherCAT, and EtherNet/IP - A10[3] access level: value = Access level via PROFINET
Record parameters	Consist of a group, a line and multiple sequential (listed) elements, which can have different properties and different values.	A00 Save values - A00[0] Start: Value = Start action - A00[1] Progress: Value = Display action progress - A00[2] Result: Value = Display action result

Tab. 3: Parameter types

4.2.4 Parameter structure

Every parameter has specific coordinates with the following structure.



- Axis (optional)
In case of multiple axes, the one to which a parameter is assigned; not applicable for global parameters (value range: 1 – 4).
- Group
The thematic group to which a parameter belongs (value range: A – Z).
- Line
Distinguishes the parameters within a parameter group (value range: 0 – 999).
- Element (optional)
Elements of an array or record parameter (value range: 0 – 16000).
- Bit (optional)
Selection of a single bit for complete data addressing; depends on the data type (value range: 0 – 31).

4.2.5 Parameter visibility

The visibility of a parameter is controlled by the access level you set in DriveControlSuite and by the properties you project for the respective drive controller (e.g. hardware, firmware and application). A parameter can also be shown or hidden depending on other parameters or settings. For example, the parameters of an additional function are only shown as soon as you activate the relevant additional function.

Access level

The access options for the individual software parameters are ranked hierarchically and divided into individual levels. This means that parameters can be hidden for a specific purpose and, relatedly, their configuration options can be locked starting from a specific level.

Each parameter has one access level for read access (visibility) and one access level for write access (editability). The following levels are present:

- Level 0
Elementary parameters
- Level 1
Important parameters of an application
- Level 2
Important parameters for service with extensive diagnostic options
- Level 3
All parameters needed for commissioning and optimizing an application

The parameter A10 Access level controls general access to parameters:

- Over CANopen, EtherCAT or EtherNet/IP (A10[2])
- Over PROFINET (A10[3])

Hardware

Which parameters are available to you in DriveControlSuite is determined by which series you select in the configuration dialog for the drive controller, for example, or whether you project an option module. Basically, the only parameters that are displayed are the ones you need to parameterize the configured hardware.

Firmware

Due to the further development and updating of functions for the drive controllers, new parameters and also new versions of existing parameters are continuously being implemented in DriveControlSuite and in the firmware. The parameters are displayed in the software according to the DriveControlSuite version used and the configured firmware version of the respective drive controller.

Applications

Applications generally differ in terms of functions and their control. For this reason, different parameters are available with each application.

Information

If a parameter exists in the configuration due to the projected properties of the drive controller (e.g. hardware, firmware, application), it is generally available for fieldbus communication. The visibility of a generally available parameter has no effect on its availability for fieldbus communication (e.g. through access level or parameter selection).

4.3 Signal sources

Drive controllers are controlled either by a fieldbus, by terminals or by mixed operation consisting of a fieldbus system and terminals. You can use the corresponding selection parameters, referred to as signal sources, to configure in DriveControlSuite whether the control signals and set values of the application are obtained over a fieldbus or using terminals.

In case of activation by using terminals, the respective analog or digital inputs are specified directly as the source. In case of activation by fieldbus, parameters are selected as sources for control signals and set values, which must be part of the process data mapping between the controller and drive controller so that the controller can write to them via fieldbus.

4.4 Non-volatile memory

All project configurations, parameterizations and related changes to parameter values are in effect after transmission to the drive controller, but are only stored in volatile memory.

Saving to a drive controller

To save the configuration in non-volatile memory on a drive controller, you have the following options:

- Saving the configuration using the **Save values wizard**:
Project menu > Wizards area > Projected axis > Save values wizard:
Select the Save values action
- Saving the configuration using the parameter list:
Project menu > Parameter list area > Projected axis > Group A: Drive controller > A00 Save values:
Set the parameter A00[0] to the value 1: Active
- Saving the configuration using the S1 operating button:
Press and hold the operating button for 3 s

Saving to all drive controllers within a project

To save the configuration in non-volatile memory on several drive controllers, you have the following options:

- Saving the configuration using the toolbar:
Toolbar > Save values icon: Click the Save values icon
- Saving the configuration using the Online functions window:
Project menu > Online connection button > Online functions window: Click on Save values (A00)

Information

Do not shut off the drive controller while saving. If the supply voltage to the control unit is interrupted while saving, the drive controller will start with the last correct configuration that was saved and with fault 40: Invalid data the next time it is switched on. In order to acknowledge the fault and successfully complete the saving process, the configuration must be stored again in non-volatile memory and the action must be completed.

5 Commissioning

The following chapters describe commissioning your EtherCAT network using DriveControlSuite in combination with the TwinCAT 3 Engineering automation software or CODESYS Development System V3.

We assume the following system environment as an example so that you can follow the individual commissioning steps exactly:

- Drive controllers from the SC6 or SI6 series with firmware version 6.7-D-EC or later
- DriveControlSuite commissioning software, version 6.7-D or later

in conjunction with

- Beckhoff controller
- Beckhoff TwinCAT 3 Engineering automation software

or in combination with

- CODESYS SoftMotion controller with SM3_Drive_ETC_STOEBER_SB6_SC6_SI6 device driver
- CODESYS Development System automation software, V3.5 SP21 or later

Commissioning is divided into the following steps:

1. DriveControlSuite
Project all of the drive controllers, i.e. set the application type, device control, process data for fieldbus communication and mechanical axis model in DriveControlSuite.
Depending on the selected axis control (controller-based or drive-based), scale your axis models on either the drive controller side or the controller side.
In both cases, transfer your configuration to the drive controllers of the system network.
2. TwinCAT 3 Engineering or CODESYS Development System V3
Map the hardware environment of your EtherCAT network in the automation software and scale your axis model based on the selected axis control.
Synchronize the operation of local clocks (distributed clocks) in all EtherCAT nodes and configure the settings for the respective connection and the communication between the drive controller and the controller.
Transfer the configuration to the controller and then verify that the axis is functioning properly.

5.1 DS6: Configuring the drive controller

Project and configure all drive controllers for your drive system in DriveControlSuite (see also [DS6 program interface \[► 11\]](#)).

Information

Always perform the steps described below in the specified order!

Some parameters are interdependent and do not become accessible to you until you have first configured certain settings. Follow the steps in the specified sequence so that you can finish the parameterization completely.

5.1.1 Initiating the project

In order to be able to configure all drive controllers and axes of your drive system using DriveControlSuite, you must record them as part of a project.

5.1.1.1 Projecting the drive controller and axis

Create a new project and project the first drive controller along with the accompanying axis.

Information

Make sure that you project the correct series in the **Drive controller** tab. The projected series cannot be changed afterwards.

Creating a new project

1. Start DriveControlSuite.
2. On the start screen, click **Create new project**.
 - ⇒ The new project is created and the configuration dialog for the first drive controller opens.
 - ⇒ The **Drive controller** button is active.

Projecting the drive controller

1. Properties tab:

Establish the relationship between your circuit diagram and the drive controller to be projected in DriveControlSuite.

1.1. Reference:

Define the reference code (equipment code) of the drive controller.

1.2. Designation:

Give the drive controller a unique name.

1.3. Version:

Version your project configuration.

1.4. Description:

If necessary, save additional supporting information (e.g., the change history).

2. Drive controller tab:

Select the series, device type and firmware version of the drive controller.

2.1. Firmware:

Select the EtherCAT version 6.x -EC.

3. Option modules tab, Safety module:

If the drive controller is part of a safety circuit, select the corresponding safety module.

4. Device control tab:

Project the basic control of the drive controller.

4.1. Device control:

Select CiA 402 device control.

4.2. Process data Rx, process data Tx:

Select EtherCAT Rx SDO Info and EtherCAT Tx for transmission of the EtherCAT process data.

Projecting the axis

1. Click Axis A.
2. **Properties tab:**
Establish the relationship between your circuit diagram and the axis to be projected in DriveControlSuite.
 - 2.1. **Reference:**
Define the reference code (equipment code) of the axis.
 - 2.2. **Designation:**
Give the axis a unique name.
 - 2.3. **Version:**
Version your project configuration.
 - 2.4. **Description:**
If necessary, save additional supporting information (e.g., the change history).
3. **Application tab:**
Select the controller-based CiA 402 application.
4. **Motor tab:**
Select the type of motor you operate with this axis. If you are working with motors from third-party suppliers, enter the accompanying motor data later.
5. Repeat the steps for axis B (only for double-axis controllers).
6. Confirm with OK.

5.1.1.2 Configuring safety technology

If the drive controller is part of a safety circuit, you must configure the safety technology in accordance with the commissioning steps outlined in the corresponding manual in the next step (see [Detailed information](#) [► 165]).

5.1.1.3 Creating other drive controllers and modules

In DriveControlSuite, all drive controllers within a project are grouped using modules. If you add a new drive controller to your project, be sure to always assign it to an existing module. Group drive controllers in a module if, for example, they are located in the same control cabinet or work together to operate the same machine part.

Creating a drive controller

1. In the project tree, select your project P1 > module M1 > context menu **Create new drive controller**.
⇒ The drive controller is created in the project tree and the configuration dialog opens.
2. Project the drive controller as described in [Projecting the drive controller and axis](#).
3. Repeat the steps for all other drive controllers that you want to project.

Creating a module

1. In the project tree, select your project P1 > context menu **Create new module**.
⇒ The module is created in the project tree.
2. Project the module as described in [Projecting the module](#) [► 26].
3. Repeat the steps for all other modules that you want to project.

5.1.1.4 Projecting the module

Give your module a unique name, enter the reference code and, as an option, store additional information like the version and change history of the module.

1. Select the module in the project tree and click on **Project configuration** in the project menu.
⇒ The configuration dialog for the module opens.
2. Establish the relationship between your circuit diagram and the module in DriveControlSuite.
 - 2.1. **Reference:**
Define the reference code (equipment code) of the module.
 - 2.2. **Designation:**
Give the module a unique name.
 - 2.3. **Version:**
Version the module.
 - 2.4. **Description:**
If necessary, save additional supporting information (e.g., the change history).
3. Confirm with **OK**.

5.1.1.5 Projecting the project

Give your project a unique name, enter the reference code and, as an option, store additional information like the version and change history of the project.

1. Select the project in the project tree and click on **Project configuration** in the project menu.
⇒ The configuration dialog for the project opens.
2. Establish the relationship between your circuit diagram and the project in DriveControlSuite.
 - 2.1. **Reference:**
Define the reference code (equipment code) of the project.
 - 2.2. **Designation:**
Give the project a unique name.
 - 2.3. **Version:**
Version the project.
 - 2.4. **Description:**
If necessary, save additional supporting information (e.g., the change history).
3. Confirm with **OK**.

5.1.2 Mapping the mechanical axis model

To be able to put your real drive train with one or more drive controllers into operation, you must map your complete mechanical environment in DriveControlSuite.

The 6th generation of STOBER drive controllers are specially developed for communication between the drive controller and controller on the basis of the actual variables at the output (° or mm of real axis movement). The scaling of the axis model is calculated by the drive controller's firmware independently of the encoder model, without any rounding errors or drift.

Information

The scaling of the axis depends on the selected axis control. In controller-based operation, the scaling of the axis is done primarily within the controller. The necessary settings in DriveControlSuite are described below. In drive-based operation, the scaling of the axis is done entirely within DriveControlSuite. In this case, the axis model can be freely parameterized to meet the requirements of drive-based operation.

When scaling the axis, follow the instructions for the axis control you have selected.

5.1.2.1 Parameterizing the motor

You have projected one of the following motors:

Synchronous servo motor with EnDat 2.2 digital encoder or EnDat 3 (with optional brake)

When the project is configured for the corresponding motor, limiting values for currents and torques as well as associated temperature data are automatically transferred to the respective parameters of the individual wizards. All additional data on the brake and encoder is transferred at the same time.

Lean motor (with optional brake)

When the project is configured for the corresponding motor, limiting values for currents and torques as well as associated temperature data are automatically transferred to the respective parameters of the individual wizards. You only have to parameterize the cable length in use and the mass inertia ratio of the load to the motor. Even the brake purging and engaging times are already stored. You just have to activate the brake.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the **Motor** wizard.
3. **B101 Cable length:**
Select the cable length of the power cable in use.
4. Select the wizard **Control cascade > Control mode**.
5. **C30 J-Last / J-Motor:**
Parameterize the mass inertia ratio of the load to the motor.
6. Repeat the steps for the 2nd axis (only for double-axis controllers).

Then activate the brake.

1. Select the relevant drive controller in the project tree and click on the first projected axis in the Project menu > Wizard area.
2. Select the **Brake** wizard.
3. **F00 Brake:**
Select 1: Active.
4. Repeat the steps for the 2nd axis (only for double-axis controllers).

Information

If you are operating only one axis on a double-axis controller, the second axis must be parameterized correctly, and the motor temperature sensor must be bypassed to prevent faults. For further information, refer to the manual of your drive controller.

5.1.2.2 Parameterizing the axis model

Parameterize the setup of your drive in this order:

- Define the axis model
- Scale the axis
- Parameterize the position and velocity window
- Limit the axis (optional)
 - Limit the position
 - Limit the velocity, acceleration and jerk
 - Limit the torque and force

Information

If you are using a double-axis controller with two projected axes, you must parameterize the axis model for each axis individually.

5.1.2.2.1 Defining the axis model

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the **Axis model wizard**.
3. I05 Type of axis:
Select 0: User defined, rotational.
4. B26 Motor encoder:
Select the interface to which the motor encoder is connected.
5. I02 Position encoder:
Select the interface to which the position encoder is connected.
6. I00 Position range:
Select 0: Limited.

Information

If you do not parameterize it differently for I02 Position encoder, B26 Motor encoder is used for position control as standard.

5.1.2.2.2 CiA 402: Scaling an axis

If you have projected CiA 402, you mainly scale the axis in the control software and only define the polarity of the axis in DriveControlSuite (parameter: A571).

✓ You have projected CiA 402

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model > Axis: Scaling wizard.
3. A585[0] Feed constant. Feed¹, A585[1] Feed constant.Shaft revolutions²:
Leave the presets of A585[0] at 1048576 inc (= 20 bits = 2²⁰) and the presets of A585[1] at 1 U and adjust the corresponding value in the controller software.
4. I06 Decimal places position:
Leave the default value at 0.
5. A571 Polarity:
Use the polarity to define the direction of interpretation between the axis movement and motor movement.

5.1.2.2.3 Parameterizing the position and velocity window

Enter position limits and velocity zones for actual values. To do so, parameterize boundary values for reaching a position or velocity.

1. Select the Axis model wizard > Window position, velocity.
2. C40 Velocity window:
Parameterize a tolerance window for velocity tests.
3. I22 Target window:
Parameterize a tolerance window for position tests.
4. I87 Actual position in window time:
Parameterize how long a drive must stay in the specified position window before a corresponding status message is output.
5. A546 Following error window:
Parameterize a tolerance window for lag tests.

For more information on monitoring the lag, see [Lag monitoring](#) [► 147].

¹Corresponds to object 6092 hex, subindex 1 hex for axis A and object 6892 hex, subindex 1 hex for axis B

²Corresponds to object 6092 hex, subindex 2 hex for axis A and 6892 hex, subindex 2 hex for axis B

5.1.2.2.4 Limiting the axis

As an option, you can limit the maximum permitted motion variables of position, velocity, acceleration, jerk and torque/force according to your application.

Information

To simplify the scaling and limiting of the axis, the *Axis model wizard* > *Axis: Scaling* provides you with the **Conversion of position, velocities, accelerations, torque/force** scaling calculator, which calculates the effects of changed motion variables on the entire system. You can use the scaling calculator to enter values for motion variables of the motor, gearbox output and axis in order to convert the values to all other locations in the axis model.

Position limiting

To secure the travel range of the axis, you have the option to limit the permitted positions using a software or hardware limit switch.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the *Axis model* > *Limit: Position wizard*.
3. I101 Source positive /limit switch, I102 Source negative /limit switch:
To limit the travel range of the axis in the positive or negative direction of motion via hardware limit switches, select the source of the digital signal that is used to evaluate a limit switch at the positive or negative end of the travel range.
 - 3.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
4. A570[1], A570[0] Software position limit:
If you selected 0: Limited for I00 and would like to limit the travel range of the axis via software limit switches, define the largest or smallest permitted position for software position limiting.

ATTENTION!

Material damage due to leaving the permitted travel range

When a limit switch is overshot, the axis at the end of the permitted travel range will go into a fault condition, either with or without a quick stop depending on the parameterization of the device control, so it may come to a standstill past the limit switch and outside the permitted travel range.

- Allow sufficient space past the limit switch for your application to bring the axis to a standstill.

Velocity, acceleration, jerk limiting

As an option, you can limit the motion variables of velocity, acceleration and jerk and define the quick stop deceleration according to your application. The default values are designed for slow velocities without gearboxes.

1. Select the **Motor wizard**.
2. **B83 v-max motor**:
Determine the maximum permitted velocity of the motor.
3. Select the **Axis model > Axis: Scaling wizard**.
4. **Conversion of positions, velocities, accelerations, torque/force section**:
Use the scaling calculator and the maximum permitted velocity of the motor to determine the maximum permitted velocity of the output as a reference value.
5. Select the **Axis model > Limit: Velocity, acceleration, jerk wizard**.
6. **I10 Maximal speed**:
Define the maximum permitted velocity of the output according to your application (taking into account the previously determined reference value).
7. **A604 Max acceleration, A605 Max deceleration**:
Set the maximum permitted acceleration for the output.
8. **I16 Maximal jerk**:
Define the maximum permitted jerk for the output.
9. **A578 Quick stop deceleration**:
Set the desired quick stop deceleration for the output.
10. Repeat the steps for axis B (only for double-axis controllers).

Limiting the torque/force

As an option, you can limit the torque/force according to your application. The default values take into account the rated operation together with the overload reserves.

1. Select the wizard **Axis model > Limit: Torque/force**.
2. **C03 Maximum positive torque/force, C05 Maximum negative torque/force**:
Set the maximum permitted set torque/maximum permitted set force.
3. **C08 Maximum torque/force for quick stop**:
Set the maximum permitted set torque/maximum permitted set force in case of a quick stop and in case of drive-controlled emergency stop, SS1 and SS2.

Information

When using the CiA 402 application, the following additional parameters are available for torque limiting: A559 Maximum torque, A315 Max. torque positive, and A316 Max. torque negative. The effective limit is determined in each case by the minimum of the values from C03/C05 and A559/A315/A316.

5.1.3 Configuring EtherCAT settings

In DriveControlSuite, configure the general EtherCAT settings, customize the process data mapping for each axis as needed for your application, synchronize your EtherCAT nodes, and, if applicable, create the device description (ESI file) for import into the automation software.

5.1.3.1 Parameterizing general EtherCAT settings

- ✓ You have projected a device control with the process data as part of drive controller and axis projecting.
- 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
- 2. Select the EtherCAT wizard.
- 3. A258 EtherCAT PDO-Timeout:
To allow detection of a communication failure, define the tolerated failure time for monitoring the receive PDO (value range: 0 – 65535 ms). For more information, see the parameter description for A258.

5.1.3.2 Configuring PDO transmission

PDO channels provide real-time transmission of control and status information, as well as actual and set values, from an EtherCAT MainDevice to EtherCAT SubDevices and vice versa.

PDO communication allows for several PDO channels to be operated simultaneously per transmission and sending direction. The channels for axes A and B each include a PDO with a defined sequence of up to 24 parameters to be transmitted. These are free to be configured in any way. One channel is reserved for FSoE communication and is parameterized automatically.

In order to guarantee error-free communication between the controller and the drive controller, STOBER offers application-dependent pre-assignment of the process data channels, which you can adapt to your specific application. Further information on standard mapping can be found in the manual for the respective application.

If you make changes to the process data mapping, you must make these both on the drive controller in the DriveControlSuite and in the control software.

Information

When using CODESYS SoftMotion, no manual adjustment of the PDO mapping is required. The standard STOBER mapping is tailored to the requirements of CODESYS SoftMotion and is stored in the SoftMotion driver. This allows the controller and the drive controller to communicate correctly with each other without any additional configuration. The same applies to TwinCAT 3 with NC or CNC axes: Here, too, the standard mapping is ready for immediate use.

5.1.3.2.1 Adapting RxPDO

- ✓ You have configured the global EtherCAT settings.
- 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
- 2. Select the EtherCAT wizard > Received process data RxPDO.
- 3. Check the preset process data and configure it according to your requirements if necessary by adding or removing parameters.

5.1.3.2.2 Adapting TxPDO

- ✓ You have configured the global EtherCAT settings.
- 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
- 2. Select the EtherCAT wizard > Transmitted process data TxPDO.
- 3. Check the preset process data and configure it according to your requirements if necessary by adding or removing parameters.

5.1.3.3 Synchronizing EtherCAT nodes

Precise synchronization of the EtherCAT nodes is absolutely required for spatially distributed processes that require simultaneous actions (path interpolation). EtherCAT provides the distributed clocks (DC-Sync) method for this. Synchronization using distributed clocks is more precise than SyncManager event synchronization (SM-Sync) because it is subject to fewer fluctuations. For this reason, DC-Sync is preconfigured in the EtherCAT MainDevice and SubDevices.

DC-Sync optimization wizard

In the first step, leave the presets as they are. As soon as you have commissioned the EtherCAT network, you can use the wizard to evaluate the quality of communication and optimize it if necessary.

More information on synchronization and how to adjust it afterwards can be found in the manual for communication with EtherCAT.

5.1.3.4 Creating an ESI file

Depending on which automation software you are using, you must create a device description (ESI file) in DriveControlSuite.

5.1.3.4.1 Creating an ESI file for TwinCAT 3

If you are working with TwinCAT 3, create an ESI file to import the STÖBER drive controllers into the device directory of the automation software, as described below.

The ESI file is the same for the SI6 drive controllers STÖBER SB6, and SC6 and contains all the relevant data of an EtherCAT SubDevice that must be imported into the EtherCAT MainDevice to configure the EtherCAT system with the automation software.

TwinCAT 3 can only import one ESI file, i.e. if you use different applications or PDO mappings for the same drive controllers, you must extend your ESI file accordingly.

If you make changes to the project configuration or the PDO mapping in the meantime, you must generate a new ESI file and make it available to TwinCAT 3.

Information

For a user-specific PDO mapping, the corresponding identifier can be entered in the E72 parameter. TwinCAT 3 can use this to uniquely identify the drive controller's PDO mapping.

Further information on ESI files can be found in the EtherCAT manual.

- ✓ You have completed the configuration of PDO mapping.
- 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
- 2. Select the EtherCAT wizard.
- 3. Click Export ESI.
 - ⇒ The Write ESI dialog box opens.
- 4. Save the ESI file (*.xml) in the directory where the controller will read it from (TwinCAT 3 Engineering default installation: C:\TwinCAT\3.1\Config\IO\EtherCAT).
- ⇒ The ESI file is imported the next time TwinCAT 3 Engineering is started.

5.1.3.4.2 Creating an ESI file for CODESYS Development System V3

Information

When using the CODESYS SoftMotion extension, there is no need to create an ESI file. A compatible ESI file is already included in the SoftMotion driver.

If you are working with CODESYS Development System V3 without SoftMotion, create an ESI file to import the STOBER drive controllers into the device directory of the automation software, as described below.

The ESI file is the same for the SI6 drive controllers STOBER SB6, and SC6 and contains all the relevant data of an EtherCAT SubDevice that must be imported into the EtherCAT MainDevice to configure the EtherCAT system with the automation software.

CODESYS Development System V3 can import multiple ESI files with different mappings. In other words, if you use different applications or PDO mappings for the same drive controller, you can create a new ESI file and import it into CODESYS Development System V3.

Further information on ESI files can be found in the EtherCAT manual.

✓ You have completed the configuration of PDO mapping.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the EtherCAT wizard.
3. Click Export ESI.
⇒ The Write ESI dialog box opens.
4. Save the ESI file (*.xml) to a directory of your choice.
⇒ The ESI file is ready for import into CODESYS Development System V3.

5.1.4 Transmitting and saving a configuration

In order to transmit and save the configuration to one or more drive controllers, you must connect your PC and the drive controllers over the network.

WARNING!

Injury to persons and material damage due to axis movement!

If there is an online connection between DriveControlSuite and the drive controller, changes to the configuration can lead to unexpected axis movements.

- Only change the configuration if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.
- For access via remote maintenance, there must be a communication link between you and a person on site with eye contact to the axis.

Information

During the search, all drive controllers within the broadcast domain are found via IPv4 limited broadcast.

Requirements for finding a drive controller in the network:

- Network supports IPv4 limited broadcast
- All drive controllers and the PC are in the same subnet (broadcast domain)

5.1.4.1 Transmitting the configuration

The steps for transmitting the configuration vary depending on the safety technology.

Drive controller without SX6 option (without extended safety technology)

- ✓ The drive controllers are switched on and can be found in the network.
- 1. In the project tree, select the module under which you have recorded your drive controller and click **Online connection** in the project menu.
 - ⇒ The **Add connection** dialog box opens. All drive controllers found via IPv4 limited broadcast are displayed.
- 2. **Direct connection** tab, **IP address** column:
 - Activate the IP addresses in question and confirm your selection with **OK**.
 - ⇒ The **Online functions** window opens. All drive controllers connected through the selected IP addresses are displayed.
- 3. Select the module and the drive controller to which you would like to transfer the configuration. Change the selection of transmission type from **Read** to **Send**.
- 4. Change the selection **Create new drive controller**:
 - Select the configuration that you would like to transfer to the drive controller.
- 5. Repeat steps 3 and 4 for all other drive controllers to which you would like to transfer your configuration.
- 6. **Online** tab:
 - Click **Establish online connections**.
 - ⇒ The configurations are transferred to the drive controllers.

Drive controller with SX6 option (with advanced safety technology)

- ✓ The drive controllers are switched on and can be found in the network.
- 1. In the project tree, select the module under which you have recorded your drive controller and click **Online connection** in the project menu.
 - ⇒ The **Add connection** dialog box opens. All drive controllers found via IPv4 limited broadcast are displayed.
- 2. **Direct connection** tab, **IP address** column:
Activate the IP addresses in question and confirm your selection with **OK**.
 - ⇒ The **Online functions** window opens. All drive controllers connected through the selected IP addresses are displayed.
- 3. Select the module and the drive controller to which you would like to transfer the configuration. Change the selection of transmission type from **Read** to **Send**.
- 4. Change the selection **Create new drive controller**:
Select the configuration that you would like to transfer to the drive controller.
- 5. Repeat steps 3 and 4 for all other drive controllers to which you would like to transfer your configuration.
- 6. **Online** tab:
Click **Establish online connections**.
 - ⇒ The configurations are transferred to the drive controllers.
 - ⇒ The **PASmotion Safety Configurator** configuration tool opens.

The safety configuration is transferred using the **PASmotion Safety Configurator**:

1. In the project administration, navigate from **PASmotion Safety Configurator** to the safety module for the drive controller and double-click to open it.
 - ⇒ The wizard for device synchronization opens.
 - ⇒ The project configuration and device configuration are checked against each other.
2. If the configurations match, click **Done** after device synchronization has finished.
3. Optional: If the configurations do not match, click **Next** after device synchronization has finished.
 - 3.1. Confirm the serial number of the safety module and click **Next**.
 - 3.2. Enter the password for the configuration on the safety module and click **Next**.
 - 3.3. Click **Download**.
 - ⇒ The project configuration is transferred to the safety module.
 - 3.4. After the successful transfer, click **Done**.
4. **Home** page, **CRC safety configuration**:
Document the checksum of the safety functions in the machine documentation.
5. Repeat the steps for each additional safety module in your project.
6. Exit **PASmotion Safety Configurator**.
 - ⇒ Transfer of the configuration is completed.

Information

If you do not know the password for the configuration on the safety module and would like to send a new safety configuration, you can delete the safety configuration on the safety module in **DriveControlSuite** using parameter **S33**.

5.1.4.2 Saving the configuration

- ✓ You have successfully transferred the configuration.
- 1. Online functions **window**, Online **tab**, Actions for drive controller in online operation **area**:
Click Save values (A00).
 - ⇒ The Save values (A00) window opens.
- 2. Select on which drive controllers you want to save the configuration.
- 3. Click Start action.
 - ⇒ The configuration is stored on the drive controllers in non-volatile memory.
- 4. Close the Save values (A00) window.

Information

For the configuration to take effect on the drive controller, a restart may be required in certain cases, such as after the configuration is saved on the drive controller for the first time or when changes are made to the firmware or process data mapping.

Restarting a drive controller

- ✓ You have stored the configuration on the drive controller in non-volatile memory.
- 1. Online functions **window**, Online **tab**:
Click Restart (A09).
 - ⇒ The Restart (A09) window opens.
- 2. Select which of the connected drive controllers you want to restart.
- 3. Click Start action.
- 4. Confirm the safety note with OK.
 - ⇒ The Restart (A09) window closes.
- ⇒ The fieldbus communication and connection between DriveControlSuite and drive controllers are interrupted.
- ⇒ The selected drive controllers restart.

5.1.5 Testing the configuration

After you have transferred the configuration to the drive controller, first check your projected axis model and the parameterized electrical and mechanical data for plausibility before continuing with the parameterization.

Test the configuration with drive-based control. For controller-based operating modes, this step is optional.

Information

Make sure that the values of the control panel are compatible with your projected axis model in order to obtain useful test results with which you can optimize your configuration for the respective axis.

The scaling calculator is available under the **Axis model > Axis: Scaling wizard** in order to recalculate the values for the control panel according to your projected axis model.



WARNING!

Injury to persons and material damage due to axis movement!

When you activate the control panel, DriveControlSuite gives you sole control over the axis movements. If you are using a controller, it cannot intervene to prevent collisions while the control panel is active. The controller takes over control again when the control panel is deactivated, which can cause unexpected axis movements.

- Do not switch to other windows when the control panel is active.
- Only use the control panel if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.

Testing the configuration using the jog control panel

- ✓ There is an online connection between DriveControlSuite and the drive controller.
 - ✓ You have successfully stored the configuration on the drive controller.
 - ✓ No safety function is active.
1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Jog control panel wizard.
 3. Click Control panel on and then Enable.
 - ⇒ The axis is monitored via the active control panel.
 4. Check the default values of the control panel and adjust them to your projected axis model if necessary.
 5. To test the configuration of your projected axis for direction of motion, velocity, etc., move the axis gradually using the Jog+, Jog-, Jog step+ and Jog step- buttons.
 6. Use your test results to optimize your configuration as necessary.
 7. To deactivate the control panel, click on Control panel off.

Information

Jog+ and Jog- cause a continual manual movement in the positive or negative direction. Jog step+ and Jog step- move the axis relative to the current actual position by the increment specified in I14.

Jog+ and Jog- have a higher priority than Jog step+ and Jog step-.

5.2 TwinCAT 3: Setting up an EtherCAT system

If you are using a Beckhoff controller, you map the hardware environment of your EtherCAT network in the TwinCAT 3 automation software. Configure all EtherCAT nodes, parameterize the necessary connection settings, and configure the data exchange between the controller and the drive controller. Then transfer the configuration to the controller and check the cyclic communication.

Before commissioning your EtherCAT network using the automation software, you must have physically connected all system devices and projected the relevant drive controllers in DriveControlSuite. For commissioning purposes, we assume in the example that you have projected the CiA 402 application for controller-based operation.

Drive-based operation requires manual programming in the automation software. For more information, see [Drive-based axis control](#) [► 48].

Information

Always perform the steps described below in the specified order!

Some parameters are interdependent and do not become accessible to you until you have first configured certain settings. Follow the steps in the specified sequence so that you can finish the parameterization completely.

5.2.1 Creating a standard project

- ✓ You have saved the ESI file generated from the DriveControlSuite in the specified directory (C:\TwinCAT\3.1\Config\IO\EtherCAT).
- 1. Start TwinCAT 3 Engineering.
 - ⇒ TwinCAT 3 Engineering opens; the Start Page tab is active.
 - ⇒ The stored ESI file is read in upon program start.
- 2. Select File > New > Project....
 - ⇒ The New Project window opens.
- 3. Select Installed > Templates > TwinCAT Projects > TwinCAT XAE Project (XML format).
- 4. Name, Location, Solution name:
Label the project and enter a save location and an internal project name.
- 5. Confirm with OK.

5.2.2 EtherCAT MainDevice: Activating Config-Modus

To be able to configure the EtherCAT MainDevice online, after creating a project, add a connection between TwinCAT 3 Engineering and the controller (EtherCAT MainDevice) if necessary and activate Config mode.

Adding routing to the controller

If Run-Time and TwinCAT 3 Engineering are installed on different PCs, first establish a connection by adding a routing rule to the controller.

- ✓ The EtherCAT MainDevice is connected to the network, all system components are energized and the infrastructure is ready for operation.
- 1. In TwinCAT 3 Engineering, click on the <Local> list field in the toolbar and select Choose Target System....
 - ⇒ The Choose Target System window opens.
- 2. Click on Search (Ethernet)....
 - ⇒ The Add Route Dialog window opens.
- 3. Click on Broadcast Search.
 - ⇒ The Select Adapter(s) window opens.
- 4. Highlight the adapter that is connected with your controller and confirm with OK.
 - ⇒ All available controllers are listed.
- 5. Highlight the desired controller and confirm with Add Route.
 - ⇒ The Add Remote Route window opens.
- 6. Remote User Credentials:
 Log in to the controller with your login credentials or Beckhoff's default login credentials:
 User name: Administrator
 Password: 1
- 7. Confirm with OK.
 - ⇒ The routing to the controller is added.
 - ⇒ The controller is available as a target system for Config mode.
- 8. Close the Add Route Dialog and Choose Target System windows.

Activating Config mode

If necessary, select the controller as the target system and activate Config mode so you can configure the EtherCAT system online.

- ✓ The EtherCAT MainDevice is connected to the network, all system components are energized and the infrastructure is ready for operation.
- ✓ If necessary, you have added a routing rule to the controller.
- 1. If the Run-Time and TwinCAT 3 Engineering toolbar are installed on different PCs, click the <Local> list field in the TwinCAT 3 Engineering toolbar and select the desired controller.
 - ⇒ The controller is saved as a target system.
- 2. To activate Config mode, select the TWINCAT > Restart TwinCAT (Config Mode) menu.
 - ⇒ The Restart TwinCAT System in Config Mode dialog box opens.
- 3. Confirm with OK.
 - ⇒ TwinCAT 3 Engineering is in Config mode for the selected controller.

5.2.3 Scanning the hardware environment

If all system components are connected to the EtherCAT network and the network is energized, it is possible to scan for connected system devices automatically. In this scenario, TwinCAT 3 Engineering searches for connected devices and terminals and integrates them into the existing project in accordance with their configuration entries in the accompanying ESI files.

If the actual EtherCAT infrastructure is not available, i.e. you are configuring in offline mode, you must map and project all connected devices manually in TwinCAT 3 Engineering. You can find more detailed information on this in the online help tool of the TwinCAT 3 Engineering software.

✓ You have activated Config mode.

1. In the solution explorer, navigate to I/O > Devices > Scan context menu.
2. Confirm the HINT: Not all types of devices can be found automatically dialog box with OK.
 - ⇒ TwinCAT 3 Engineering scans the EtherCAT system for the EtherCAT MainDevice.
 - ⇒ The ... new I/O devices found dialog box opens.
3. Activate the relevant EtherCAT MainDevice and confirm with OK.
 - ⇒ The EtherCAT MainDevice is created in the solution explorer under I/O > Devices as a device (EtherCAT).
 - ⇒ The Scan for boxes? dialog box opens.
4. Confirm with Yes.
 - ⇒ TwinCAT 3 Engineering scans the EtherCAT system for the EtherCAT SubDevices.
 - ⇒ The EtherCAT drive(s) added dialog box opens.
5. Append linked axis to:

If you are using controller-based operation for the drive controller, select the desired option and confirm with OK in order to activate the NC or CNC function. For drive-based control, you prevent the creation of an axis using Cancel.

 - ⇒ The EtherCAT SubDevices are created in the solution explorer.
 - The Activate Free Run dialog box opens.
6. In order to shift the system components during configuration into free run mode and thereby enable verification of the signal exchange, confirm with Yes.
 - ⇒ EtherCAT MainDevice and SubDevices are created in TwinCAT 3 Engineering.

5.2.4 Configuring synchronization using distributed clocks

As the more precise of the two sync methods, synchronization using distributed clocks (DC-Sync) is pre-configured in the EtherCAT SubDevices. Check the associated settings for EtherCAT MainDevice and SubDevices.

✓ You have fully configured the associated axis model in the DriveControlSuite.

1. Navigate to the EtherCAT MainDevice in the solution explorer.
2. Double-click on the EtherCAT MainDevice.
⇒ The settings open in the main window.
3. In the main window, switch to the EtherCAT tab and click Advanced Settings....
⇒ The Advanced Settings window opens.
4. In the tree view, select Distributed Clocks.
5. Automatic DC Mode Selection:
This option must be activated.
6. Close the window.
7. Navigate to the first EtherCAT SubDevice in the solution explorer.
8. In the main window, switch to the DC tab and click Advanced Settings....
⇒ The Advanced Settings window opens.
9. Enable:
This option must be activated.
10. DC enabled (multiplier = 1):
This list entry must be selected.
11. Sync Unit Cycle (μ s):
Check the default value for the cycle time of the controller and change it if necessary.
12. Enable SYNC 0:
This option must be activated.
13. Close the window.
14. Repeat steps 7 – 13 for each additional SubDevice in your EtherCAT network.
⇒ The EtherCAT MainDevice and SubDevices will now be synchronized with the first EtherCAT SubDevice that has the distributed clocks option enabled.

5.2.5 Controller-based axis control

In order to control one or more drive controllers with controller-based operation when using the CiA 402 application, start by parameterizing the axes and then program your control.

5.2.5.1 Parameterizing an axis

1. In the solution explorer, navigate to Motion > NC-Task 1 SAF > Axes > Axis 1.
2. Double-click the axis.
⇒ The settings open in the main window.
3. In the main window, switch to the Settings tab.
4. Unit:
Select the unit based on the actual motion of your mechanical system (e.g. degrees (°) for a rotational axis).
5. Switch to the Parameters tab.
6. Open the Maximum Dynamics parameter list.
7. Parameterize associated limit values for velocity, acceleration and deceleration.
8. Open the Limit Switches parameter list.
9. Soft Position Limit Minimum Monitoring:
If you want to put a lower negative limit on position values, select the True list entry and enter the associated value in Minimum Position.
10. Soft Position Limit Maximum Monitoring:
If you want to put an upper positive limit on position values, select the True list entry and enter the associated value in Maximum Position.
11. In the solution explorer, navigate to Axis > Enc.
12. In the main window, switch to the Parameters tab.
13. Open the Encoder Evaluation parameter list.
14. Scaling Factor Numerator:
Specify the value 0.000343322 ($360 \div 1048576$) – in accordance with the parameterization of A585[0] = 1048576 inc for the feed factor in DriveControlSuite.
15. Repeat the steps for each additional axis.
⇒ The axes are parameterized.

5.2.5.2 Programming axis control

You can program control of the axes in TwinCAT 3 using the MC_Power block.

In order to control the drive controller with controller-based operation, the following operating modes are available to you in the A541 Modes of operation parameter:

- 6: Homing mode
- 8: Cyclic synchronous position mode
- 9: Cyclic synchronous velocity mode
- 10: Cyclic synchronous torque mode

For more information on the operating modes, see [Operating modes in detail](#) [► 118].

For more information on device control, see [CiA 402 device control](#) [► 111].

In addition, the set values and actual values are available in the standard mapping; see [EtherCAT and CiA 402 standard mapping](#) [► 148].

Gravity-loaded axis with brake



Risk of fatal injury due to gravity-loaded axis!

On a gravity-loaded axis with brake, the load may drop until the brake is fully engaged. This can result in serious injury or death.

- Always shut down the drive using a quick stop (state transition 11 in the device state machine).

In the MC_Power block, first deactivate the bDriveStart bit and then, after a delay, deactivate the bRegulatorOn bit to bring the drive to a controlled standstill.

5.2.6 Transmitting the configuration

Transfer the configuration to the EtherCAT MainDevice.

1. Select the menu TWINCAT > Activate Configuration.
2. Confirm the transfer of the project configuration to the EtherCAT MainDevice with OK.
 - ⇒ The Restart TwinCAT System in Run Mode dialog box opens.
3. Confirm with OK.
 - ⇒ The configuration was transferred to the EtherCAT MainDevice.

5.2.7 Checking the functionality of the axes

If you use controller-based operation for the drive controller, check the functionality of the axes before operation in production.

Information

Ensure that a suitable safety application that ensures safe shut-off of the axis (emergency off, safety switch, etc.) exists before the start of testing.

Information

In order to be able to check the function of the axes, the **A541 Modes of operation** parameter of the respective axis must be set to the value **8** (default value).

1. In the solution explorer, navigate to Motion > NC-Task 1 SAF > Axes > Axis 1.
2. Double-click the axis.
⇒ The settings open in the main window.
3. In the main window, switch to the Online tab.
4. In the Enabling area, click Set.
⇒ The Set Enabling window opens.
5. Enable the Controller, Feed Fw and Feed Bw options.
6. Override:
Specify a value for the override (e.g. 100).
7. Confirm with OK.
⇒ The axis is monitored via the active control panel.
8. F1 – F4:
Move the axis step-by-step and test the movement direction, velocity, etc. using the corresponding buttons.
9. To deactivate the enable signal, click on Set Enabling and deactivate the options Controller, Feed Fw, Feed Bw.
10. Repeat the steps for each additional axis of your system.

5.2.8 More information about TwinCAT 3?

The following chapters summarize additional information, functions, and procedures in the TwinCAT 3 automation software that go beyond the commissioning process described in the CiA 402 application example.

5.2.8.1 Drive-based axis control

Drive-based axis control in the CiA 402 application requires manual programming in the automation software. The following operating modes are available to you in the parameter A541 Modes of operation:

- -1: Jog
- 1: Profile position mode
- 2: Velocity mode
- 3: Profile velocity mode
- 4: Profile torque mode
- 6: Homing mode

For more information on the operating modes, see [Operating modes in detail \[► 118\]](#).

The axes are controlled using control word A515. The device state machine has to receive certain commands for starting operation and the associated state transitions. These commands are the result of a bit combination in the control word. The order of the commands is specified by the device state machine in accordance with CiA 402.

For more information on device control, see [CiA 402 device control \[► 111\]](#).

In addition, the set values and actual values are available in the standard mapping; see [EtherCAT and CiA 402 standard mapping \[► 148\]](#).

Gravity-loaded axis with brake



Risk of fatal injury due to gravity-loaded axis!

On a gravity-loaded axis with brake, the load may drop until the brake is fully engaged. This can result in serious injury or death.

- Always shut down the drive using a quick stop (state transition 11 in the device state machine).

In A515, first deactivate bit 2, and then, after a delay, deactivate bits 1 and 0 to bring the drive to a controlled standstill.

5.2.8.2 Expanding the startup list

With the help of the startup list, you can change the values of objects during startup of the EtherCAT state machine via the CoE protocol. The objects are accessed in the order in which they appear in the startup list.

Information

Start parameters are transferred via SDO communication. The scaling of the SDO communication is configured via A213[1] (default value: 0: Integer without point). In contrast, the ESI file is based on the scaling of the PDO communication, which is configured via A213[0] (default value: 1: Native).

If objects from the ESI file are added to the startup list without first setting A213[1] to the value of A213[0], start parameters in TwinCAT 3 are defined with a different scaling from the one that is subsequently transferred.

We recommend setting A213[1] = A213[0] = 1: Native in the drive controller.

Adding an object from the ESI file

Objects contained in the object directory of the ESI file are displayed in TwinCAT 3 Engineering in list form for selection.

✓ EtherCAT MainDevice and SubDevices are created in TwinCAT 3 Engineering.

1. In the solution explorer, navigate to the EtherCAT SubDevice for which you want to add an object to the startup list.
2. Double-click on the EtherCAT SubDevice.
 - ⇒ The settings open in the main window.
3. In the main window, switch to the **Startup** tab.
 - ⇒ The current startup list is displayed.
4. Click **New....**
 - ⇒ The **Edit CANopen Startup Entry** window opens.
 - ⇒ All available objects are listed.
5. In the list, double-click on the object that you want to add to the startup list.
 - ⇒ The **Set Value Dialog** window opens.
6. **Dec, Hex, Float:**
Enter the value to be written to the object in the field corresponding to the desired data format (decimal specification, hexadecimal specification or selection from list).
7. Confirm the value with **OK**.
8. **Transition:**
In the **Edit CANopen Startup Entry** window, activate the state change upon which the object is to be written:
 - 8.1. I -> P: State change from Init to Pre-Operational
 - 8.2. P -> S: State change from Pre-Operational to Safe-Operational
 - 8.3. S -> O: State change from Safe-Operational to Operational
 - 8.4. O -> S: State change from Operational to Safe-Operational
 - 8.5. S -> P: State change from Safe-Operational to Pre-Operational
9. **Comment:**
If required, store a comment that will be displayed in the startup list for the object.
10. Confirm with **OK**.
 - ⇒ The object is added to the startup list.

Adding an object via CoE Online and SDO Info

The advantage of this approach is that it does not use the static ESI file, but instead reads the current parameter settings directly from the drive controller, including the set scaling.

For more information on activating SDO Info, see chapter "SDO: Service Data Objects" in the corresponding EtherCAT manual.

✓ EtherCAT MainDevice and SubDevices are created in TwinCAT 3 Engineering.

1. In the solution explorer, navigate to the EtherCAT SubDevice for which you want to add an object to the startup list.
2. Double-click on the EtherCAT SubDevice.
 - ⇒ The settings open in the main window.
3. In the main window, switch to the CoE Online tab.
 - ⇒ The current startup list is displayed.
4. Make sure that the indicated status is Online Data (green), not Offline Data (red).
5. Select the desired item from the list and click Add to Startup....
 - ⇒ The Edit CANopen Startup Entry window opens with the selected object.
6. Dec, Hex, Float:

Enter the value to be written to the object in the field corresponding to the desired data format (decimal specification, hexadecimal specification or selection from list).
7. Confirm the value with OK.
8. Transition:

In the Edit CANopen Startup Entry window, activate the state change upon which the object is to be written:

 - 8.1. I -> P: State change from Init to Pre-Operational
 - 8.2. P -> S: State change from Pre-Operational to Safe-Operational
 - 8.3. S -> O: State change from Safe-Operational to Operational
 - 8.4. O -> S: State change from Operational to Safe-Operational
 - 8.5. S -> P: State change from Safe-Operational to Pre-Operational
9. Comment:

If required, store a comment that will be displayed in the startup list for the object.
10. Confirm with OK.
 - ⇒ The object is added to the startup list.

Adding an object via index and subindex

Alternatively, you can add objects to the startup list by their index and subindex. Use this procedure for parameters that are not included in the ESI file's object directory.

For manufacturer-specific parameters, calculate the index and subindex of the object from the parameter coordinate in advance (see [Manufacturer-specific parameters: 2000 hex – 53FF hex \[► 161\]](#) for axis A and [Manufacturer-specific parameters: A000 hex – D3FF hex \[► 163\]](#) for axis B).

✓ EtherCAT MainDevice and SubDevices are created in TwinCAT 3 Engineering.

1. In the solution explorer, navigate to the EtherCAT SubDevice for which you want to add an object to the startup list.
2. Double-click on the EtherCAT SubDevice.
 - ⇒ The settings open in the main window.
3. In the main window, switch to the **Startup** tab.
 - ⇒ The current startup list is displayed.
4. Click **New...**
 - ⇒ The **Edit CANopen Startup Entry** window opens.
5. **Transition:**
Activate the state change upon which the object is to be written:
 - 5.1. I -> P: State change from Init to Pre-Operational
 - 5.2. P -> S: State change from Pre-Operational to Safe-Operational
 - 5.3. S -> O: State change from Safe-Operational to Operational
 - 5.4. O -> S: State change from Operational to Safe-Operational
 - 5.5. S -> P: State change from Safe-Operational to Pre-Operational
6. **Index (hex):**
Enter the index of the object (hexadecimal specification).
7. **Complete Access:**
 - ⇒ Activate the option to be able to enter a subindex.
8. **Sub-index (dec):**
Enter the subindex of the object (decimal specification).
9. **Data (hexbin):**
Enter the value to be written to the object.
10. **Comment:**
If required, store a comment that will be displayed in the startup list for the object.
11. **Confirm with OK.**
 - ⇒ The object is added to the startup list.

5.2.8.3 Configuring synchronization via SyncManager event

Synchronization using distributed clocks (DC-Sync) is pre-configured in the EtherCAT SubDevices. You have the option of switching the synchronization for one or more EtherCAT SubDevices manually to synchronization via SyncManager event (SM-Sync).

1. In the solution explorer, navigate to the EtherCAT SubDevice for which you want to change the synchronization to SM-Sync.
2. Double-click on the EtherCAT SubDevice.
⇒ The settings open in the main window.
3. In the main window, switch to the DC tab.
4. Operation Mode:
Select the *SM Synchronous* list entry from the picklist.
⇒ You have changed the synchronization for the EtherCAT SubDevice.
5. Repeat the steps for each additional SubDevice of your EtherCAT network for which you want to switch the synchronization to SM-Sync.

5.2.8.4 Configuring EoE communication

1. Navigate to the EtherCAT MainDevice in the solution explorer.
2. Double-click on the EtherCAT MainDevice.
⇒ The settings open in the main window.
3. In the main window, switch to the EtherCAT tab and click Advanced Settings....
⇒ The Advanced Settings window opens.
4. Select EoE Support from the tree view on the left.
5. Virtual Ethernet Switch > Enable:
This option must be activated.
6. Close the window.
7. Navigate to the first EtherCAT SubDevice in the solution explorer.
8. In the main window, switch to the EtherCAT tab and click Advanced Settings....
⇒ The Advanced Settings window opens.
9. In the tree view on the left, navigate to Mailbox > EoE.
10. Virtual Ethernet Port:
This option must be activated.
11. IP Port:
Activate this option.
12. Define the type of address assignment:
 - 12.1. Activate the DHCP option if an IP address is to be assigned to the EtherCAT SubDevice automatically via DHCP.
 - 12.2. Activate the IP Address option in order to assign a fixed IP address to the EtherCAT SubDevice in accordance with the subnet of your EoE device group. When assigning a fixed IP address for EoE, note that the first and last host address in a subnet must not be used. If one of these addresses is configured in TwinCAT 3, it is not accepted by the drive controller.
13. Default gateway:
When assigning a fixed IP address, the IP address of the EtherCAT network interface of the EtherCAT MainDevice must be specified as the default gateway.
14. Close the window.
15. Repeat steps 7 – 14 for each additional SubDevice in your EtherCAT system.
⇒ EoE communication is enabled for EtherCAT MainDevice and SubDevices.

Information

Depending on your EoE network structure, you may have to set the routing on your EtherCAT MainDevice PC manually to connect the Ethernet and EtherCAT networks. For more information on the different use cases, refer to the manual for communication with EtherCAT.

Information

Address assignment via DHCP is possible via either a DHCP server or DriveControlSuite. This requires the DHCP server or DriveControlSuite to be installed directly on the controller PC. Furthermore, the IP address reference must be correctly defined in the drive controller (A166 = 2: DHCP + DS6, default value).

The correct assignment of the drive controllers in DriveControlSuite is ensured for TwinCAT 3 by the STOBES_BoxName function block.

For more information on the topology and function blocks, refer to the manual for communication with EtherCAT.

5.2.8.5 Configuring a Station Alias

You have the option of assigning an EtherCAT Station Alias to each EtherCAT SubDevice. This address is stored in the EEPROM of the respective drive controller. The drive controller can thus be connected to any open port within the network and identified using the Station Alias.

1. In the solution explorer, navigate to the EtherCAT SubDevice to which you want to assign a Station Alias.
2. Double-click on the EtherCAT SubDevice.
⇒ The settings open in the main window.
3. In the main window, switch to the EtherCAT tab and click *Advanced Settings....*
⇒ The *Advanced Settings* window opens.
4. In the tree view on the left of the *Advanced Settings* window, navigate to *ESC Access > E2PROM > Configured Station:*
New Value: Enter the value you want to write as the Station Alias into the EEPROM.
5. Click *Write to E2PROM* to write the value to EEPROM.
6. Confirm the *Function succeeded!* dialog box with *OK*.
7. Close the *Advanced Settings* window with *OK*.
8. Repeat the steps for each additional SubDevice in your EtherCAT system to which you want to assign a Station Alias.
⇒ The configuration of the Station Alias is complete.
⇒ The change of addresses will take effect upon the next start of TwinCAT 3.

Information

In DriveControlSuite, the Station Alias can be read out via parameter A254.

5.3 CODESYS Development System V3: Setting up the EtherCAT system

If you are using a CODESYS SoftMotion controller, you map the hardware environment of your EtherCAT network in the CODESYS Development System V3 automation software. Configure all EtherCAT nodes, parameterize the necessary connection settings, and configure the data exchange between the controller and the drive controller. Then transfer the configuration to the controller and check the cyclic communication.

Before commissioning your EtherCAT network using the automation software, you must have physically connected all system devices and projected the relevant drive controllers in DriveControlSuite. For commissioning, we assume in the example that you have projected the CiA 402 application and are using the CODESYS SoftMotion extension (version 4.21.0.0 or later) for controller-based operation.

Drive-based operation requires manual programming in the automation software. For more information, see [Drive-based axis control without SoftMotion](#) [► 61].

Information

Always perform the steps described below in the specified order!

Some parameters are interdependent and do not become accessible to you until you have first configured certain settings. Follow the steps in the specified sequence so that you can finish the parameterization completely.

5.3.1 Creating a standard project

1. Start the CODESYS Development System V3 automation software.
2. Select **Basic Operations > New Project**.
⇒ The **New Project** window opens.
3. Select a standard project that corresponds to your hardware version. Give it a name and save it in a location of your choice.

5.3.2 Adding an EtherCAT MainDevice

- ✓ The EtherCAT MainDevice is connected to the network, all system components are energized and the infrastructure is ready for operation.

1. In the device tree, navigate to the module **Device** > Context menu **Add Device...**
⇒ The **Add Device** window opens.
2. For controller-based operation with SoftMotion, select **EtherCAT_Master_SoftMotion**.
3. Confirm with **Add Device**.
⇒ The EtherCAT MainDevice has been created in the device tree.

5.3.2.1 Assigning control communication

1. In the device tree, navigate to the **Device** module and double-click to open it.
⇒ **Device** tab > **Communication Settings** opens.
2. Open the **Scan Network** tab.
3. Select the network path to the controller.
4. Confirm with **OK**.
⇒ The control device is assigned to the project.

5.3.2.2 EtherCAT MainDevice: Assigning a network interface

1. In the device tree, navigate to the module EtherCAT_Master_SoftMotion (EtherCAT Master SoftMotion) and double-click to open it.
 - ⇒ The EtherCAT_Master_SoftMotion > General tab opens.
2. Specify the source address (MAC) of the network interface that is to be used as the EtherCAT interface. Alternatively, you can read out the device's available interfaces directly and copy the MAC address, provided you have the necessary login credentials.
 - 2.1. Click Select....
 - 2.2. Enter your login credentials.
 - 2.3. Select the desired interface.

5.3.3 Adding a SoftMotion axis

If all system components are connected to the EtherCAT network and the network is energized, it is possible to search for connected system devices automatically. In this scenario, CODESYS Development System V3 searches for connected devices and terminals and integrates them into the existing project.

If the actual EtherCAT infrastructure is not available, i.e. you are configuring in offline mode, you must map and project all connected system devices manually in CODESYS Development System V3.

Scanning the hardware environment and automatically adding an axis

- ✓ You have selected the CiA 402 application.
 - ✓ The desired drive controller is connected to the network and linked to it.
1. In the device tree, navigate to the module EtherCAT_Master_SoftMotion (EtherCAT Master SoftMotion) > Context menu Search for Device....
 - ⇒ The Search for Device window opens.
 - ⇒ Available STOBER drive controllers are automatically detected and displayed with one or two mappings and SoftMotion axes, depending on the number of axes.
 2. Click Copy All Devices to Project.
- ⇒ The STOBER drive controllers are stored in the project and displayed in the device tree under EtherCAT_Master_SoftMotion (EtherCAT Master SoftMotion).

Adding an axis manually

- ✓ You have selected the CiA 402 application.
1. In the device tree, navigate to the module EtherCAT_Master_SoftMotion (EtherCAT Master SoftMotion) > Context menu Add Device....
 - ⇒ The Add Device window opens.
 2. Fieldbuses > EtherCAT > Slave section:
Select STOEBER ANTRIEBSTECHNIK GmbH & Co. KG as the manufacturer and open the folder of the same name.
 3. Open the Drives folder contained within it.
 - ⇒ All drive controllers that can be mapped are displayed.
 4. Highlight the desired drive controller and confirm with Add Device.
 - 4.1. If you are using SoftMotion, select the drive controller with SoftMotion in its designation.
 - 4.2. If you are using a safety module with FSoE (e.g. SX6), select the Y version of the drive controller.
 5. Repeat step 4 for all other drive controllers in your EtherCAT system.

5.3.4 Configuring synchronization using distributed clocks

- ✓ As the more precise of the two sync methods, synchronization using distributed clocks (DC-Sync) is pre-configured in the EtherCAT SubDevices.

In order to reduce jitter in general, we recommend setting data transfer (I/O) of the controller to task start in the EtherCAT configuration.

1. In the device tree, navigate to the EtherCAT_MainDevice (EtherCAT MainDevice) module and double-click to open it.
⇒ EtherCAT_MainDevice tab > General opens in the editor window.
 2. Distributed Clock area > Cycle Time and Sync Offset:
Check the default values and change them if necessary.
 3. To set the data transfer to task start, select Tools > Options > Device editor.
 4. Activate the Show generic configuration editors option and confirm with OK.
 5. Switch to the EtherCAT parameters vertical tab.
 6. Navigate to the FrameAtTaskStart parameter and set the Value of the parameter to True.
⇒ From now on, controller data transfer will take place at the start of the task.
 7. In the device tree, double-click the first of the added drive controllers.
⇒ SI6_SC6_Single(Double)Ax_SoftMotion_HiRes tab > General opens in the editor window.
 8. Distributed Clocks area > Select DC:
DC enabled (multiplier = 1) and Sync 0 as a sync event are enabled by default.
 9. If you want to change the presets, enable the Additional > Enable Expert Settings option and change the settings.
 10. Repeat steps 7 – 9 for each additional drive controller in your EtherCAT network.
- ⇒ The EtherCAT MainDevice and SubDevices will now be synchronized with the first EtherCAT SubDevice that has the distributed clocks option enabled.

5.3.5 Controller-based axis control

In order to control one or more drive controllers with controller-based operation when using the CiA 402 application, start by parameterizing the axes and then program your control.

5.3.5.1 Parameterizing an axis

When operating in controller-based mode with SoftMotion, the mechanical system must be scaled in CODESYS Development System V3.

In the default settings, both the drive controller and the CodeSys driver are configured with a scaling of 1,048,576 increments ≤ 1 motor revolution (equivalent to 20 bits). STÖBER recommends this configuration because it is suitable for common motion profiles and available EnDat encoders (20-bit ST, 12-bit MT).

1. In the device tree, navigate to the module for the respective SoftMotion axis and double-click it to open it.
⇒ The General tab opens.
2. Open the Scaling/Mapping tab.
3. Parameterize motor revolutions $\leq$ gearbox output revolutions and gearbox output revolutions $\leq$ units in the application according to your mechanical system.
4. Repeat the steps for axis B (only for double-axis controllers).

5.3.5.2 Programming axis control

You program the control of the axes in the automation software. See the CODESYS Development System V3 documentation for the required information.

In order to control the drive controller with controller-based operation, the following operating modes are available to you in the A541 Modes of operation parameter:

- 6: Homing mode
- 7: Interpolated position mode or
- 8: Cyclic synchronous position mode
- 9: Cyclic synchronous velocity mode
- 10: Cyclic synchronous torque mode

For more information on the operating modes, see [Operating modes in detail](#) [► 118].

For more information on device control, see [CiA 402 device control](#) [► 111].

In addition, the set values and actual values are available in the standard mapping; see [EtherCAT and CiA 402 standard mapping](#) [► 148].

Gravity-loaded axis with brake

**DANGER!**

Risk of fatal injury due to gravity-loaded axis!

On a gravity-loaded axis with brake, the load may drop until the brake is fully engaged. This can result in serious injury or death.

- Always shut down the drive using a quick stop (state transition 11 in the device state machine).

In the MC_Power block, start by deactivating the `Enable_Positive` and `Enable_Negative` bits and then, after a delay, the `Enable` bit in order to bring the drive to a controlled standstill.

5.3.6 Transmitting the configuration

Transfer the project to your CODESYS SoftMotion controller and start CODESYS Development System V3.

5.3.7 Checking the functionality of the axes

If you use controller-based operation for the drive controller, check the functionality of the axes before operation in production.

Information

Ensure that a suitable safety application that ensures safe shut-off of the axis (emergency off, safety switch, etc.) exists before the start of testing.

Information

If you want to perform referencing to test the functionality, a reference method must be parameterized for parameter A586.

For an initial functional test, the online configuration mode included in the base system is available:

- ✓ You have commissioned the EtherCAT network.
 - ✓ You have verified the quality of communication using the DC-Sync Optimization wizard; see [Synchronizing EtherCAT nodes](#) [► 34].
1. In the device tree, navigate to the **Device** module.
 2. Click the icon for **online configuration mode** on the toolbar.
 - ⇒ Online configuration mode is activated.
 3. In the device tree, navigate to the module for the respective SoftMotion axis and double-click it to open it.
 - ⇒ The **General** tab opens.
 4. Open the **Commissioning** tab.
 5. Check the available functions of the axis.
 - 5.1. **Switch on:**
Supply power to the drive.
 - 5.2. **Error reset:**
Reset the drive after an error
 - 5.3. **Optional: Start referencing:**
Perform referencing of the drive with the parameters set in the drive.
 - 5.4. **Jog mode:**
Enter values for distance, velocity, acceleration, deceleration and jerk, and to move the drive forward or backward with < and >.

Further testing requires programming the controller. For example, you can create a test project based on the SMC_StartupDrive function block or work directly with PLCopen function blocks.

5.3.8 More information about CODESYS Development System V3?

The following chapters summarize additional information, functions, and procedures in the CODESYS Development System V3 that go beyond the commissioning process described in the example with the CiA 402 application and the CODESYS SoftMotion extension.

5.3.8.1 Drive-based axis control without SoftMotion

If you are using CODESYS Development System V3 without SoftMotion and with drive-based axis control, you must perform additional steps, or different steps, when commissioning the EtherCAT system. These are summarized in the following chapters.

5.3.8.1.1 Importing an ESI file

- ✓ You have created and saved the ESI file in DS6.
- ✓ You have started CODESYS Development System V3.
- 1. In the menu bar, click **Tools > Device Repository....**
 - ⇒ The Device Repository window opens.
- 2. Click **Install....**
 - ⇒ The search window in Windows Explorer opens.
- 3. Select the location where the ESI file is stored.
- 4. Select the corresponding ESI file and install it.
- ⇒ The ESI file has been imported into CODESYS Development System V3.

5.3.8.1.2 Adding an EtherCAT MainDevice without SoftMotion

- ✓ The EtherCAT MainDevice is connected to the network, all system components are energized and the infrastructure is ready for operation.
- ✓ You have created an ESI file for the drive controller and imported it into CODESYS Development System V3.
- ✓ You have created a project in CODESYS Development System V3; see [Creating a standard project \[► 55\]](#).
- 1. In the device tree, navigate to the module **Device >** Context menu **Add Device....**
 - ⇒ The Add Device window opens.
- 2. For drive-based operation without SoftMotion, select **EtherCAT Master (EtherCAT Master)**.
- 3. Confirm with **Add Device**.
- ⇒ The EtherCAT MainDevice has been created in the device tree.

5.3.8.1.2.1 Assigning control communication

1. In the device tree, navigate to the **Device** module and double-click to open it.
 - ⇒ **Device tab > Communication Settings** opens.
2. Open the **Scan Network** tab.
3. Select the network path to the controller.
4. Confirm with **OK**.
- ⇒ The control device is assigned to the project.

5.3.8.1.2.2 EtherCAT MainDevice: Assigning a network interface without SoftMotion

1. In the device tree, navigate to the module EtherCAT_Master (EtherCAT Master) and double-click to open it.
⇒ EtherCAT_Master tab > General opens.
2. Specify the source address (MAC) of the network interface that is to be used as the EtherCAT interface.
Alternatively, you can read out the device's available interfaces directly and copy the MAC address, provided you have the necessary login credentials.
 - 2.1. Click Select....
 - 2.2. Enter your login credentials.
 - 2.3. Select the desired interface.

5.3.8.1.3 Adding an axis without SoftMotion

If all system components are connected to the EtherCAT network and the network is energized, it is possible to search for connected system devices automatically. In this scenario, CODESYS Development System V3 searches for connected devices and terminals and integrates them into the existing project.

If the actual EtherCAT infrastructure is not available, i.e. you are configuring in offline mode, you must map and project all connected system devices manually in CODESYS Development System V3.

Scanning the hardware environment and automatically adding an axis

- ✓ You have selected the CiA 402 application.
 - ✓ The desired drive controller is connected to the network and linked to it.
1. In the device tree, navigate to the module EtherCAT_Master (EtherCAT Master) > Context menu Search for Device....
 - ⇒ The Search for Device window opens.
 - ⇒ Available STOBER drive controllers are automatically detected and displayed with one or two mappings and SoftMotion axes, depending on the number of axes.
 2. Click Copy All Devices to Project.
- ⇒ The STOBER drive controllers are stored in the project and displayed in the device tree under EtherCAT_Master (EtherCAT Master).

Adding an axis manually

When adding an axis manually, an additional ESI file is required. For information on importing the ESI file, see chapter [Importing an ESI file \[► 61\]](#).

- ✓ You have selected the CiA 402 application.
- ✓ You have imported the ESI file.
- 1. In the device tree, navigate to the module EtherCAT_Master (EtherCAT Master) > Context menu Add Device....
 - ⇒ The Add Device window opens.
- 2. Fieldbuses > EtherCAT > Slave section:
Select STOEBER ANTRIEBSTECHNIK GmbH & Co. KG as the manufacturer and open the folder of the same name.
- 3. Open the Drives folder contained within it.
 - ⇒ Drive controllers that you have imported using the ESI file are displayed.
- 4. Highlight the desired drive controller and confirm with Add Device.
 - 4.1. If you are using a safety module with FSoE (e.g. SX6), select the Y version of the drive controller.
- 5. Repeat step 4 for all other drive controllers in your EtherCAT system.

Information

The ESI file contains multiple mappings, known as modules. When a drive controller is added manually, the module with the STOBER CiA 402 mapping is always used. If you want to switch to a different mapping, follow the steps below.

- 1. In the device tree, navigate to the corresponding axis > context menu Update Device....
 - ⇒ The Update Device window opens.
- 2. Select the desired module for this axis.
- 3. Confirm with Update Device.

5.3.8.1.4 Drive-based axis control

Drive-based axis control in the CiA 402 application requires manual programming in the automation software. The following operating modes are available to you in the parameter A541 Modes of operation:

- -1: Jog
- 1: Profile position mode
- 2: Velocity mode
- 3: Profile velocity mode
- 4: Profile torque mode
- 6: Homing mode

For more information on the operating modes, see [Operating modes in detail](#) [► 118].

The axes are controlled using control word A515. The device state machine has to receive certain commands for starting operation and the associated state transitions. These commands are the result of a bit combination in the control word. The order of the commands is specified by the device state machine in accordance with CiA 402.

For more information on device control, see [CiA 402 device control](#) [► 111].

In addition, the set values and actual values are available in the standard mapping; see [EtherCAT and CiA 402 standard mapping](#) [► 148].

Gravity-loaded axis with brake



DANGER!

Risk of fatal injury due to gravity-loaded axis!

On a gravity-loaded axis with brake, the load may drop until the brake is fully engaged. This can result in serious injury or death.

- Always shut down the drive using a quick stop (state transition 11 in the device state machine).

In A515, first deactivate bit 2, and then, after a delay, deactivate bits 1 and 0 to bring the drive to a controlled standstill.

5.3.8.2 CODESYS SoftMotion: Additional information

The following chapters summarize further information and additional features for using SoftMotion.

5.3.8.2.1 Outdated device driver

The CODESYS device driver SM_Drive_ETC_STOEBER_SI6_SC6 is outdated and has been replaced by the new device driver SM3_Drive_ETC_STOEBER_SB6_SC6_SI6. If you are using the old device driver, we recommend that you switch to the new one. To commission the SB6 drive controller and the SX6 safety module with CODESYS Development System V3, you must install the new device driver.

The new device driver, SM3_Drive_ETC_STOEBER_SB6_SC6_SI6, introduces the following changes:

- The SX6 safety module, in combination with SC6 and SI6, is supported.
- The SB6 drive controller is supported, even in combination with SX6.
- The CiA 402 HiRes Motion application is no longer supported. Instead, use the CiA 402 application (incremental version).
- EtherCAT Rx SDO Info device control is required for the new device driver; EtherCAT Rx was required for the previous device driver.

5.3.8.2.2 Special case: Adding to the PDO transmission

- ✓ Are you working with a controller-based operating mode (SoftMotion) and need expanded PDO transmission? Proceed as described in the following steps. Note that you can transmit a maximum of 24 CiA objects or drive controller parameters per channel.
 - 1. In the device tree, navigate to the drive controller whose PDO transmission you would like to expand and double-click to open it.
 - ⇒ SI6_SC6_Single(Double)Ax_SoftMotion_HiRes tab > General opens in the editor window.
 - 2. Additional area > Enable Expert Settings:
Activate this option.
 - 3. Switch to the Expert Process Data Mode vertical tab.
 - 4. PDO list:
The list contains one transmit and one receive channel for each parameterized SoftMotion axis.
Highlight the channel whose PDO transmission you would like to expand.
 - ⇒ PDO Content: This area shows all PDOs that are exchanged between the controller and drive controller over the selected channel.
 - 5. Click on Insert.
 - ⇒ The Select Item from Object Directory dialog box opens. The directory contains a selection of available CiA objects (along with the coordinates and the name of the corresponding drive controller parameter from STOBER).
 - 6. Highlight the CiA object for which you would like to extend PDO transmission and confirm with OK.
If the desired CiA object is not included in the directory, enter its index and subindex in the corresponding fields. For manufacturer-specific parameters, calculate the two indices from the parameter coordinate in advance (see [Manufacturer-specific parameters: 2000 hex – 53FF hex \[► 161\]](#) and [Manufacturer-specific parameters: A000 hex – D3FF hex \[► 163\]](#)). Also select the data type that matches the data type of the drive controller parameter and confirm with OK.
 - ⇒ The selected CiA object or the specified drive controller parameter has been added to the PDO content of the selected channel.
 - 7. Repeat steps 5 – 6 for all other CiA objects for which you would like to extend PDO transmission for the selected channel.
 - 8. Switch to the associated DriveControlSuite project and add to the PDO transmission there in the same way as the additions in CODESYS V3 (see Configuring PDO transmission).
- ⇒ The expansion of PDO transmission takes effect the next time the EtherCAT MainDevice is started.

5.3.8.2.3 SMC3_BrakeControl

The CODESYS device driver supports the `SMC3_BrakeControl` function block, which the controller can use to control the motor holding brake. This allows the holding brake to be opened or closed as needed via the controller.

The block interacts with parameter `F108` and control word `A515`.

Behavior in the drive controller

Enabling the axis is not possible when the holding brake is released or the release override is active (switch-on lockout `E47/E49`). If `SMC3_BrakeSetState` is set to `SMC_BRAKE_AUTO`, the bit for the `Release Brake` function is set to inactive, provided it was previously set via `SMC_BRAKE_OPEN`.

Brake status in CODESYS Development System V3

The `BrakeClosed` block output reflects the internal CODESYS control status, not the mechanical state of the holding brake. You can determine the actual braking state from bits 3 and 4 of status word `E201`, which are part of the CiA 402 standard mapping.

Information

We recommend using this feature only for functional testing or service purposes. During normal operation, the holding brake should be controlled by the drive controller.

5.3.8.2.4 MC_TouchProbe

The CODESYS device driver supports the `MC_TouchProbe` function block, which detects and stores the current axis position in the drive controller as soon as a configured trigger is activated.

To use the function block, the following prerequisites must be met:

CODESYS Development System V3

A value of type `RECORD` must be created at the `TriggerInput` input and configured as follows:

- `TriggerNumber`: 0 for Touch probe 1 or 2 for Touch probe 2
- `FastLatching`: `TRUE`

DriveControlSuite

Further parameterization of the Touch probe function is performed using the corresponding `Touch probe wizard`.

5.3.8.2.5 EtherCAT encoder

Via the drive controller's context menu, an encoder connected to the drive controller can be added to the device tree as a freely configurable EtherCAT encoder and used freely in CODESYS Development System V3, for example, as a master value for motion control.

Settings in DriveControlSuite

The encoder type (SMC_Encoder_ETC_STOEBER_SB6_SC6_SI6) is based on parameter A620, which is part of the PDO mapping. The drive controller provides the actual position G122 of its master encoder via A620. The parameterization of the master encoder is done using G104 in the *Encoder* and *Master Encoder: Scaling* wizards, as well as through the wizard for the specific hardware interface to which the encoder is connected.

Settings in CODESYS Development System V3

1. In the device tree, navigate to the drive controller module > context menu *Add Device...*
⇒ The *Add Device* window opens.
2. Select SMC_Encoder_ETC_STOEBER_SB6_SC6_SI6.
3. Confirm with *Add Devices*.
⇒ The encoder is added to the device tree.
4. In the device tree, navigate to the module SMC_Encoder_ETC_STOEBER_SB6_SC6_SI6 and double-click to open it.
⇒ The *Encoder* tab opens.
5. Make the appropriate settings for the encoder.

5.3.8.3 Expanding the startup list

You can use the startup list to set start parameters for an EtherCAT SubDevice. The specified values are written to the corresponding parameters when communication via SDO begins.

There are two ways to add objects: From the object directory of the ESI file, or by directly entering the index, subindex, and bit length. The ESI file contains only a portion of the object directory: Either the scope included in the SoftMotion device driver for SoftMotion axes, or the contents of an imported ESI file. For parameters not included in the ESI file, the index and subindex can be calculated.

Information

Start parameters are transferred via SDO communication. The scaling of the SDO communication is configured via A213[1] (default value: 0: Integer without point). This setting directly affects the correct transfer of start parameters.

We recommend setting A213[1] = 1: Native in the drive controller.

Adding an object from the ESI file

Objects included in the object directory of the ESI file are displayed in CODESYS Development System V3 in list form for selection.

- ✓ The EtherCAT MainDevice and SubDevices are created in CODESYS Development System V3.
- ✓ A213[1] is set to 1: Native.
- 1. In the device tree, navigate to the EtherCAT SubDevice for which you want to add an object to the startup list.
- 2. Double-click on the EtherCAT SubDevice.
 - ⇒ The EtherCAT SubDevice > General tab opens.
- 3. Go to the Start Parameters tab.
- 4. Click Add.
 - ⇒ The object directory containing selectable objects from the ESI file is displayed.
- 5. Select the item you want to add to the startup list.
- 6. Value:
 - Enter the value to be written to the object.
- 7. Confirm the value with OK.
 - ⇒ The object is added to the startup list.

Adding an object via index and subindex

Alternatively, you can add objects to the startup list by their index and subindex. Use this procedure for parameters that are not included in the ESI file's object directory.

For manufacturer-specific parameters, calculate the index and subindex of the object from the parameter coordinate in advance (see [Manufacturer-specific parameters: 2000 hex – 53FF hex \[▶ 161\]](#) for axis A and [Manufacturer-specific parameters: A000 hex – D3FF hex \[▶ 163\]](#) for axis B).

- ✓ The EtherCAT MainDevice and SubDevices are created in CODESYS Development System V3.
- ✓ A213[1] is set to 1: Native.
- ✓ You have calculated the index and subindex and determined the data type.
- 1. In the device tree, navigate to the EtherCAT SubDevice for which you want to add an object to the startup list.
- 2. Double-click on the EtherCAT SubDevice.
 - ⇒ The EtherCAT SubDevice > General tab opens.
- 3. Go to the Start Parameters tab.
- 4. Click Add.
 - ⇒ The object directory is displayed.
- 5. Index and SubIndex:
 - Enter the calculated value to be written to the object.
- 6. Select the appropriate data type.
 - ⇒ The bit length is adjusted automatically.
- 7. Confirm the value with OK.
 - ⇒ The object is added to the startup list.

5.3.8.4 Configuring EoE communication

STOBER 6th generation drive controllers support EoE communication. For information on whether your controller also supports EoE and how the packets are transferred from your controller to the service PC, refer to the documentation for your controller.

Information

Depending on your EoE network structure, you may have to set the routing on your EtherCAT MainDevice PC manually to connect the Ethernet and EtherCAT networks. For more information on the different use cases, refer to the manual for communication with EtherCAT.

6 More about CiA 402?

The following chapters summarize the important terms and relationships concerning CiA 402.

6.1 CiA 402 – Concept

Using the controller-based operating modes of CiA 402, you can implement applications with synchronized, cyclic set value specification by a motion controller. In addition, the drive controllers can also independently handle motion tasks such as referencing and jogging during commissioning.

The drive-based operating modes of CiA 402, on the other hand, offer complete movement calculation and design in the drive. In this setup, the set values for position, velocity and torque/force are converted into movements accurately and precisely. Referencing and jogging are performed, like all movements, with jerk limitation during commissioning. The drive controllers are either networked over a fieldbus or obtain signals and set values using analog and digital hardware inputs.

The following graphic shows the components and configuration steps for the CiA 402 application. The elements shown in a lighter shade are optional.

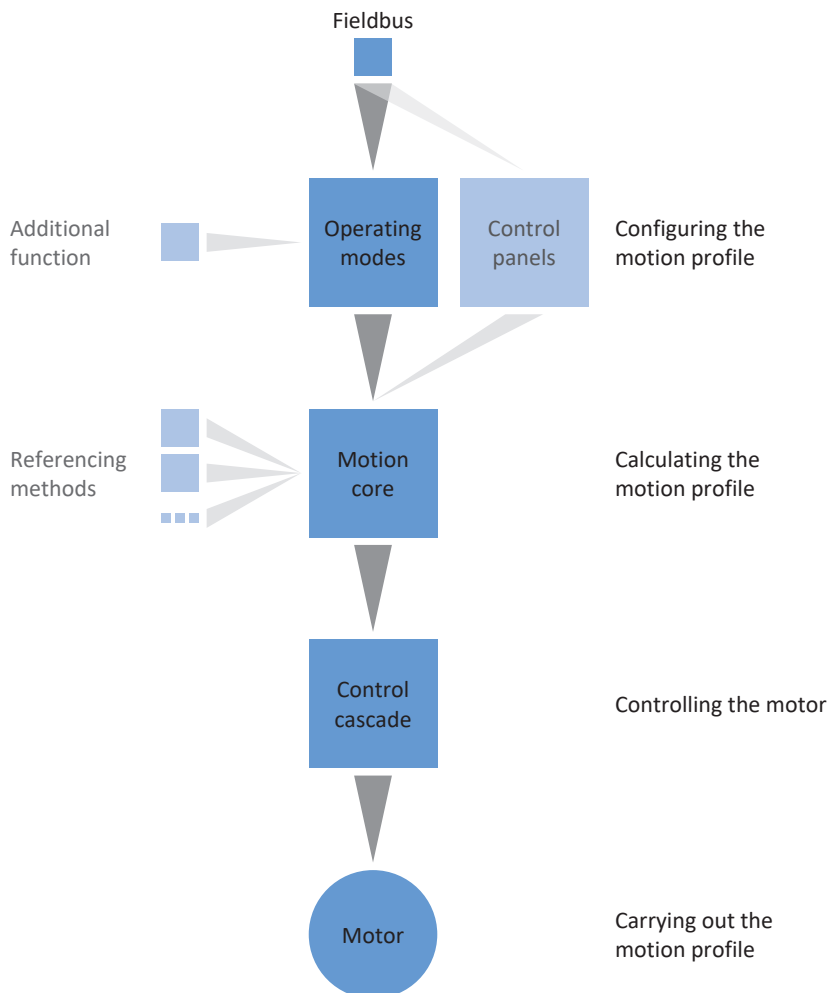


Fig. 5: Components and configuration steps

6.1.1 Activation

The application is based on the internationally standardized CiA 402 CANopen device profile for electric drives. Accordingly, the drive control is controlled using CiA 402-compliant control and status words.

6.1.2 Operating modes

The CiA 402 application provides 3 controller-based operating modes in production operations. In these modes, the controller handles the cyclical position, velocity or torque/force specification:

- **7: Interpolated position mode or 8: Cyclic synchronous position mode** for position-controlled drives
- **9: Cyclic synchronous velocity mode** for velocity-controlled drives
- **10: Cyclic synchronous torque mode** for torque/force-controlled drives

In addition, 4 drive-based operating modes are available in production operations. For these modes, the drive calculates the necessary motion profiles by itself:

- **1: Profile position mode** for target position specification by a controller
- **2: Velocity mode** for set velocity specification by a controller
- **3: Profile velocity mode** for set velocity specification by a controller
- **4: Profile torque mode** for set torque/force specification by a controller

Using the **6: Homing mode** operating mode, you can reference the drive independently of the controller.

For commissioning, emergency operation and maintenance or repair work, the STOBER-specific **-1: Jog** operating mode (manual movement) is available. Using this mode, you can move the drive independently of the controller.

For more information on the operating modes, see [Operating modes in detail](#) [► 118].

6.1.3 Data sources

The signals for controlling the drive controller, i.e. control signals for the start of motion, motion limits or set value specifications, can be obtained from various external sources.

A fieldbus is usually acts as the signal source, but digital hardware inputs, or even mixed operation of the above-listed sources, can also be used to control the drive controller.

Values that are obtained from external data sources are usually adapted to the stored reference values automatically, i.e. calibrated and calculated with scaling.

6.1.4 User-defined status word

In the user-defined status word, you have the option of defining the meaning of each individual bit yourself. You can subsequently define a total of 16 sources in A66 that are to be exchanged between drive controller and controller and about which you require information.

6.1.5 Additional functions

The additional Touch probe 1 and Touch probe 2 functions enable position measurement on a digital signal.

For more detailed information on how it works along with examples, see [Touch probe – Examples](#) [► 144].

6.1.6 Control panels

The control panels are special wizards in DriveControlSuite, which you can use to take control of the axis. With the help of the control panels, you can manually release and move an axis even if the drive controller has no operating unit or is difficult to access.

Using control panels, you can check the connection wiring, the project configuration of your physical axis model or the parameterization of your application before you start commissioning the controller or switch to normal operation.

The following control panels are available:

- Jog control panel is used to check the projected axis model in jog mode.
- Control panel motion provides you with a standard set of motion commands based on PLCopen. Via the control panel, independent of the application and fieldbus interface, you can parameterize a motion profile directly for the motion core of the axis in order to check the basic functions of the drive controller.

For a comparison of operating modes in accordance with CiA 402 and the PLCopen commands, see [Operating modes in accordance with CiA 402 – Commands](#) [► 137].

Because control panels override normal operation, they can be activated only with a deactivated enable signal and should be operated by experienced users only.

6.1.7 Motion core

Based on the projected and parameterized data, the motion core calculates a motion profile and associated detail movements as a basis for the drive controller as well as mandatory set values for the control cascade.

6.2 Axis model

With the help of the axis model, you map the real mechanical environment of your drive project in DriveControlSuite by parameterizing the axis type and the arrangement of existing encoders. The parameterization of the axis model is required for the smooth operation and easy diagnosis of your drive train.

With I05 Type of axis, you select a rotational or a translational axis model and define whether the scaling of the axis is done using either predefined or individually configured units of measure. With I00 Position range, you parameterize either an endless (modulo) or a limited travel range. You parameterize the encoder arrangement by using B26 Motor encoder and I02 Position encoder.

The 6th generation of STOBER drive controllers are specially developed for communication between the drive controller and controller on the basis of the actual variables at the output (° or mm of real axis movement). The scaling of the axis model is calculated by the drive controller's firmware independently of the encoder model, without any rounding errors or drift.

If your axis model is not followed by any other gear ratios, you can operate the axis with output-side motion variables where all nominal and actual values correspond to the real axis movement.

Information

The firmware processes values for the motion variables velocity, acceleration and jerk in the data type REAL32 (floating-point number, 32 bits). Position values are processed in the INT32 data type (integer, 32 bits) to eliminate rounding errors and enable precise movements.

Abbreviation	Meaning
LinM	Linear Motor
M	Motor
MEnc	Motor Encoder
PEnc	Position Encoder

Rotational axis models

The following figures each show a rotational axis model consisting of a motor, a gearbox and a rotary table (endless rotational) or a pointer (limited rotational). Rotational axis models support rotational motor encoders as well as rotational position encoders.

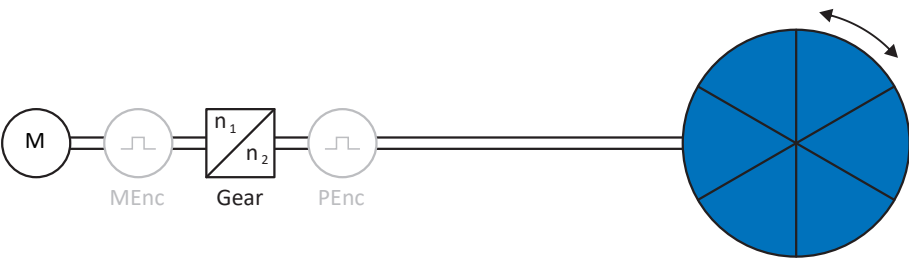


Fig. 6: Endless-rotational motion: Rotary table

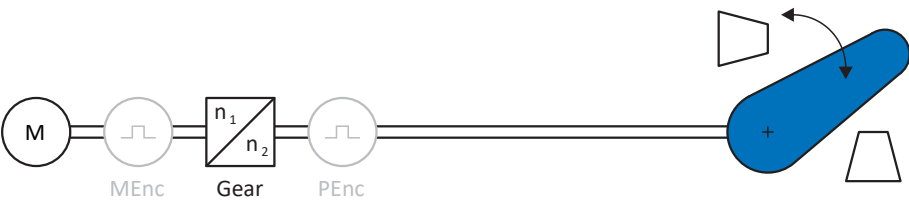


Fig. 7: Limited-rotational motion: Pointer

Translational axis models

The following figures each show a translational axis model consisting of a motor, gearbox, feed and conveyor belt (endless translational) or tool slide (limited translational). Translational axis models support rotational motor encoders as well as rotational or translational position encoders.

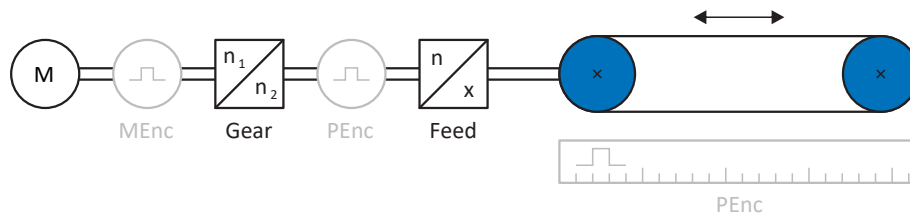


Fig. 8: Endless translational motion: Conveyor belt

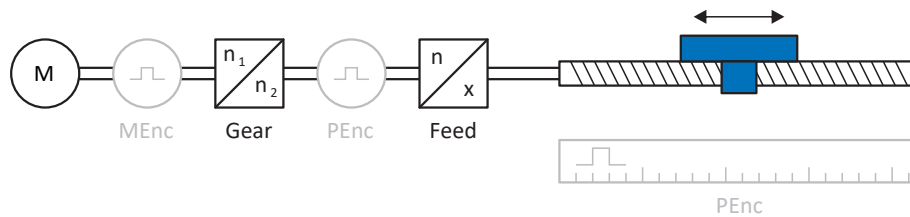


Fig. 9: Limited translational motion: Tool slide

Translational axis model: Linear motor

The following figure shows a limited translation axis model using the example of a linear motor. Linear motors only support translational motor encoders and translational position encoders.

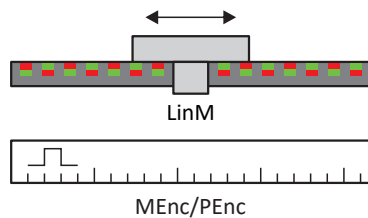
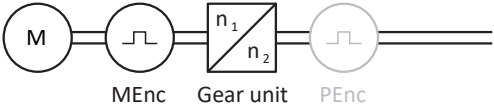
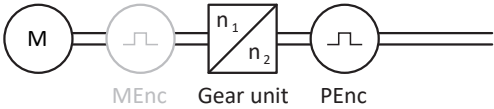
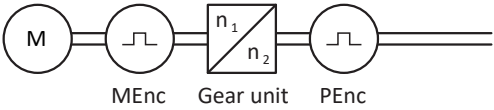


Fig. 10: Limited translational motion: Linear motor

Encoder arrangement

You parameterize the encoder arrangement by using B26 Motor encoder and I02 Position encoder. The motor encoder for velocity control is located on the motor shaft. The position encoder for position control is located on the gearbox output. If you use only one of the two encoders, it will be used for both velocity and position control.

Encoder	Parameterization	Encoder arrangement
Motor encoder	B26 ≠ 0: Inactive I02 = 0: Motor encoder	
Position encoder	B26 ≠ 0: Inactive I02 = B26	
Motor encoder & position encoder	B26 ≠ 0: Inactive I02 ≠ B26	

6.3 Limit switches

With limit switches, you can secure the travel range by selecting the respective end of the travel range and thus limiting the permitted positions of the axis. Hardware limit switches and software limit switches are available to limit the travel range. Hardware and software limit switches can be used individually or in combination with each other.

Hardware limit switches are physical sensors (hardware) that are triggered when a certain position is reached, while software limit switches are purely software-implemented functions for position limiting that are only available when the travel range is limited (I00 = 0: Limited).

Function

When a limit switch is overshoot, event 53: Limit switch is triggered and the axis goes into a fault condition with or without a quick stop, depending on the parameterization. As soon as a limit switch is triggered, the motion core interrupts the current motion command and refuses further commands in the same direction of motion. The direction lock ensures that the axis can only move away from the limit switch in the opposite direction and return to the permitted travel range. As soon as the direction lock is deactivated, event 53: Limit switch can be triggered again.

Further information on event 53: Limit switch, possible causes and actions can be found in the drive controller manual (see [Detailed information](#) [► 165]).

ATTENTION!

Material damage due to leaving the permitted travel range

When a limit switch is overshoot, the axis at the end of the permitted travel range will go into a fault condition, either with or without a quick stop depending on the parameterization of the device control, so it may come to a standstill past the limit switch and outside the permitted travel range.

- Allow sufficient space past the limit switch for your application to bring the axis to a standstill.

Restriction of the direction of motion

With an endless travel range, a direction lock can also be triggered by a positioning command outside the revolution length or by restricting the direction of motion using I04 (travel range: I00 = 1: Endless; revolution length: I01; permitted direction: I04). A restriction of the direction of motion is also indicated in the operating state of the axis (operating state: E80).

Since restricting the direction of motion using I04 prevents the hardware limit switch from moving in the opposite direction of motion, it is generally not advisable to combine the two functions.

6.3.1 Hardware limit switches

Hardware limit switches can be used for both limited and endless travel ranges (travel range: I00). A digital input can serve as the source for the digital signal for evaluating the respective hardware limit switch (source: I101, I102).

The digital signals for evaluating the hardware limit switches are low-active (normally closed contact), i.e. the limit switch is triggered when the digital signal is interrupted (signal: I441, I442 = 0: Inactive). Ensure that the hardware limit switches are connected correctly to prevent false triggering of event 53: Limit switch.

Relevant parameters

- I101 Source positive /limit switch
- I102 Source negative /limit switch
- I441 Signal /HW limit switch positive
- I442 Signal /HW limit switch negative
- I805 Actual value HW-Limit switch positive
- I806 Actual value HW-Limit switch negative
- I52 Delete limit switch memory
- I196 Direction blocking

Normal operation

When a hardware limit switch is overshoot, event 53: Limit switch is triggered with a fault reaction.

Causes

- 1: Hardware-Limit-Switch positive (prerequisite: I441 = 0: Inactive)
- 2: Hardware-Limit-Switch negative (prerequisite: I442 = 0: Inactive)

A hardware limit switch is triggered when the digital signal for evaluating the hardware limit switches is interrupted. The motion core interrupts the current motion command and refuses further commands in the same direction of motion; a direction lock becomes active (denial: I91 = 1: Active; cause: I90; direction lock: I196).

The direction lock is deactivated as soon as the limit switch memory is reset.

Jog mode

With hardware limit switches, the behavior of the axis is independent of the control type selected for jog mode (I26). When a hardware limit switch is overshoot, the axis behaves the same way in jog mode as in normal operation.

Referencing

During referencing, hardware limit switches can be evaluated like reference switches by using the corresponding referencing type (referencing type: I30 = 2: Limit switch; referencing methods: A, C, E, G).

With other referencing types, a triggered hardware limit switch reverses the direction of motion if the limit switch matches the direction of motion (e.g. positive limit switch and positive direction of motion). If a triggered hardware limit switch does not match the direction of motion, fault 53: Limit switch is triggered (e.g. positive limit switch with negative direction of motion).

Example

At the start of referencing, the axis is between the reference switch and positive limit switch. The referencing direction is positive. The axis moves in the positive direction and first encounters the positive limit switch, not the reference switch. The axis reverses and searches for the reference switch in the other direction.

Limit switch memory

When a hardware limit switch is overshoot, the position of the limit switch is saved internally and the triggered limit switch is recorded as active in the limit switch memory as long as the axis is at or past the limit switch (limit switch memory: I805, I806). Based on the stored position of the limit switch and the digital signals for evaluating the hardware limit switches, the limit switch memory is automatically reset in normal operation as soon as the axis has returned to the permitted travel range.

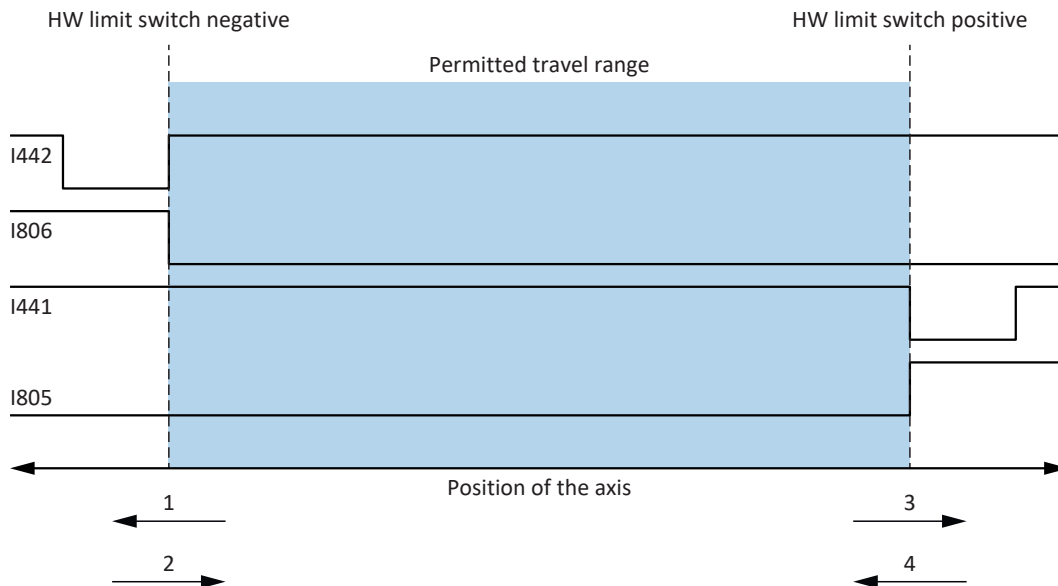


Fig. 11: Hardware limit switch: Limit switch memory

- 1 Negative HW limit switch triggered, limit switch memory set
- 2 Negative HW limit switch exited, limit switch memory reset
- 3 Positive HW limit switch triggered, limit switch memory set
- 4 Positive HW limit switch exited, limit switch memory reset

Example

A positive hardware limit switch ranges from position 100 to 120; when moving in the positive direction of motion, it is triggered at position 100 (signal: I441 = 1: Active → 0: Inactive). On the return journey, the tolerance may cause the real hardware limit switch to be left at position 101 (signal: I441 = 0: Inactive → 1: Active). However, in order for the effective limit switch or limit switch memory to be reset, position 100 must be reached or undershot.

The position of the limit switch is not stored in non-volatile memory, i.e. if the axis is outside the limit switches after switching on, it must first be moved back into the permitted travel range.

During commissioning or in case of defective hardware limit switch connections, problems can arise with the saved positions. These can be deleted with parameter I52. However, deletion only takes place if the corresponding limit switch signal is inactive. Stored positions can also be deleted by referencing, setting a reference can be deleted using I452 or by restarting the drive controller.

6.3.2 Software limit switches

Software limit switches are only available with a limited travel range (I00 = 0: Limited). To be able to check the actual position of the axis using software limit switches, the axis must be referenced.

The position of the negative software limit switch must be a smaller value than the position of the positive software limit switch. If the same value is defined for both software limit switches, the function is deactivated.

Relevant parameters

- A570[0] Software position limit: Lowest permissible position
- A570[1] Software position limit: Greatest permitted position
- I22 Target window
- I80 Current position
- I196 Direction blocking

Normal operation

When a software limit switch is overshot, event 53: Limit switch is triggered with a fault reaction.

Causes

- 3: SW-limit switch positive (prerequisite: I80 > A570[1] + I22)
- 4: SW-limit switch negative (prerequisite: I80 < A570[0] + I22)

A software limit switch is triggered if the actual position of the axis is outside the permitted travel range, taking the position window into account. The motion core interrupts the current motion command and refuses further commands in the same direction of motion; a direction lock becomes active (denial: I91 = 1: Active; cause: I90; direction lock: I196).

The direction lock is deactivated as soon as the actual position of the axis is back within the permitted travel range, taking the position window into account.

Jog mode

With software limit switches, the behavior of the axis depends on the control type selected for jog mode (I26).

If I26 = 0: Velocity control, a quick stop is triggered instead of a fault reaction when a software limit switch is overshot in jog mode, so that the axis comes to a standstill outside the permitted travel range depending on the quick stop deceleration (deceleration: I17). The direction lock ensures that the axis can only move away from the limit switch in the opposite direction and return to the permitted travel range.

When I26 = 1: Position control and the set position is outside the permitted travel range, the axis is braked with jog deceleration and tip jerk so that the axis still comes to a standstill within the permitted travel range shortly before the software limit switch (deceleration: I45; jerk: I18).

Referencing

Software limit switches are not evaluated during referencing (motion command I401 = 6: MC_Home).

6.4 Referencing

To be able to work with a system with absolute positions, the relation of a measured axis position to an actual axis position must be determined.

During initial commissioning or after changes to the axis model, the actual position of the axis is unknown. A defined initial position is necessary. As a rule, this is either identified through a reference search or referencing setting. The associated process is referred to as referencing.

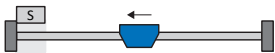
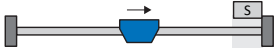
Absolute movements can be performed only in a referenced state. For relative movements, referencing is only required if the software limit switch function is used at the same time.

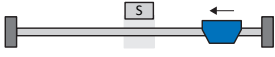
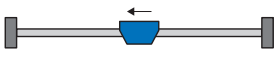
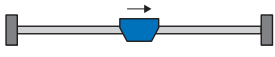
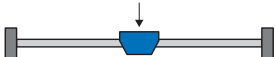
When replacing a drive controller, the reference can be transferred to the replacement drive controller via an SD card. The information is also saved on the SD card if the action A00 Save values is executed for the referenced axis. For more information on replacing a drive controller, refer to the manual for the respective drive controller.

6.4.1 Referencing methods

The following table shows an overview of the possible referencing methods.

Abbreviation	Meaning				
S	Switch				
M/F	Torque or force				

	Method	CiA 402 Homing method	Initial movement	Zero pulse	Feature
	A	1: Homing on negative limit switch and index pulse	Negative	Yes	Negative limit switch
	B	5: Homing on negative home switch and index pulse	Negative	Yes	Negatively positioned reference switch
	C	17: Homing on negative limit switch	Negative	–	Negative limit switch
	D	21: Homing on negative home switch	Negative	–	Negatively positioned reference switch
	E	2: Homing on positive limit switch and index pulse	Positive	Yes	Positive limit switch
	F	3: Homing on positive home switch and index pulse	Positive	Yes	Positively positioned reference switch
	G	18: Homing on positive limit switch	Positive	–	Positive limit switch
	H	19: Homing on positive home switch	Positive	–	Positively positioned reference switch
	I	7: Homing on home switch and index pulse initial positive	Positive	Yes	Centrally positioned reference switch
	J	23: Homing on home switch initial positive	Positive	–	Centrally positioned reference switch

	Method	CiA 402 Homing method	Initial movement	Zero pulse	Feature
	K	11: Homing on home switch and index pulse initial negative	Negative	Yes	Centrally positioned reference switch
	L	27: Homing on home switch initial negative	Negative	–	Centrally positioned reference switch
	M	33: Homing on index pulse negative	Negative	Yes	–
	N	34: Homing on index pulse positive	Positive	Yes	–
	O	37: Homing on current position (35: Homing on current position)	–	–	Setting the reference
	P	-1: Homing on torque/force limit initial positive	Positive	–	Torque/force stop
	Q	-3: Homing on torque/force limit and index pulse initial positive	Positive	Yes	Torque/force stop
	R	-2: Homing on torque/force limit initial negative	Negative	–	Torque/force stop
	S	-4: Homing on torque/force limit and index pulse initial negative	Negative	Yes	Torque/force stop

Tab. 4: Referencing methods

6.4.1.1 Referencing methods in detail

The following chapters show the individual referencing methods in detail.

The following abbreviations are used in the graphs on the referencing methods:

Abbreviation	Meaning
ALT	Alternative
LS	Limit Switch
RS	Reference Switch
ZP	Zero Pulse

Information

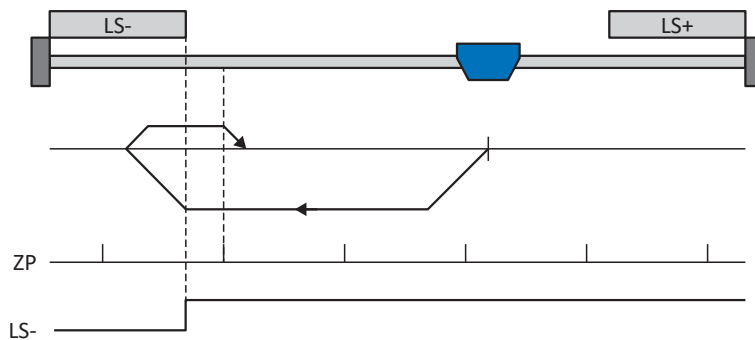
For graphically representing axes, the left is the smallest and the right is the largest position value. This means a positive movement goes to the right and a negative to the left.

The following parameters are used in the descriptions of the referencing methods:

Coordinate	Name
A569	Home offset
A586	Homing method
A587[0]	Speed during search for switch
A587[1]	Speed during search for zero
A588	Homing acceleration
I28	Homing torque/force limit
I29	Time homing torque/force limit
I43	Move to reference position
I44	Referencing jerk
I53	Index search offset
I101	Source positive /limit switch
I102	Source negative /limit switch
I103	Source reference switch

6.4.1.1.1 Referencing method A

Referencing method A determines a reference using a run to the negative limit switch and zero pulse.



This method corresponds to the CiA 402 Homing method, 1: Homing on negative limit switch and index pulse operating mode.

Preparation

1. Activate referencing method A by setting A586 to a value of 1: Homing on negative limit switch and index pulse.
2. I102:
Enter the data source for the negative limit switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search.

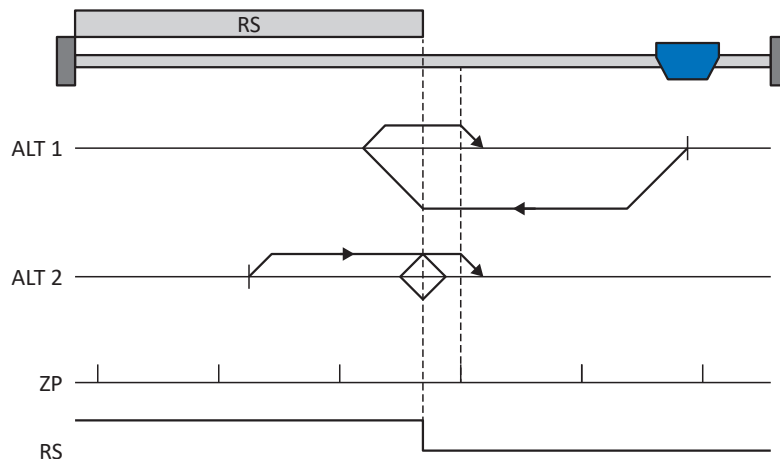
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the negative limit switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the limit switch.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.2 Referencing method B

Referencing method B determines the reference using a run to the negatively positioned reference switch and zero pulse.



This method corresponds to the CiA 402 Homing method, 5: Homing on negative home switch and index pulse operating mode.

Preparation

1. Activate referencing method B by setting parameter A586 to the value 5: Homing on negative home switch and index pulse.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between two referencing variants.

✓ Alternative 1: Drive has not actuated reference switch

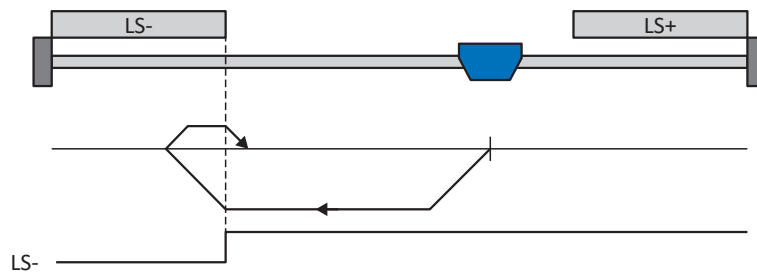
1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts in the positive direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity A587[1] until it reaches the zero pulse.
4. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.3 Referencing method C

Referencing method C determines the reference using a run to the negative limit switch.



This method corresponds to the CiA 402 Homing method, 17: Homing on negative limit switch operating mode.

Preparation

1. Activate referencing method C by setting parameter A586 to the value 17: Homing on negative limit switch.
2. I102:
Enter the data source for the negative limit switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.

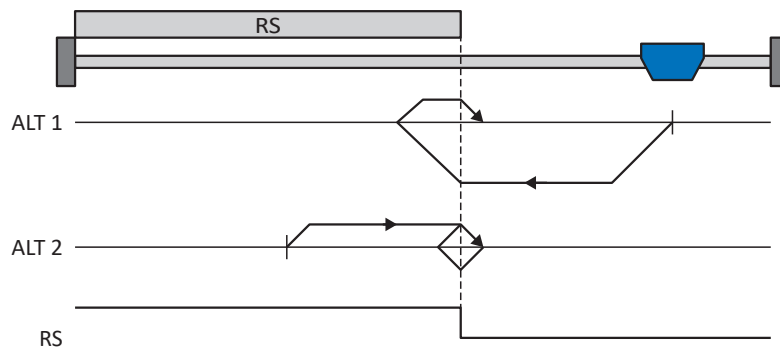
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the negative limit switch and continues its run with velocity A587[1] until it leaves the limit switch again.
3. The current actual position is set to the value of the reference position A569 upon leaving the limit switch.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.4 Referencing method D

Referencing method D determines the reference using a run to the negatively positioned reference switch.



This method corresponds to the CiA 402 Homing method, 21: Homing on negative home switch operating mode.

Preparation

1. Activate referencing method D by setting parameter A586 to the value 21: Homing on negative home switch.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between two referencing variants.

✓ Alternative 1: Drive has not actuated reference switch

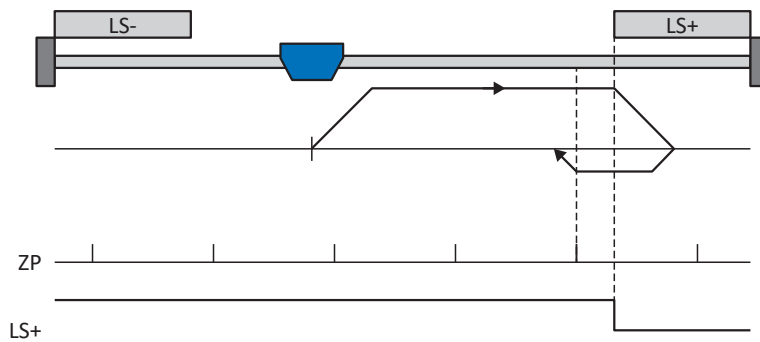
1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it leaves the reference switch again.
3. The current actual position is set to the value of the reference position A569 upon leaving the reference switch.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts with acceleration A588 and velocity A587[1] in the positive direction until it leaves the reference switch.
2. It changes its direction upon leaving the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity A587[1] until it reaches the reference switch again.
4. The current actual position is set to the value of the reference position A569 upon reaching the reference switch.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.5 Referencing method E

Referencing method E determines the reference using a run to the positive limit switch and zero pulse.



This method corresponds to the CiA 402 Homing method, 2: Homing on positive limit switch and index pulse operating mode.

Preparation

1. Activate referencing method E by setting parameter A586 to the value 2: Homing on positive limit switch and index pulse.
2. I101:
Enter the data source for the positive limit switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search.

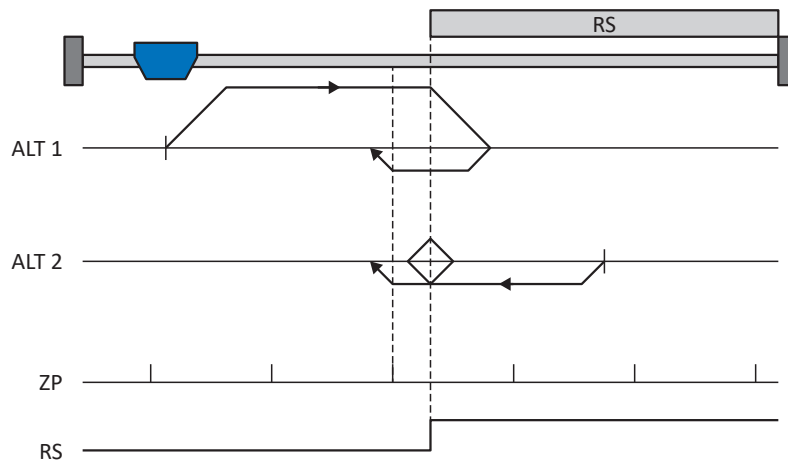
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the positive limit switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the limit switch.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.6 Referencing method F

Referencing method F determines the reference using a run to the positively positioned reference switch and zero pulse.



This method corresponds to the CiA 402 Homing method, 3: Homing on positive home switch and index pulse operating mode.

Preparation

1. Activate referencing method F by setting parameter A586 to the value 3: Homing on positive home switch and index pulse.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569: Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between two referencing variants.

✓ Alternative 1: Drive has not actuated reference switch

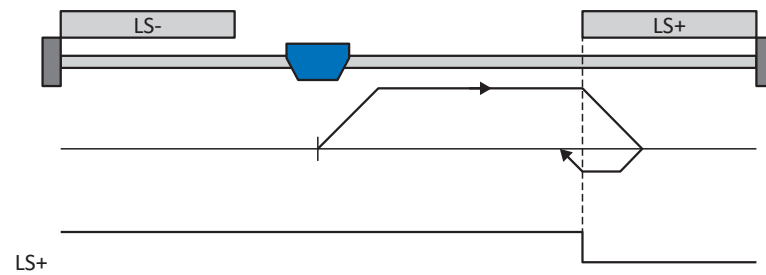
1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity A587[1] until it reaches the zero pulse.
4. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.7 Referencing method G

Referencing method G determines the reference using a run to the positive limit switch.



This method corresponds to the CiA 402 Homing method, 18: Homing on positive limit switch operating mode.

Preparation

1. Activate referencing method G by setting parameter A586 to the value 18: Homing on positive limit switch.
2. I102:
Enter the data source for the positive limit switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.

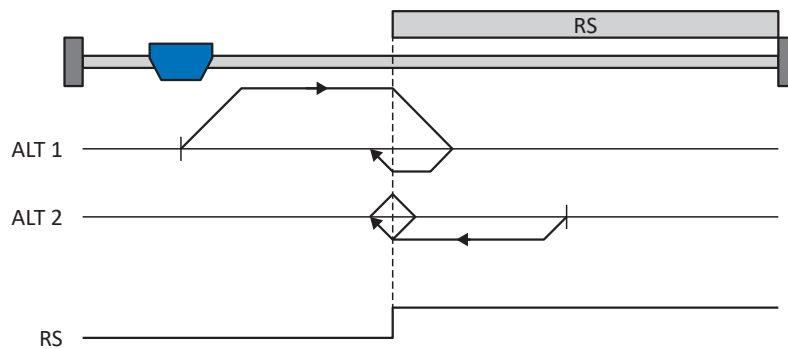
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the positive limit switch and continues its run with velocity A587[1] until it leaves the limit switch again.
3. The current actual position is set to the value of the reference position A569 upon leaving the limit switch.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.8 Referencing method H

Referencing method H determines the reference using a run to the positively positioned reference switch.



This method corresponds to the CiA 402 Homing method, 19: Homing on positive home switch operating mode.

Preparation

1. Activate referencing method H by setting parameter A586 to the value 19: Homing on positive home switch.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between two referencing variants.

✓ Alternative 1: Drive has not actuated reference switch

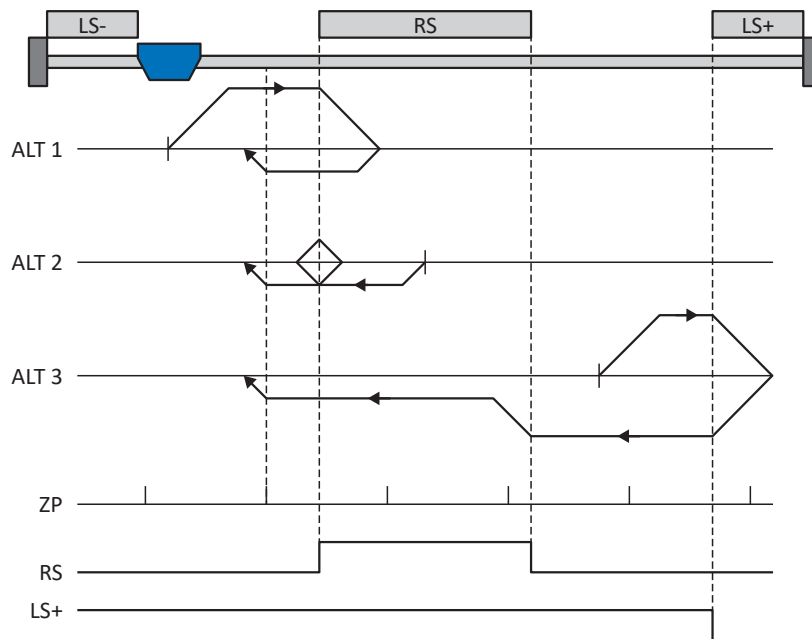
1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it leaves the reference switch again.
3. The current actual position is set to the value of the reference position A569 upon reaching the reference switch.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts with acceleration A588 and velocity A587[1] in the negative direction until it leaves the reference switch.
2. It changes its direction upon leaving the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity A587[1] until it reaches the reference switch again.
4. The current actual position is set to the value of the reference position A569 upon reaching the reference switch.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.9 Referencing method I

Referencing method I determines the reference using a run to the centered reference switch and zero pulse.



This method corresponds to the CiA 402 Homing method, 7: Homing on home switch and index pulse initial positive operating mode.

Preparation

1. Activate referencing method I by setting parameter A586 to the value 7: Homing on home switch and index pulse initial positive.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between three referencing variants.

✓ Alternative 1: Drive is positioned between the negative limit and reference switches

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

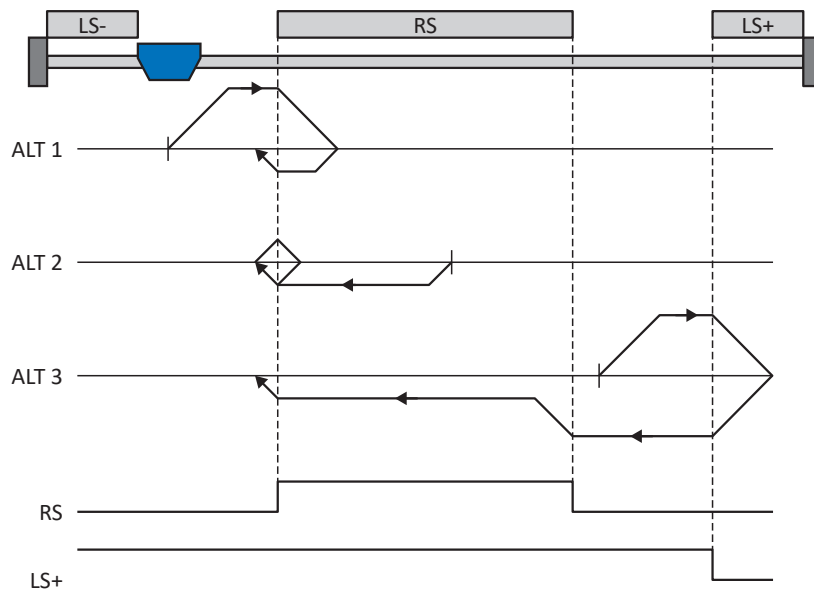
1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity A587[1] until it reaches the zero pulse.
4. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 3: Drive is positioned between the reference and positive limit switches

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the positive limit switch and continues its run until the reference switch is reached.
3. Upon reaching the reference switch, the drive changes its velocity to A587[1] until it leaves the reference switch again.
4. The current actual position is set to the value of the reference position A569 after leaving the reference switch and upon reaching the next zero pulse.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.10 Referencing method J

Referencing method J determines the reference using a run to the centered reference switch.



This method corresponds to the CiA 402 Homing method, 23: Homing on home switch initial positive operating mode.

Preparation

1. Activate referencing method J by setting parameter A586 to the value 23: Homing on home switch initial positive.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between three referencing variants.

✓ Alternative 1: Drive is positioned between the negative limit and reference switches

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it leaves the reference switch again.
3. The current actual position is set to the value of the reference position A569 upon leaving the reference switch.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

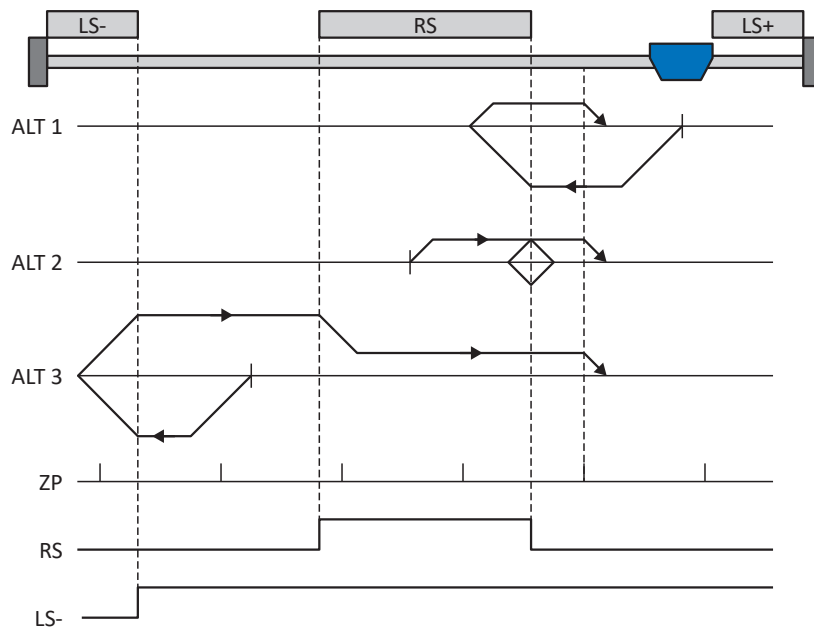
1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0] until it leaves the reference switch.
3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity A587[1] until it reaches the reference switch again.
4. The current actual position is set to the value of the reference position A569 upon reaching the reference switch.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 3: Drive is positioned between the reference and positive limit switches

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. Upon reaching the positive limit switch, the drive changes its direction and continues its run until it reaches the reference switch.
3. Upon reaching the reference switch, the drive changes its velocity to A587[1] until it leaves the reference switch again.
4. The current actual position is set to the value of the reference position A569 upon leaving the reference switch.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.11 Referencing method K

Referencing method K determines the reference using a run to the centered reference switch and zero pulse.



This method corresponds to the CiA 402 Homing method, 11: Homing on home switch and index pulse initial negative operating mode.

Preparation

1. Activate referencing method K by setting parameter A586 to the value 11: Homing on home switch and index pulse initial negative.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between three referencing variants.

✓ Alternative 1: Drive is positioned between the reference and positive limit switches

1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

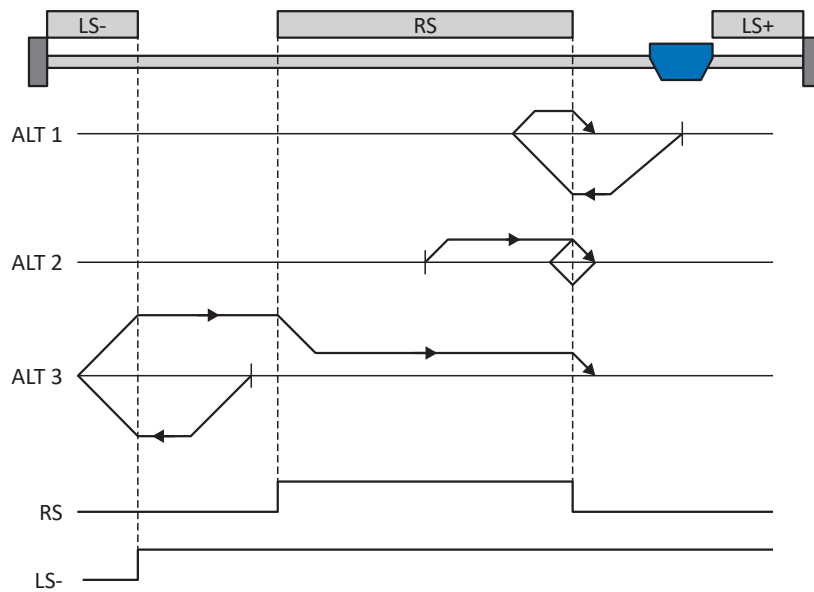
1. The drive starts in the positive direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity A587[1] until it reaches the zero pulse.
4. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 3: Drive is positioned between the negative limit and reference switches

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the negative limit switch and continues its run until the reference switch is reached.
3. Upon reaching the reference switch, the drive changes its velocity to A587[1] and continues its run until it reaches the next zero pulse after leaving the reference switch.
4. The current actual position is set to the value of reference position A569 upon reaching the next zero pulse.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.12 Referencing method L

Referencing method L determines the reference using a run to the centered reference switch.



This method corresponds to the CiA 402 Homing method, 27: Homing on home switch initial negative operating mode.

Preparation

1. Activate referencing method L by setting parameter A586 to the value 27: Homing on home switch initial negative.
2. I103:
Enter the data source for the reference switch.
3. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.

Referencing

If the CiA 402 homing operating mode is active, a distinction is made between three referencing variants.

✓ Alternative 1: Drive is positioned between the reference and positive limit switches

1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it leaves the reference switch again.
3. The current actual position is set to the value of the reference position A569 upon leaving the reference switch.
4. The drive comes to a standstill with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 2: Drive stopped at reference switch

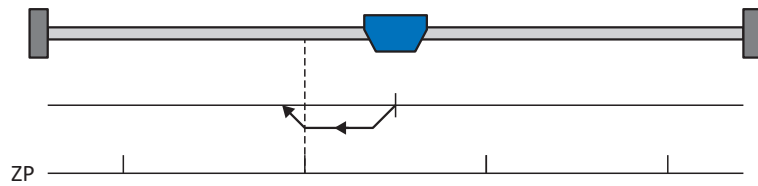
1. The drive starts with acceleration A588 and velocity A587[1] in the positive direction until it leaves the reference switch.
2. It changes its direction upon leaving the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity A587[1] until it reaches the reference switch again.
4. The current actual position is set to the value of the reference position A569 upon reaching the reference switch.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

✓ Alternative 3: Drive is positioned between the negative limit and reference switches

1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the negative limit switch and continues its run until the reference switch is reached.
3. When the reference switch is reached, the drive changes its velocity to A587[1] and continues its run until it leaves the reference switch again.
4. The current actual position is set to the value of the reference position A569 upon leaving the reference switch.
5. The drive comes to a standstill with deceleration A588.
6. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.13 Referencing method M

This method determines the reference using a run to the zero pulse.



This method corresponds to the CiA 402 Homing method, 33: Homing on index pulse negative operating mode.

Preparation

1. Activate referencing method M by setting parameter A586 to the value 33: Homing on index pulse negative.
2. A587[1], A588, I44, A569:
Define the set values necessary for the referencing.

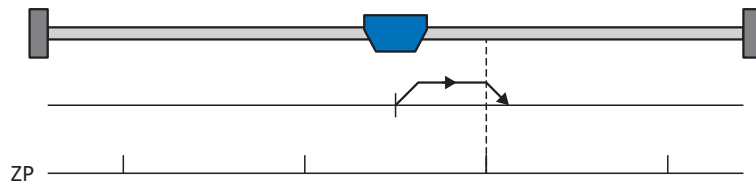
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
3. The drive comes to a standstill with deceleration A588.
4. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.14 Referencing method N

Referencing method N determines the reference using a run to the zero pulse.



This method corresponds to the CiA 402 Homing method, 34: Homing on index pulse positive operating mode.

Preparation

1. Activate referencing method N by setting parameter A586 to the value 34: Homing on index pulse positive.
2. A587[1], A588, I44, A569:
Define the set values necessary for the referencing.

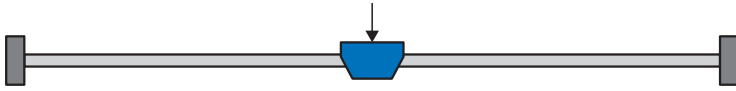
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the positive direction with acceleration A588 and velocity A587[1].
2. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
3. The drive comes to a standstill with deceleration A588.
4. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.15 Referencing method O

Referencing method O determines the reference by setting the reference to any position.



This method corresponds to the CiA 402 Homing method, 37: Homing on current position operating mode and the 35: Homing on current position operating mode currently reserved in CiA 402.

Preparation

1. Activate referencing method O by setting parameter A586 to the value 37: Homing on current position.
2. A569:
Define the reference position.

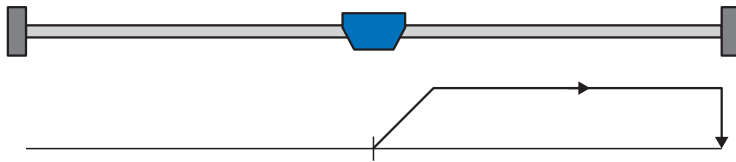
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

The current actual position is set to the value of the reference position A569.

6.4.1.1.16 Referencing method P

Referencing method P determines the reference using a run with a torque/force stop.



Preparation

1. Activate referencing method P by setting parameter A586 to the value -1: Homing on torque/force limit initial positive.
2. A587[0], A588, I44, A569:
Define the set values necessary for the referencing.
3. I28, I29:
Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.

Information

In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

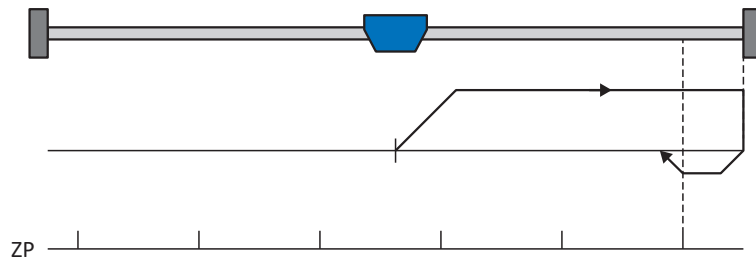
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. The current actual position is set to the value of the reference position A569 when the torque/force limit is reached and the time stored in I29 has elapsed.
3. The set values are set to the value 0 with deceleration A588.
4. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.17 Referencing method Q

Referencing method Q determines the reference using a run with a torque/force stop and zero pulse.



Preparation

1. Activate referencing method Q by setting parameter A586 to the value -3: Homing on torque/force limit and index pulse initial positive.
2. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.
3. I28, I29:
Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.
4. I53:
Define the start of the search for the zero pulse search.

Information

In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

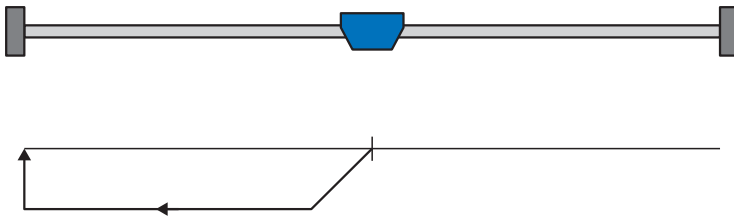
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the torque/force limit and when the time stored in I29 has passed, and continues its run with velocity A587[1] until it reaches the next zero pulse.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The set values are set to the value 0 with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.18 Referencing method R

Referencing method R determines the reference using a run with a torque/force stop.



Preparation

1. Activate referencing method R by setting parameter A586 to the value -2: Homing on torque/force limit initial negative.
2. A587[1], A588, I44, A569:
Define the set values necessary for the referencing.
3. I28, I29:
Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.

Information

In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

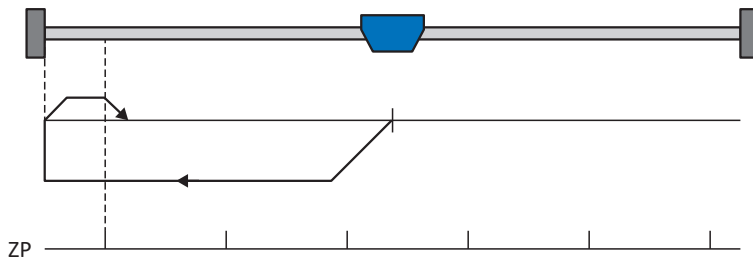
Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. The current actual position is set to the value of the reference position A569 when the torque/force limit is reached and the time stored in I29 has elapsed.
3. The set values are set to the value 0 with deceleration A588.
4. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.1.1.19 Referencing method S

Referencing method S determines the reference using a run with torque/force stop and zero pulse.



Preparation

1. A586:
Activate referencing method S by setting this parameter to the value -4: Homing on torque/force limit and index pulse initial negative.
2. A587[0], A587[1], A588, I44, A569:
Define the set values necessary for referencing.
3. I28, I29:
Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.
4. I53:
Define the start of the search for the zero pulse search.

Information

In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

Referencing

If the CiA 402 homing operating mode is active, referencing takes place as follows:

1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the torque/force limit and when the time stored in I29 has passed, and continues its run with velocity A587[1] until it reaches the next zero pulse.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse.
4. The set values are set to the value 0 with deceleration A588.
5. If I43 is set to 1: Active, the drive is positioned at reference position A569.

6.4.2 Reference position

Depending on the referencing method A586, the actual position I80 is replaced by the reference position A569 during the referencing event.

6.4.3 Reference retaining

STOBER offers a convenient drive-based referencing system. Depending on the encoder model used and the reference management, different types of reference retaining (I46) are offered.

If the reference is to be processed using controller-based operation and the drive controller is to provide absolute positions based on the motor shaft, you have to take the following settings into account so that the reference can be reliably processed by the controller. These settings provide you with the absolute position of the encoder based on the motor shaft in the data format of the axis model, which you map in DriveControlSuite during commissioning.

6.4.3.1 Settings for control-based reference management

For reference retaining I46, select the 5: Direct absolute option.

This means that the absolute position of the encoder is directly mapped in the actual position (prerequisite: I02 = 0: Motor encoder, actual position: A545).

Information

Afterwards, you must not perform any drive-based referencing using a referencing method or by setting the reference directly, as this can cause the actual position to shift.

The axis model must be in relation to the output shaft of the motor und not the output after a mechanism such as a gearbox.

Positions are generally provided in a 32-bit integer data format.

6.4.3.2 Preset – Scaling to 20 bits

Scaling to 20 bits corresponds to the preset for axis mode scaling in the CiA 402 application. If the position range represented by 32 bits is exceeded, the controller must save the number of multi-turn overruns and take them into account after a power-off/power-on for controller-based reference retaining.

6.4.3.3 Preset with greater dynamic range

In order to get greater dynamic range for the derived variables of velocity, acceleration, deceleration and jerk, you need the following parameters:

- A310: Number of decimal places for the scaled transmission of the process data for velocities
- A311: Number of decimal places for the scaled transmission of the process data for acceleration, deceleration and jerk
- I06: Number of decimal places for positions

If, for example, the velocity, acceleration and jerk values are to be weighted more heavily by a factor of 100, the following steps are necessary:

- Change the value of A310 and A311 to 0
- Change the number of decimal places for the position I06 to 2

6.4.4 Reference loss

Axis

In certain cases, an axis loses its reference, and its state changes from I86 = 1: Active to I86 = 0: Inactive.

Normal operation (axis)

During normal operation, encoder faults or actions can lead to a loss of the reference. If the reference was deleted by an encoder event, it can subsequently be restored.

Information

Before restoring the reference, check the actual position displayed (I80). If in doubt, do the referencing again. If relative encoders are used or if the axis was still moving during the encoder fault, the actual position displayed may deviate from the real actual position of the axis.

Cause		Check and action
Event 76: Position encoder	Encoder fault	Actual position may still be correct, restoring the reference is possible: ▪ Check the actual position (I80). ▪ Confirm reference (I459) or re-reference axis
Event 37: Motor encoder	Encoder fault	When using the motor encoder as a position encoder (I02 = 0: Motor encoder), 2 faults are triggered (37: Motor encoder and 76: Position encoder), but it is possible that only one fault is displayed in display parameter E82 and in the fault memory. Actual position may still be correct, restoring the reference is possible: ▪ Check the actual position (I80). ▪ Confirm reference (I459) or re-reference axis
Action I38	Deleted reference	Action I38 deletes the reference but does not change the actual position displayed. Actual position may still be correct, restoring the reference is not possible: ▪ Check the actual position (I80). ▪ Reference the axis again

Tab. 5: Loss of reference of the axis in normal operation

Parameter changes (axis)

Changing individual parameters or transferring a new configuration with changed settings can lead to a loss of the reference.

Cause		Check and action
Changing parameters	Changed axis model	Actual position is undefined if one of the following parameters was changed: ▪ A571 Polarity ▪ A584 Gear ratio ▪ A585 Feed constant ▪ B26 Motor encoder ▪ I00 Position range ▪ I01 Circular length ▪ I02 Position encoder ▪ I05 Type of axis It is not possible to restore the reference: ▪ Reference the axis again
Changing parameters	Changed encoder interface	Actual position is undefined if a parameter of the H group was changed. It is not possible to restore the reference: ▪ Reference the axis again

Tab. 6: Reference loss of the axis due to parameter changes

Restarting the drive controller (axis)

Depending on the encoder type and method for reference retaining (I46), the reference of a previously referenced axis is restored or deleted after a restart.

By default (I46 = 0: Normal), the reference for a multi-turn absolute encoder is retained after a restart if the axis was referenced with this encoder. In all other cases, the reference is deleted as soon as the drive controller is switched off.

The method for reference retaining can be adapted in I46. In addition to the preset, the following further options are available:

- Reference is retained if the measuring range covers the entire travel range
- Reference is retained as long as the position change in the switched-off state is less than the reference retention window (I48)
- Reference is retained regardless of the presence of an encoder
- Reference is retained regardless of the encoder type
- Reference is deleted when the drive controller is switched off

Special case of incorrect motor connection (axis)

If a motor is inadvertently connected to the incorrect axis or the incorrect drive controller, such as after a case of service, the drive controller behaves as follows after it is switched on:

- The axis changes to the non-referenced state (I86 = 0: Inactive)
- The actual position displayed is undefined

However, the information from the original motor and the associated reference data are stored in the drive controller. After switching off the drive controller, connecting the correct motor and restarting the drive controller, the reference is restored and the actual position is displayed correctly (prerequisites: multi-turn absolute encoder, referenced axis and preset for reference retaining I46 = 0: Normal).

Master encoder

Actual positions of the master encoder are used in the Drive Based Synchronous application for synchronous operation.

In all applications except the Drive Based Center Winder application, the actual position of the master encoder can be used to pass on the position of another encoder attached to the machine to the control system. The drive controller transfers the position from the encoder interface to the respective fieldbus.

Master encoders behave in a similar way to position encoders. In certain cases, a master encoder loses its reference, and its state changes from G89 = 1: Active to G89 = 0: Inactive.

Normal operation (master encoder)

During normal operation, encoder faults can lead to a loss of the reference.

Cause		Check and action
Event 77: Master encoder	Encoder fault	Actual position of the master encoder is undefined, it is not possible to restore the reference: ▪ Reference the master encoder again

Tab. 7: Reference loss of the master encoder in normal operation

Parameter changes (master encoder)

Changing individual parameters or transferring a new configuration with changed settings can lead to a loss of the reference.

Cause		Check and action
Changing parameters	Changed axis model	Actual position of the master encoder is undefined if one of the following parameters was changed: ▪ G30 Position range master ▪ G40 Circular length master ▪ G47 Distance factor numerator master ▪ G48 Distance factor denominator master ▪ G104 Source master encoder It is not possible to restore the reference: ▪ Reference the master encoder again
Changing parameters	Changed encoder interface	Actual position of the master encoder is undefined if a parameter of the H group was changed. It is not possible to restore the reference: ▪ Reference the master encoder again

Tab. 8: Reference loss of the master encoder due to parameter changes

Restarting the drive controller (master encoder)

Depending on the encoder type and method for reference retaining (G35), the reference of a previously referenced master encoder is restored or deleted after a restart.

By default (G35 = 0: Normal), the reference for a multi-turn absolute encoder is retained after a restart if this encoder was used for referencing. In all other cases, the reference is deleted as soon as the drive controller is switched off.

The method for reference retaining can be adapted in G35. In addition to the preset, the following further options are available:

- Reference is retained if the measuring range covers the entire travel range
- Reference is retained as long as the position change in the switched-off state is less than the reference retention window (I48)
- Reference is retained regardless of the presence of an encoder
- Reference is retained regardless of the encoder type
- Reference is deleted when the drive controller is switched off

6.5 CiA 402 device control

The CiA 402 device control is based on the internationally standardized CiA 402 CANopen device profile for electric drives. This profile describes the control sequence of a drive controller based on a device state machine. Each device state represents a specific behavior.

The device state machine has to receive certain commands for state transitions. These commands are the result of a bit combination in the control word in accordance with CiA 402. The bit combination in the status word indicates the current state of the drive controller.

The following chapters describe the device states and the possible state transitions associated with them. You will also learn whether certain actions are necessary on your part to reach the individual device states, what those actions are and which user-specific factors you can parameterize yourself.

6.5.1 CiA 402 device state machine

The device state machine describes the different device states of the drive controller along with the possible state changes.

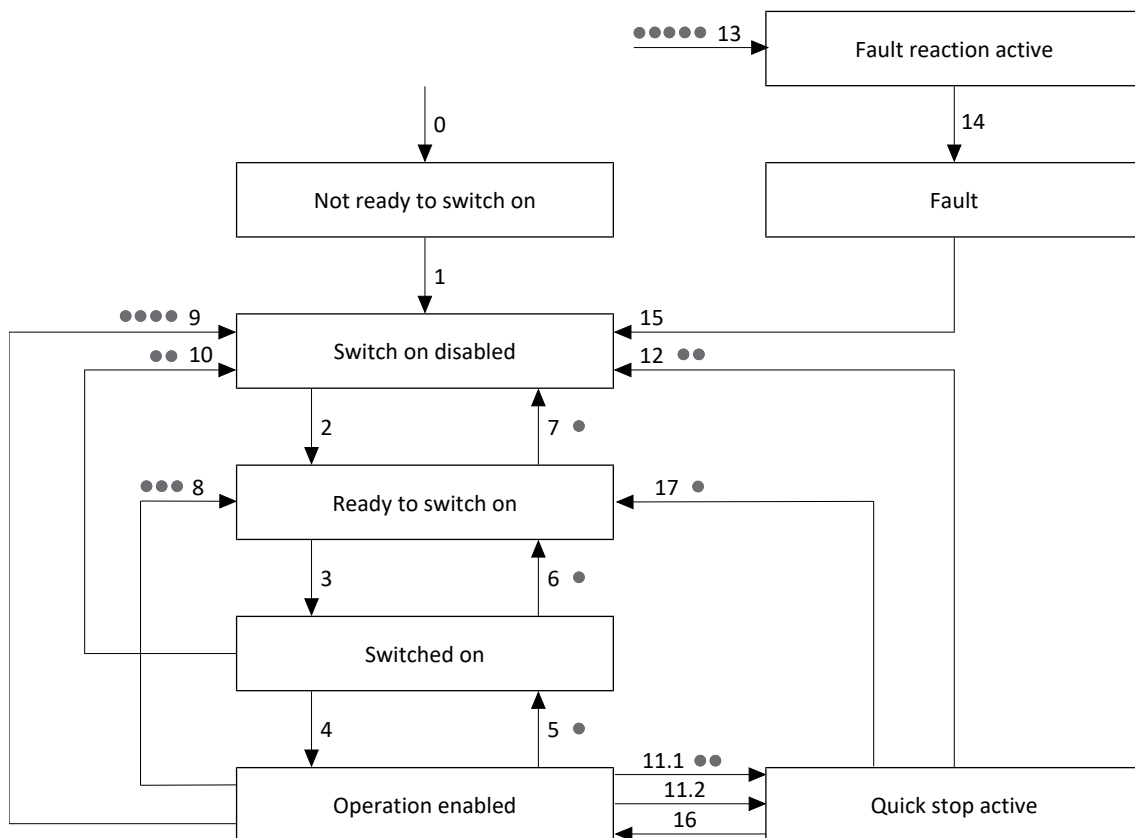


Fig. 12: CiA 402 device state machine: Device states and state changes

Priority levels are marked by dots. The more dots a state change has, the higher its priority. Accordingly, a state change without any dots has the lowest priority.



DANGER!

Risk of fatal injury due to gravity-loaded axis!

On a gravity-loaded axis with brake, the load may drop until the brake is fully engaged. This can result in serious injury or death.

- Always shut down the drive using a quick stop (state transition 11 in the device state machine).

6.5.2 Commands, states and transitions

The states provided by CiA 402 device control all have specific features. A state either transitions into another state automatically or requires specific commands.

Parameter E48 shows the current device state of a drive controller.

6.5.2.1 Key

The following terms are used in the descriptions of the states and transitions:

Term	Meaning
Enable active	Additional enable signal is active (A300 = 1: Active, source: A60)
Enable inactive	Additional enable signal is inactive (A300 = 0: Inactive, source: A60)
Quick stop in case of Enable-off active	Quick stop in case of Enable-off is active (A44 = 1: Active)
Quick stop in case of Enable-off inactive	Quick stop in case of Enable-off is inactive (A44 = 0: Inactive)
Quick stop end	Standstill reached (I199 = 1: Active) OR Maximum quick stop duration has expired (A39)
Quick stop duration expired	Maximum quick stop duration has expired (A39)
Quick stop option code = 2	A536 = 2: Slow down on quick stop ramp and transit into Switch On Disabled
Quick stop option code = 6	A536 = 6: Slow down on quick stop ramp and stay in Quick Stop Active (quick stop deceleration: A578)

Tab. 9: States, transitions and conditions: Terms

6.5.2.2 Commands

The device state machine has to receive certain commands for state changes. The commands are the result of a bit combination in the control word A515 in accordance with CiA 402. The following table shows the states of the bits in A515 and their combination for the commands. Bits marked with X are irrelevant.

Command	Bit 7 Fault reset	Bit 3 Enable operation	Bit 2 Quick stop	Bit 1 Enable voltage	Bit 0 Switch on
Shutdown	0	X	1	1	0
Switch on	0	0	1	1	1
Disable voltage	0	X	X	0	X
Quick stop	0	X	0	1	X
Disable operation	0	0	1	1	1
Enable operation	0	1	1	1	1
Fault reset	Rising edge	X	X	X	X

Tab. 10: Bit combinations in the control word in accordance with CiA 402

Information

The STO safety function is mapped by the **Disable voltage** command, while drive-based braking within the SS1 safety function is mapped by the **Quick stop** command.

Information

The brake control is not defined by the CiA 402 device profile. The brake is either controlled by the drive controller (prerequisite: F00 = 1: Active and F108 = 0: Internal (automatic)) or externally by a controller (prerequisite: F00 = 1: Active and F108 = 1: External (plc), source: A515, bit 14).

6.5.2.3 Not ready to switch on

Features

- Drive controller and safety module are initialized and tested
- The power unit as well as drive and activation functions are locked
- Brakes remain engaged

Transition to Switch on disabled (1)

The drive controller switches to the *Switch on disabled* state automatically after initialization and successful completion of the self-test (typically approx. 30 s).

Transition to Fault reaction active (13), priority: ● ● ● ● ●

Fault with or without fault reaction

6.5.2.4 Switch on disabled

Features

- Initialization has been completed
- Self-test was successfully completed
- The power unit as well as drive and activation functions are locked
- Brakes remain engaged, release override can be requested with a rising edge

Possible causes of a switch-on lockout include:

1. Insufficient or no supply voltage/DC link feed-in
2. STO safety function is active
3. Disable voltage command
4. Quick stop command (cause only in the *Switch on disabled*, *Ready to switch on* and *Switched on* states)
5. Release override is active (source: F06; cause only in the *Switch on disabled*, *Ready to switch on* and *Switched on* states)
6. Control panel or local operation is active (cause only in the *Switch on disabled*, *Ready to switch on* and *Switched on* states)

For the exact cause of a switch-on lockout, refer to parameter E49.

Transition to Ready to switch on (2)

Shutdown command

AND

Enable active

AND

No cause for a switch-on lockout

Transition to Fault reaction active (13), priority: ● ● ● ● ●

Fault with or without fault reaction

6.5.2.5 Ready to switch on

Features

- Power unit and drive function are disabled
- Drive controller is ready to switch on
- Brakes remain engaged, release override can be requested with a rising edge

Transition to Switched on (3)

Switch on command

Transition to Switch on disabled (7), priority: ●

Quick stop command

OR

Disable voltage command

OR

Enable inactive

OR

Cause for a switch-on lockout

Transition to Fault reaction active (13), priority: ● ● ● ● ●

Fault with or without fault reaction

6.5.2.6 Switched on

Features

- Power unit is ready for operation
- Drive function is locked; set values are not processed
- Brakes remain engaged, release override can be requested with a rising edge

Transition to Operation enabled (4)

Enable operation command

Transition to Ready to switch on (6), priority: ●

Shutdown command

Transition to Switch on disabled (10), priority: ● ●

Disable voltage command

OR

Enable inactive

OR

Cause for a switch-on lockout

Transition to Fault reaction active (13), priority: ● ● ● ● ●

Fault with or without fault reaction

6.5.2.7 Operation enabled


DANGER!

Risk of fatal injury due to gravity-loaded axis!

On a gravity-loaded axis with brake, the load may drop until the brake is fully engaged. This can result in serious injury or death.

- Always shut down the drive using a quick stop (state transition 11 in the device state machine).

Features

- Power unit is switched on
- Drive function is enabled, set values are processed
- Brakes are released:
 - csp, csv, cst, vl, pv or pt operating mode: upon first active motion command
 - pp, ip or homing operating mode: depending on the operating mode-specific bit 4

Transition to Switched on (5), priority: ●

This transition brings the drive to an uncontrolled standstill. The power unit is disabled and axis movement is no longer controlled by the drive controller.

Disable operation command

Transition to Ready to switch on (8), priority: ● ● ●

This transition brings the drive to an uncontrolled standstill. The power unit is disabled and axis movement is no longer controlled by the drive controller.

Shutdown command AND quick stop in case of Enable-off inactive

Transition to Switch on disabled (9), priority: ● ● ● ●

This transition brings the drive to an uncontrolled standstill. The power unit is disabled and axis movement is no longer controlled by the drive controller.

Disable voltage command

OR

Enable inactive AND quick stop in case of Enable-off inactive

OR

Cause for a switch-on lockout

In addition to CiA 402: Transition to Quick stop active (11.1), priority: ● ●

Shutdown command AND quick stop in case of Enable-off active

Transition to Quick stop active (11.2)

Quick stop command

OR

Enable inactive AND quick stop in case of Enable-off active

OR

Request for drive-controlled SS1

OR

Request for drive-controlled SS2

Transition to Fault reaction active (13), priority: ● ● ● ● ●

Fault with or without fault reaction

6.5.2.8 Quick stop active

Features

- Power unit is switched on; drive function is enabled
- A quick stop is executed
- Brakes remain or are released

Transition to Switch on disabled (12), priority: ● ●

Quick stop option code = 2 AND one of the following conditions:

Disable voltage command

OR

Enable inactive AND quick stop in case of Enable-off inactive

OR

Enable inactive AND quick stop in case of Enable-off active AND quick stop end

OR

No SS1 AND no SS2 AND quick stop end

OR

SS1 AND standstill

OR

Cause for a switch-on lockout

Information

With SS1, the maximum quick stop duration (A39) is not evaluated.

Transition to Fault reaction active (13), priority: ● ● ● ● ●

Fault with or without fault reaction

Transition to Operation enabled (16)

Enable operation command

AND

Enable active

AND

Quick stop option code = 6

AND

No SS1

AND

No SS2

In addition to CiA 402: Transition to Ready to switch on (17), priority: ●

Shutdown command AND Enable active AND quick stop end

OR

Shutdown command AND Enable active AND quick stop duration expired

6.5.2.9 Fault reaction active

- A drive error has occurred
- Fault reaction is executed depending on the respective fault event
- Brakes are activated depending on the respective fault reaction

Transition to Fault (14)

After a completed fault reaction, the drive controller automatically switches to the Fault state.

6.5.2.10 Fault

- A drive error has occurred
- The fault reaction is finished
- The power unit as well as drive and activation functions are locked
- Brakes engage

Transition to Switch on disabled (15)

Fault reset command (rising edge)

After the fault cause is eliminated and the fault message is acknowledged, the drive controller automatically switches to the Switch on disabled state.

6.6 Operating modes in detail

The following sections provide on the operating modes of the CiA 402 application.

Switching between a controller-based operating mode (ip, csp, csv or cst) and a drive-based operating mode (pp, vl, pv, pt) is only possible at a standstill (I199 = 1: Active), i.e. the absolute value of the actual velocity A553 is less than the velocity window C40.

6.6.1 Interpolated position mode (ip)

Interpolated position mode lets you implement a cyclical position specification through the controller. Position control takes place in the drive. The position control receives a set position (with time stamp) and, if applicable, a set velocity processed as a feedforward control. The application interpolates the set values and passes them to the position control. Acceleration and braking ramps or jerk limits are not taken into account.

Information

Note that it is not possible to specify a set of set values for interpolated position mode, just an individual set value.

6.6.1.1 Input and output signals

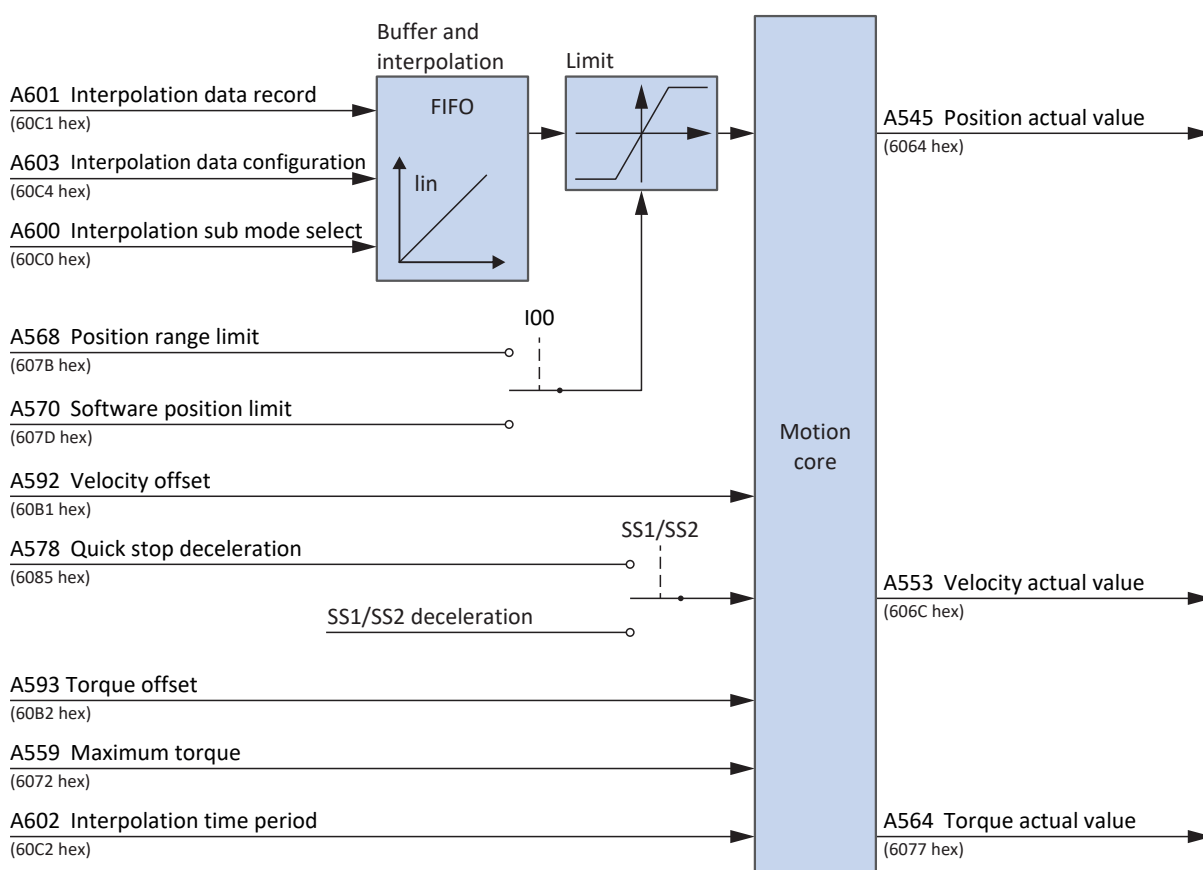


Fig. 13: Interpolated position mode: Input and output signals

6.6.1.2 Control commands and status information

For the interpolated position operating mode, parameter A541 must be set to 7. The active operating mode is output in the parameter A542. If the interpolated position operating mode is active, the information 7: Interpolated position mode is there.

In control word A515, the following operating mode-specific bit is assigned:

Bit	Designation	Comment
4	Enable interpolation	Enable interpolation: 0 = inactive; 1 = active

Tab. 11: Interpolated position mode: Operating mode-specific bits in the control word

The following operating mode-specific bit is assigned in status word A516:

Bit	Designation	Comment
12	IP mode active	Interpolation and transmission of cyclic set position active (source: A927): 0 = inactive; 1 = active

Tab. 12: Interpolated position mode: Operating mode-specific bits in the status word

6.6.1.3 Process data mapping

Interpolated position mode cannot be used with the standard mapping of the CiA 402 application.

The following tables show the operating mode-specific mapping for the receive process data. For fieldbus communication via EtherCAT, the process data mapping for axis A is done via parameter A225 and for axis B via A226. If necessary, the mapping can be customized.

Byte	Data type	Name	Parameter
0 – 1	WORD	Control word	1.A515 Communication object 6040 hex in accordance with CiA 402
2	INT8	Modes of operation	1.A541 Communication object 6060 hex in accordance with CiA 402
3 – 6	INT32	Interpolation data record	1.A601 Communication object 60C1 hex in accordance with CiA 402

Tab. 13: SC6, SI6: Receive process data (Interpolated position mode); first part, axis A

Byte	Data type	Name	Parameter
0 – 1	WORD	Control word	2.A515 Communication object 6840 hex in accordance with CiA 402
2	INT8	Modes of operation	2.A541 Communication object 6860 hex in accordance with CiA 402
3 – 6	INT32	Interpolation data record	2.A601 Communication object 68C1 hex in accordance with CiA 402

Tab. 14: SC6, SI6: Receive process data (Interpolated position mode); second part, axis B

The following tables show the operating mode-specific mapping for the transmit process data. For fieldbus communication via EtherCAT, the process data mapping for axis A is done via parameter A233 and for axis B via A234. If necessary, the mapping can be customized.

Byte	Data type	Name	Parameter
0 – 1	WORD	Statusword	1.A516 Communication object 6041 hex in accordance with CiA 402
2	INT8	Modes of operation display	1.A542 Communication object 6061 hex in accordance with CiA 402
3 – 6	INT32	Position actual value	1.A545 Communication object 6064 hex in accordance with CiA 402

Tab. 15: SC6, SI6: Transmit process data (Interpolated position mode); first part, axis A

Byte	Data type	Name	Parameter
0 – 1	WORD	Statusword	2.A516 Communication object 6841 hex in accordance with CiA 402
2	INT8	Modes of operation display	2.A542 Communication object 6861 hex in accordance with CiA 402
3 – 6	INT32	Position actual value	2.A545 Communication object 6864 hex in accordance with CiA 402

Tab. 16: SC6, SI6: Transmit process data (Interpolated position mode); second part, axis B

6.6.2 Cyclic synchronous position mode (csp)

Cyclic synchronous position mode lets you implement a cyclical position specification through a controller. Position control takes place in the drive. The position control receives a set position (with time stamp) and, if applicable, a set velocity processed as a feedforward control. The application interpolates the set values and passes them to the position control. Acceleration, braking or jerk limits are not taken into account.

6.6.2.1 Input and output signals

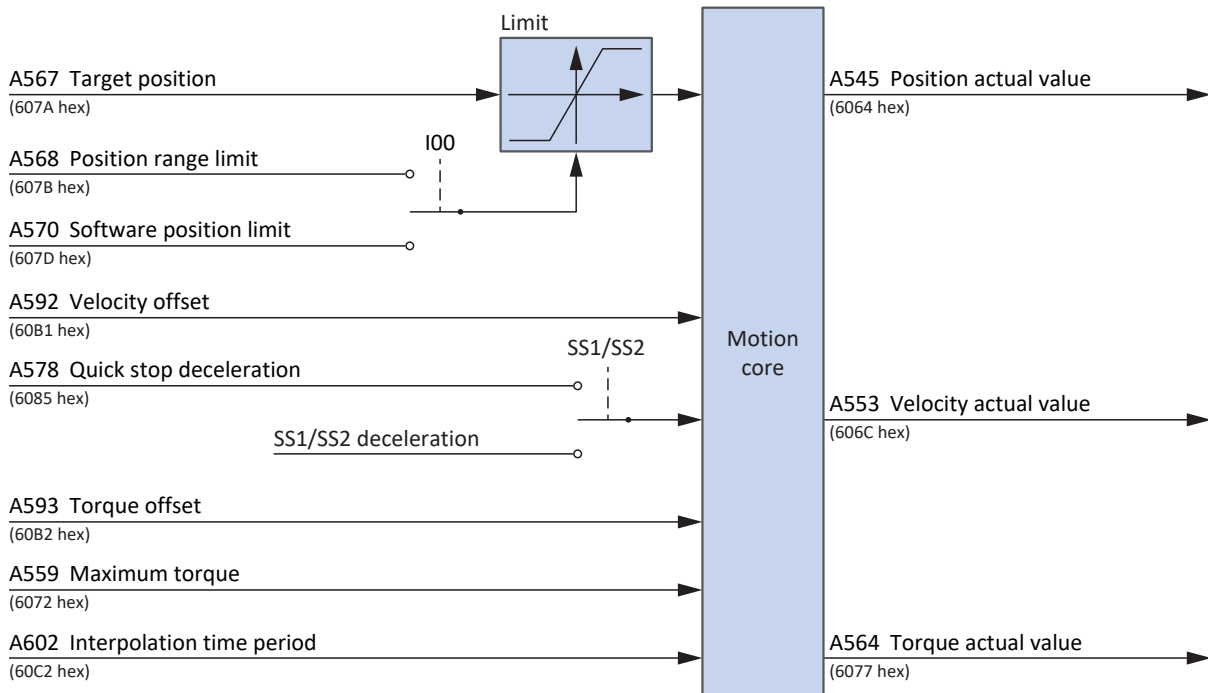


Fig. 14: Cyclic synchronous position mode: Input and output signals

6.6.2.2 Control commands and status information

For Cyclic synchronous position mode, parameter A541 must be set to 8. The active operating mode is output in the parameter A542. If the cyclic synchronous position operating mode is active, the information 8: Cyclic synchronous position mode is there.

No operating mode-specific control commands have to be specified in control word A515.

The following operating mode-specific bits are assigned in status word A516:

Bit	Designation	Comment
12	Target position	Drive follows set position (source: A927): 0 = inactive; 1 = active
13	Following error	Following error (source: A928): 0 = inactive; 1 = active

Tab. 17: Cyclic synchronous position mode: Operating mode-specific bits in the status word

6.6.3 Cyclic synchronous velocity mode (csv)

Cyclic synchronous velocity mode lets you implement a cyclical velocity specification through a controller. Velocity control takes place in the drive. The velocity control receives a set velocity (with time stamp). The application interpolates the set values and passes them to the velocity control. Acceleration, braking or jerk limits are not taken into account.

6.6.3.1 Input and output signals

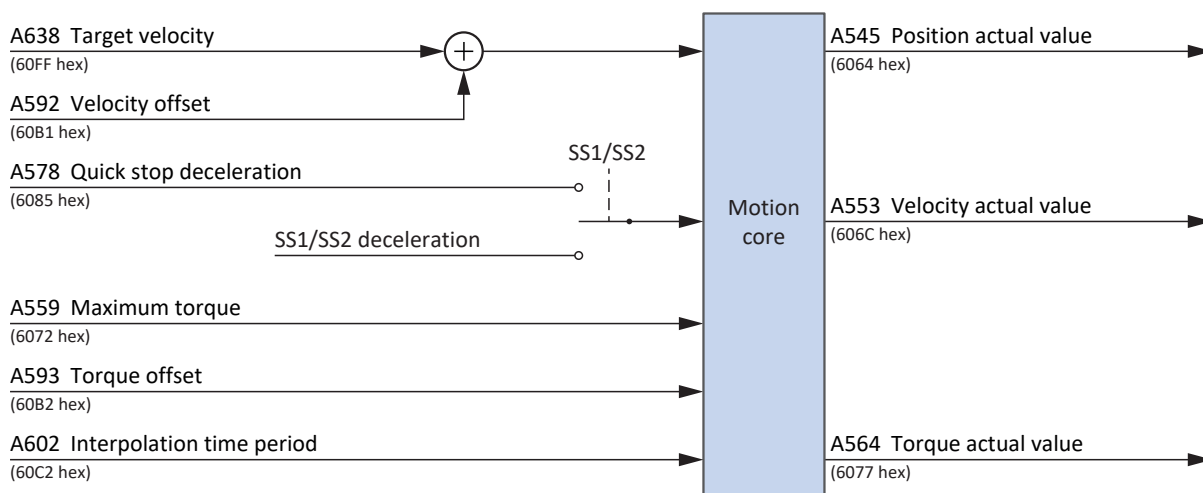


Fig. 15: Cyclic synchronous velocity mode: Input and output signals

6.6.3.2 Control commands and status information

For Cyclic synchronous velocity mode, parameter A541 must be set to 9. The active operating mode is output in the parameter A542. If the cyclic synchronous velocity operating mode is active, the information 9: Cyclic synchronous velocity mode is there.

No operating mode-specific control commands have to be specified in control word A515.

The following operating mode-specific bits are assigned in status word A516:

Bit	Designation	Comment
12	Target velocity	Drive follows set velocity (source: A927): 0 = inactive; 1 = active

Tab. 18: Cyclic synchronous velocity mode: Operating mode-specific bits in the status word

6.6.4 Cyclic synchronous torque mode (cst)

Cyclic synchronous torque mode lets you implement a cyclical torque/force specification through a controller. Torque/force control takes place in the drive. The control receives a set torque or set force (with time stamp). The application interpolates the set values and passes them to the torque/force control. Acceleration, braking or jerk limits are not taken into account. A velocity limit is available.

6.6.4.1 Input and output signals

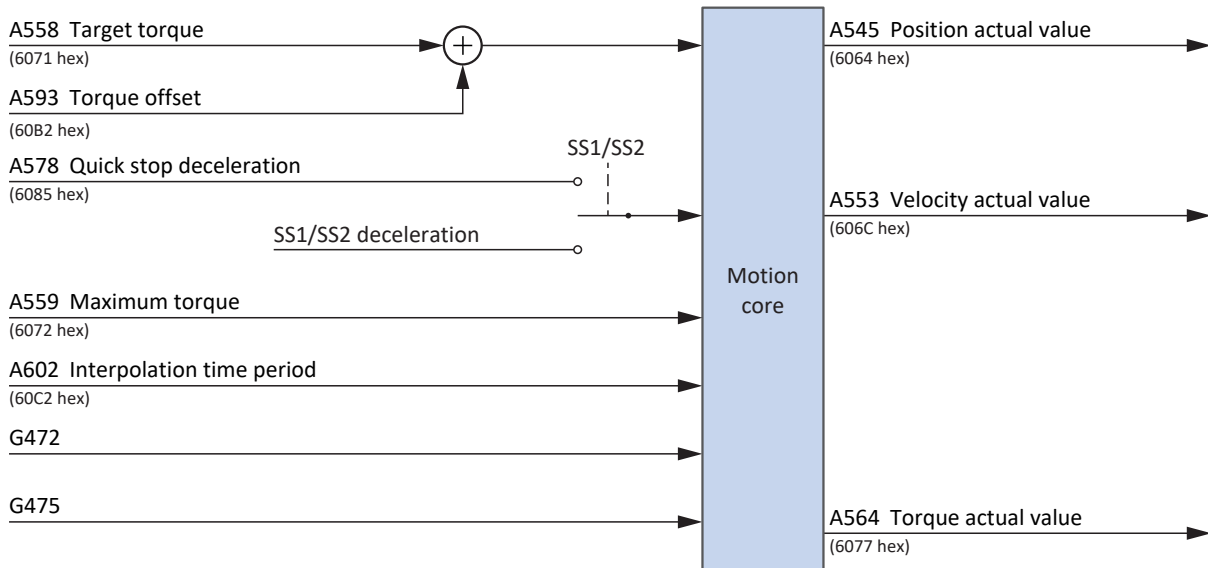


Fig. 16: Cyclic synchronous torque mode: Input and output signals

Information

By default, the velocity limit is active (G472 and G475 = 0%). Increase the values according to the requirements of your application to enable motion.

6.6.4.2 Control commands and status information

For Cyclic synchronous torque mode, parameter A541 must be set to 10. The active operating mode is output in the parameter A542. If the cyclic synchronous torque operating mode is active, the information 10: Cyclic synchronous torque mode is there.

No operating mode-specific control commands have to be specified in control word A515.

The following operating mode-specific bits are assigned in status word A516:

Bit	Designation	Comment
12	Target torque	Drive follows set force/torque (source: A927): 0 = inactive; 1 = active

Tab. 19: Cyclic synchronous torque mode: Operating mode-specific bits in the status word

6.6.5 Profile position mode (pp)

Profile position mode lets you implement a target position specification through a controller. The calculation of the motion profile and position control takes place in the drive. The drive receives a target position, set velocity, set acceleration, set deceleration and set jerk. The application uses this to calculate the motion profile and passes it to the position control.

6.6.5.1 Input and output signals

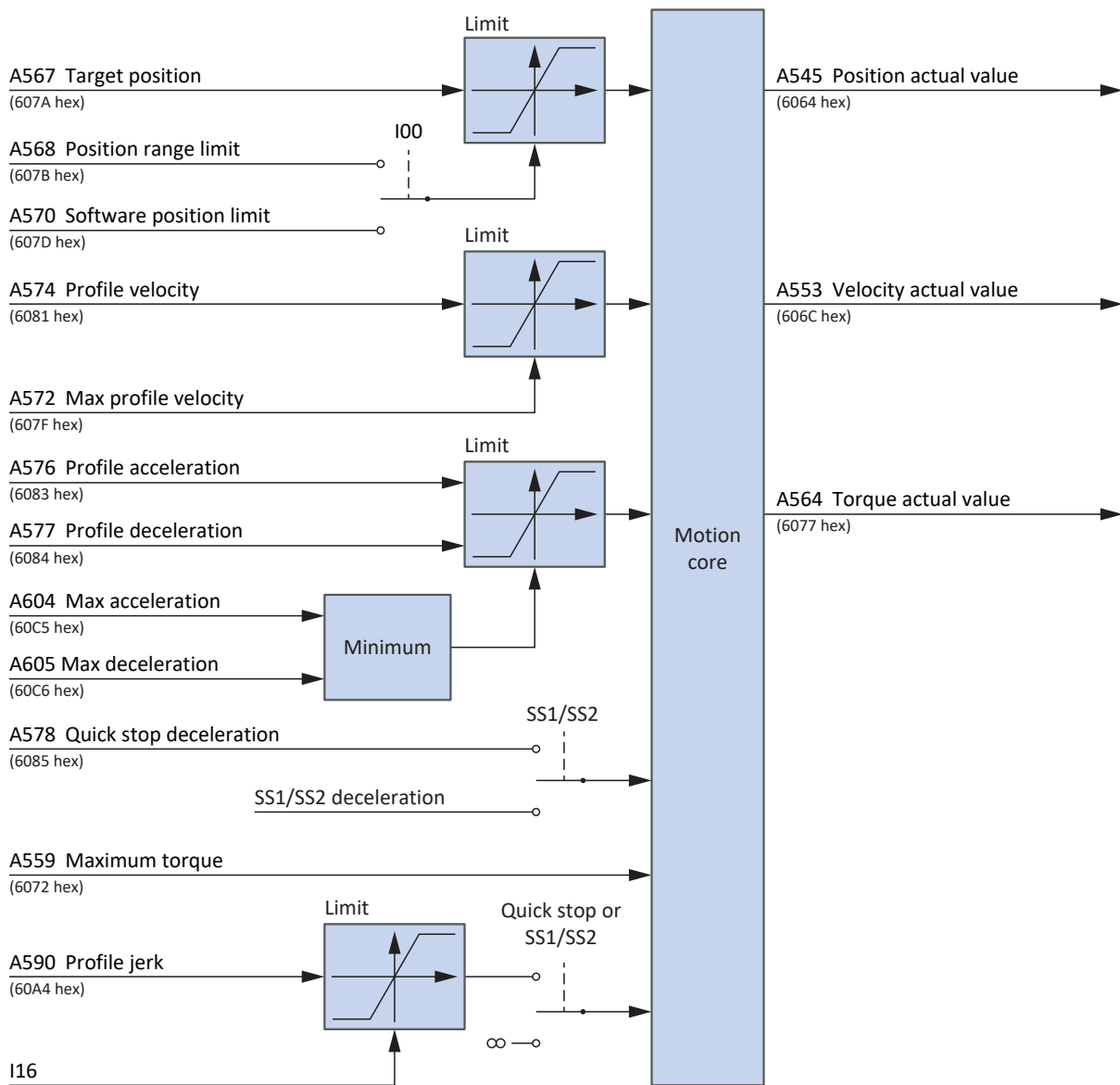


Fig. 17: Profile position mode: Input and output signals

Information

A velocity override via A340 is taken into account.

6.6.5.2 Control commands and status information

For Profile position mode, parameter A541 must be set to 1. The active operating mode is output in the parameter A542. If the profile position operating mode is active, the information 1: Profile position mode is there.

In control word A515, the following bits are used specific to the operating mode:

Bit	Designation	Comment
4	New setpoint	Accept new set position and start motion: 0, 1 = inactive; 0 → 1 = depending on bit 5, accept set position and start motion
5	Change set immediately	Accept new set position with active motion 0 = accept set position with bit 4 only at standstill and start motion (prerequisite: I199 = 1: Active); 1 = accept set position with bit 4 immediately and start motion; active motion is interrupted
6	Abs/rel	Absolute set position/relative distance: 0 = set position is an absolute position; 1 = set position is a distance relative to the last set position A567 (prerequisite: A621 = X...X00 bin) or 1 = set position is a distance relative to the current actual position A545 (prerequisite: A621 = X...X10 bin)
8	Stop	Execute stop: 0 = inactive; 1 = stop axis with deceleration of the motion profile until standstill (deceleration: A577; standstill: I199 = 1: Active) 0 → 1 = motion is not continued

Tab. 20: Profile position mode: Operating mode-specific bits in the control word

The following operating mode-specific bits are assigned in status word A516:

Bit	Designation	Comment
10	Target reached	Target position reached (source: A925): 0 = inactive; 1 = active
12	Setpoint acknowledge	Target position accepted by motion core (source: A927): 0 = inactive; 1 = active
13	Following error	Following error (source: A928): 0 = inactive; 1 = active

Tab. 21: Profile position mode: Operating mode-specific bits in the status word

6.6.6 Velocity mode (vl)

Velocity mode lets you implement a set velocity specification through a controller. The velocity ramp calculation and velocity control take place in the drive. The drive receives a set velocity, set acceleration and set deceleration. The application uses this to calculate the motion profile and passes it to the velocity control.

6.6.6.1 Input and output signals

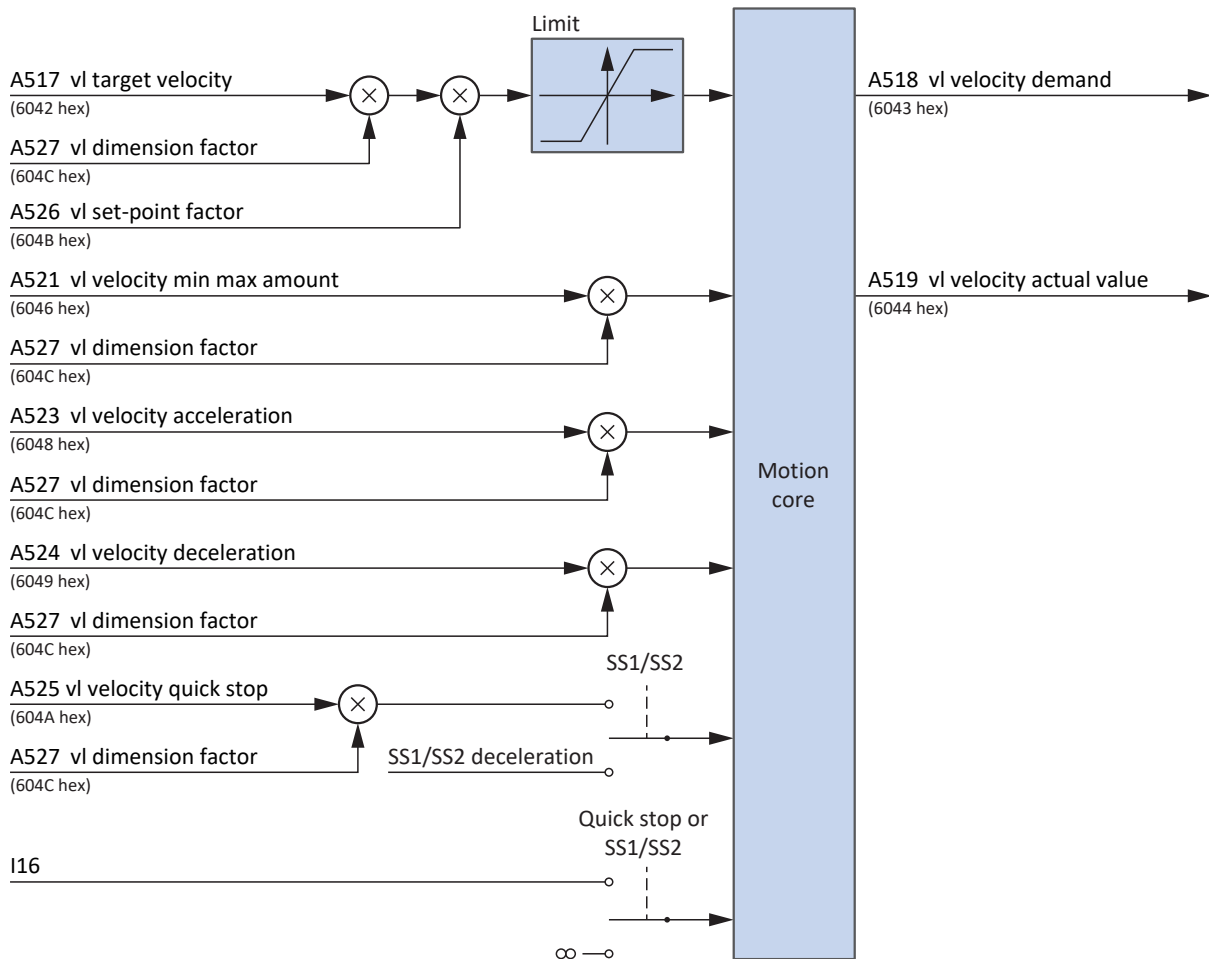


Fig. 18: Velocity mode: Input and output signals

Information

A velocity override via A340 is taken into account.

6.6.6.2 Control commands and status information

For Velocity mode, parameter A541 must be set to 2. The active operating mode is output in the parameter A542. If Velocity mode is active, the information 2: Velocity mode is there.

In control word A515, the following bits are used specific to the operating mode:

Bit	Designation	Comment
4	Enable ramp	Activate velocity control: 0 = velocity specification = 0; 1 = velocity specification by A517
5	Unlock ramp	Unlock locked velocity control: 0 = velocity specification = last velocity specification before freeze; 1 = velocity specification by A517
6	Reference ramp	Deactivate set velocity: 0 = velocity specification = 0; 1 = velocity specification by A517
8	Stop	Execute stop: 0 = inactive; 1 = stop axis with deceleration of the motion profile until standstill (deceleration: A577; standstill: I199 = 1: Active) 0 → 1 = motion is continued immediately

Tab. 22: Velocity mode: Operating mode-specific bits in the control word

No operating mode-specific bits are assigned in status word A516.

6.6.6.3 Process data mapping

Velocity mode cannot be used with the standard mapping of the CiA 402 application.

The following tables show the operating mode-specific mapping for the receive process data. For fieldbus communication via EtherCAT, the process data mapping for axis A is done via parameter A225 and for axis B via A226. If necessary, the mapping can be customized.

Byte	Data type	Name	Parameter
0 – 1	WORD	Control word	1.A515 Communication object 6040 hex in accordance with CiA 402
2	INT8	Modes of operation	1.A541 Communication object 6060 hex in accordance with CiA 402
3 – 4	INT	vl target velocity	1.A517 Communication object 6042 hex in accordance with CiA 402

Tab. 23: SC6, SI6: Receive process data (Velocity mode); first part, axis A

Byte	Data type	Name	Parameter
0 – 1	WORD	Control word	2.A515 Communication object 6840 hex in accordance with CiA 402
2	INT8	Modes of operation	2.A541 Communication object 6860 hex in accordance with CiA 402
3 – 4	INT	vl target velocity	2.A517 Communication object 6842 hex in accordance with CiA 402

Tab. 24: SC6, SI6: Receive process data (Velocity mode); second part, axis B

The following tables show the operating mode-specific mapping for the transmit process data. For fieldbus communication via EtherCAT, the process data mapping for axis A is done via parameter A233 and for axis B via A234. If necessary, the mapping can be customized.

Byte	Data type	Name	Parameter
0 – 1	WORD	Statusword	1.A516 Communication object 6041 hex in accordance with CiA 402
2	INT8	Modes of operation display	1.A542 Communication object 6061 hex in accordance with CiA 402
3 – 4	INT	vl velocity actual value	1.A519 Communication object 6044 hex in accordance with CiA 402

Tab. 25: SC6, SI6: Transmit process data (Velocity mode); first part, axis A

Byte	Data type	Name	Parameter
0 – 1	WORD	Statusword	2.A516 Communication object 6841 hex in accordance with CiA 402
2	INT8	Modes of operation display	2.A542 Communication object 6861 hex in accordance with CiA 402
3 – 4	INT	vl velocity actual value	2.A519 Communication object 6844 hex in accordance with CiA 402

Tab. 26: SC6, SI6: Transmit process data (Velocity mode); second part, axis B

6.6.7 Profile velocity mode (pv)

Profile velocity mode lets you implement a set velocity specification through a controller. The drive calculates the motion profile and performs the velocity control. The drive receives a set velocity, set acceleration, set deceleration and set jerk. The application uses this to calculate the motion profile and passes it to the velocity control.

6.6.7.1 Input and output signals

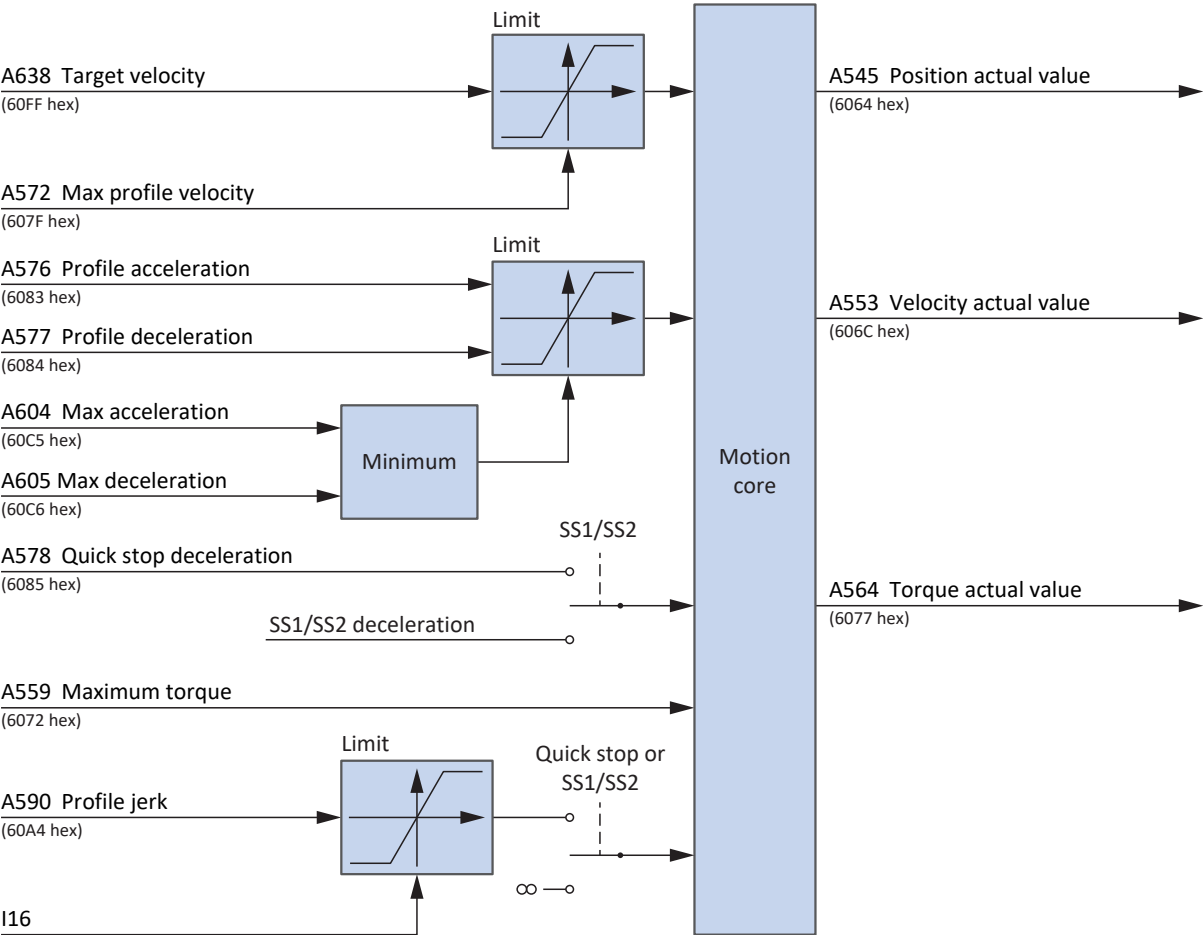


Fig. 19: Profile velocity mode: Input and output signals

Information

A velocity override via A340 is taken into account.

6.6.7.2 Control commands and status information

For Profile velocity mode, parameter A541 must be set to 3. The active operating mode is output in the parameter A542. If the Profile velocity mode is active, the information 3: Profile velocity mode is there.

In control word A515, the following bits are used specific to the operating mode:

Bit	Designation	Comment
8	Stop	Execute stop: 0 = inactive; 1 = stop axis with deceleration of the motion profile until standstill (deceleration: A577; standstill: I199 = 1: Active) 0 → 1 = motion is continued immediately

Tab. 27: Profile velocity mode: Operating mode-specific bits in the control word

The following operating mode-specific bits are assigned in status word A516:

Bit	Designation	Comment
10	Target reached	Set velocity reached (source: A925): 0 = inactive; 1 = active
12	Speed	Velocity is zero (source: A927): 0 = inactive; 1 = active
13	Max slippage error	Velocity error (difference in the amount of the effective set and actual velocities greater than velocity window C40; signal source: A928): 0 = inactive; 1 = active

Tab. 28: Profile velocity mode: Operating mode-specific bits in the status word

6.6.8 Profile torque mode (pt)

Profile torque mode lets you implement a reference set torque/force specification through a controller. The calculation of the motion profile and torque/force control takes place in the drive. The drive receives a set torque or set force. The application uses this to calculate the motion profile and passes it to the torque/force control.

6.6.8.1 Input and output signals

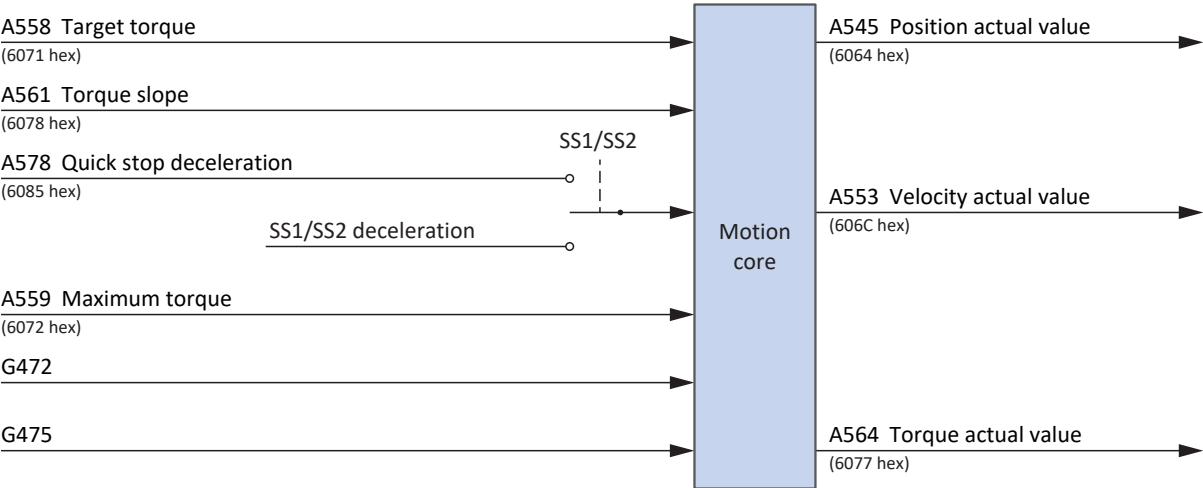


Fig. 20: Profile torque mode: Input and output signals

Information

By default, the velocity limit is active (G472 and G475 = 0%). Increase the values according to the requirements of your application to enable motion.

6.6.8.2 Control commands and status information

For Profile torque mode, parameter A541 must be set to 4. The active operating mode is output in the parameter A542. If the Profile torque mode is active, the information 4: Profile torque mode is there.

In control word A515, the following bits are used specific to the operating mode:

Bit	Designation	Comment
8	Stop	Execute stop: 0 = inactive; 1 = stop axis with deceleration of the motion profile until standstill (deceleration: A577; standstill: I199 = 1: Active) 0 → 1 = motion is continued immediately

Tab. 29: Profile torque mode: Operating mode-specific bits in the control word

The following operating mode-specific bits are assigned in status word A516:

Bit	Designation	Comment
10	Target reached	Set force/set torque reached (difference in the amount of effective set force/set torque and actual force/actual torque greater than torque/force window, source: A925): 0 = inactive; 1 = active

Tab. 30: Profile torque mode: Operating mode-specific bits in the status word

10/2026 | ID 443080.04

6.6.9 Homing mode

During referencing in the homing operating mode, the machine and controller reference system are coordinated together using absolute values. The drive can perform absolute movements, i.e. movements to defined positions, only in the referenced state.

For more detailed information on the individual referencing methods, see [Referencing methods](#) [► 80].

6.6.9.1 Input and output signals

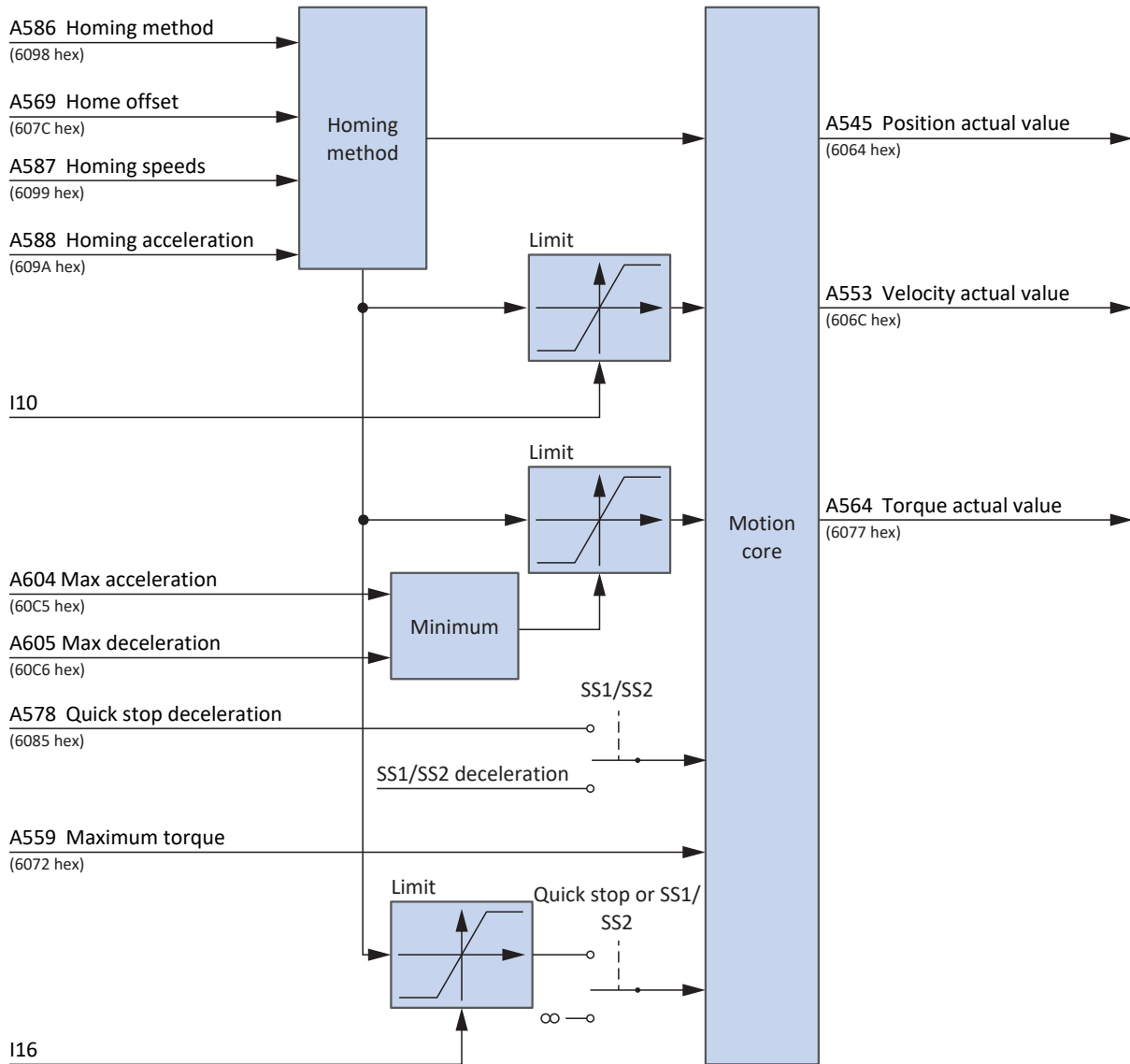


Fig. 21: Homing mode: Input and output signals

6.6.9.2 Control commands and status information

For the homing operating mode, parameter A541 must be set to 6. The active operating mode is output in the parameter A542. If the Homing mode is active, the information 6: Homing Mode is there.

In control word A515, the following bit is assigned specific to the operating mode:

Bit	Designation	Comment
4	Homing operation start	Start referencing: 0, 1 = inactive; 0 → 1 = active

Tab. 31: Homing mode: Operating mode-specific bits in the control word

The following operating mode-specific bits are assigned in status word A516:

Bit	Designation	Comment
10	Target reached	Referencing completed (source: A925): 0 = inactive; 1 = active
12	Homing attained	Reference set (source: A926): 0 = inactive; 1 = active
13	Homing error	Error during referencing (source: A927): 0 = inactive; 1 = active

Tab. 32: Homing mode: Operating mode-specific bits in the status word

Bit 13	Bit 12	Bit 10	Meaning
0	0	0	Referencing active
0	0	1	Referencing canceled or not started
0	1	0	Reference set, but referencing not finished
0	1	1	Referencing completed successfully
1	0	0	Error during referencing, velocity is not zero
1	1	0	Error during referencing, velocity is zero

Tab. 33: Homing mode: Meaning of operating mode-specific bit combinations in the status word

6.6.10 Jog

The STÖBER-specific jog operating mode lets you move the drive during commissioning, in emergency operation or during setup and repair work, including independently of the controller.

You can use jog mode either through the jog control panel or using a controller that applies the manual movement.

6.6.10.1 Input and output signals

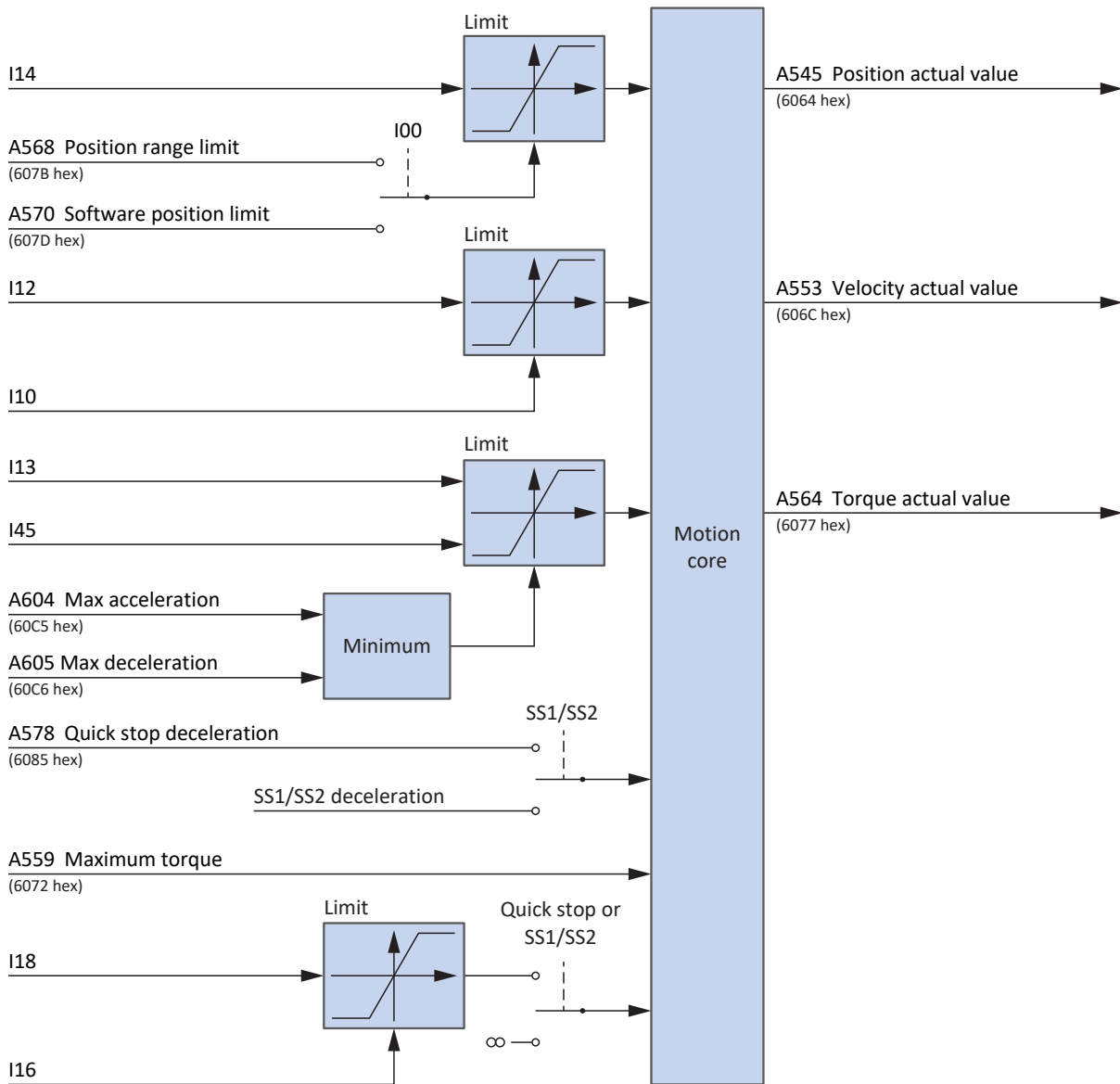


Fig. 22: Jog: Input and output signals

6.6.10.2 Function

If the Jog positive bit is set to 1, the drive accelerates to velocity $I12$ with the ramp $I13$ and the jerk limit $I18$. If the Jog positive bit is set to 0, however, the drive brakes to a standstill with the ramp $I45$. In the event of an active Jog negative signal, the drive accelerate to $-I12$. Acceleration and jerk apply in the same way.

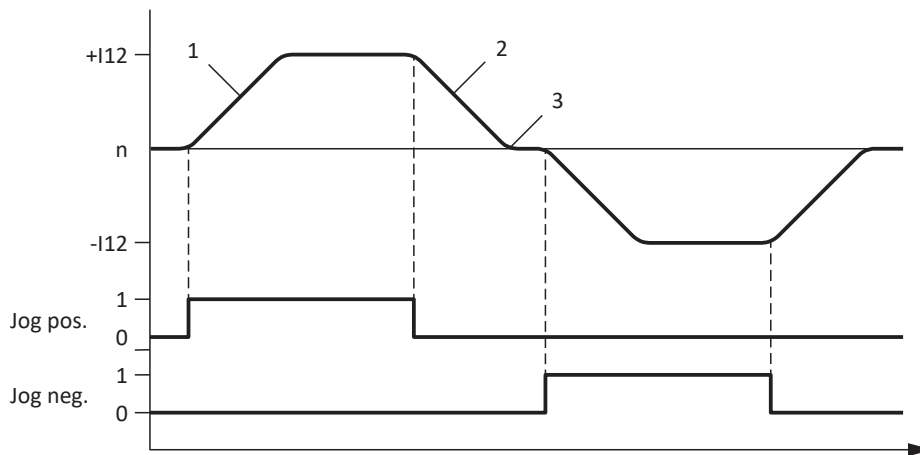


Fig. 23: Jog – Accelerating and braking

- 1 $I13$ Jog acceleration
- 2 $I45$ Jog deceleration
- 3 $I18$ Jog jerk

If the Jog positive and Jog negative bits are set to 1 simultaneously, the drive brakes to a standstill.

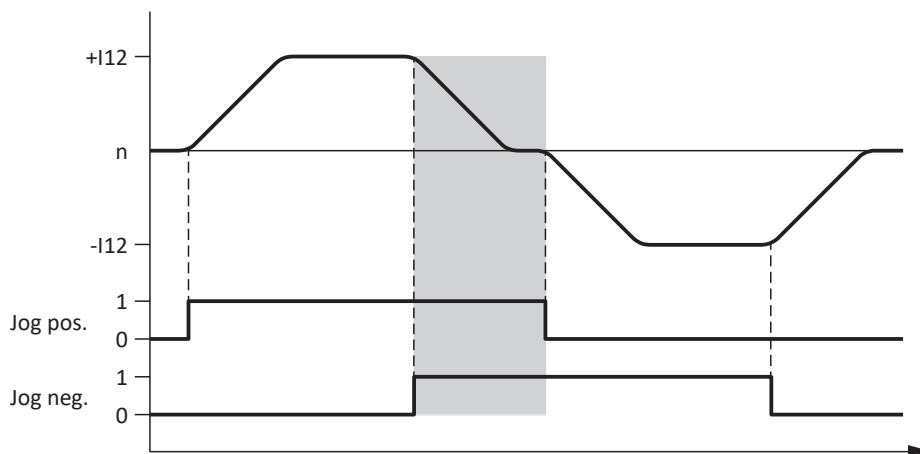


Fig. 24: Jog – Response to contradictory signals

6.6.10.3 Control commands and status information

For the jog operating mode, parameter A541 must be set to -1. The active operating mode is output in the parameter A542. If the jog operating mode is active, the information -1: Jog is there.

In control word A515, the following bits are used specific to the operating mode:

Bit	Designation	Comment
4	Jog positive	Jog in the positive direction: 0 = inactive; 1 = active
5	Jog negative	Jog in the negative direction: 0 = inactive; 1 = active
8	Stop	Execute stop: 0 = inactive; 1 = stop axis with deceleration of the motion profile until standstill (deceleration: A577; standstill: I199 = 1: Active) 0 → 1 = motion is continued immediately

Tab. 34: Jog: Operating mode-specific bits in the control word

No operating mode-specific bits are assigned in status word A516.

6.7 Operating modes in accordance with CiA 402 – Commands

The individual operating modes feature special motion commands that are based on the PLCopen standard and are supplemented by 4 manufacturer-specific motion commands (MC_DoNothing, MC_MoveSpeed, MC_CyclicSpeed and MC_CyclicPosition). Any motion command – except MC_Stop – can be overwritten during execution.

In order to execute a command, the drive controller must not be in the Fault or Quick stop active device state.

Operating mode in accordance with CiA 402 (A541)	Additional settings	Command	Feature	Control type	Necessary motion variables
-1: Jog	—	—	Controller-dependent movement of the drive	Position or velocity	Alternative parameters and motion variables
0: No mode	—	0: MC_DoNothing	—	—	—
1: Profile position mode	A515, bit 6 = 0	1: MC_MoveAbsolute	Move to an absolute position	Position	- Position - Velocity, override - Acceleration - Deceleration - Jerk
	A515, bit 6 = 1 A621, bit 0 = 0 and bit 1 = 1	2: MC_MoveRelative	Move a distance relative to the actual position	Position	- Position - Velocity, override - Acceleration - Deceleration - Jerk
	A515, bit 6 = 1 A621, bit 0 = 0 and bit 1 = 0	3: MC_MoveAdditive	Move a distance relative to the set position	Position	- Position - Velocity, override - Acceleration - Deceleration - Jerk

Operating mode in accordance with CiA 402 (A541)	Additional settings	Command	Feature	Control type	Necessary motion variables
2: Velocity mode	A515, bits 4, 5 and 6	8: MC_MoveSpeed	Move without end with set velocity	Velocity	▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Maximum velocity ▪ Minimum velocity
3: Profile velocity mode	—	8: MC_MoveSpeed	Move without end with set velocity	Velocity	▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk
6: Homing mode	—	6: MC_Home	Reference the axis	Depending on selected homing method	▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk ▪ Torque/force
4: Profile torque mode ^{a)}	—	9: MC_TorqueControl	Move infinitely at set torque/force	Torque/force	▪ Torque/force ▪ Torque/force ramps
7: Interpolated position mode ^{a)}	—	10: MC_CyclicPosition	Move to the next position value without calculating ramps	Position	Position
8: Cyclic synchronous position mode ^{a)}	—	10: MC_CyclicPosition	Move to the next position value without calculating ramps	Position	Position
9: Cyclic synchronous velocity mode ^{a)}	—	7: MC_CyclicSpeed	Move without end with set velocity without calculating ramps	Velocity	Velocity

Operating mode in accordance with CiA 402 (A541)	Additional settings	Command	Feature	Control type	Necessary motion variables
10: Cyclic synchronous torque mode ^{a)}	—	9: MC_TorqueControl	Move without end with set torque/force without calculating ramps	Torque/force	Torque/force

Tab. 35: Comparison of the operating modes in accordance with CiA 402 and the commands based on PLCopen

a) Can only be used in combination with cyclic set values

For detailed information on the motion commands, refer to the associated manual (see [Detailed information](#) [► 165]).

6.8 Feedforward control

The feedforward control lessens the work for the position controller and reduces slip error.

A feedforward control lets you run the following operating modes:

- Interpolated position mode
- Cyclic synchronous position mode

You can select various settings:

- With controller-generated (external) feedforward control
- With drive-generated (internal) feedforward control
- Without feedforward control

Preferable to use movement with feedforward control.

Information

In the case of multi-axis motion, a feedforward control of nearly 100% is necessary for the most accurate trajectory.

If your controller calculates set velocities, you should use it for feedforward control. If your controller only calculates set positions, you can fall back on the drive-led feedforward control. You may have to determine the optimal feedforward control value for your system using an iterative process.

Always refer to the manual for the drive controller for optimization of the regulating behavior (see [Detailed information \[p. 165\]](#)).

6.8.1 Controller-generated external feedforward control

You use controller-generated feedforward control if the feedforward control is to be changed during operation in order to achieve movement optimization.

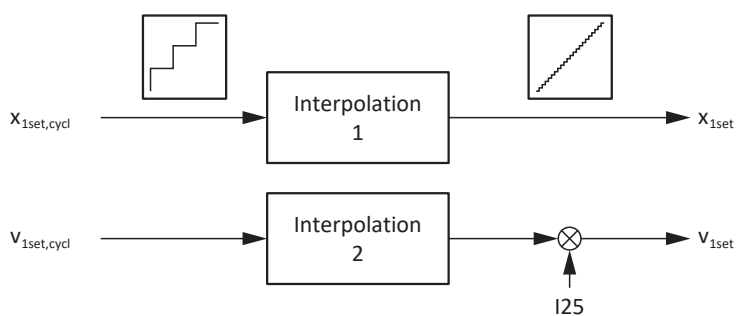


Fig. 25: Controller-generated external feedforward control

6.8.2 Drive-generated internal feedforward control

Use drive-generated internal feedforward control if the feedforward control set once delivers the desired results for all operation.

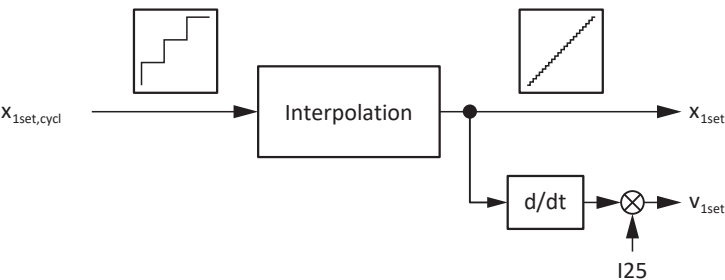


Fig. 26: Drive-generated internal feedforward control

6.8.3 Without feedforward control

Select this setting if the overall setting of the drive controller is optimized to the extent that active feedforward control leads to worse overall regulation.

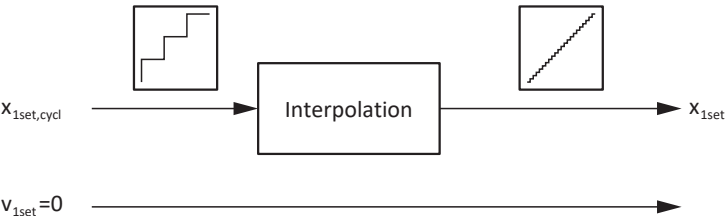


Fig. 27: Without feedforward control

6.8.4 Setting up feedforward control in DS6

The following chapters cover the setup of feedforward control with the aid of the DriveControlSuite commissioning software.

6.8.4.1 Setting up controller-generated external feedforward control

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Control cascade wizard > Position controller.
 3. I25 Speed feedforward:
Enter a suitable value for your application, e.g. 80%.
 4. Select the Motion core wizard > Fine interpolator.
 5. I425 Speed feed forward - interpolation:
Select 1: External - linear $v+x$ or 3: External - quadratic x .
- ⇒ The controller transfers the values A601 Interpolation data record and A592 Velocity offset.

6.8.4.2 Setting up drive-generated internal feedforward control

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Control cascade wizard > Position controller.
 3. I25 Speed feedforward:
Enter a suitable value for your application, e.g. 80%.
 4. Select the Motion core wizard > Fine interpolator.
 5. I425 Speed feed forward - interpolation:
Select 0: Internal - linear $v+x$, 4: Internal - quadratic x or 5: Internal - cubic x .
- ⇒ The controller transfers the value A601 Interpolation data record.

6.8.4.3 Without internal feedforward control

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Control cascade wizard > Position controller.
 3. I25 Speed feedforward:
Enter a value of 0.
- ⇒ The controller only transfers the value A601 Interpolation data record.

6.9 Interpolation

The interpolation is relevant to the following operating modes:

- Interpolated position mode
- Cyclic synchronous position mode
- Cyclic synchronous velocity mode

For interpolation, each bus cycle has to return new set values. A602 Interpolation time period is used to define the interval for this. The value in A602 is transferred to A291 PLC Cycle time. Note that A291 must be equal to or a multiple of A150 Cycle time. A291, A150 and the cycle time of the interpolation routine are used to compute how many single steps a set value difference has to be divided into. The interpolation routine changes the set value for the internal control every cycle by a single step.

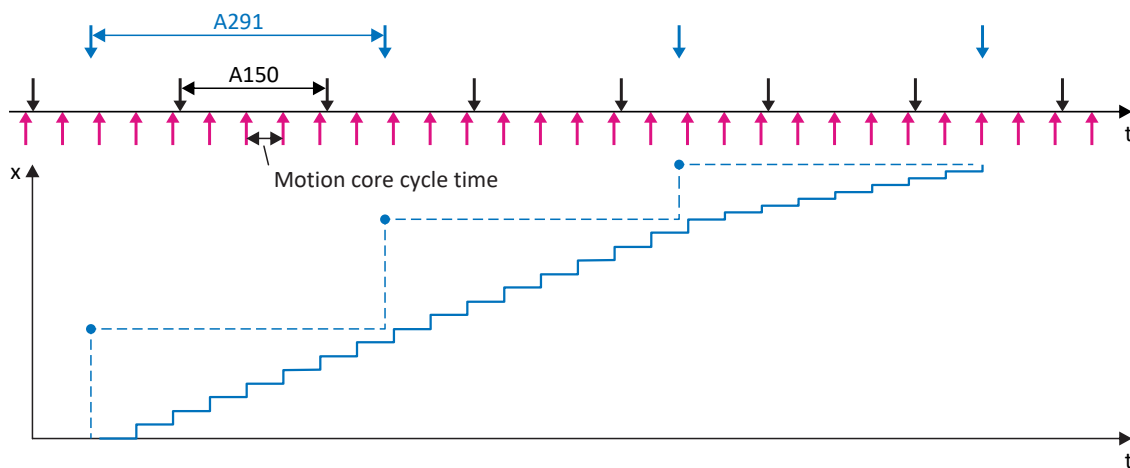


Fig. 28: Interpolation: Ideal behavior over time, no jitter

If a new set value is still not present in the first cycle after the last single step, the set value for the internal control is extrapolated using the previous increment. The extrapolation ends if a new set value is present or if the specified maximum extrapolation period I423 has been exceeded. In the second case, the drive controller changes to the Fault device state. The event 78:Cyclical position limit is shown on the display with cause 3:Overrun of max. extrapolation period I423.

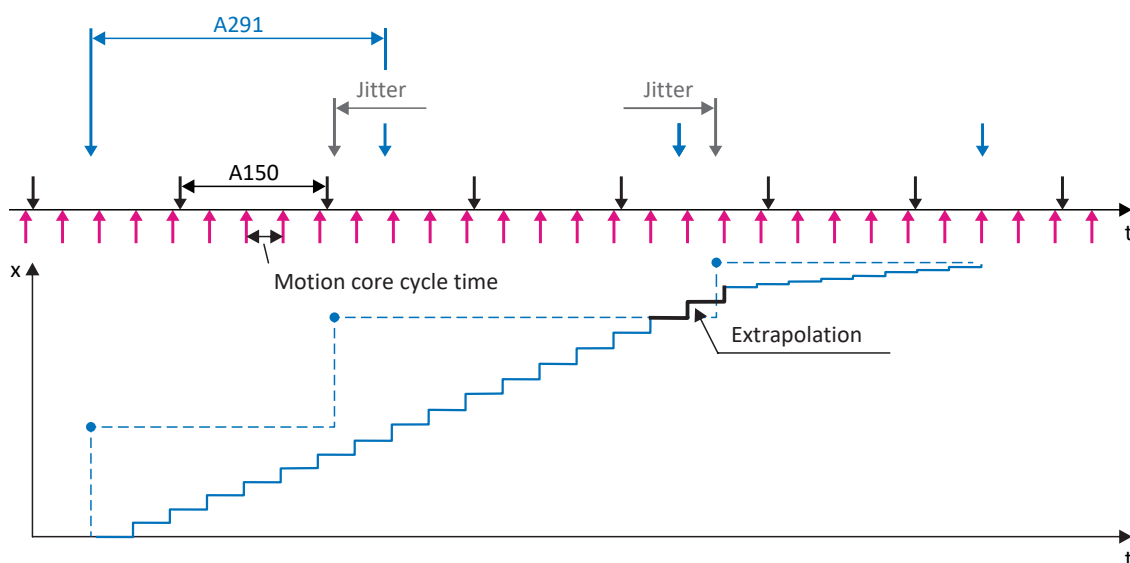


Fig. 29: Interpolation: Controller jitter over time

6.10 Touch probe – Examples

The source of the signal for position measurement 1 is set in I110 (e.g. DI1) and in I126 for position measurement 2.

The touch probe function is explained using position measurement 1, which is activated using bit 0 to 7. The function of position measurement 2 is identical and is activated at bit 8 to 15 in parameter A594. The function of position measurement 1 is shown in the following paragraphs using some examples.

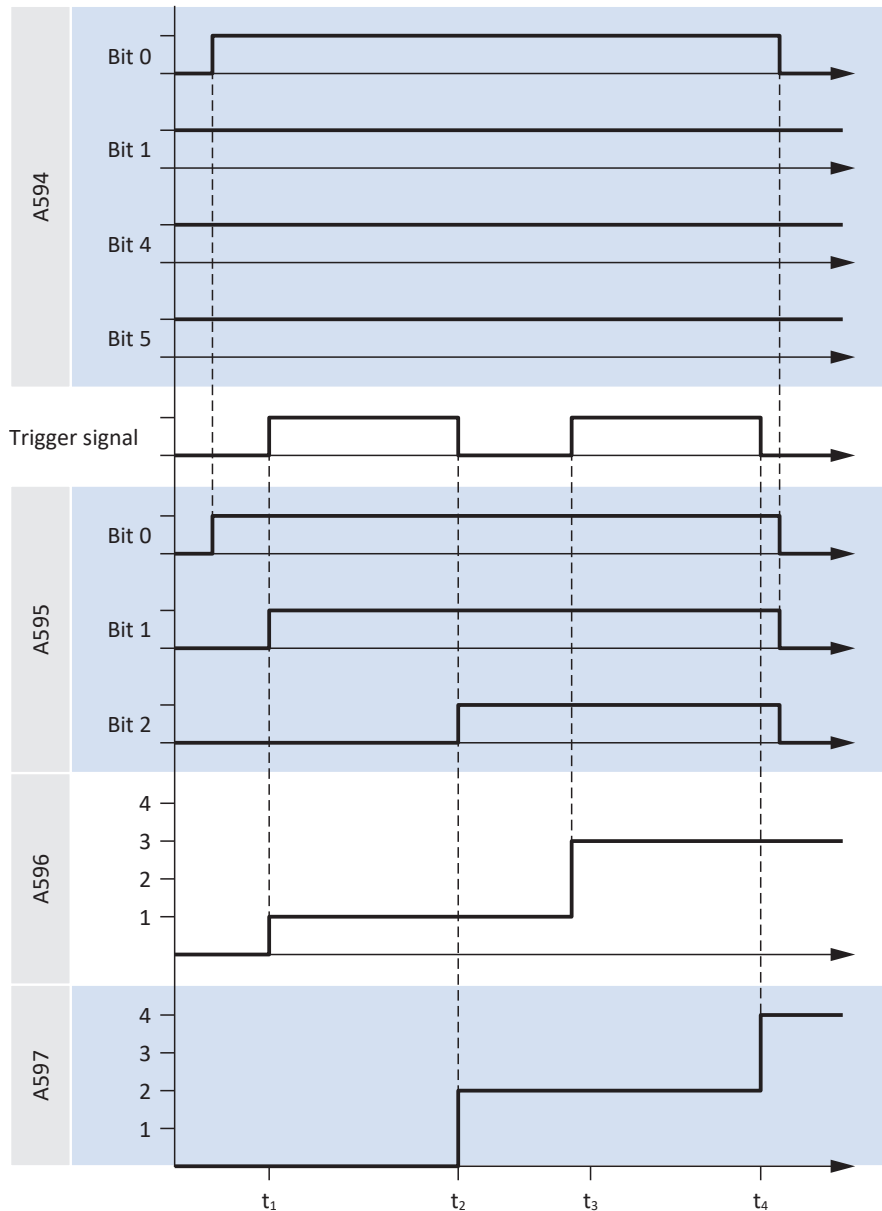


Fig. 30: Touch probe: Example 1

If position measurement 1 is switched on (A594 bit 0 = 1), feedback takes place over A595 (bit 0 = 1). The function is configured to trigger at each event (A594 bit 1 = 1). Triggering at the rising edge of the trigger signal and the falling edge are both switched on (A594 bit 4 = 1 and bit 5 = 1). Accordingly, the position in A596 (rising edge) and A597 (falling edge) are written for each edge of the trigger signal. The initial write processes in each case are shown in A595. Bit 1 = 1 means that triggering occurred at a rising edge; bit 2 = 1 means that triggering occurred at a falling edge. If the position measurement 1 is switched off (A594 bit 0 = 0), all status bits are reset in A595. The positions in A596 and A597 remain intact.

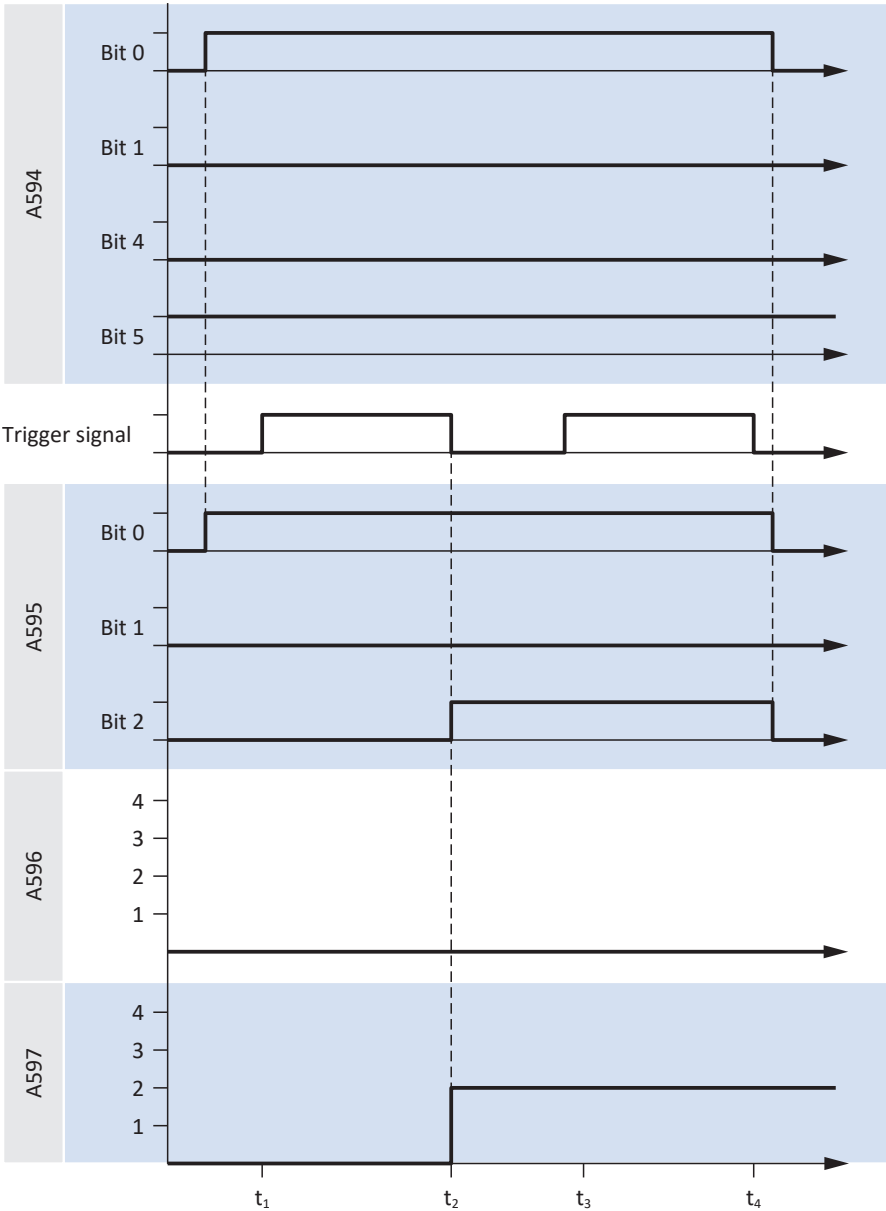


Fig. 31: Touch probe: Example 2

Unlike the first example, Example 2 is triggered only at the first event (bit 1 = 0). Triggering at the rising edge of the trigger signal is deactivated and switched on at the falling edge (bit 4 = 0 and bit 5 = 1). Accordingly, the position in A597 (falling edge) is written for the first falling edge of the trigger signal (t_2).

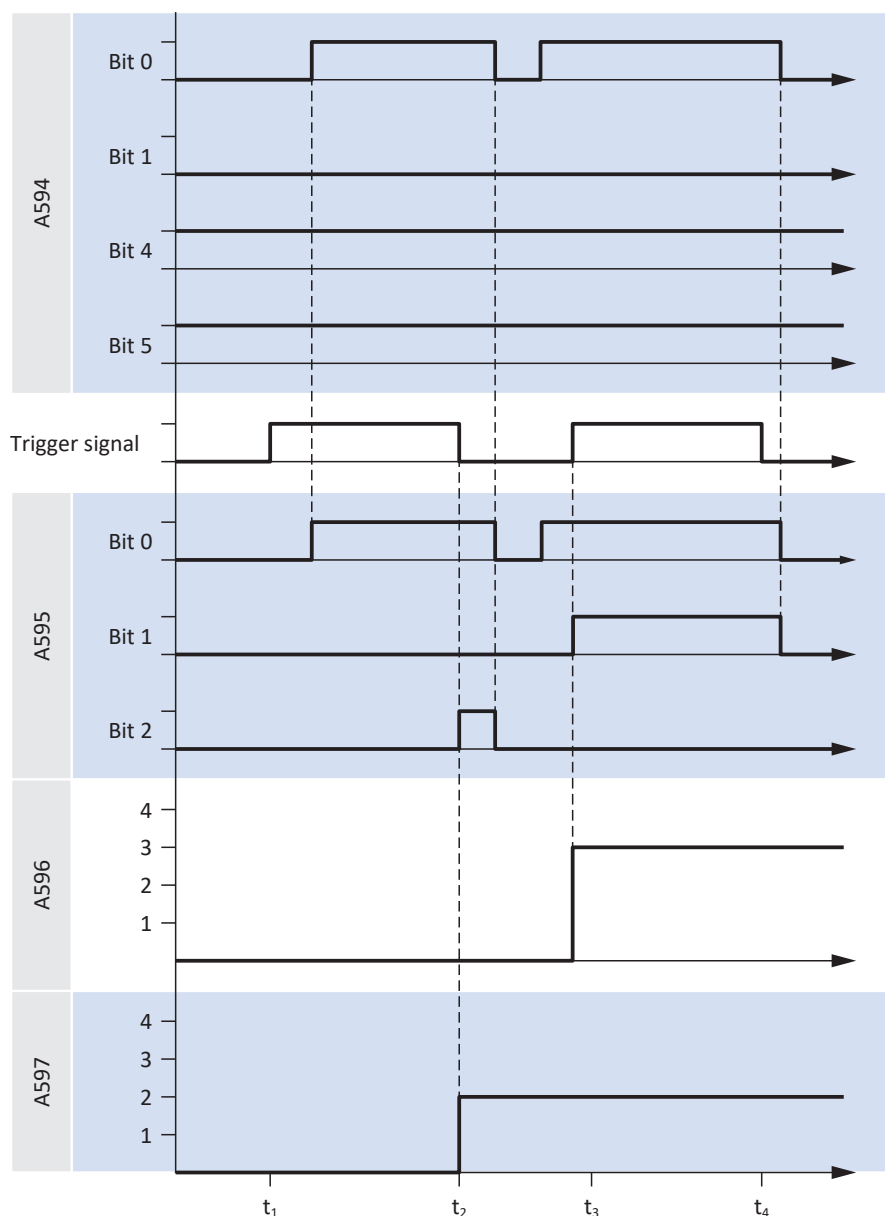


Fig. 32: Touch probe: Example 3

In Example 3, the position measurement 1 is switched on twice in the displayed time frame (bit 0 = 1), and it is triggered at the first event in each case (bit 1 = 0). Triggering at the rising and falling edge of the trigger signal is switched on (bit 4 = 1 and bit 5 = 1). Accordingly, the position is written in A597 (falling edge) for the first falling edge of the trigger signal (t_2) and in A596 for the second rising edge (t_3).

6.11 Cycle times

Possible cycle times can be found in the following table.

Type	Cycle times	Relevant parameters
Application	250 µs, 500 µs, 1 ms, 2 ms, 4 ms, 8 ms	Adjustable in A150
Digital inputs	250 µs, 500 µs, 1 ms, 2 ms, 4 ms, 8 ms	Adjustable in A150 ³

Tab. 36: Cycle times

6.12 Lag monitoring

In CiA 402 type applications, you can monitor the lag of the axis with active position control in order to detect increasing position deviations at an early stage. Monitoring the lag allows you to react quickly before material damage can occur, e.g. in the event of stiffness or a mechanical blockage of the output.

To monitor the lag, the difference between the actual position of the axis and the set position x_{2_set} of the control is determined and compared with the maximum permitted lag (actual position: A545; set position: I96; permitted lag: A546; result: I187). If the permitted lag is exceeded, event 54: Following error is triggered with the corresponding protection level (protection level: U22).

³Partially with timestamp correction with a resolution of 1/64 µs

7 Appendix

7.1 EtherCAT and CiA 402 standard mapping

7.1.1 SC6, SI6: CiA 402 receive process data

The following tables show the standard mapping for the receive process data in the CiA 402 application. For fieldbus communication via EtherCAT, the process data mapping for axis A is done via parameter A225 and for axis B via A226. If necessary, the standard mapping can be customized.

Information

The standard mapping cannot be used for Interpolated position mode and Velocity mode. For more information on process data mapping for these two operating modes, refer to the descriptions of the respective operating mode (see [Operating modes in detail](#) [► 118]).

Byte	Data type	Name	Parameter
0 – 1	WORD	Control word	1.A515 Communication object 6040 hex in accordance with CiA 402
2	INT8	Modes of operation	1. A541 Communication object 6060 hex in accordance with CiA 402
3 – 6	INT32	Target position	1. A567 Communication object 607A hex in accordance with CiA 402
7 – 10	INT32	Target velocity	1. A638 Communication object 60FF hex in accordance with CiA 402
11 – 14	INT32	Velocity offset	1. A592 Communication object 60B1 hex in accordance with CiA 402
15 – 16	INT16	Torque offset	1. A593 Communication object 60B2 hex in accordance with CiA 402
17 – 18	INT16	Target torque	1. A558 Communication object 6071 hex in accordance with CiA 402
19 – 22	DWORD	Digital outputs	1. A637 Communication object 60FE hex in accordance with CiA 402

Tab. 37: SC6, SI6: receive process data (standard mapping); first part, axis A

Byte	Data type	Name	Parameter
0 – 1	WORD	Control word	2.A515 Communication object 6840 hex in accordance with CiA 402
2	INT8	Modes of operation	2. A541 Communication object 6860 hex in accordance with CiA 402
3 – 6	INT32	Target position	2. A567 Communication object 687A hex in accordance with CiA 402
7 – 10	INT32	Target velocity	2. A638 Communication object 68FF hex in accordance with CiA 402
11 – 14	INT32	Velocity offset	2. A592 Communication object 68B1 hex in accordance with CiA 402
15 – 16	INT16	Torque offset	2. A593 Communication object 68B2 hex in accordance with CiA 402
17 – 18	INT16	Target torque	2. A558 Communication object 6871 hex in accordance with CiA 402
19 – 22	DWORD	Digital outputs	2. A637 Communication object 68FE hex in accordance with CiA 402

Tab. 38: SC6, SI6: receive process data (standard mapping); second part, axis B

7.1.2 SC6, SI6: CiA 402 transmit process data

The following tables show the standard mapping for the transmit process data in the CiA 402 application. For fieldbus communication via EtherCAT, the process data mapping for axis A is done via parameter A233 and for axis B via A234. If necessary, the standard mapping can be customized.

Information

The standard mapping cannot be used for Interpolated position mode and Velocity mode. For more information on process data mapping for these two operating modes, refer to the descriptions of the respective operating mode (see [Operating modes in detail](#) [► 118]).

Byte	Data type	Name	Parameter
0 – 1	WORD	Statusword	1.A516 Communication object 6041 hex in accordance with CiA 402
2	INT8	Modes of operation display	1.A542 Communication object 6061 hex in accordance with CiA 402
3 – 6	INT32	Position actual value	1.A545 Communication object 6064 hex in accordance with CiA 402
7 – 10	INT32	Velocity actual value	1.A553 Communication object 606C hex in accordance with CiA 402
11 – 12	INT16	Torque actual value	1.A564 Communication object 6077 hex in accordance with CiA 402
13 – 16	INT32	Following error actual value	1.A632 Communication object 60F4 hex in accordance with CiA 402
17 – 20	DWORD	Digital inputs	1.A636 Communication object 60FD hex in accordance with CiA 402
21 – 24	INT32	Additional position actual value / 1st value	1.A620 Communication object 60E4 hex in accordance with CiA 402
25 – 26	WORD	Status word 2	1.E201
27 – 28	WORD	Status word user-defined	1.A67

Tab. 39: SC6, SI6: transmit process data (standard mapping); first part, axis A

Byte	Data type	Name	Parameter
0 – 1	WORD	Statusword	2.A516 Communication object 6841 hex in accordance with CiA 402
2	INT8	Modes of operation display	2.A542 Communication object 6861 hex in accordance with CiA 402
3 – 6	INT32	Position actual value	2.A545 Communication object 6864 hex in accordance with CiA 402
7 – 10	INT32	Velocity actual value	2.A553 Communication object 686C hex in accordance with CiA 402
11 – 12	INT16	Torque actual value	2.A564 Communication object 6877 hex in accordance with CiA 402
13 – 16	INT32	Following error actual value	2.A632 Communication object 68F4 hex in accordance with CiA 402
17 – 20	DWORD	Digital inputs	2.A636 Communication object 68FD hex in accordance with CiA 402
21 – 24	INT32	Additional position actual value / 1st value	2.A620 Communication object 68E4 hex in accordance with CiA 402
25 – 26	WORD	Status word 2	2.E201
27 – 28	WORD	Status word user-defined	2.A67

Tab. 40: SC6, SI6: transmit process data (standard mapping); second part, axis B

7.2 Supported communication objects

7.2.1 CiA 402 Drives and motion control: 6000 hex – 65FF hex

The following table includes the communication objects of the standardized profile CiA 402 Drives and motion control device profile – Part 2: Operation modes and application data supported by axis A for motion control, as well as how they are mapped to the corresponding parameters of STÖBER.

The communication objects are used in applications CiA 402 and CiA 402 HiRes Motion.

For information on the communication objects supported by axis B, see [CiA 402 Drives and motion control: 6800 hex – 6DFF hex](#) [► 156]. The axes are differentiated by a standard offset of 800 hex.

Index	Subindex	TxPDO	RxPDO	Name	Comment
603F hex	0 hex	✓	—	Error code	A514
6040 hex	0 hex	✓	✓	Control word	A515
6041 hex	0 hex	✓	—	Statusword	A516
6042 hex	0 hex	✓	✓	vl target velocity	A517
6043 hex	0 hex	✓	—	vl velocity demand	A518
6044 hex	0 hex	✓	—	vl velocity actual value	A519
6046 hex				vl velocity min max amount	Array with 2 elements
6046 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6046 hex	1 hex	✓	✓	vl velocity min amount	A521[0]
6046 hex	2 hex	✓	✓	vl velocity max amount	A521[1]
6048 hex				vl velocity acceleration	Structure with 2 elements
6048 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6048 hex	1 hex	✓	✓	Delta Speed	A523[0]
6048 hex	2 hex	✓	✓	Delta Time	A523[1]
6049 hex				vl velocity deceleration	Structure with 2 elements
6049 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6049 hex	1 hex	✓	✓	Delta Speed	A524[0]
6049 hex	2 hex	✓	✓	Delta Time	A524[1]
604A hex				vl velocity quick stop	Structure with 2 elements
604A hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
604A hex	1 hex	✓	✓	Delta Speed	A525[0]
604A hex	2 hex	✓	✓	Delta Time	A525[1]
604B hex				vl set-point factor	Array with 2 elements
604B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
604B hex	1 hex	✓	✓	vl set-point factor numerator	A526[0]
604B hex	2 hex	✓	✓	vl set-point factor denominator	A526[1]
604C hex				vl dimension factor	Array with 2 elements

Index	Subindex	TxPDO	RxPDO	Name	Comment
604C hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
604C hex	1 hex	✓	✓	vl dimension factor numerator	A527[0]
604C hex	2 hex	✓	✓	vl dimension factor denominator	A527[1]
605A hex	0 hex	—	✓	Quick stop option code	A536
605E hex	0 hex	—	✓	Fault reaction option code	A540
6060 hex	0 hex	✓	✓	Modes of operation	A541
6061 hex	0 hex	✓	—	Modes of operation display	A542
6064 hex	0 hex	✓	—	Position actual value	A545
6065 hex	0 hex	✓	✓	Following error window	A546
6066 hex	0 hex	✓	✓	Following error time out	A547
606C hex	0 hex	✓	—	Velocity actual value	A553
6071 hex	0 hex	✓	✓	Target torque	A558
6072 hex	0 hex	✓	✓	Maximum torque	A559
6076 hex	0 hex	✓	—	Rated torque	A563
6077 hex	0 hex	✓	—	Torque actual value	A564
6078 hex	0 hex	✓	—	Current actual value	A565
6079 hex	0 hex	✓	—	DC link circuit voltage	A566
607A hex	0 hex	✓	✓	Target position	A567
607B hex				Position range limit	Record with 2 elements
607B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
607B hex	1 hex	✓	✓	Min. position range limit	A568[0]; no function
607B hex	2 hex	✓	✓	Max. position range limit	A568[1]; used as revolution length
607C hex	0 hex	✓	✓	Home offset	A569
607D hex				Software position limit	Record with 2 elements
607D hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
607D hex	1 hex	✓	✓	Min. position range limit	A570[0]
607D hex	2 hex	✓	✓	Max. position range limit	A570[1]
607E hex	0 hex	—	✓	Polarity	A571; only bit 7 has function
607F hex	0 hex	✓	✓	Max profile velocity	A572
6081 hex	0 hex	✓	✓	Profile velocity	A574
6083 hex	0 hex	✓	✓	Profile acceleration	A576
6084 hex	0 hex	✓	✓	Profile deceleration	A577
6085 hex	0 hex	✓	✓	Quick stop deceleration	A578
6087 hex	0 hex	✓	✓	Torque slope	A561
6091 hex				Gear ratio	Record with 2 elements
6091 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex

Index	Subindex	TxPDO	RxPDO	Name	Comment
6091 hex	1 hex	✓	✓	Motor revolutions	A584[0]
6091 hex	2 hex	✓	✓	Shaft revolutions	A584[1]
6092 hex				Feed constant	Record with 2 elements
6092 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6092 hex	1 hex	✓	✓	Feed	A585[0]
6092 hex	2 hex	✓	✓	Shaft revolutions	A585[1]
6098 hex	0 hex	—	—	Homing method	A586
6099 hex				Homing speeds	Record with 2 elements
6099 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6099 hex	1 hex	✓	✓	Speed during search for switch	A587[0]
6099 hex	2 hex	✓	✓	Speed during search for zero	A587[1]
609A hex	0 hex	✓	✓	Homing acceleration	A588
60A3 hex	0 hex	—	—	Profile jerk use	A589
60A4 hex				Profile jerk	Array with 1 element
60A4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60A4 hex	1 hex	✓	✓	Profile jerk, Profile jerk 1	A590
60B1 hex	0 hex	✓	✓	Velocity offset	A592
60B2 hex	0 hex	✓	✓	Torque offset	A593
60B8 hex	0 hex	✓	✓	Touch probe function	A594
60B9 hex	0 hex	✓	—	Touch probe status	A595
60BA hex	0 hex	✓	—	Touch probe position 1 positive value	A596
60BB hex	0 hex	✓	—	Touch probe position 1 negative value	A597
60BC hex	0 hex	✓	—	Touch probe position 2 positive value	A598
60BD hex	0 hex	✓	—	Touch probe position 2 negative value	A599
60C0 hex	0 hex	—	✓	Interpolation sub mode select	A600
60C1 hex				Interpolation data record	Record with 1 element
60C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60C1 hex	1 hex	✓	✓	1st set-point	A601
60C2 hex				Interpolation time period	Record with 2 elements
60C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
60C2 hex	1 hex	—	✓	Interpolation time period value	A602[0]
60C2 hex	2 hex	—	✓	Interpolation time index	A602[1]

Index	Subindex	TxPDO	RxPDO	Name	Comment
60C4 hex				Interpolation data configuration	Record with 5 elements
60C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5 hex
60C4 hex	1 hex	—	—	Maximum buffer size	A603[0]; no function
60C4 hex	2 hex	—	—	Actual buffer size	A603[1]; no function
60C4 hex	3 hex	—	—	Buffer organisation	A603[2]; no function
60C4 hex	4 hex	—	—	Buffer position	A603[3]; no function
60C4 hex	5 hex	—	—	Size of data record	A603[4]; no function
60C4 hex	6 hex	—	—	Buffer clear	A603[5]; no function
60C5 hex	0 hex	✓	✓	Max acceleration	A604
60C6 hex	0 hex	✓	✓	Max deceleration	A605
60D0 hex				Touch probe source	Array with 3 elements (see also Touch probe source: Values [► 160])
60D0 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
60D0 hex	1 hex	—	—	Touch probe 1 source	I110
60D0 hex	2 hex	—	—	Touch probe 2 source	I126
60E3 hex				Supported homing methods	Record with 20 elements
60E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 14 hex
60E3 hex	1 hex – 14 hex	—	—	1st - 20th supported homing method	A619[0] – A619[19]
60E4 hex				Additional position actual value / 1st value	Record with 1 element
60E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60E4 hex	1 hex	✓	—	1st additional position actual value	A620
60F2 hex	0 hex	—	✓	Positioning option code	A621
60F4 hex	0 hex	✓	—	Following error actual value	A632
60FD hex	0 hex	✓	—	Digital inputs	A636
60FF hex	0 hex	✓	✓	Target velocity	A638
6502 hex	0 hex	—	—	Supported drive modes	

Tab. 41: CiA 402-2 communication objects: 6000 hex – 65FF hex

7.2.2 CiA 402 Drives and motion control: 6800 hex – 6DFF hex

The following table includes the communication objects of the standardized profile CiA 402 Drives and motion control device profile – Part 2: Operation modes and application data supported by axis B for motion control, as well as how they are mapped to the corresponding parameters of STOBER.

The communication objects are used in the applications CiA 402 and CiA 402 HiRes Motion.

Index	Subindex	TxPDO	RxPDO	Name	Comment
683F hex	0 hex	✓	—	Error code	A514
6840 hex	0 hex	✓	✓	Control word	A515
6841 hex	0 hex	✓	—	Statusword	A516
6842 hex	0 hex	✓	✓	vl target velocity	A517
6843 hex	0 hex	✓	—	vl velocity demand	A518
6844 hex	0 hex	✓	—	vl velocity actual value	A519
6846 hex				vl velocity min max amount	Array with 2 elements
6846 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6846 hex	1 hex	✓	✓	vl velocity min amount	A521[0]
6846 hex	2 hex	✓	✓	vl velocity max amount	A521[1]
6848 hex				vl velocity acceleration	Structure with 2 elements
6848 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6848 hex	1 hex	✓	✓	Delta Speed	A523[0]
6848 hex	2 hex	✓	✓	Delta Time	A523[1]
6849 hex				vl velocity deceleration	Structure with 2 elements
6849 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6849 hex	1 hex	✓	✓	Delta Speed	A524[0]
6849 hex	2 hex	✓	✓	Delta Time	A524[1]
684A hex				vl velocity quick stop	Structure with 2 elements
684A hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
684A hex	1 hex	✓	✓	Delta Speed	A525[0]
684A hex	2 hex	✓	✓	Delta Time	A525[1]
684B hex				vl set-point factor	Array with 2 elements
684B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
684B hex	1 hex	✓	✓	vl set-point factor numerator	A526[0]
684B hex	2 hex	✓	✓	vl set-point factor denominator	A526[1]
684C hex				vl dimension factor	Array with 2 elements
684C hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
684C hex	1 hex	✓	✓	vl dimension factor numerator	A527[0]

Index	Subindex	TxPDO	RxPDO	Name	Comment
684C hex	2 hex	✓	✓	vl dimension factor denominator	A527[1]
685A hex	0 hex	—	✓	Quick stop option code	A536
685E hex	0 hex	—	✓	Fault reaction option code	A540
6860 hex	0 hex	✓	✓	Modes of operation	A541
6861 hex	0 hex	✓	—	Modes of operation display	A542
6864 hex	0 hex	✓	—	Position actual value	A545
6865 hex	0 hex	✓	✓	Following error window	A546
6866 hex	0 hex	✓	✓	Following error time out	A547
686C hex	0 hex	✓	—	Velocity actual value	A553
6871 hex	0 hex	✓	✓	Target torque	A558
6872 hex	0 hex	✓	✓	Maximum torque	A559
6876 hex	0 hex	✓	—	Rated torque	A563
6877 hex	0 hex	✓	—	Torque actual value	A564
6878 hex	0 hex	✓	—	Current actual value	A565
6879 hex	0 hex	✓	—	DC link circuit voltage	A566
687A hex	0 hex	✓	✓	Target position	A567
687B hex				Highest subindex supported	Record with 2 elements
687B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
687B hex	1 hex	✓	✓	Min. position range limit	A568[0]; no function
687B hex	2 hex	✓	✓	Max. position range limit	A568[1]; used as revolution length
687C hex	0 hex	✓	✓	Home offset	A569
687D hex				Software position limit	Record with 2 elements
687D hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
687D hex	1 hex	✓	✓	Min. position range limit	A570[0]
687D hex	2 hex	✓	✓	Max. position range limit	A570[1]
687E hex	0 hex	—	✓	Polarity	A571; only bit 7 has function
687F hex	0 hex	✓	✓	Max profile velocity	A572
6881 hex	0 hex	✓	✓	Profile velocity	A574
6883 hex	0 hex	✓	✓	Profile acceleration	A576
6884 hex	0 hex	✓	✓	Profile deceleration	A577
6885 hex	0 hex	✓	✓	Quick stop deceleration	A578
6887 hex	0 hex	✓	✓	Torque slope	A561
6891 hex				Gear ratio	Record with 2 elements
6891 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6891 hex	1 hex	✓	✓	Motor revolutions	A584[0]
6891 hex	2 hex	✓	✓	Shaft revolutions	A584[1]

Index	Subindex	TxPDO	RxPDO	Name	Comment
6892 hex				Feed constant	Record with 2 elements
6892 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6892 hex	1 hex	✓	✓	Feed	A585[0]
6892 hex	2 hex	✓	✓	Shaft revolutions	A585[1]
6898 hex	0 hex	—	—	Homing method	A586
6899 hex				Homing speeds	Record with 2 elements
6899 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6899 hex	1 hex	✓	✓	Speed during search for switch	A587[0]
6899 hex	2 hex	✓	✓	Speed during search for zero	A587[1]
689A hex	0 hex	✓	✓	Homing acceleration	A588
68A3 hex	0 hex	—	—	Profile jerk use	A589
68A4 hex				Profile jerk	Array with 1 element
68A4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
68A4 hex	1 hex	✓	✓	Profile jerk 1	A590
68B1 hex	0 hex	✓	✓	Velocity offset	A592
68B2 hex	0 hex	✓	✓	Torque offset	A593
68B8 hex	0 hex	✓	✓	Touch probe function	A594
68B9 hex	0 hex	✓	—	Touch probe status	A595
68BA hex	0 hex	✓	—	Touch probe position 1 positive value	A596
68BB hex	0 hex	✓	—	Touch probe position 1 negative value	A597
68BC hex	0 hex	✓	—	Touch probe position 2 positive value	A598
68BD hex	0 hex	✓	—	Touch probe position 2 negative value	A599
68C0 hex	0 hex	—	✓	Interpolation sub mode select	A600
68C1 hex				Interpolation data record	Array with 1 element
68C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
68C1 hex	1 hex	✓	✓	1st set-point	A601
68C2 hex				Interpolation time period	Array with 2 elements
68C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
68C2 hex	1 hex	—	✓	Interpolation time period value	A602[0]
68C2 hex	2 hex	—	✓	Interpolation time index	A602[1]
68C4 hex				Interpolation data configuration	Array with 5 elements

Index	Subindex	TxPDO	RxPDO	Name	Comment
68C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5 hex
68C4 hex	1 hex	—	—	Maximum buffer size	A603[0]; no function
68C4 hex	2 hex	—	—	Actual buffer size	A603[1]; no function
68C4 hex	3 hex	—	—	Buffer organisation	A603[2]; no function
68C4 hex	4 hex	—	—	Buffer position	A603[3]; no function
68C4 hex	5 hex	—	—	Size of data record	A603[4]; no function
68C4 hex	6 hex	—	—	Buffer clear	A603[5]; no function
68C5 hex	0 hex	✓	✓	Max acceleration	A604
68C6 hex	0 hex	✓	✓	Max deceleration	A605
68D0 hex				Touch probe source	Array with 3 elements (see also Touch probe source: Values [► 160])
68D0 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
68D0 hex	1 hex	—	—	Touch probe 1 source	I110
68D0 hex	2 hex	—	—	Touch probe 2 source	I126
68E3 hex				Supported homing methods	Record with 20 elements
68E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 14 hex
68E3 hex	1 hex – 14 hex	—	—	1 st - 20 th supported homing method	A619[0] – A619[19]
68E4 hex				Additional position actual value / 1st value	Record with 1 element
68E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
68E4 hex	1 hex	✓	—	1st additional position actual value	A620
68F2 hex	0 hex	—	✓	Positioning option code	A621
68F4 hex	0 hex	✓	—	Following error actual value	A632
68FD hex	0 hex	✓	—	Digital inputs	A636
68FF hex	0 hex	✓	✓	Target velocity	A638
6D02 hex	0 hex	—	—	Supported drive modes	

Tab. 42: CiA 402-2 communication objects: 6800 hex – 6DFF hex

7.2.3 Touch probe source: Values

The following table shows possible values for communication objects 60D0 hex and 68D0 hex in column 1 in accordance with EtherCAT specification ETG.6010. Column 2 contains the values transferred to DriveControlSuite (Touch probe 1 source: I110, Touch probe 2 source: I126).

Value (EtherCAT)	Value (DS6)	Definition
-100	0: Low	Manufacturer-specific
-101	1: High	Manufacturer-specific
1	3: DI1	Digital input 1 (Touch probe input)
-1	4: DI1 inverted	Manufacturer-specific
2	5: DI2	Digital input 2 (Touch probe input)
-2	6: DI2 inverted	Manufacturer-specific
3	7: DI3	Digital input 3 (Touch probe input)
-3	8: DI3 inverted	Manufacturer-specific
4	9: DI4	Digital input 4 (Touch probe input)
-4	10: DI4 inverted	Manufacturer-specific
-105	11: DI5	Manufacturer-specific
-5	12: DI5 inverted	Manufacturer-specific
-106	13: DI6	Manufacturer-specific
-6	14: DI6 inverted	Manufacturer-specific
-107	15: DI7	Manufacturer-specific
-7	16: DI7 inverted	Manufacturer-specific
-108	17: DI8	Manufacturer-specific
-8	18: DI8 inverted	Manufacturer-specific
-109	19: DI9	Manufacturer-specific
-9	20: DI9 inverted	Manufacturer-specific
-110	21: DI10	Manufacturer-specific
-10	22: DI10 inverted	Manufacturer-specific
-111	23: DI11	Manufacturer-specific
-11	24: DI11 inverted	Manufacturer-specific
-112	25: DI12	Manufacturer-specific
-12	26: DI12 inverted	Manufacturer-specific
-113	27: DI13	Manufacturer-specific
-13	28: DI13 inverted	Manufacturer-specific

Tab. 43: Values for 60D0 hex and 68D0 hex

7.2.4 Manufacturer-specific parameters: 2000 hex – 53FF hex

Index, subindex and calculation example (axis A)

Information	
-------------	--

Index and Subindex must be specified in the format required by the controller.

Information	
-------------	--

The calculation described below is only valid for converting the manufacturer-specific parameters.

The index is calculated from the group and line of the parameter according to the following formula:

$$\text{Index} = 8192 + (\text{number of the group} \times 512) + \text{number of the line}$$

The subindex for simple parameters is always 0.

For array or record parameters, the subindex for EtherCAT Rx corresponds to the element number of the parameter.

The subindex for EtherCAT Rx SDO Info corresponds to the element number of the parameter + 1 for array or record parameters.

	Simple parameters	Array or record parameter
Index	$8192 + (\text{number of the group} \times 512) + \text{number of the line}$	
Subindex for EtherCAT Rx	0	Element number
Subindex for EtherCAT Rx SDO Info	0	Element number + 1

Tab. 44: Index and subindex for manufacturer-specific parameters

Calculation example

Calculation for parameter E200[0]:

Number of the group = 4

Number of the line = 200

$$\text{Index} = 8192 + (4 \times 512) + 200 = 10440 = 28C8 \text{ hex}$$

$$\text{Subindex for EtherCAT Rx} = 0 = 0 \text{ hex}$$

$$\text{Subindex for EtherCAT Rx SDO Info} = 1 = 1 \text{ hex}$$

Communication objects (axis A)

The following table includes the manufacturer-specific communication objects of axis A and their mapping to the corresponding parameters of STOBER.

For information on the manufacturer-specific communication objects of axis B, see [Manufacturer-specific parameters: A000 hex – D3FF hex \[► 163\]](#). The axes are differentiated by an offset of 8000 hex.

Index	Group	Number	Parameters
2000 hex – 21FF hex	A: Drive controller	0	A00 – A511
2200 hex – 23FF hex	B: Motor	1	B00 – B511
2400 hex – 25FF hex	C: Machine	2	C00 – C511
2600 hex – 27FF hex	D: Set value	3	D00 – D511
2800 hex – 29FF hex	E: Show	4	E00 – E511
2A00 hex – 2BFF hex	F: Terminals	5	F00 – F511
2C00 hex – 2DFF hex	G: Technology	6	G00 – G511
2E00 hex – 2FFF hex	H: Encoders	7	H00 – H511
3000 hex – 31FF hex	I: Motion	8	I00 – I511
3200 hex – 33FF hex	J: Motion blocks	9	J00 – J511
3400 hex – 35FF hex	K: Control panel	10	K00 – K511
3600 hex – 37FF hex	L: Technology 2	11	L00 – L511
3800 hex – 39FF hex	M: Profile	12	M00 – M511
3E00 hex – 3FFF hex	P: Customer-specific parameters	15	P00 – P511
4000 hex – 41FF hex	Q: Customer-specific parameters, instance-dependent	16	Q00 – Q511
4200 hex – 43FF hex	R: Production data	17	R00 – R511
4400 hex – 45FF hex	S: Safety	18	S00 – S511
4600 hex – 47FF hex	T: Scope	19	T00 – T511
4800 hex – 49FF hex	U: Protection functions	20	U00 – U511
5200 hex – 53FF hex	Z: Fault counter	25	Z00 – Z511

Tab. 45: Manufacturer-specific communication objects: 2000 hex – 53FF hex

7.2.5 Manufacturer-specific parameters: A000 hex – D3FF hex

Index, subindex, and calculation example (axis B)

Information	
-------------	--

Index and Subindex must be specified in the format required by the controller.

Information	
-------------	--

The calculation described below is only valid for converting the manufacturer-specific parameters.

The index is calculated from the group and line of the parameter according to the following formula:
Index = 40960 + (number of the group × 512) + number of the line

The subindex for simple parameters is always 0.

For array or record parameters, the subindex for EtherCAT Rx corresponds to the element number of the parameter.

The subindex for EtherCAT Rx SDO Info corresponds to the element number of the parameter + 1 for array or record parameters.

	Simple parameters	Array or record parameter
Index	40960 + (number of the group × 512) + number of the line	
Subindex for EtherCAT Rx	0	Element number
Subindex for EtherCAT Rx SDO Info	0	Element number + 1

Tab. 46: Index and subindex for manufacturer-specific parameters

Calculation example

Calculation for parameter E200[0]:

Number of the group = 4

Number of the line = 200

Index = 40960 + (4 × 512) + 200 = 43208 = A8C8 hex

Subindex for EtherCAT Rx = 0 = 0 hex

Subindex for EtherCAT Rx SDO Info = 1 = 1 hex

Communication objects (axis B)

The following table includes the vendor-specific communication objects supported by axis B and how they are mapped to the corresponding parameters of STOBBER.

Index	Group	Number	Parameters
A000 hex – A1FF hex	A: Drive controller	0	A00 – A511
A200 hex – A3FF hex	B: Motor	1	B00 – B511
A400 hex – A5FF hex	C: Machine	2	C00 – C511
A600 hex – A7FF hex	D: Set value	3	D00 – D511
A800 hex – A9FF hex	E: Show	4	E00 – E511
AA00 hex – ABFF hex	F: Terminals	5	F00 – F511
AC00 hex – ADFF hex	G: Technology	6	G00 – G511
AE00 hex – AFFF hex	H: Encoders	7	H00 – H511
B000 hex – B1FF hex	I: Motion	8	I00 – I511
B200 hex – B3FF hex	J: Motion blocks	9	J00 – J511
B400 hex – B5FF hex	K: Control panel	10	K00 – K511
B600 hex – B7FF hex	L: Technology 2	11	L00 – L511
B800 hex – B9FF hex	M: Profile	12	M00 – M511
BE00 hex – BFFF hex	P: Customer-specific parameters	15	P00 – P511
C000 hex – C1FF hex	Q: Customer-specific parameters, instance-dependent	16	Q00 – Q511
C200 hex – C3FF hex	R: Production data	17	R00 – R511
C400 hex – C5FF hex	S: Safety	18	S00 – S511
C600 hex – C7FF hex	T: Scope	19	T00 – T511
C800 hex – C9FF hex	U: Protection functions	20	U00 – U511
D200 hex – D3FF hex	Z: Fault counter	25	Z00 – Z511

Tab. 47: Vendor-specific communication objects: A000 hex – D3FF hex

7.3 Detailed information

The documentation listed below provides you with further relevant information on the 6th STOBER drive controller generation. The current status of the documentation can be found in our download center at:

<http://www.stoeber.de/en/downloads/>.

Enter the ID of the documentation in the search field.

Title	Documentation	Contents	ID
SC6 drive controller	Manual	System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	442790
Multi-axis drive system with SI6 and PS6	Manual	System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	442728
EtherCAT communication – SC6, SI6	Manual	Electrical installation, data transfer, commissioning, diagnostics, detailed information	443025
SX6 safety technology – Safe drive monitoring using FSoE	Manual	Technical data, installation, commissioning, diagnostics	443337
SY6 safety technology – STO and SS1 via FSoE	Manual	Technical data, installation, commissioning, diagnostics, detailed information	442744
SR6 safety technology – STO via terminals	Manual	Technical data, installation, commissioning, diagnostics, detailed information	442741
EtherNet/IP communication – SC6, SI6	Manual	Electrical installation, data transfer, commissioning, diagnostics, detailed information	443538
Motion commands	Manual	Control and status information, denials and limitations, motion	443349

Additional information and sources that form the basis of this documentation or are referenced by the documentation:

Beckhoff Automation GmbH & Co. KG (publisher): *EtherCAT System Documentation*. Version 5.1. Verl, 2016.

A free basic version of the TwinCAT 3 automation software is available at

<https://www.beckhoff.com/en-us/products/automation/twincat/te1xxx-twincat-3-engineering/te1000.html>.

EtherCAT Technology Group (ETG), 2015. *ETG.1300: EtherCAT Indicator and Labeling*. ETG.1300 S (R) V1.1.0. Specification. 2015-07-03.

7.4 Formula symbols

Symbol	Unit	Explanation
F	N	Force
M	Nm	Torque
n	rpm	Speed
n_1	rpm	Speed at gearbox input
n_2	rpm	Speed at the gearbox output
$v_{1set,cycl}$	m/min	Set velocity transmitted cyclically by the fieldbus and interpolated and processed directly
v_{1set}	m/min	Set velocity for calculating the motion profile
x	mm	Position
$x_{1set,cycl}$	m	Set position transmitted cyclically by the fieldbus and interpolated and processed directly
x_{1set}	m	Set position for calculating the motion profile

7.5 Abbreviations

Abbreviation	Meaning
ALT	Alternative
CAN	Controller Area Network
CiA	CAN in Automation
CNC	Computerized Numerical Control
csp	Cyclic synchronous position mode
cst	Cyclic synchronous torque mode
csv	Cyclic synchronous velocity mode
DI	Digital Input
EoE	Ethernet over EtherCAT
ESI	EtherCAT SubDevice Information (device description of an EtherCAT SubDevice)
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control Automation Technology
I/O	Input/Output
ip	Interpolated position mode
IP	Internet Protocol
LinM	Linear Motor
LS	Limit Switch
LSB	Least Significant Bit
M/F	Torque or force
MAC	Media Access Control
MDevice	MainDevice
MEnc	Motor Encoder
NC	Numerical Control
PDO	Process Data Objects
PEnc	Position Encoder
PLL	Phase Locked Loop
pp	Profile position mode
PROFINET	Process Field Network
pt	Profile torque mode
pv	Profile velocity mode
RS	Reference Switch
RxPDO	Receive PDO (receive process data)
S	Switch
SubDevice	SubordinateDevice
SYNC	Synchronization
TwinCat	The Windows Control and Automation Technology (automation software from Beckhoff Automation GmbH)
TxPDO	Transmit PDO (transmit process data)
vl	Velocity mode
ZP	Zero Pulse

8 Contact

8.1 Consultation, service and address

We would be happy to help you!

We offer a wealth of information and services to go with our products on our website:

<http://www.stoeber.de/en/service>

For additional or personalized information, contact our consultation and support service:

<http://www.stoeber.de/en/support>

If you need our system support:

Phone: +49 7231 582-3060

systemsupport@stoeber.de

If you need a replacement device:

Phone: +49 7231 582-1128

replace@stoeber.de

Call our 24-hour service hotline:

Phone: +49 7231 582-3000

Our address is:

STÖBER Antriebstechnik GmbH + Co. KG

Kieselbronner Strasse 12

75177 Pforzheim, Germany

8.2 Your opinion is important to us

We created this documentation to the best of our knowledge with the goal of helping you build and expand your expertise productively and efficiently with our products.

Your suggestions, opinions, wishes and constructive criticism help us to ensure and further develop the quality of our documentation.

If you want to contact us for a specific reason, we would be happy to receive an e-mail from you at:

documentation@stoeber.de

Thank you for your interest.

Your STÖBER editorial team

8.3 Close to customers around the world

We offer you committed, expert advice and support in over 40 countries worldwide:

STOBER AUSTRIA

www.stoeber.at
+43 7613 7600-0
sales@stoeber.at

STOBER FRANCE

www.stober.fr
+33 478 98 91 80
sales@stober.fr

STOBER ITALY

www.stober.it
+39 02 93909570
sales@stober.it

STOBER KOREA

www.stober.kr
+82 10 5681 6298
sales@stober.kr

STOBER SWITZERLAND

www.stoeber.ch
+41 56 496 96 50
sales@stoeber.ch

STOBER TURKEY

www.stober.com
+90 216 510 2290
sales-turkey@stober.com

STOBER USA

www.stober.com
+1 606 759 5090
sales@stober.com

STOBER CHINA

www.stoeber.cn
+86 512 5320 8850
sales@stoeber.cn

STOBER GERMANY

www.stoeber.de
+49 7231 582-0
sales@stoeber.de

STOBER JAPAN

www.stober.co.jp
+81-3-5875-7583
sales@stober.co.jp

STOBER SWEDEN

www.stober.com
+46 702 394 675
neil.arstad@stoeber.de

STOBER TAIWAN

www.stober.tw
+886 4 2358 6089
sales@stober.tw

STOBER UK

www.stober.co.uk
+44 1543 458 858
sales@stober.co.uk

Glossary

Broadcast domain

Logical grouping of network devices within a local network that reaches all nodes via broadcast.

DC-Sync

Also: Synchronization using distributed clocks. Method for EtherCAT network synchronization. Each EtherCAT SubDevice with distributed clocks functionality has a local clock. Normally, the time from the first DC-Sync-capable EtherCAT SubDevice downstream of the MainDevice in the network serves as the reference time: Both the MainDevice and the SubDevices synchronize with this reference clock when prompted by the MainDevice. The event belonging to a synchronization is labeled as a Sync 0 signal and is generated cyclically by the SyncManager of each SubDevice.

ESI file

Device description file for EtherCAT SubDevices. In accordance with ETG.2000: XML file that contains all relevant data for an EtherCAT node in the EtherCAT system, such as the identity of the manufacturer, the product code, the version or the production number. The EtherCAT MainDevice requires this file to configure the EtherCAT system.

IPv4 limited broadcast

Type of broadcast in a network with IPv4 (Internet Protocol version 4). The IP address 255.255.255.255 is entered as the destination. The content of the broadcast is not forwarded by a router, which limits it to the local network.

Process Data Objects (PDO)

Communication objects in a CANopen or EtherCAT network that transmit data such as set and actual values, control commands or status information based on events or objectives, in cycles or in real time on request. PDOs are generally exchanged over the process data channel with high priority. Depending on the view of the respective node, a distinction is made between receive PDOs (RxPDO) and transmit PDOs (TxPDO).

SM-Sync

Also: Synchronization using SyncManager event. Method for EtherCAT network synchronization where the EtherCAT SubDevices synchronize with an event from incoming data.

Startup list

Predefined list of CiA objects that is processed every time EtherCAT is started. The values it contains are sent to the corresponding EtherCAT SubDevice with the defined state change.

Synchronization

Time synchronization of EtherCAT network nodes that allows the EtherCAT MainDevice and SubDevices to work synchronously with each other in the same cycle. EtherCAT provides two different methods for precisely synchronizing the MainDevice and SubDevices: a SyncManager event (SM-Sync) and distributed clocks (DC-Sync). The FreeRun state exists if the MainDevice and SubDevices are not synchronized.

List of figures

Fig. 1	DS6: Program interface	12
Fig. 2	DriveControlSuite: Navigation using text links and symbols	14
Fig. 3	TwinCAT 3 Engineering: Program interface.....	15
Fig. 4	CODESYS Development System V3: Program interface	16
Fig. 5	Components and configuration steps	70
Fig. 6	Endless-rotational motion: Rotary table	73
Fig. 7	Limited-rotational motion: Pointer	73
Fig. 8	Endless translational motion: Conveyor belt	74
Fig. 9	Limited translational motion: Tool slide.....	74
Fig. 10	Limited translational motion: Linear motor	74
Fig. 11	Hardware limit switch: Limit switch memory.....	78
Fig. 12	CiA 402 device state machine: Device states and state changes	111
Fig. 13	Interpolated position mode: Input and output signals	118
Fig. 14	Cyclic synchronous position mode: Input and output signals	121
Fig. 15	Cyclic synchronous velocity mode: Input and output signals.....	122
Fig. 16	Cyclic synchronous torque mode: Input and output signals	123
Fig. 17	Profile position mode: Input and output signals.....	124
Fig. 18	Velocity mode: Input and output signals	126
Fig. 19	Profile velocity mode: Input and output signals	129
Fig. 20	Profile torque mode: Input and output signals.....	131
Fig. 21	Homing mode: Input and output signals.....	132
Fig. 22	Jog: Input and output signals	134
Fig. 23	Jog – Accelerating and braking.....	135
Fig. 24	Jog – Response to contradictory signals.....	135
Fig. 25	Controller-generated external feedforward control.....	140
Fig. 26	Drive-generated internal feedforward control	141
Fig. 27	Without feedforward control.....	141
Fig. 28	Interpolation: Ideal behavior over time, no jitter	143
Fig. 29	Interpolation: Controller jitter over time	143
Fig. 30	Touch probe: Example 1.....	144
Fig. 31	Touch probe: Example 2.....	145
Fig. 32	Touch probe: Example 3.....	146

List of tables

Tab. 1	Parameter groups.....	17
Tab. 2	Parameters: Data types, parameter types, possible values	18
Tab. 3	Parameter types.....	19
Tab. 4	Referencing methods	80
Tab. 5	Loss of reference of the axis in normal operation.....	107
Tab. 6	Reference loss of the axis due to parameter changes	108
Tab. 7	Reference loss of the master encoder in normal operation	109
Tab. 8	Reference loss of the master encoder due to parameter changes	110
Tab. 9	States, transitions and conditions: Terms	112
Tab. 10	Bit combinations in the control word in accordance with CiA 402	112
Tab. 11	Interpolated position mode: Operating mode-specific bits in the control word	119
Tab. 12	Interpolated position mode: Operating mode-specific bits in the status word	119
Tab. 13	SC6, SI6: Receive process data (Interpolated position mode); first part, axis A.....	119
Tab. 14	SC6, SI6: Receive process data (Interpolated position mode); second part, axis B.....	119
Tab. 15	SC6, SI6: Transmit process data (Interpolated position mode); first part, axis A.....	120
Tab. 16	SC6, SI6: Transmit process data (Interpolated position mode); second part, axis B.....	120
Tab. 17	Cyclic synchronous position mode: Operating mode-specific bits in the status word.....	121
Tab. 18	Cyclic synchronous velocity mode: Operating mode-specific bits in the status word	122
Tab. 19	Cyclic synchronous torque mode: Operating mode-specific bits in the status word.....	123
Tab. 20	Profile position mode: Operating mode-specific bits in the control word.....	125
Tab. 21	Profile position mode: Operating mode-specific bits in the status word.....	125
Tab. 22	Velocity mode: Operating mode-specific bits in the control word	127
Tab. 23	SC6, SI6: Receive process data (Velocity mode); first part, axis A.....	127
Tab. 24	SC6, SI6: Receive process data (Velocity mode); second part, axis B.....	127
Tab. 25	SC6, SI6: Transmit process data (Velocity mode); first part, axis A.....	128
Tab. 26	SC6, SI6: Transmit process data (Velocity mode); second part, axis B.....	128
Tab. 27	Profile velocity mode: Operating mode-specific bits in the control word	130
Tab. 28	Profile velocity mode: Operating mode-specific bits in the status word	130
Tab. 29	Profile torque mode: Operating mode-specific bits in the control word.....	131
Tab. 30	Profile torque mode: Operating mode-specific bits in the status word.....	131
Tab. 31	Homing mode: Operating mode-specific bits in the control word.....	133
Tab. 32	Homing mode: Operating mode-specific bits in the status word	133
Tab. 33	Homing mode: Meaning of operating mode-specific bit combinations in the status word.....	133
Tab. 34	Jog: Operating mode-specific bits in the control word	136
Tab. 35	Comparison of the operating modes in accordance with CiA 402 and the commands based on PLCopen....	137

Tab. 36	Cycle times	147
Tab. 37	SC6, SI6: receive process data (standard mapping); first part, axis A	148
Tab. 38	SC6, SI6: receive process data (standard mapping); second part, axis B	149
Tab. 39	SC6, SI6: transmit process data (standard mapping); first part, axis A	150
Tab. 40	SC6, SI6: transmit process data (standard mapping); second part, axis B	151
Tab. 41	CiA 402-2 communication objects: 6000 hex – 65FF hex	152
Tab. 42	CiA 402-2 communication objects: 6800 hex – 6DFF hex	156
Tab. 43	Values for 60D0 hex and 68D0 hex	160
Tab. 44	Index and subindex for manufacturer-specific parameters	161
Tab. 45	Manufacturer-specific communication objects: 2000 hex – 53FF hex	162
Tab. 46	Index and subindex for manufacturer-specific parameters	163
Tab. 47	Vendor-specific communication objects: A000 hex – D3FF hex	164



4 4 3 0 8 0 - 0 4

10/2026

STÖBER Antriebstechnik GmbH + Co. KG
Kieselbronner Str. 12
75177 Pforzheim
Germany
Tel. +49 7231 582-0
mail@stoeber.de
www.stober.com

24 h Service Hotline
+49 7231 582-3000

Copyright © STÖBER 2018
All rights reserved.

www.stober.com